**SZABADKAI MŰSZAKI SZAKFŐISKOLA**

**PROJEKTUM**

**Haladó web programozás és Integrált web rendszerek tantárgyakból**

**HALLGATÓ(K) MENTOR**

**Huszár Bálint dr Zlatko Čović**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**SZABADKA, 2024.**

# TARTALOM

[TARTALOM 2](#_Toc6242)

[A FELADATKÖR MEGHATÁROZÁSA 3](#_Toc6661)

[Weboldal 3](#_Toc4902)

[Mobil alkalmazás 4](#_Toc10522)

[PIACKUTATÁS 5](#_Toc12519)

[KIDOLGOZÁS 6](#_Toc4981)

[Weboldal 6](#_Toc14620)

[Mobil alkalmazás 8](#_Toc28325)

[AZ ADATBÁZIS STRUKTURÁJA 12](#_Toc28481)

[A PROJEKT MŰKÖDÉSÉNEK LEÍRÁSA 13](#_Toc2997)

[Honlap 13](#_Toc5552)

[Vendég 13](#_Toc27436)

[Bejelentkezett felhasználó 16](#_Toc7196)

[Organizáció 20](#_Toc4629)

[Admin 21](#_Toc32511)

[Mobil alkalmazás 26](#_Toc20527)

[FELHASZNÁLT IRODALOM 34](#_Toc18310)

# A FELADATKÖR MEGHATÁROZÁSA

## Weboldal

A projektnek a feladata az volt, hogy egy működőképes weboldalat lehessen létrehozni, amely turistáknak célzott.

A weboldalon különböző turisztikai túrákat lehet létrehozni, amelyek létező vannak attrakciókból állnak. Az attrakciók városokra vannak osztva. Lehet ez főváros, település vagy akár falu is.

A vendég aki, még nem regisztrált megtudja tekinteni a városokat, és a városok által tud keresni attrakciókat, név és attrakció típus által tud keresni. Ha esetleg többet akarna csinálni akkor tud regisztrálni. Emellett alap információkat tud az oldalról megnézni.

A bejelentkezett felhasználó tudja az oldalon keresni az ország szerint is az attrakciókat, illetve lát több adatot egyes attrakciókról.

Miután a bejelentkezett felhasználó rámegy a ,,More information” gombra egy másik oldalra dobja a felhasználót, ahol leírást láthat róla illetve egy térképet ahol tudja látni pontosan hol található az attrakció.

A térkép alatt különböző képeket lát bizonyos látványosságról.

Ha kedve van megnézheti a kommenteket, amiket más felhasználók írtak arról a bizonyos látványosságról, és ő is kommentelhet.

Ha nagyon tetszik az attrakció akkor az város kedvencévé tudja alakítani. Városonként, csak 1 kedvenc attrakció lehet. Ezeket bírja módosítani és törölni is.

Ha nagyon akar akkor tud túrákat létrehozni, ahol kiválaszthat több attrakciót is amit meg akar tekinteni.

A bejelentkezett organizáció, amiket csak a rendszergazda tud létrehozni látja a saját attrakciókjait és annak a népszerűségeit. Minden egyes kedvenccé tételnél nől 1-el a népszerűség.

Tudja név, típus és népszerűség által is keresni az attrakciókat.

Képes új attrakciót létrehozni, ahol a nevét az attrakciónak, leírását, tudja leírni, típusát kiválasztani és a koordinátáit beírni ami decimálisan kell beírni.

Több képet is tud feltölteni az attrakcióról.

A rendszergazda képes tiltani felhasználókat a bejelentkezéstől, készíteni új városokat, organizációkat, tiltani organizációkat az attrakciók mutatásáról, a bejelentkezett felhasználók hozzáférési eszközéről képes adatokat gyűjteni. Tudja módosítani a szervezetek adatait, városokat vagy akár törölheti azokat

## Mobil alkalmazás

A mobil alkalmazásnak két módja van, van a sima vendég aki csak az attrakcióknak nevét látja és típusát, illetve a bejelntkezett felhasználó.

A bejelentkezett felhasználó tud több információt megtekinteni az attrakciókról, meg tudja a leírásokat nézni, a várost melyik organizáció a készítő kedvenccé tenni azokat, törölni a kedvencek közül, és a profiljánál bírja változtatni a nevét és vezetéknevét.

# PIACKUTATÁS

A témámhoz hasonló weboldalak amiket a világhálón találtam a következők:

* Tripadvisor: <https://www.tripadvisor.com/>
* Lonelyplanet: <https://www.lonelyplanet.com/>

A Tripadvisor-nek az oldalán tetszett a letisztultság, amit az enyémen nem annyira lehet észrevenni, emellett a valutaváltó lehetősége. Nem tetszik a zsúfolt kinézet a keresésnél, több dolgot kiír mint ami kell.

A Lonelyplanet-nél tetszettek az egyértelmű ,,trip”-ek és az is, hogy az Instagramon egy #lonelyplanet címkével meg lehet a képeket nézni a weboldalon.

A plusz három funkció amik ki lettek választva a projektre a következők:

* Kommentek
* Hírlevél
* QR-kód

Én a harmadik funkció helyett térkép integrálását próbáltam megvalósítani, mivel odaillőbnek találtam.

# KIDOLGOZÁS

## Weboldal

Ez a funkció kiszedi 1 táblából az adatokat, a táblát mi adjuk meg.

function getAllData(PDO $pdo, string $table\_name): array

{

$sql = "SELECT \* FROM $table\_name";

$stmt = $pdo->prepare($sql);

$stmt->execute();

return $stmt->fetchAll(PDO::FETCH\_ASSOC);

}

Ez a funkció kitörli 1 táblából az adatot a megadott tábla név, átadott id-név, és id által:

function deleteTableData(PDO $pdo, string $table\_name, string $id\_name, int $id): bool

{

$sql = "DELETE FROM $table\_name WHERE $id\_name = :id;";

$stmt = $pdo->prepare($sql);

$stmt->bindParam(':id', $id, PDO::PARAM\_STR);

return $stmt->execute();

}

Ezzel a funkcióval egy megadott táblából kiszedhetők az adatok.

function getTableData(PDO $pdo, string $table\_name, string $id\_name, int $id, bool $fetchAll): array|bool

{

$sql = "SELECT \* FROM $table\_name WHERE $id\_name = :id";

$stmt = $pdo->prepare($sql);

$stmt->bindParam(':id', $id, PDO::PARAM\_INT);

$stmt->execute();

if ($fetchAll)

return $stmt->fetchAll(PDO::FETCH\_ASSOC);

else

return $stmt->fetch(PDO::FETCH\_ASSOC);

}

Ezzel a funkcióval a táblába megnézhető, hogy létezik-e egy cella a táblában, a megadott paraméterrel.

function dataExists(PDO $pdo, string $selectField, string $selectTable, array $whereFields, array $whereValues): bool

{

$sql = "SELECT $selectField FROM $selectTable";

$where = " WHERE $whereFields[0] = :value0";

$params = [":value0" => $whereValues[0]];

for ($i = 1; $i < count($whereFields); $i++) {

$where .= " AND $whereFields[$i] = :value$i";

$params[":value$i"] = $whereValues[$i];

}

$sql .= $where;

$stmt = $pdo->prepare($sql);

$stmt->execute($params);

return $stmt->rowCount() > 0 ? true : false;

}

## Mobil alkalmazás

A felhasználó adatjainak a változtatása (UPDATE/PATCH):

<?php

header("Content-type: application/json; charset=UTF-8");

require\_once 'config.php';

require\_once 'functions.php';

if ($\_SERVER["REQUEST\_METHOD"] === "PATCH") {

try {

$data = json\_decode(file\_get\_contents("php://input"), true);

$id\_user = $data["id\_user"];

$firstname = $data["firstname"];

$lastname = $data['lastname'];

if (empty($id\_user) || empty($firstname) || empty($lastname)) {

http\_response\_code(400);

echo json\_encode(['status' => 'error', 'message' => 'Empty variables']);

exit();

}

$stmt = $GLOBALS['pdo']->prepare("UPDATE users SET firstname = :firstname, lastname = :lastname WHERE id\_user = :id\_user");

$stmt->bindParam(':id\_user', $id\_user, PDO::PARAM\_INT);

$stmt->bindParam(':firstname', $firstname, PDO::PARAM\_STR);

$stmt->bindParam(':lastname', $lastname, PDO::PARAM\_STR);

$stmt->execute();

sleep(1);

echo json\_encode(array("status" => "ok", "message" => "Data updated successfully"));

exit();

} catch (PDOException $e) {

http\_response\_code(500);

echo json\_encode(array("status" => "error", "message" => "Database error"));

exit();

}

} else {

http\_response\_code(405);

echo json\_encode(array("status" => "error", "message" => "Method Not Allowed"));

exit();}

Új kedvenc hozzáadása (CREATE/POST)

<?php

header("Content-type: application/json; charset=UTF-8");

require\_once 'config.php';

require\_once 'functions.php';

if ($\_SERVER['REQUEST\_METHOD'] === "POST") {

try {

$data = json\_decode(file\_get\_contents('php://input'), true);

if (!empty($data)) {

if (isset($data['id\_attraction']))

$id\_attraction = (int)$data['id\_attraction'];

if (isset($data['id\_user']))

$id\_user = (int)$data['id\_user'];

if (isset($id\_user) and isset($id\_attraction))

$simpleFavourite = insertFavouriteAttraction($id\_user, $id\_attraction);

}

if (empty($id\_attraction) || empty($id\_user)) {

http\_response\_code(400);

echo json\_encode(['status' => 'error', 'message' => 'Not enough variables given']);

exit();

}

if (empty($simpleFavourite)) {

http\_response\_code(404);

echo json\_encode(['status' => 'error', 'message' => 'Not found']);

} else {

http\_response\_code(200);

echo json\_encode(array("status" => "success", "message" => "Successful insert"));

}

exit();

} catch (PDOException $e) {

http\_response\_code(500);

echo json\_encode(array("status" => "error", "message" => "Database error!"));

exit();

}

} else {

http\_response\_code(405);

echo json\_encode(array("status" => "error", "message" => "Method Not Allowed"));

exit();

}

Attrakciók lekérése (READ/GET)

<?php

header("Content-type: application/json; charset=UTF-8");

require\_once 'config.php';

require\_once 'functions.php';

if ($\_SERVER['REQUEST\_METHOD'] === "GET") {

$fields = ['id\_attraction', 'attraction\_name', 'type'];

$attractions = getAttractions($fields);

if (empty($attractions)) {

http\_response\_code(404);

echo json\_encode(['status' => 'error', 'message' => 'No attractions exist yet!']);

} else {

http\_response\_code(200);

echo json\_encode($attractions);

}

} else {

http\_response\_code(405);

echo json\_encode(array("status" => "error", "message" => "Method Not Allowed"));

}

exit();

Kedvenc törlése (DELETE)

<?php

header("Content-type: application/json; charset=UTF-8");

require\_once 'config.php';

require\_once 'functions.php';

if ($\_SERVER['REQUEST\_METHOD'] === "DELETE") {

try {

$data = json\_decode(file\_get\_contents('php://input'), true);

$id\_favourite = (int)$data['id\_favourite'];

$simpleFavourite = deleteFavouriteAttraction($id\_favourite);

if (empty($id\_favourite)) {

http\_response\_code(400);

echo json\_encode(['status' => 'error', 'message' => 'Nonexistent favourite']);

exit();

}

if (empty($simpleFavourite)) {

http\_response\_code(404);

echo json\_encode(['status' => 'error', 'message' => 'Not found']);

} else {

http\_response\_code(200);

echo json\_encode(array("status" => "success", "message" => "Successful deletion"));

}

exit();

} catch (PDOException $e) {

http\_response\_code(500);

echo json\_encode(array("status" => "error", "message" => "Database error!"));

exit();

}

} else {

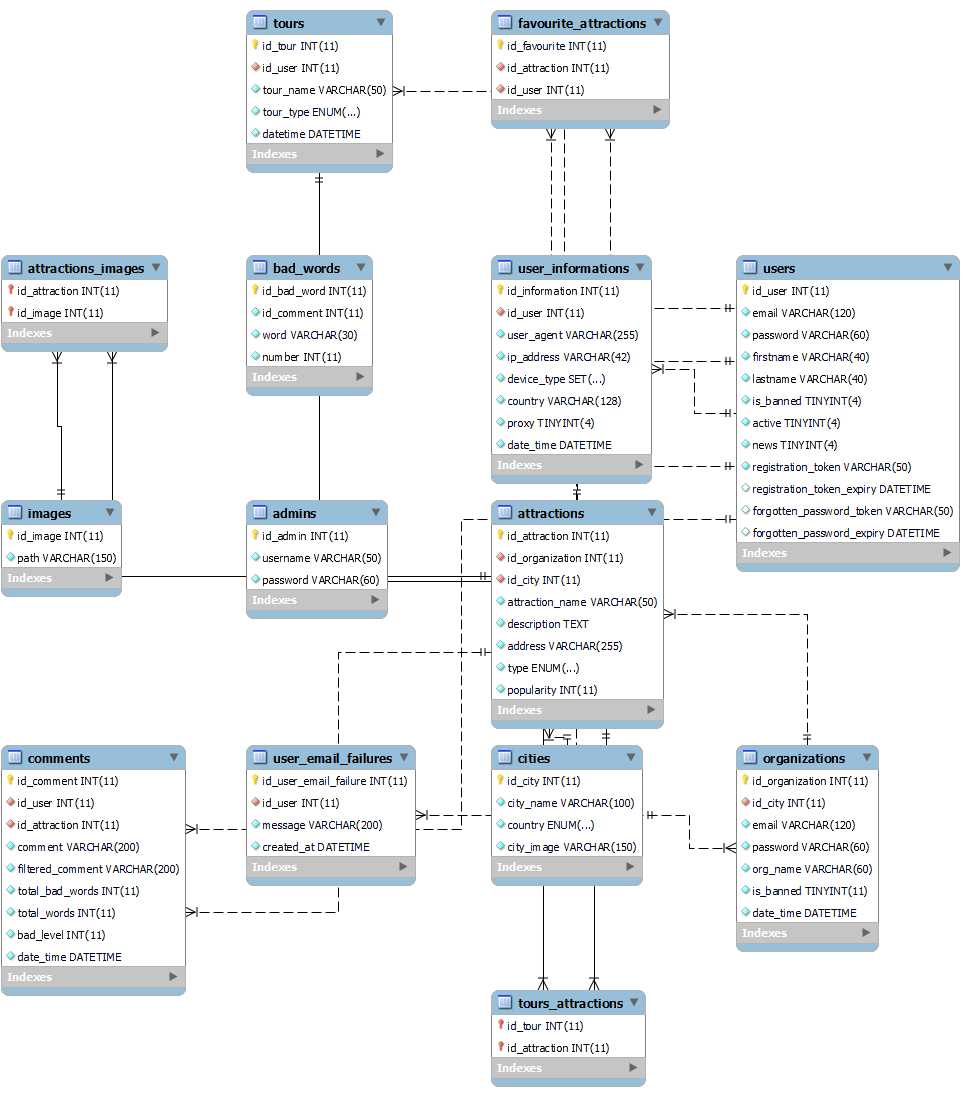
http\_response\_code(405);

echo json\_encode(array("status" => "error", "message" => "Method Not Allowed"));

exit();

}

# AZ ADATBÁZIS STRUKTURÁJA

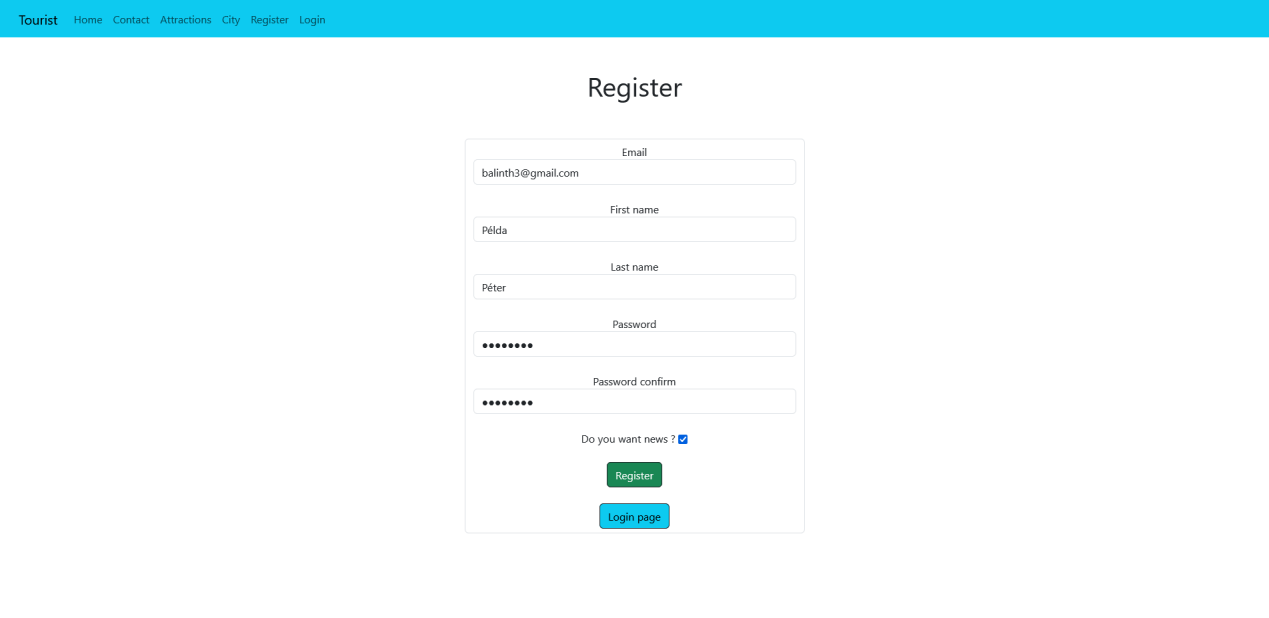


Ábra 1: Adatbázis sturktúrája

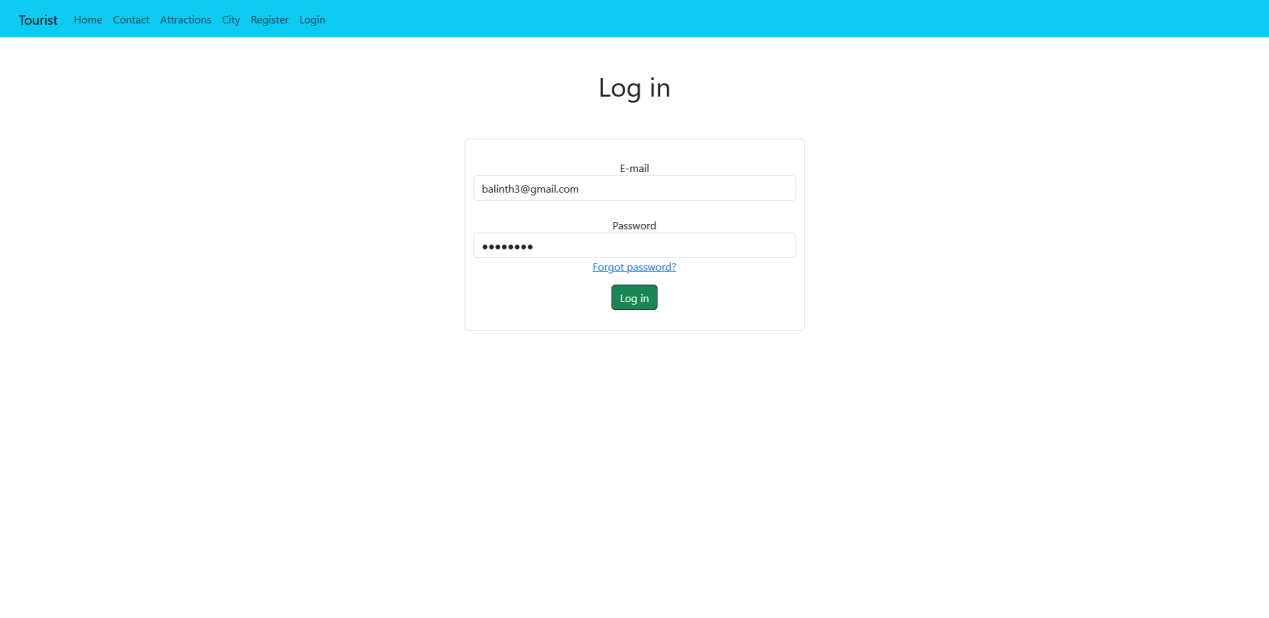
# A PROJEKT MŰKÖDÉSÉNEK LEÍRÁSA

## Honlap

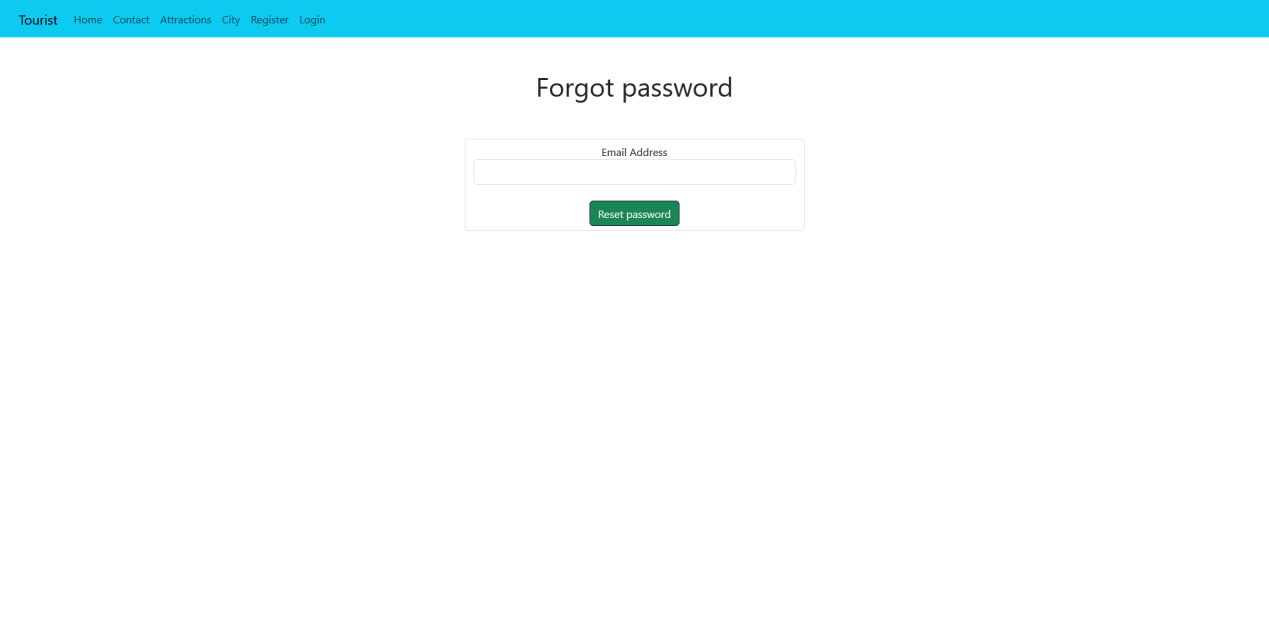
### Vendég



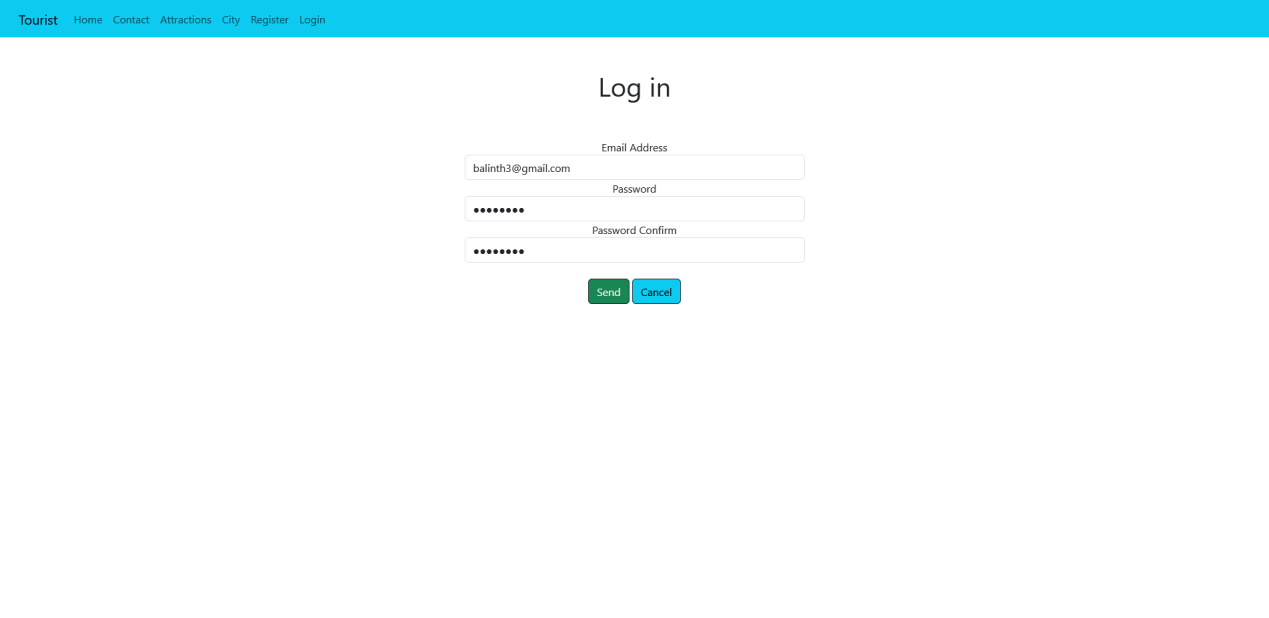
Ábra 2: Regisztrációs képernyő



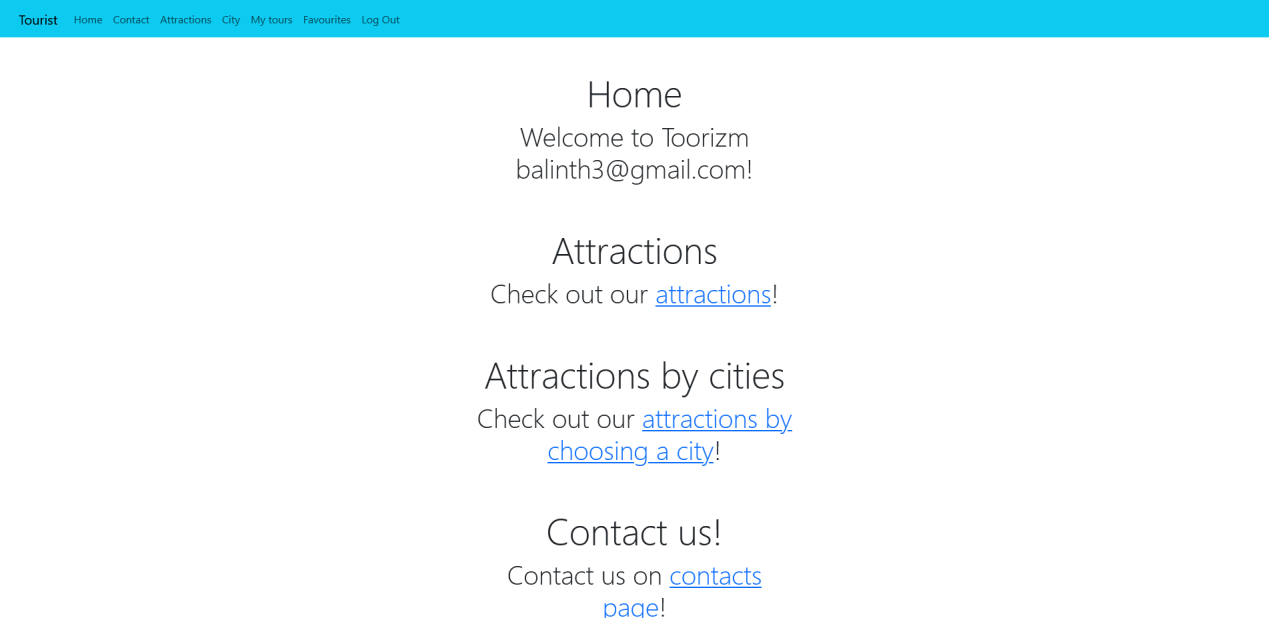
Ábra 3: Bejelentkezési képernyő



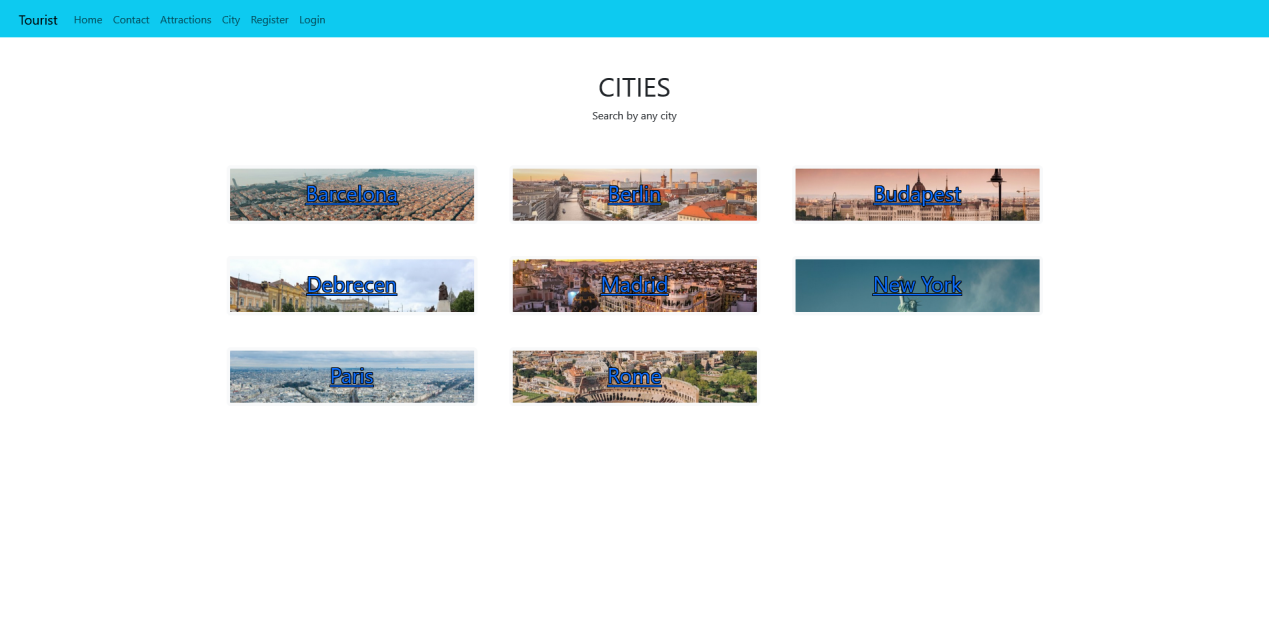
Ábra 4: Elfejetett jelszó



Ábra 5: Új jelszó beállítása



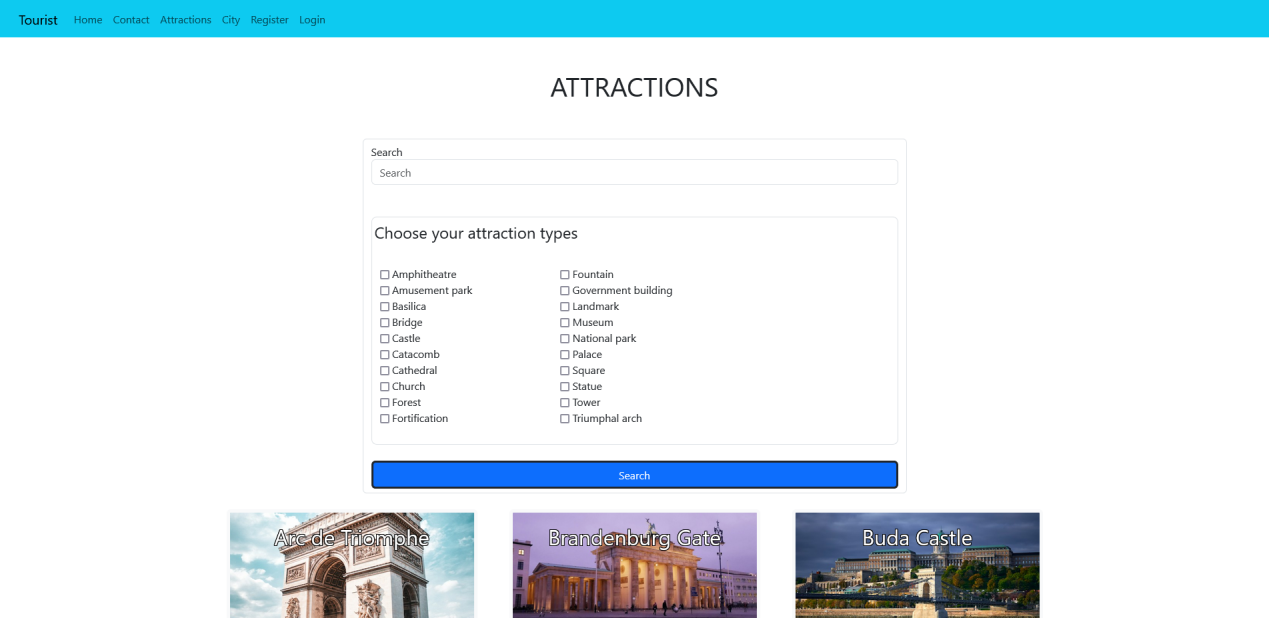
Ábra 6: Index képernyő



Ábra 7: Városok általi attrakció keresés

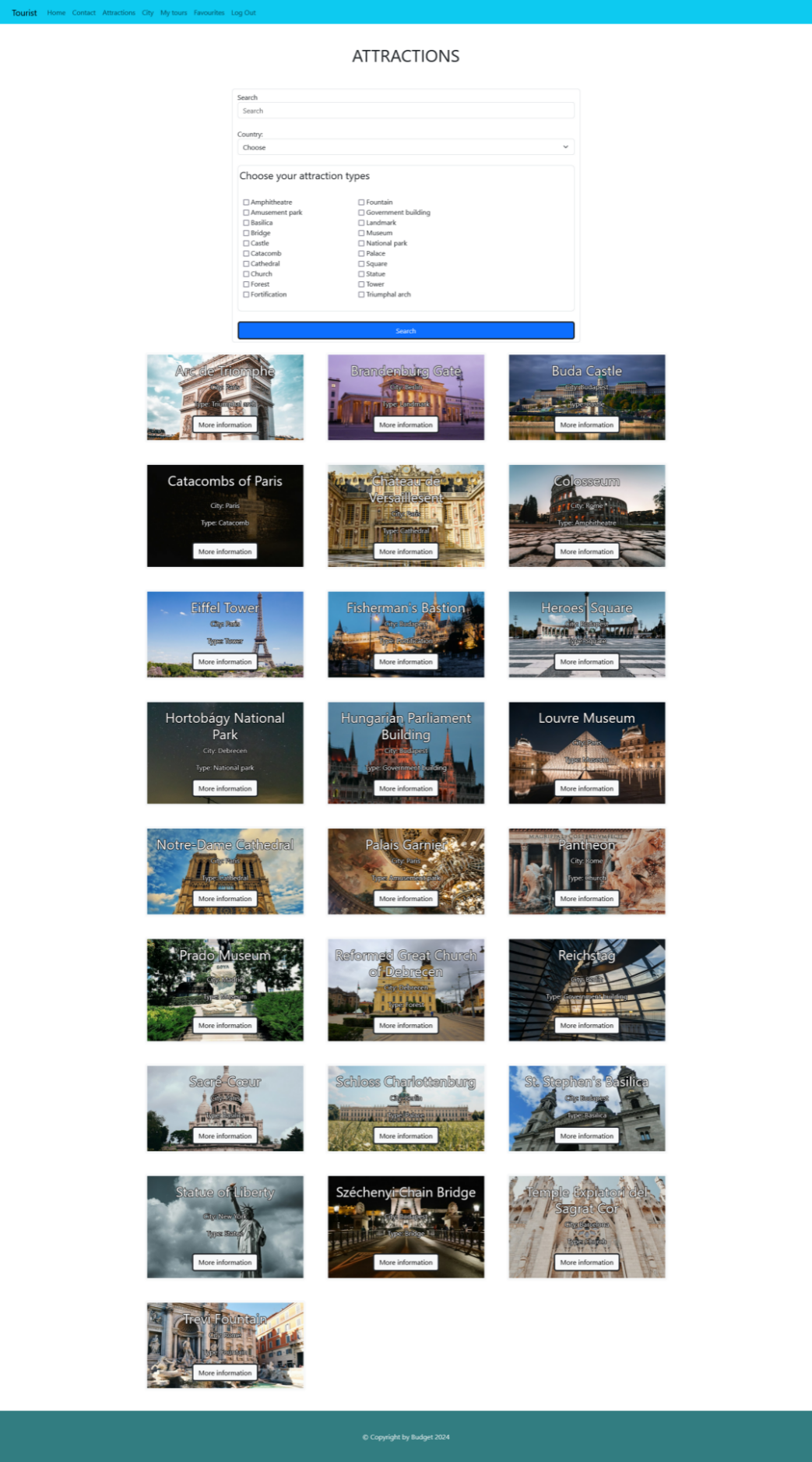


Ábra 8: Elérhetőségek oldal

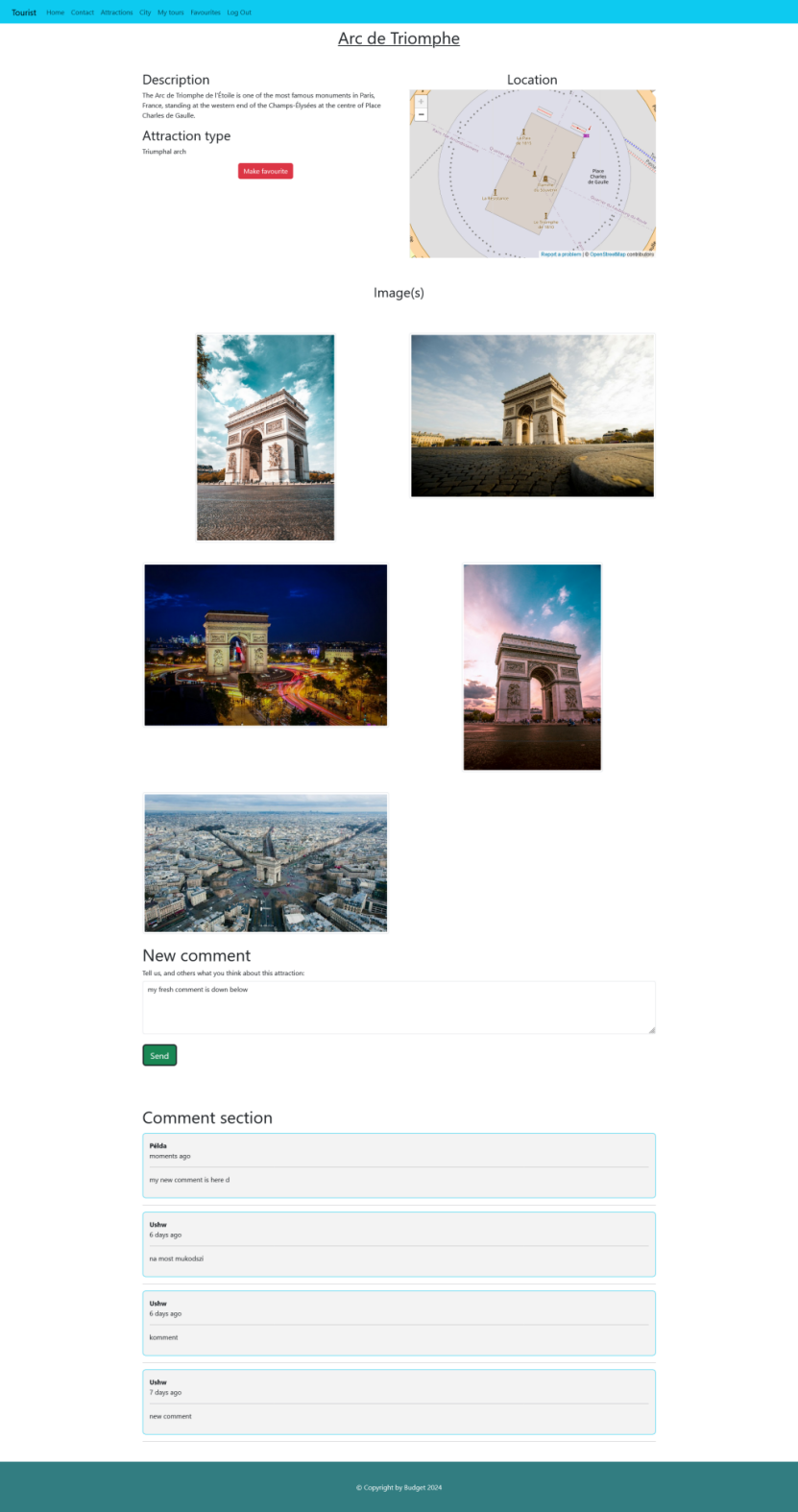


Ábra 9: Vendégnél az attrakciók

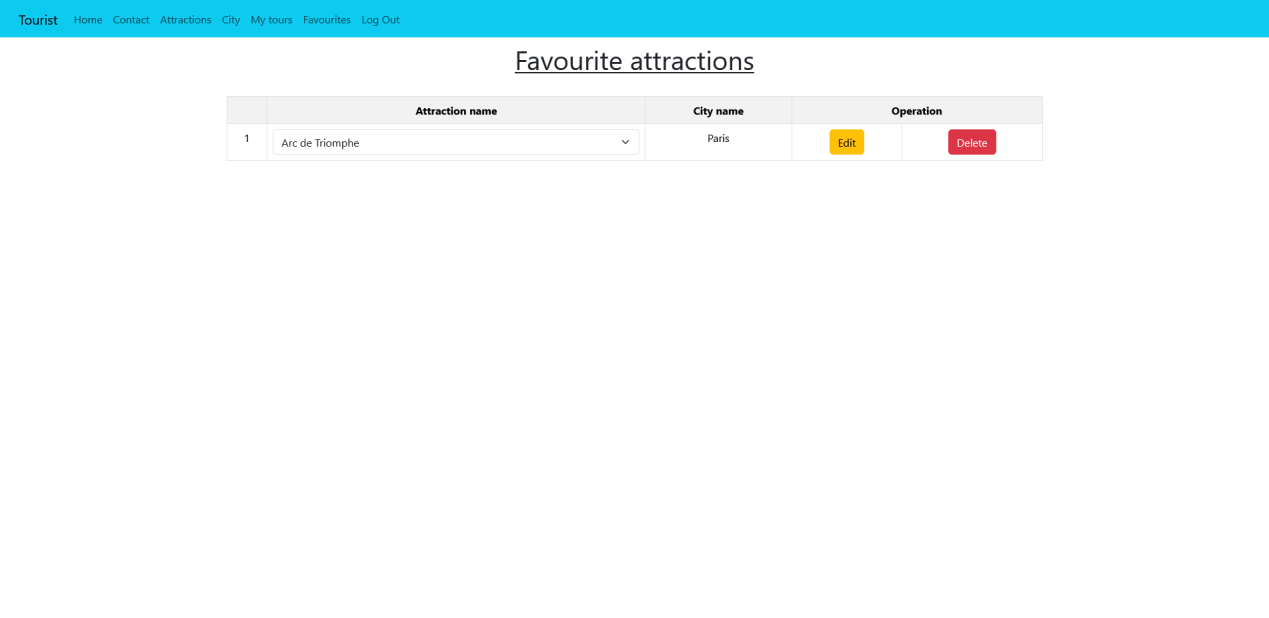
### Bejelentkezett felhasználó



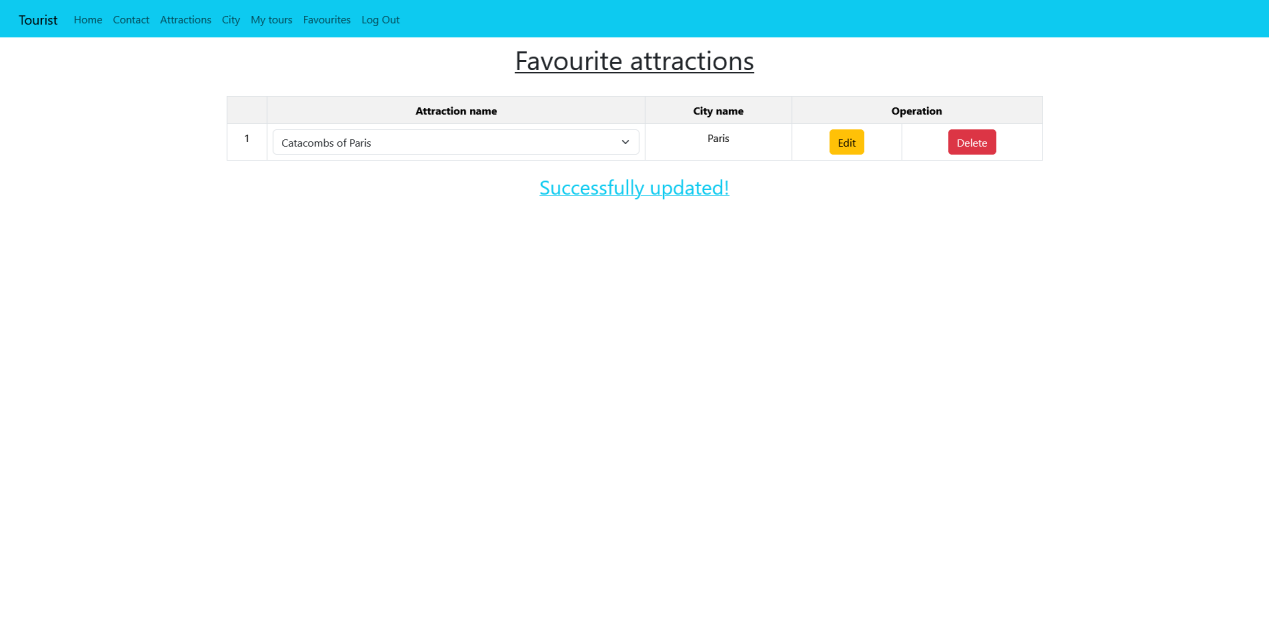
Ábra 10: Felsorolt attrakciók bejelentkezett felhasználóna



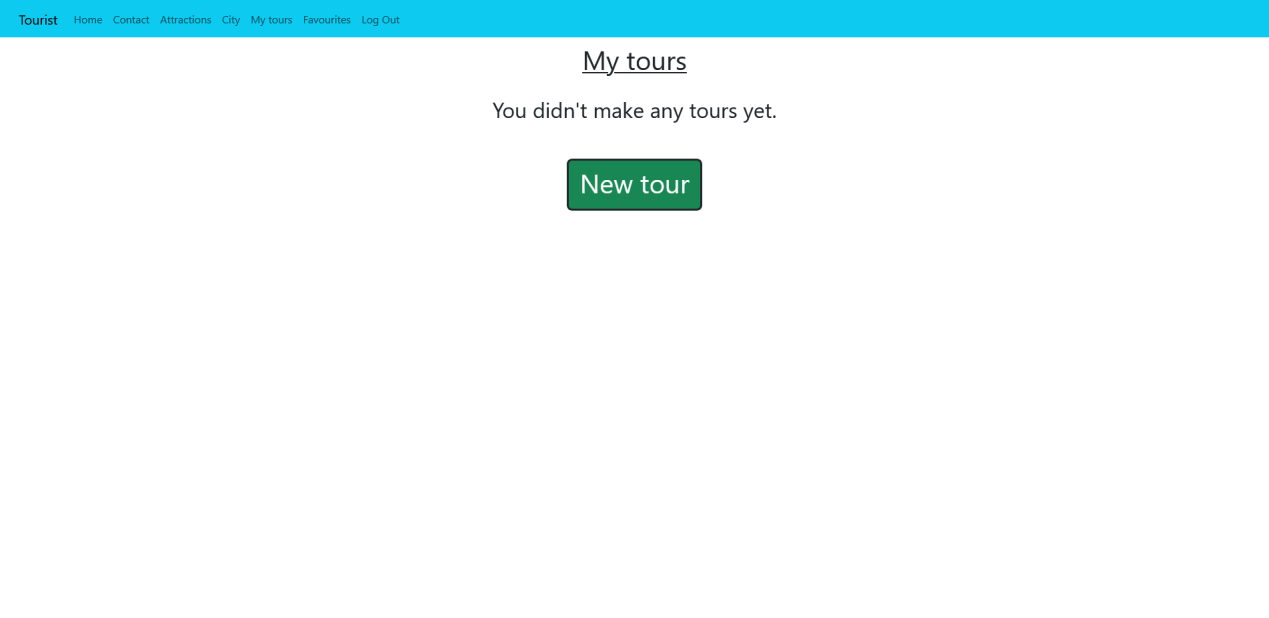
Ábra 11: Attrakcióról részletek



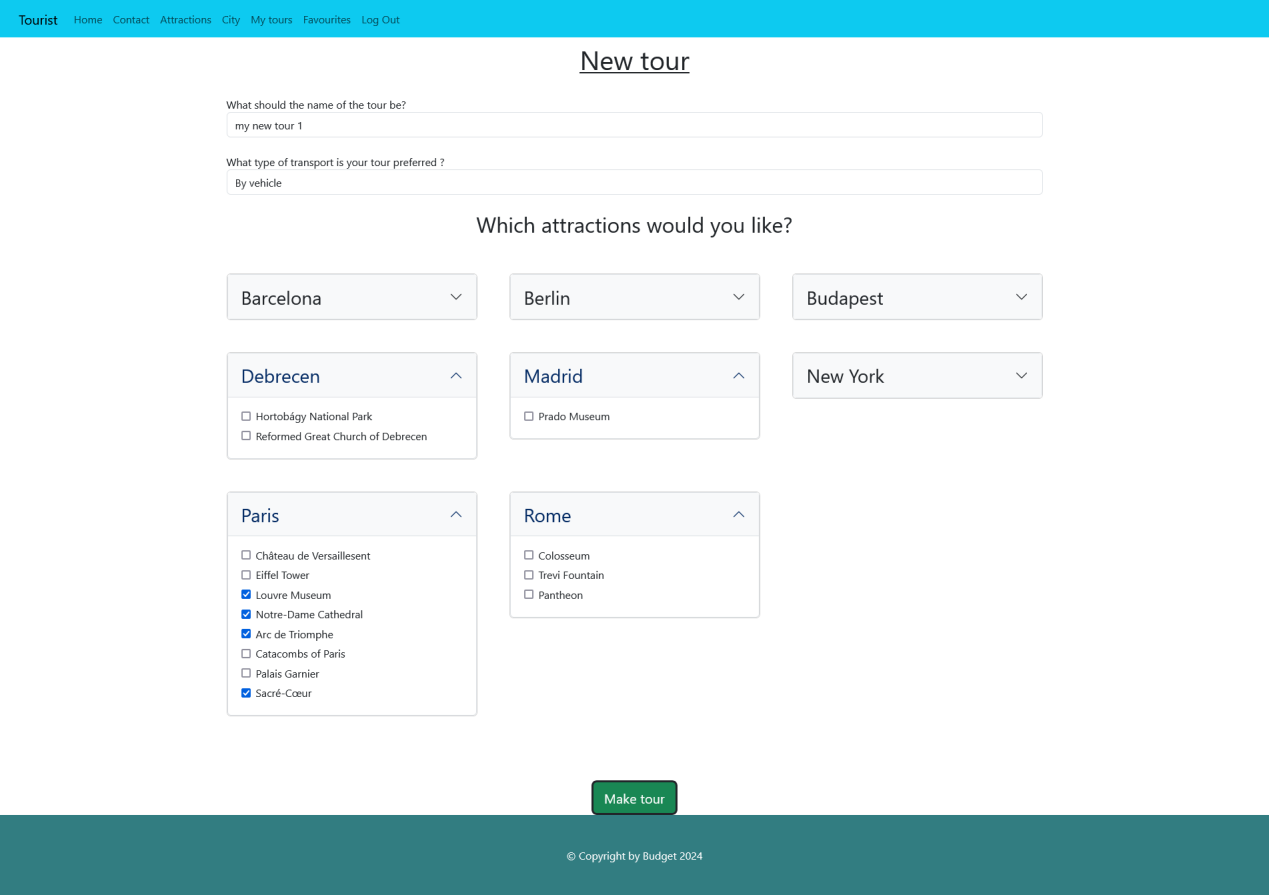
Ábra 12: Kedvenc attrakciók

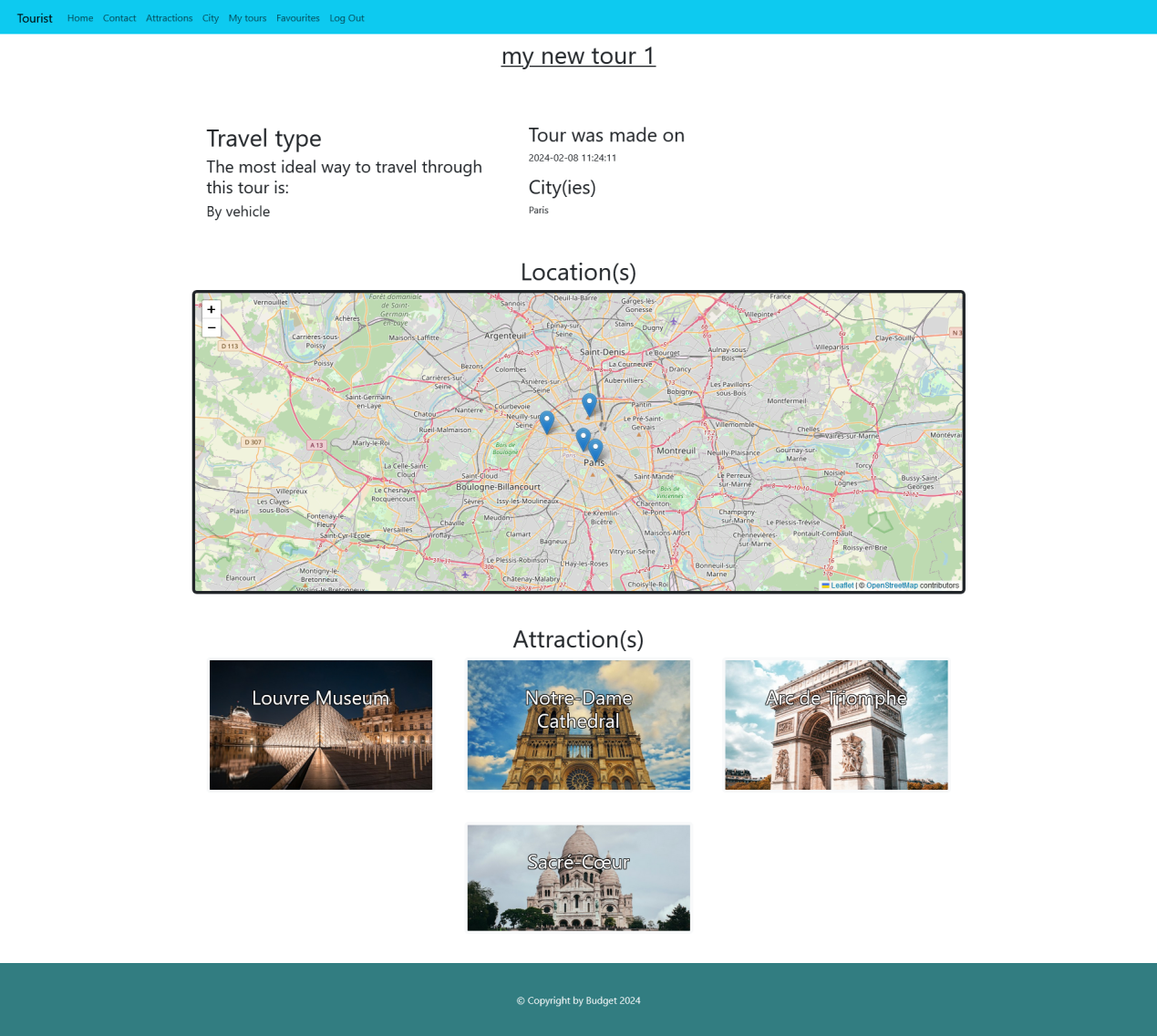


Ábra 13: Kedvenc változtatása



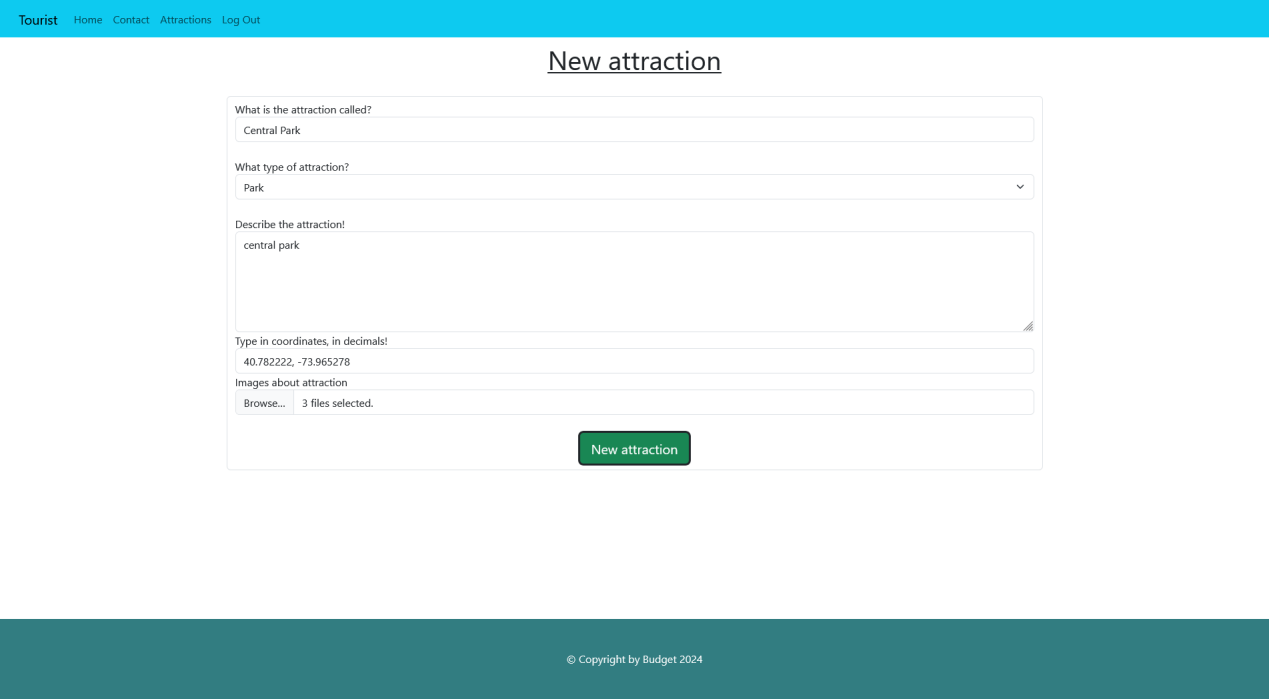
Ábra 14: Nem léteznek túrák



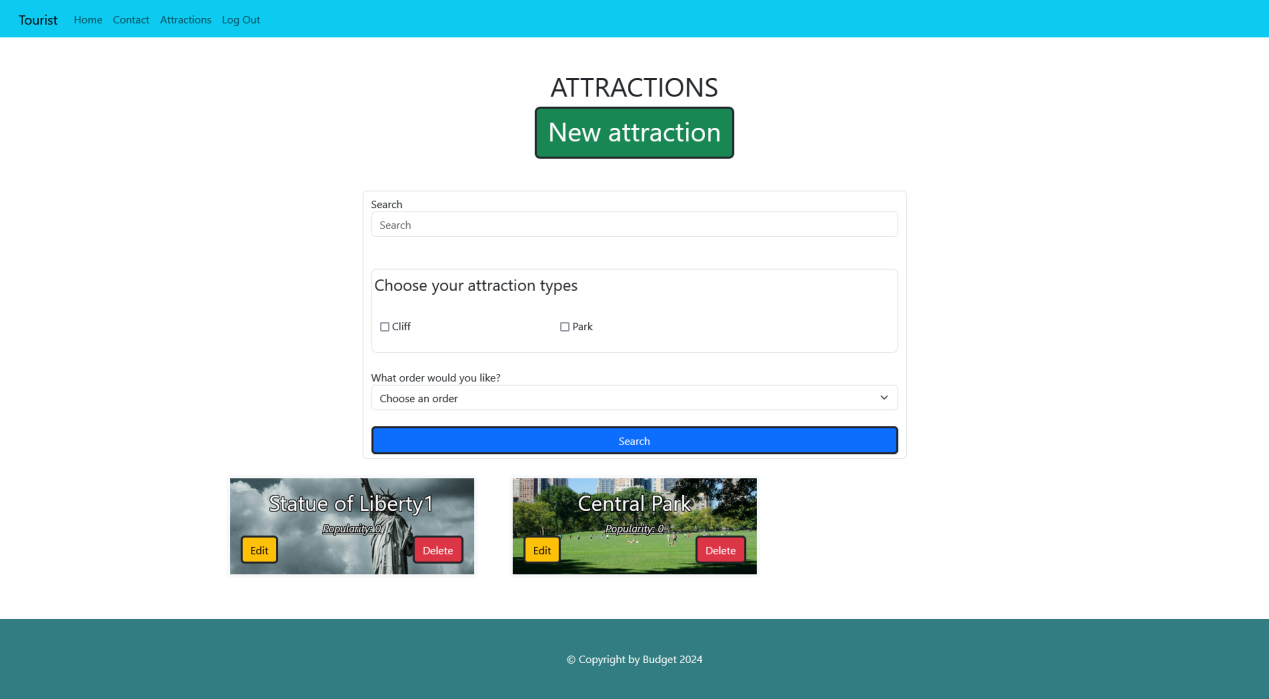
Ábra 15: Új túra létrehozása Párizsban

Ábra 16: Túra megtekintése

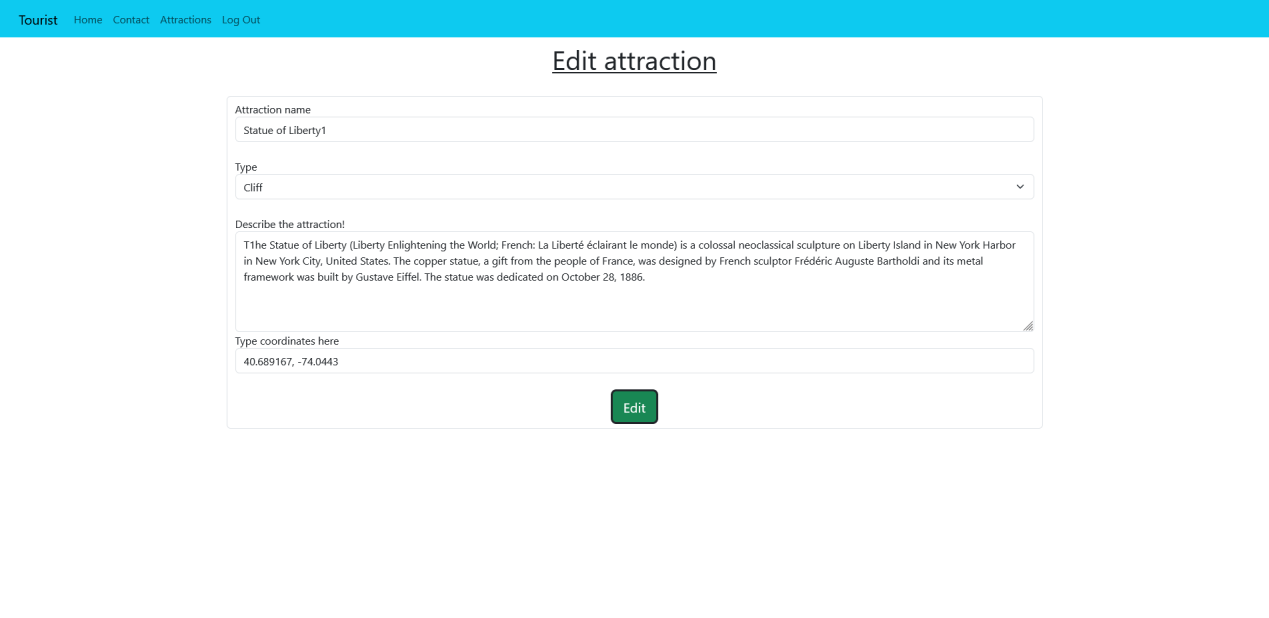
### Organizáció



Ábra 17: Új attrakció

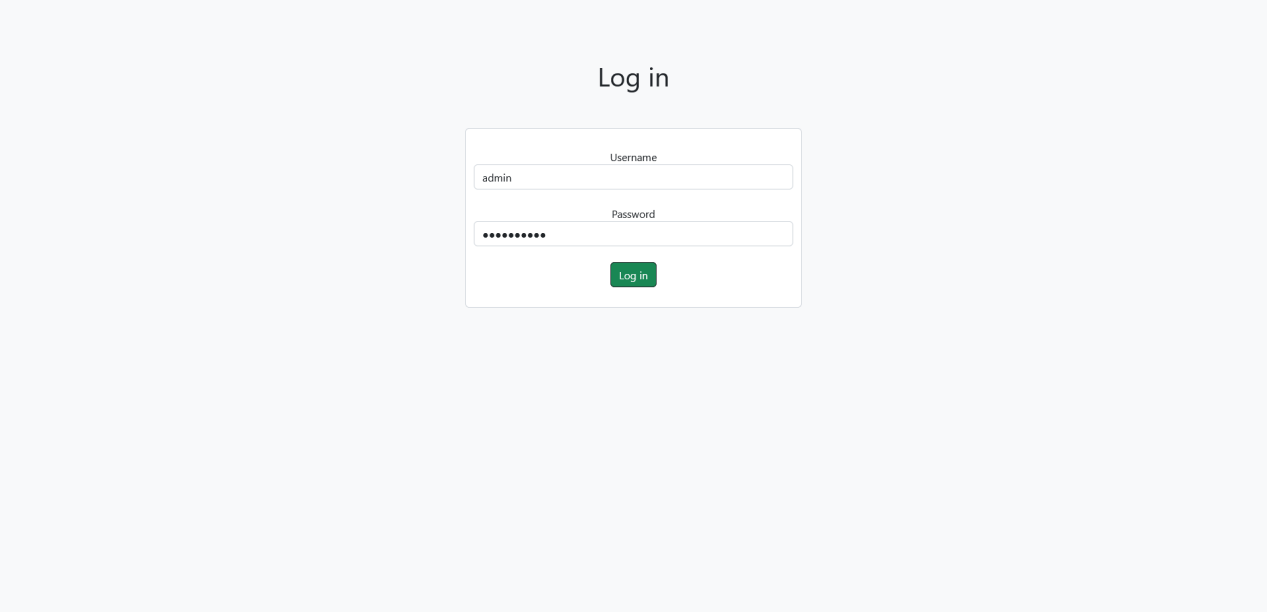


Ábra 18: Attrakciók, amiket az organizáció lát

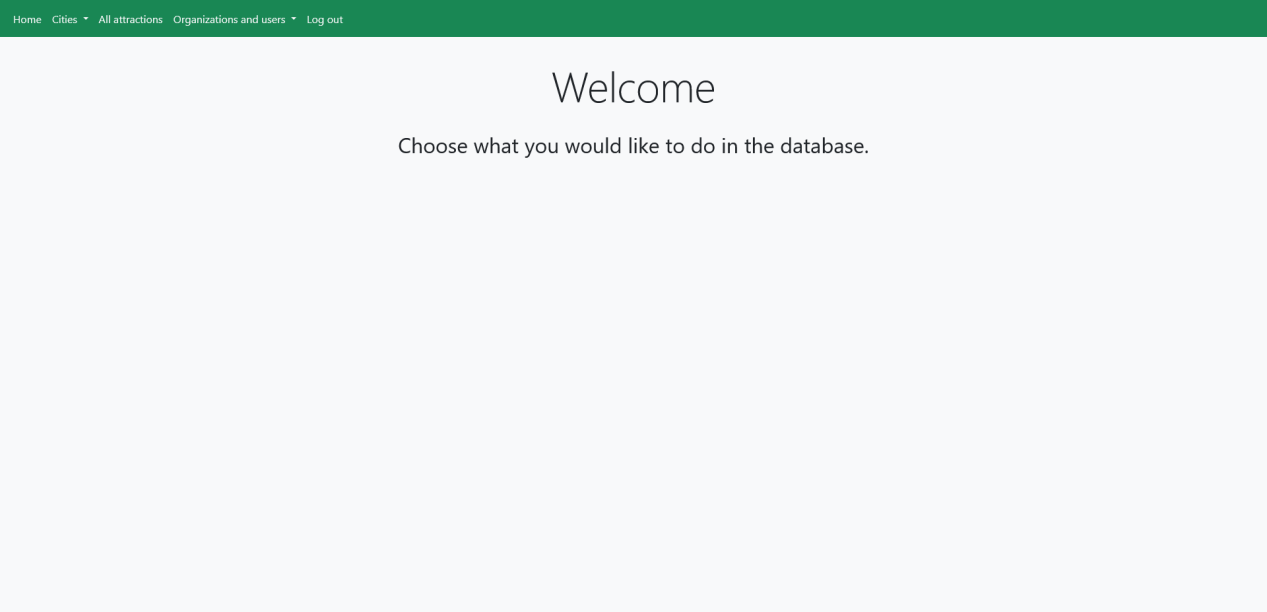


Ábra 19: Attrakció változtatása

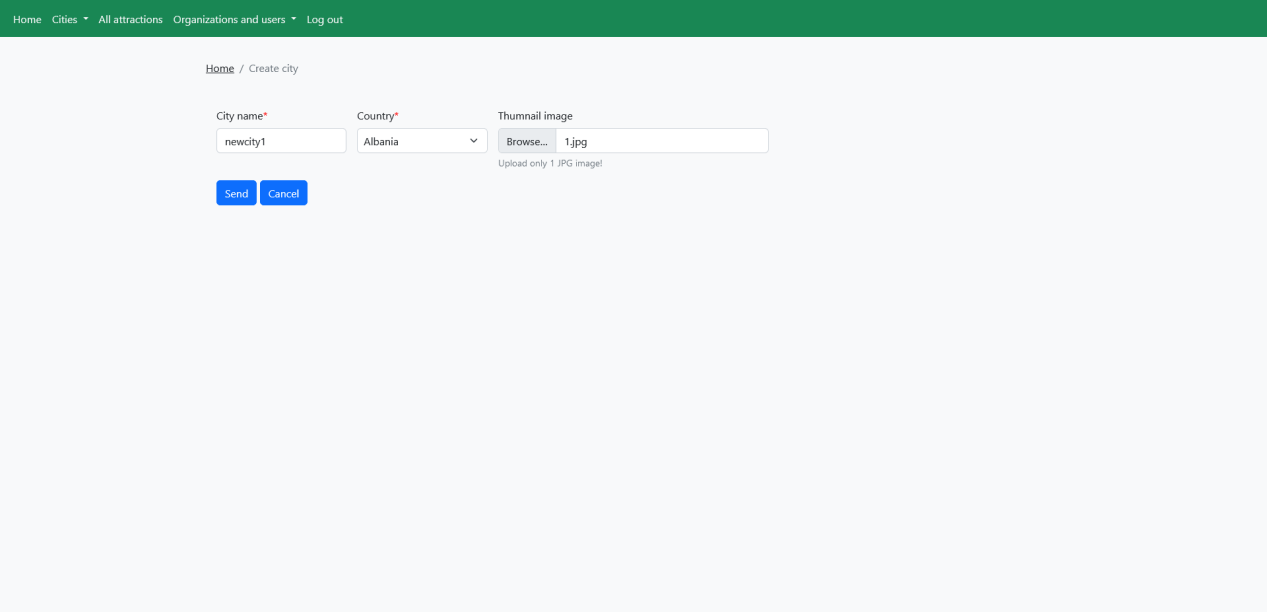
### Admin



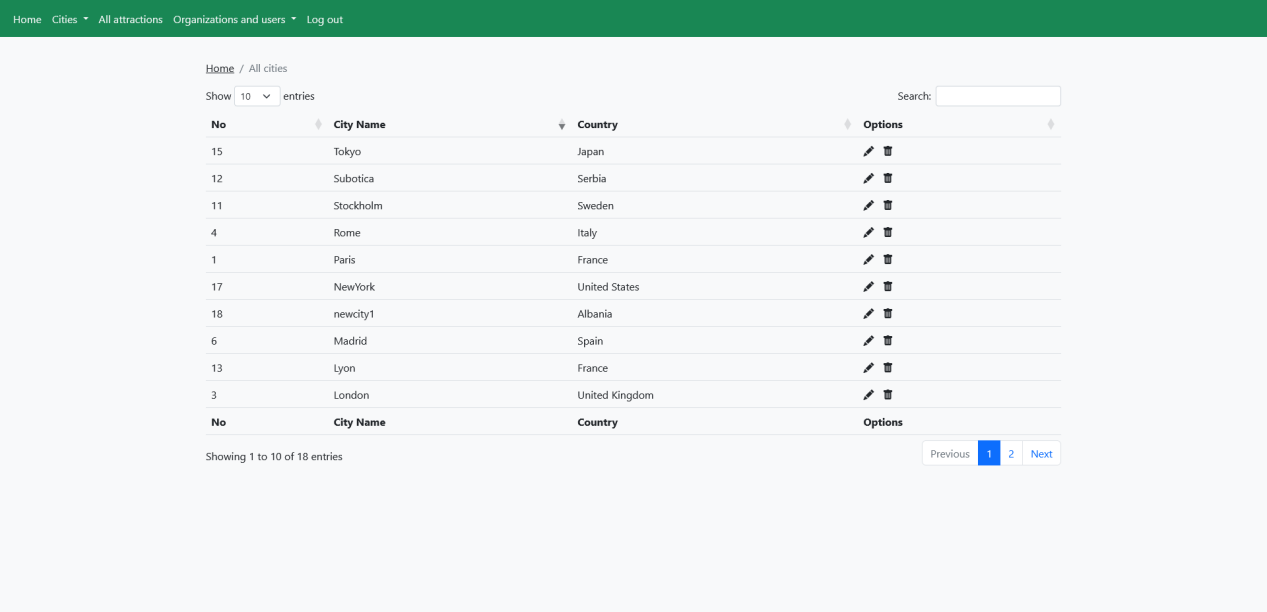
Ábra 20: Admin bejelentkezési űrlap



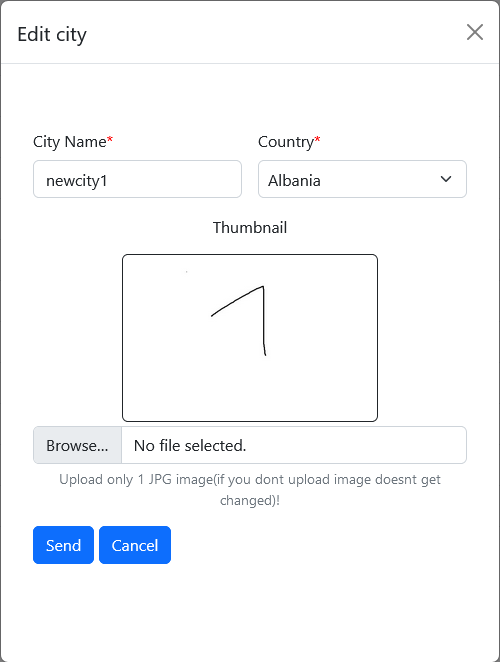
Ábra 21: Admin index oldal



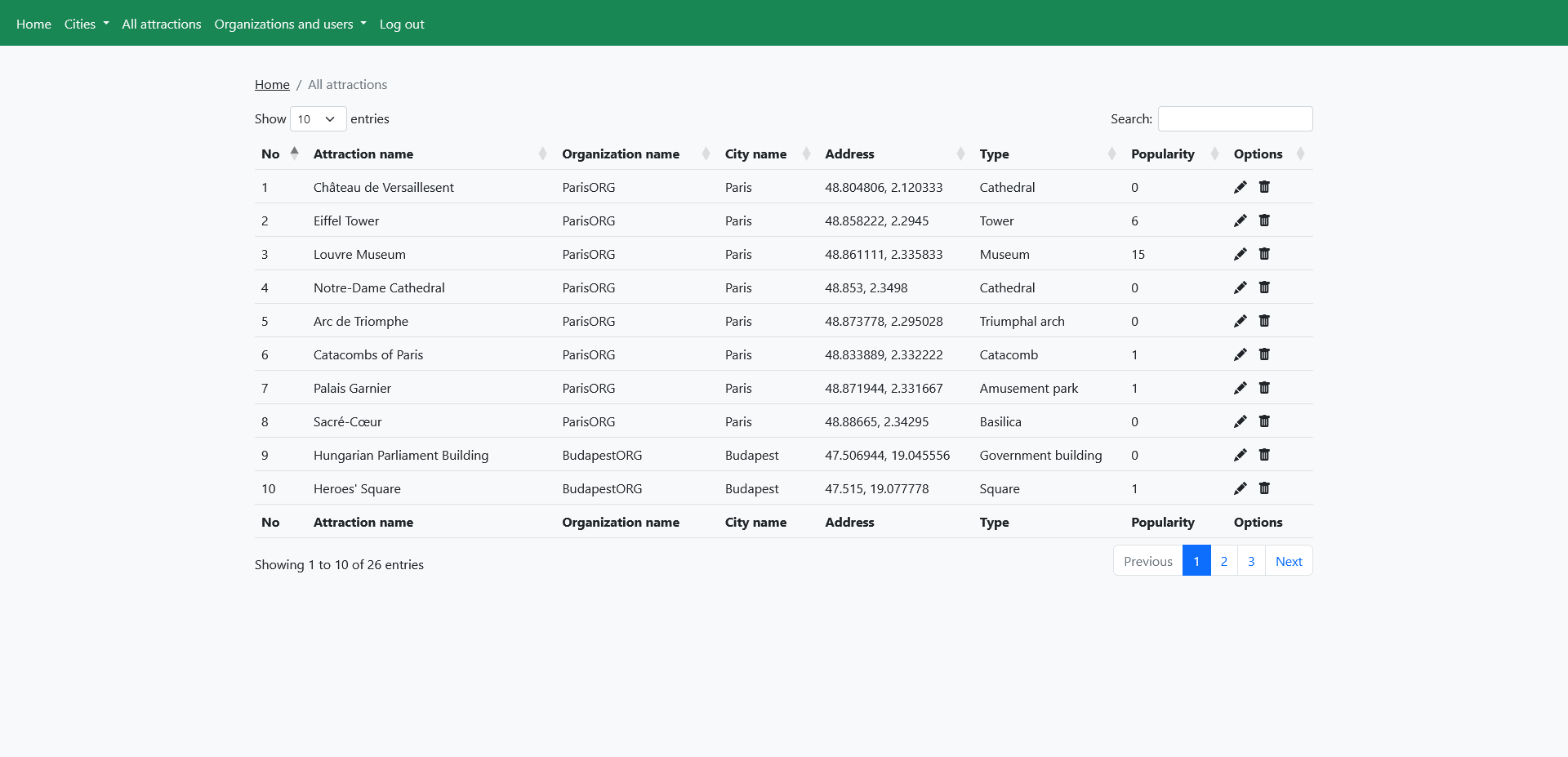
Ábra 22: Új város létrehozása



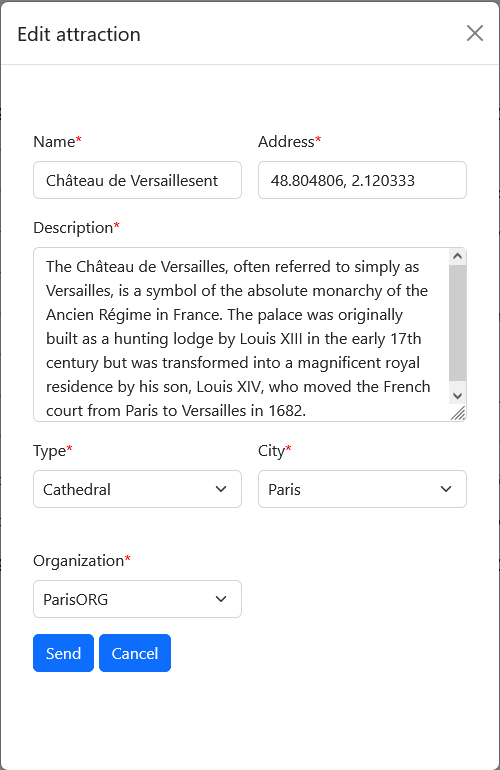
Ábra 23: Városok kilistázása



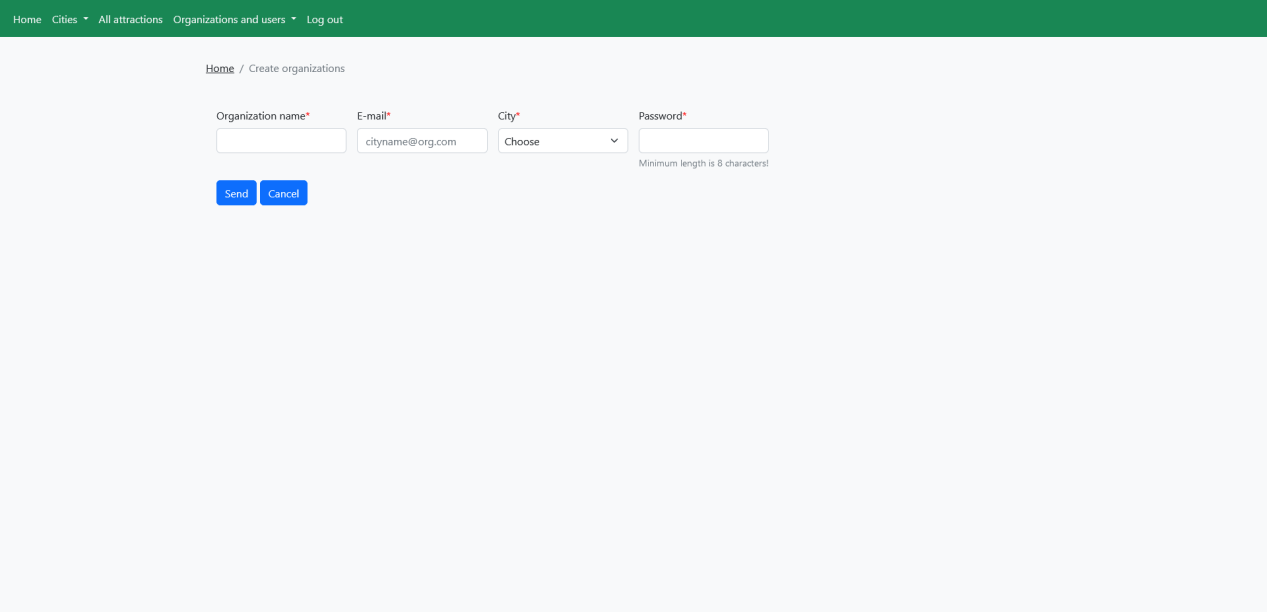
Ábra 24: Város változatása



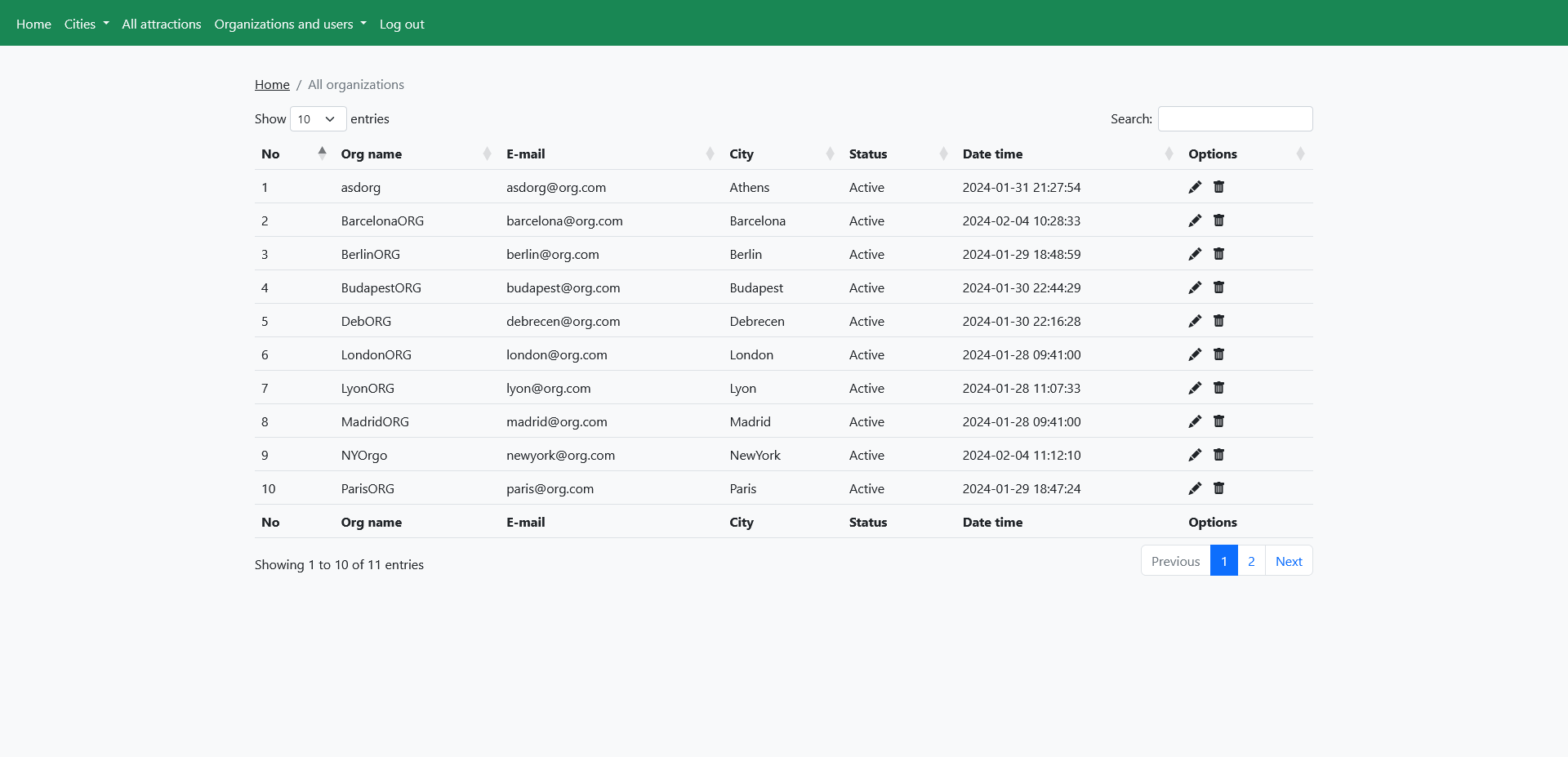
Ábra 25: Attrakciók kilistázása



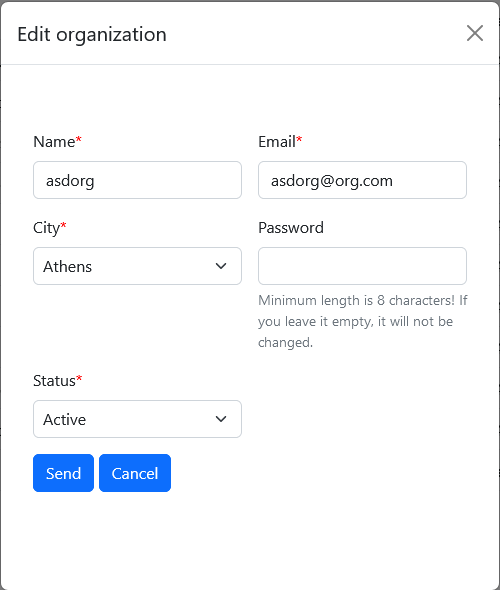
Ábra 26: Attrakció módosítása



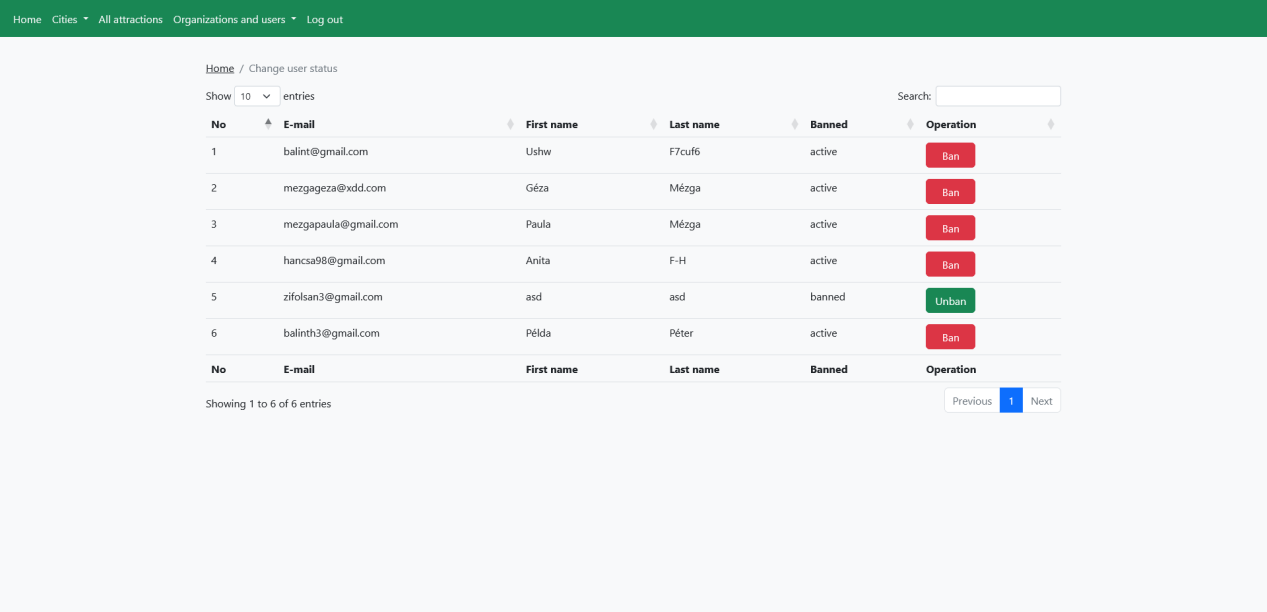
Ábra 27: Új organizáció készítése



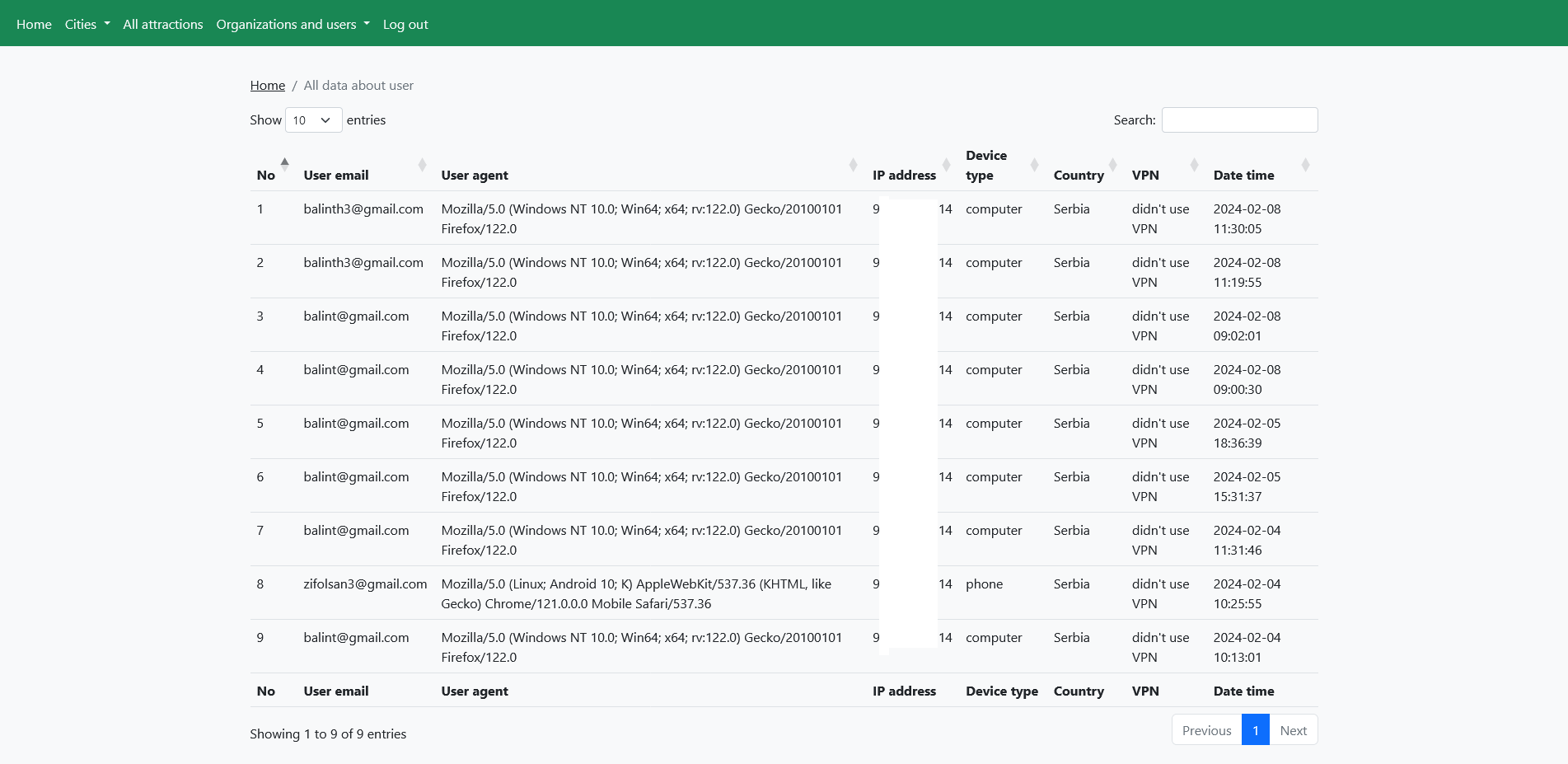
Ábra 28: Organizációk kilistázása



Ábra 29: Organizáció adatainak módosítása

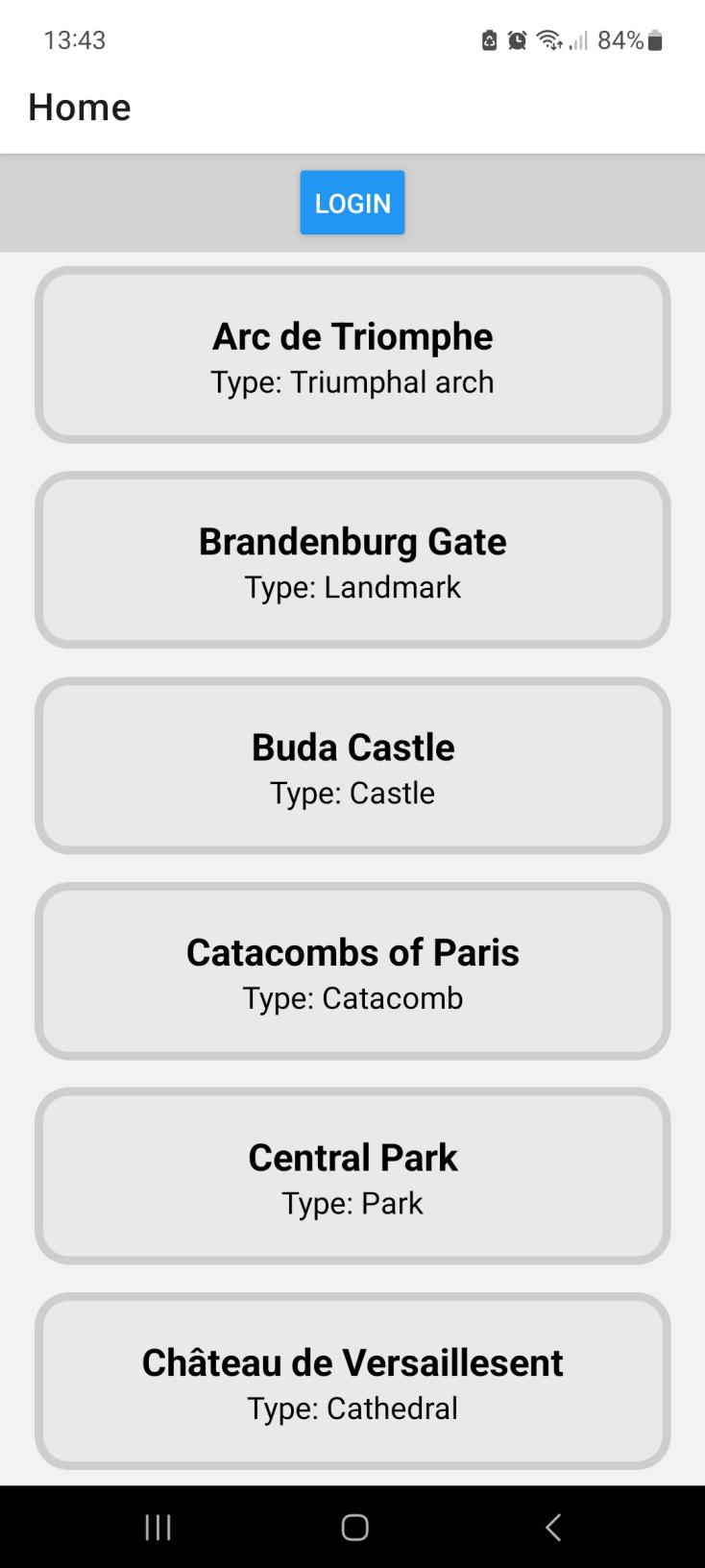


Ábra 30: Felhasználók tiltás/engedélyezés

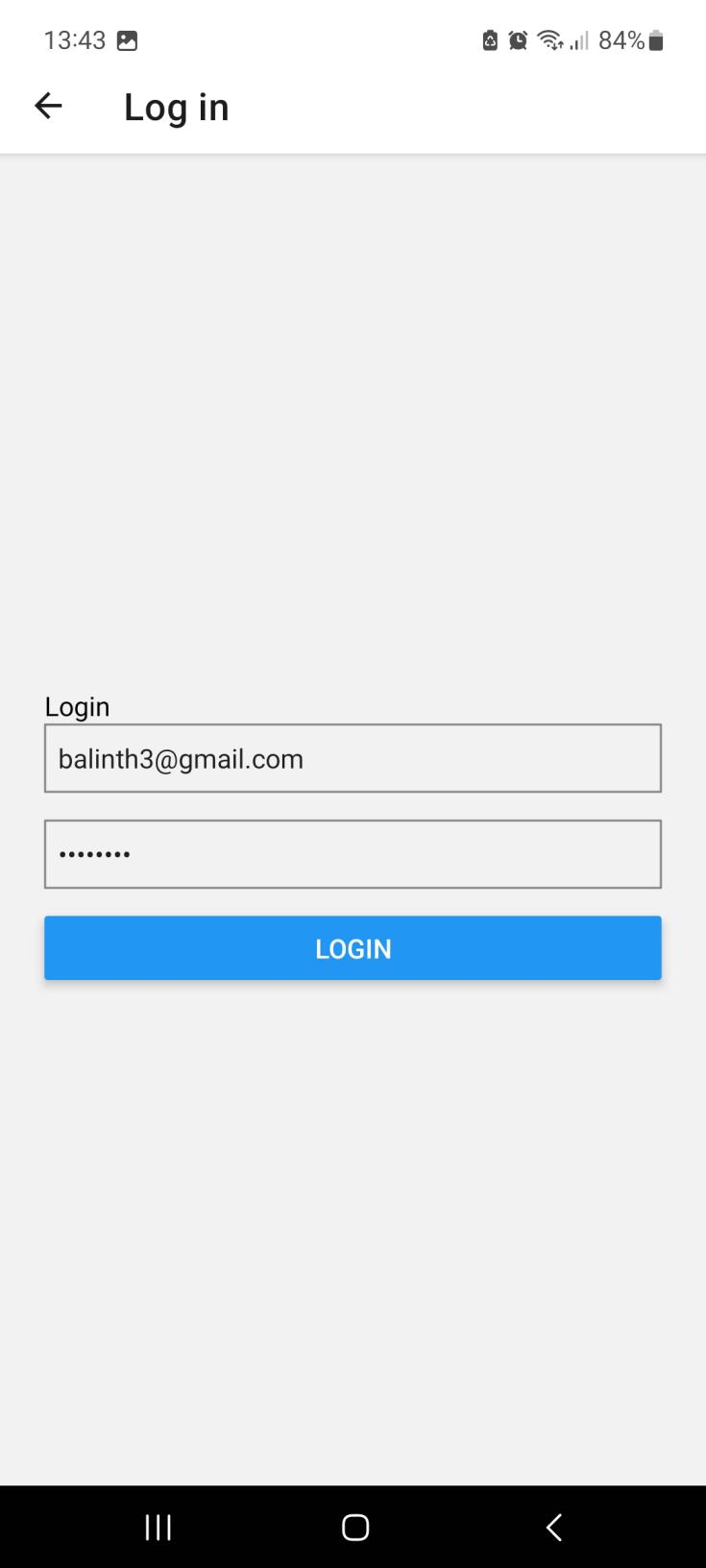


Ábra 31: Bejelentkezett felhasználók adatainak kiíratása

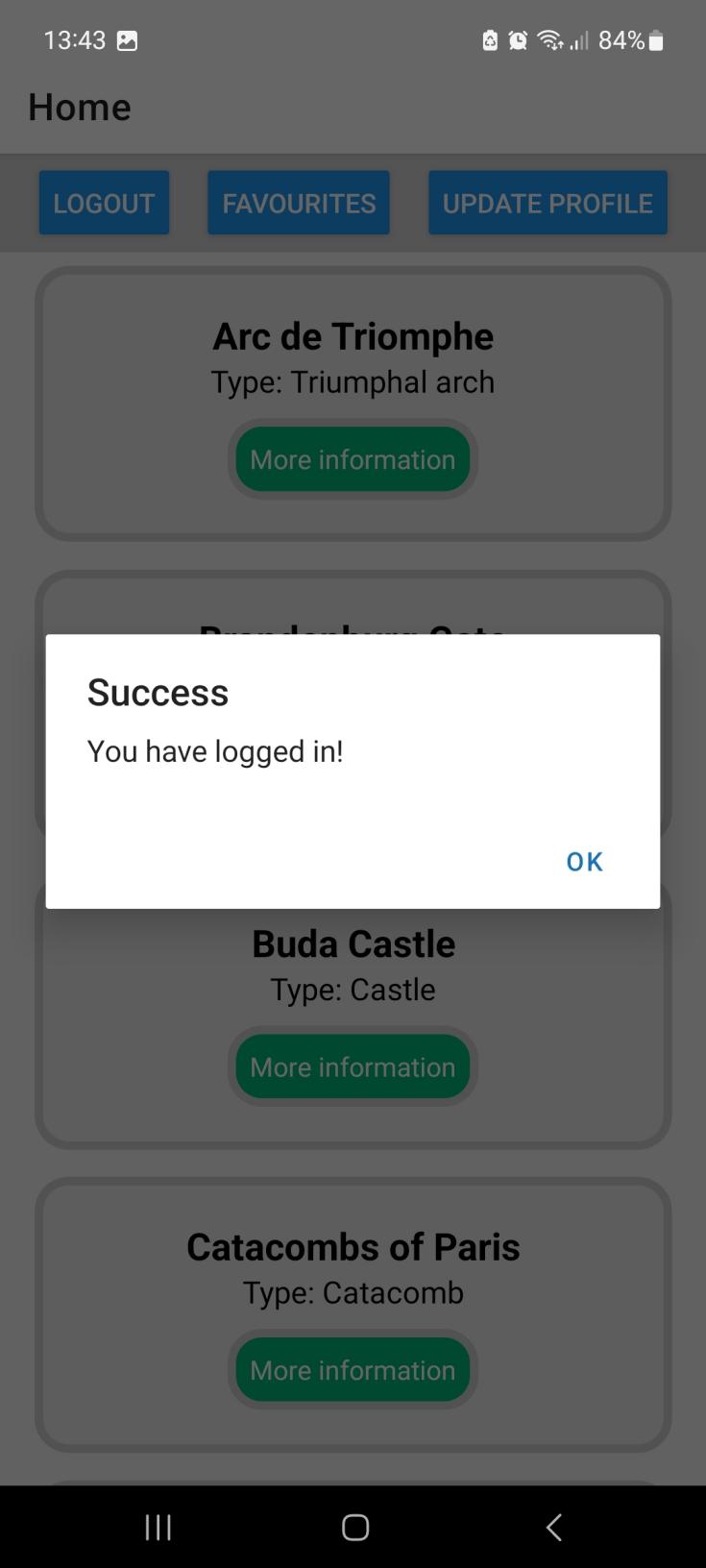
## Mobil alkalmazás



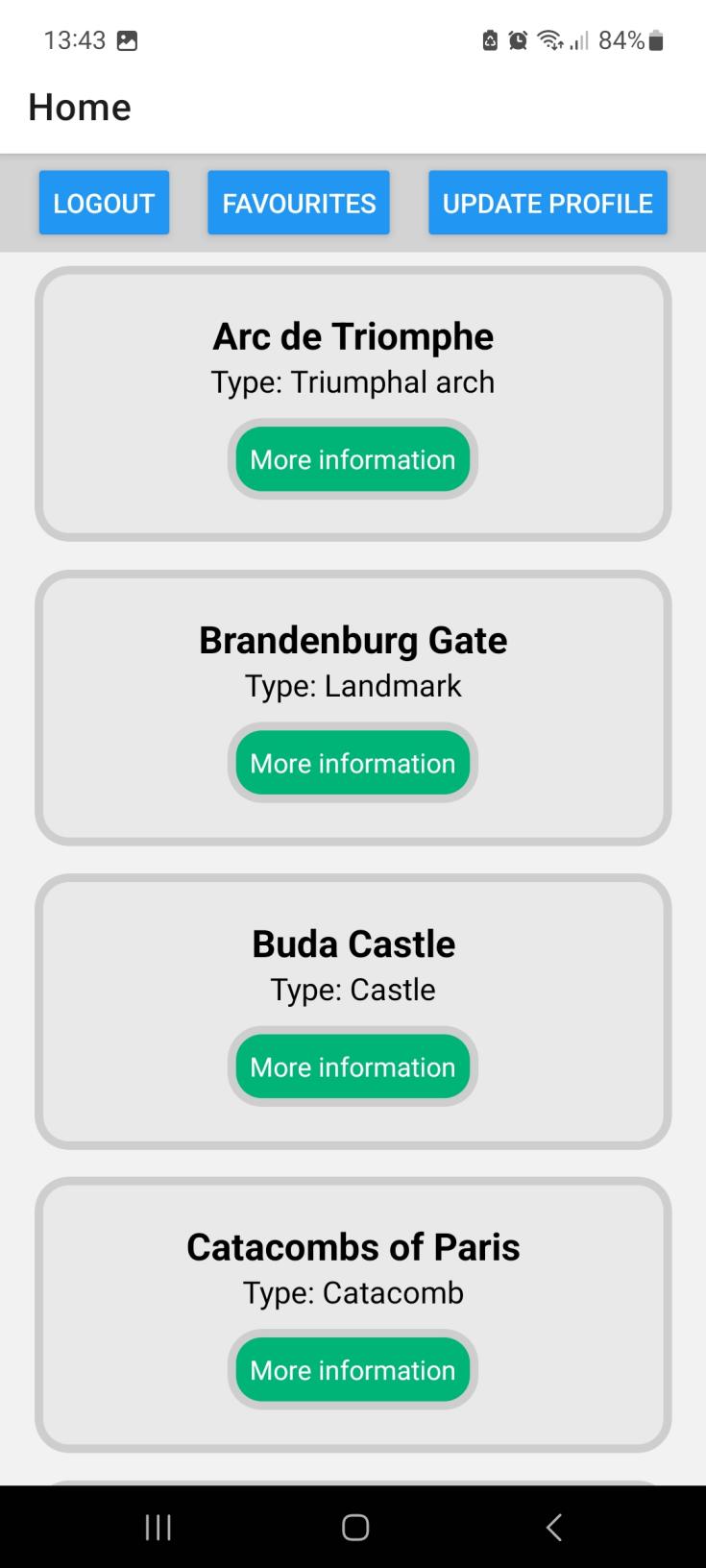
Ábra 32: Applikáció vendég nézet



Ábra 33: Bejelentkezési űrlap



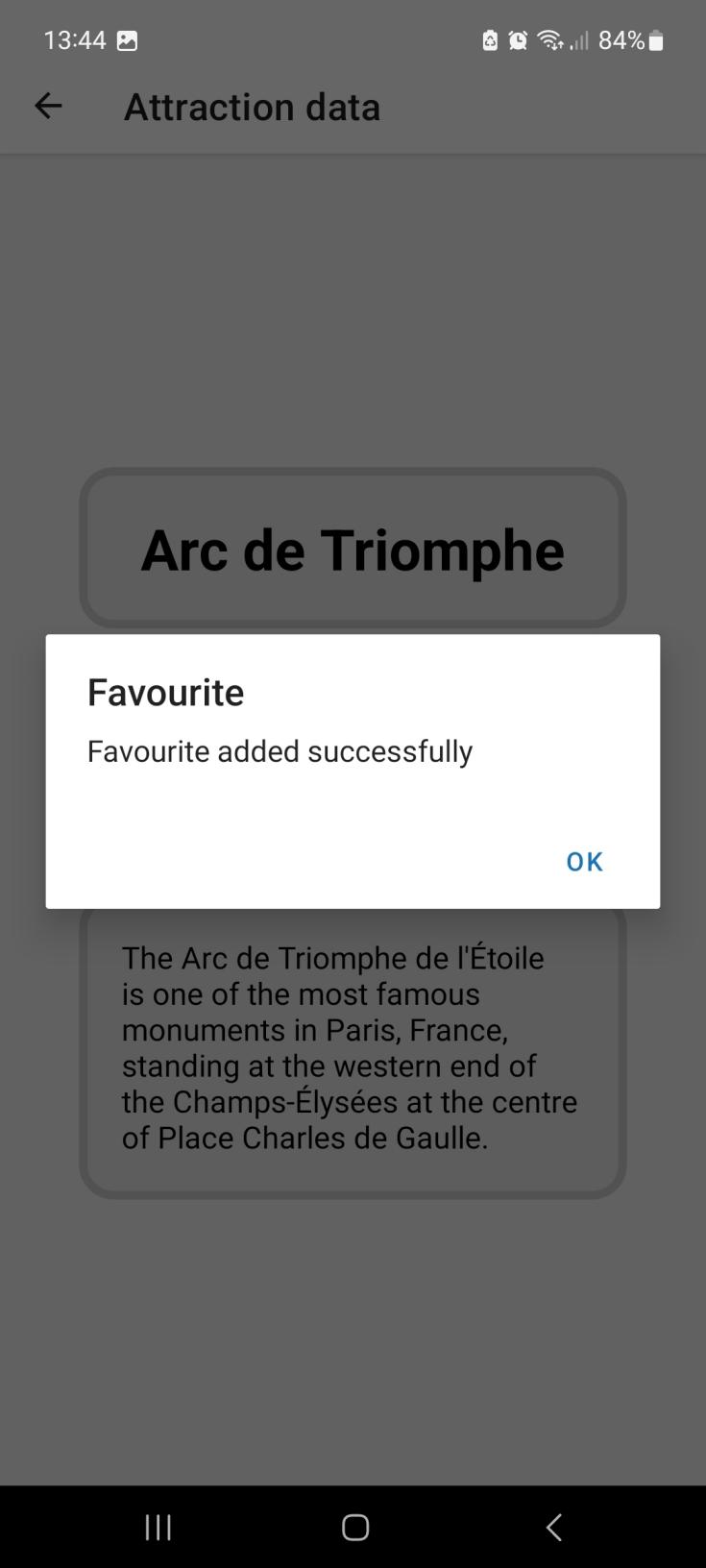
Ábra 34: Sikeres bejelentkezés



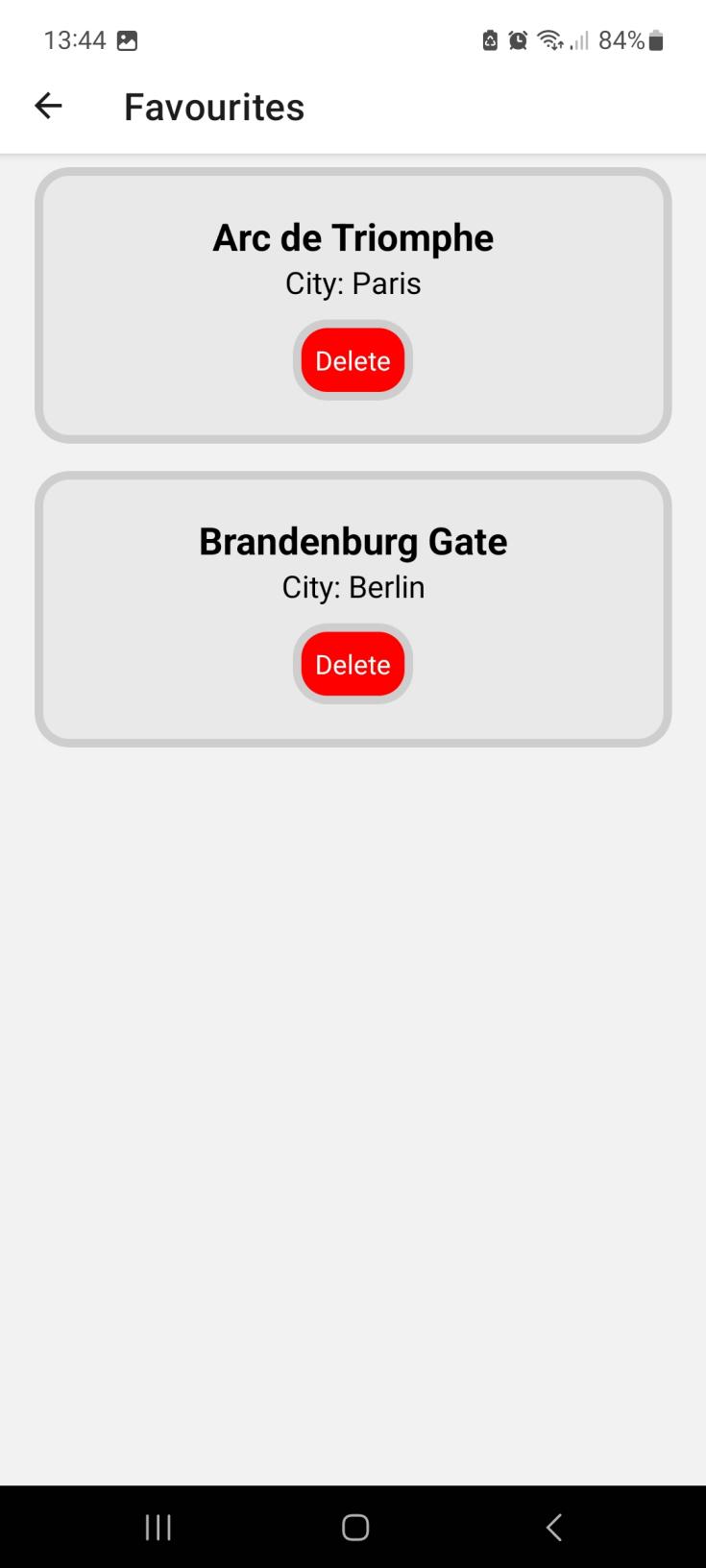
Ábra 35: Bővített attrakciók kiíratása



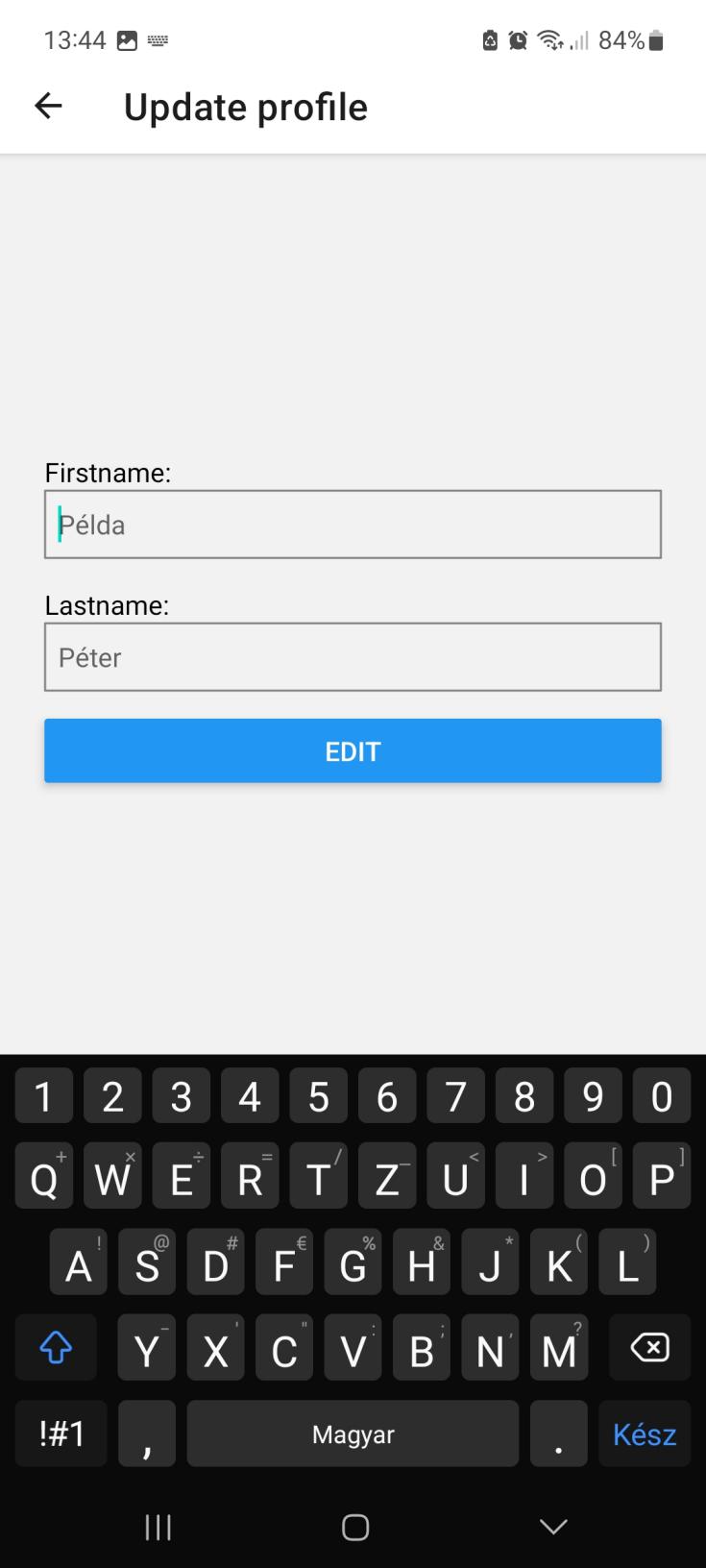
Ábra 36: Egy attrakció adatainak kiíratása



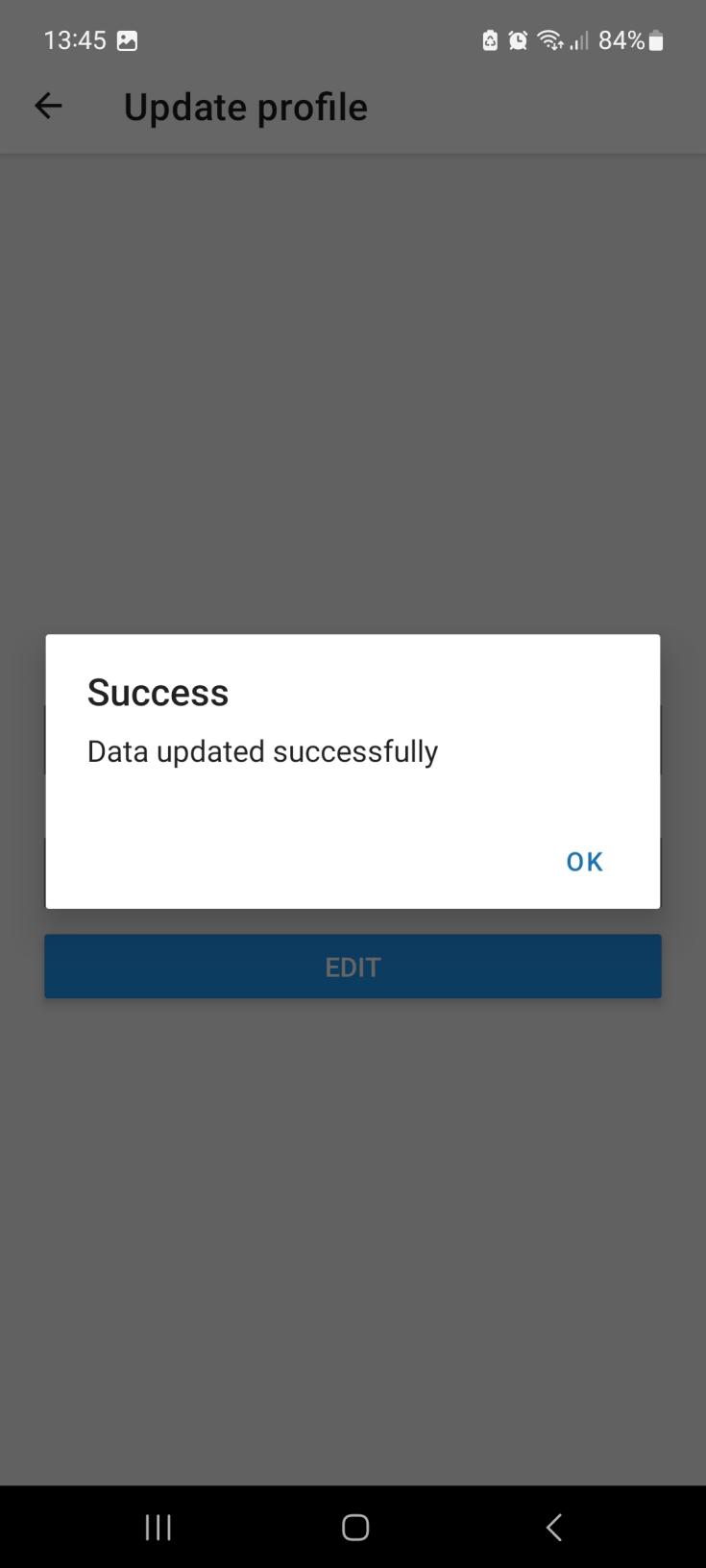
Ábra 37: Attrakció hozzáadása a kedvencekhez



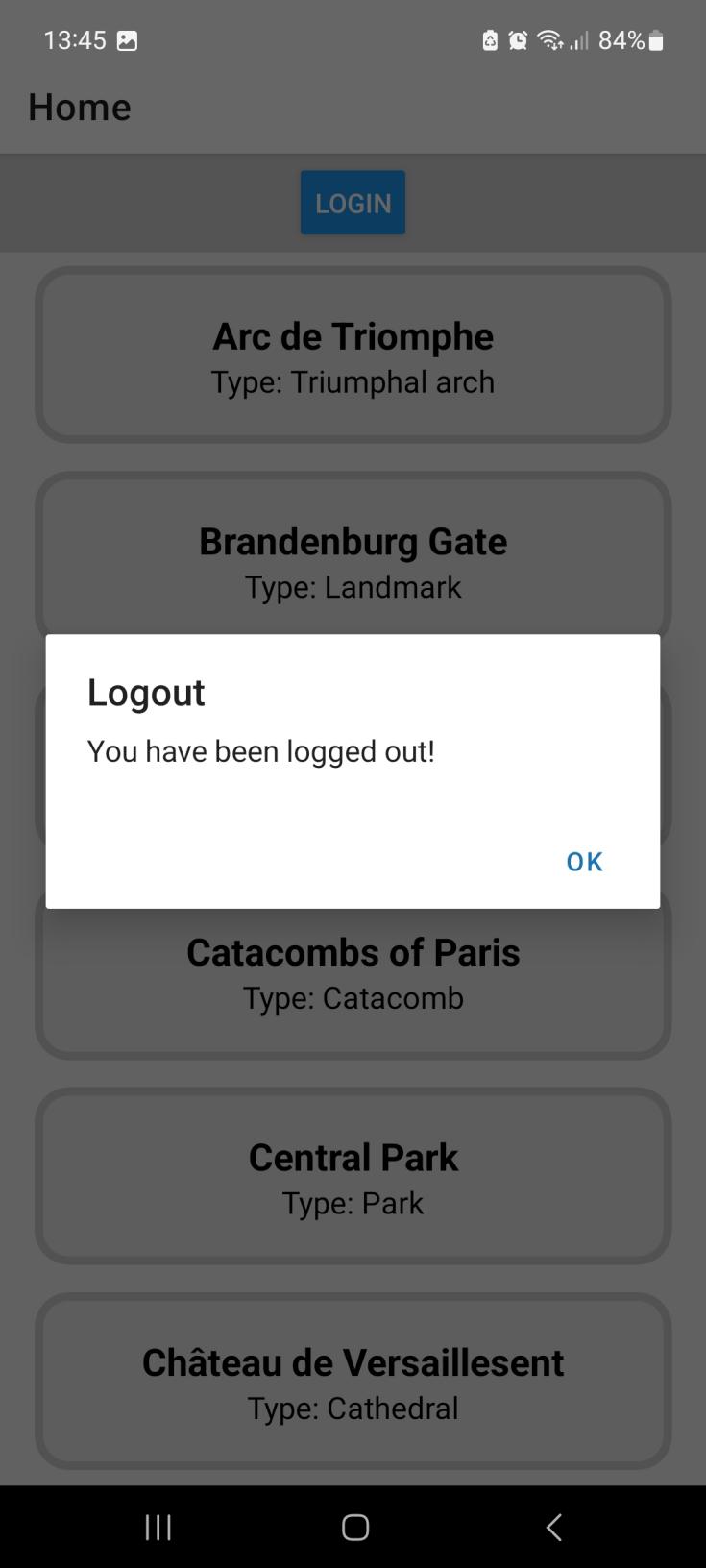
Ábra 38: Kedvencek kiíratása



Ábra 39: Adatok változtatása



Ábra 40: Sikeres változtatás



Ábra 41: Sikeres kijelentkezés

# FELHASZNÁLT KÖNYVTÁRAK

## Leaflet

A honlap készítése közben nem gondoltam, hogy fogok akármilyen külső féltől könyvtárat használni, de amikor a térképen akartam jelölni az attrakcióknak a honlétjüket, akkor bukkantam rá a Leaflet könyvtárra.

A Leaflet lehetővé teszi térképek integrálását és azon lévő helyek kitűzésének lehetőségét.

A következő kóddal lehet egy térképet megmutatni, ahol a ‘map’ az egy div-nek az id-je kell, hogy legyen.

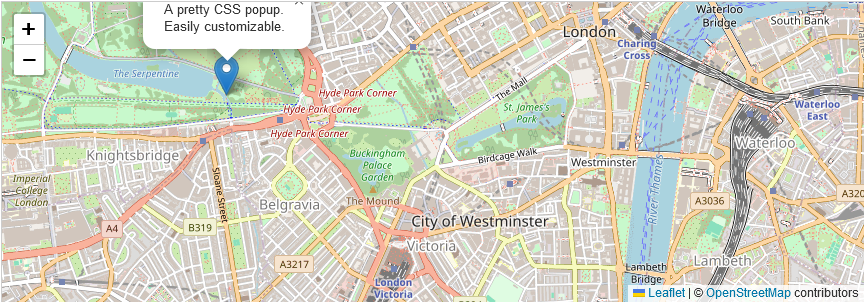
var map = L.map('map').setView([51.505, -0.09], 13);

A következő kód pedig lehetővé teszi a hely kitűzését:

L.marker([51.5, -0.09]).addTo(map)

.bindPopup('A pretty CSS popup.<br> Easily customizable.')

.openPopup();



Ábra 42: Leaflet példakép

# FELHASZNÁLT IRODALOM

W1. <https://reactnative.dev/docs/tutorial>

W2. <https://phptherightway.com/>

W3.<https://reactnavigation.org/docs/getting-started/>

W4.<https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started>

W5. <https://leafletjs.com/>