

Cáki pincesor

Szente Gábor

Szoftverfejlesztő

2024

Tartalom

1. Összefoglaló	3
2. Abstract	5
3. Implementációs leírás	6
4. Felhasználói dokumentáció.....	7
4.1 Rendszerkövetelmény	7
4.2 Ajánlott rendszerkövetelmények	7
4.3 Program megnyitása	7
4.4 Program használata.....	9
4.5 Admin felület.....	13
4.6 Összefoglaló	17
5. Fejlesztői dokumentáció	18
5.1 Adatbázis	18
5.2 Adatbázis táblák	18
5.2.1 Admin	18
5.2.2 Comments	19
5.2.3 Events	20
5.2.4 Gallery	22
5.2.5 Subscribers	22
5.2.6 Tickets	23
5.2.7 Visits	24
6. Fontosabb kódrészletek	26
7. Fejlesztési lehetőségek.....	33
8. Tesztdokumentáció	34
9. Irodalomjegyzék	37

1. Összefoglaló

A záródolgozat témája:

A program célja, hogy bemutassa a cáki pincesor történetét, különböző borászati folyamatokat, a helyi kultúrát és hagyományokat, valamint lehetőséget nyújtana interaktív élményekre és szórakoztató tevékenységekre , valamint a cáki pincesor különlegességeinek és vonzerejének megismertetésére a felhasználókkal.

Miért választottam?

A cáki pincesorral kapcsolatos projekt lehetőséget kínál számomra arra, hogy összehangoljam szakmai tudásomat és érdeklődésemet egy valós kihívással.

Személyes affinitásom van a borászat, a kultúra és a turizmus iránt, így a cáki pincesorral kapcsolatos weboldal készítése számomra inspiráló és motiváló.

Cákon olyat terveznek, ami a cáki pincesor különlegességeinek megismertetésére szolgálna. Cák polgármestere, és a képviselő-testület úgy döntött, hogy egy olyan weboldalt szeretnének létrehozni, ami bemutatja Cák különleges borospince sorait. Megkeresték nővéremet aki cáki lakos , hogy ismer-e valakit, aki elkészítené a weboldalt. Így hát rám hárult a feladat, hogy megtervezzem az oldalt. A cáki pincesorral kapcsolatos weboldal elkészítése kihívást jelent számomra, és lehetőséget ad arra, hogy megmutassam kreatív és technikai képességeimet egy érdekes és vonzó témában.

Köszönetnyilvánítás

Szeretném ezúton is megköszönni tanárainnak azt a támogatást és iránymutatást, amit a záródolgozatom elkészítése során kaptam. Külön köszönet illeti Kovács László Bálint tanár urat a cáki pincesorral kapcsolatos weboldal témaválasztásom elfogadásáért és támogatásáért. A tanárain bizalmának köszönhetően izgalmas és kihívásokkal teli lehetőségem nyílt arra, hogy elmélyedjek ebben a témában, és kreatív módon fejezzem ki a tanultakat. Hálás vagyok azért is, hogy mindig rendelkezésemre álltak, hogy válaszoljanak kérdéseimre és segítsenek megoldani az esetleges nehézségeket. A tanárain tanácsai és észrevételei nagyban hozzájárultak a munkám minőségéhez és a fejlődésemhez.

Köszönöm még egyszer az egész tanári csapatnak a támogatást és a lehetőséget, hogy ezt a projektet véghez vigyem.

2. Abstract

A program célja a cáki pincesor népszerűsítése és a látogatói élmény javítása a digitális térben. A weboldal lehetővé teszi a felhasználók számára, hogy virtuálisan felfedezzék a cáki pincsort, megismerjék történetét, borászati folyamatait és kulturális örökségét. A weboldal együttműködik az adatbázissal, amelyben tárolja a borokkal és a pincesorral kapcsolatos információkat.

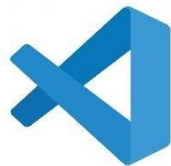
3. Implementációs leírás

Windows 11



A fejlesztéshez Windows 11 operációs rendszert használtam, mivel ez a rendszer kényelmes és jól kezelhető fejlesztői felületet biztosít.

Visual Studio Code



A kódszerkesztéshez és projektmenedzseléshez a Visual Studio Code-ot (VSCode) választottam. Ez egy könnyűsúlyú, de erős kódszerkesztő, amely sok funkcióval és kiterjesztéssel rendelkezik a fejlesztői produktivitás növelése érdekében.

Mysql



Az adatbázisok kezeléséhez MySQL-t használtam, mivel ez egy népszerű és jól dokumentált relációs adatbázis-kezelő rendszer, amely hatékonyan kezeli a nagy adatmennyiséget.

Webkiszolgáló



A fejlesztés során saját szerveren dolgoztam, amelyet a fejlesztési folyamat könnyű és gyors tesztelése érdekében állítottam fel. A saját szerver lehetőséget adott arra, hogy valós környezetben teszteljem a weboldalt, mielőtt azt éles környezetbe helyeztem volna.

4. Felhasználói dokumentáció

4.1 Rendszerkövetelmény

- Operációs rendszer: Bármelyik modern operációs rendszer támogatott, beleértve Windows 10 vagy újabb verzióit, macOS-t, valamint Linuxot.
- Processzor: Legalább 1 GHz-es vagy annál gyorsabb processzor.
- Memória (RAM): Legalább 2 GB RAM.
- Tárhely: Legalább 100 MB szabad tárhely a telepítéshez és az adatbázisnak.
- Böngésző: Támogatott legyen a legújabb verziója a modern böngészőknek, mint például a Google Chrome, Mozilla Firefox vagy Microsoft Edge.

4.2 Ajánlott rendszerkövetelmények

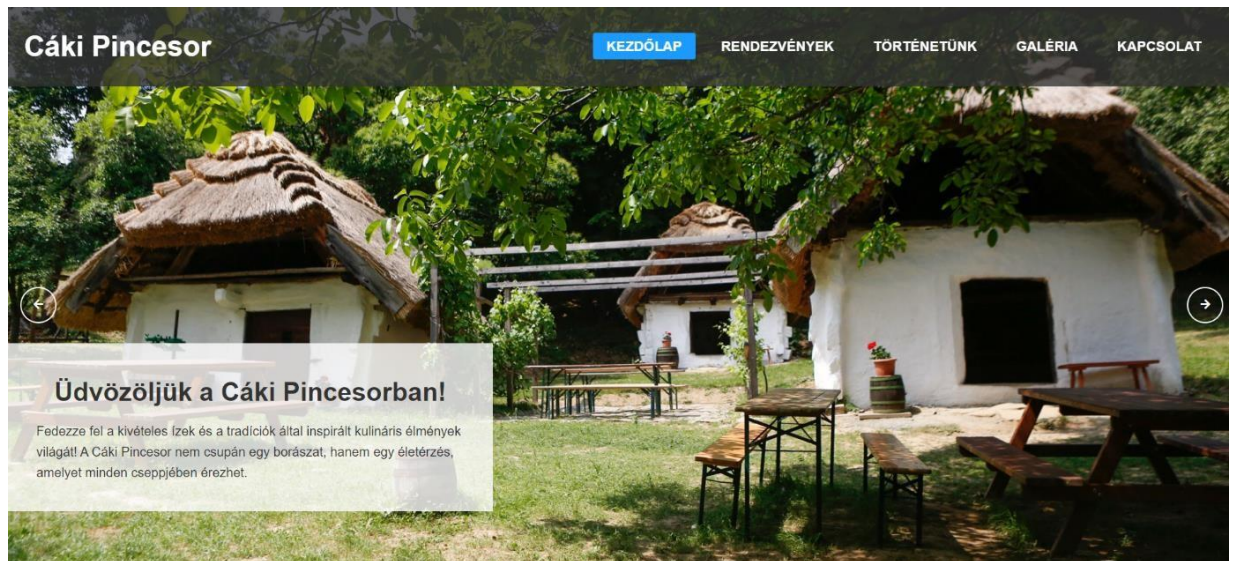
- Operációs rendszer: Windows 11, vagy a legújabb macOS vagy Linux disztribúció.
- Processzor: Dual-core vagy quad-core processzor 2 GHz vagy annál gyorsabb órajellel.
- Memória (RAM): Legalább 4 GB RAM.
- Tárhely: Legalább 500 MB szabad tárhely a telepítéshez és az adatbázisnak.
- Böngésző: A legújabb verziója a Google Chrome, Mozilla Firefox vagy Microsoft Edge böngészőknek.

4.3 Program megnyitása

- Számítógép bekapcsolása: Kezdjük azzal, hogy bekapcsoljuk a számítógépet. Ezt általában egy gomb lenyomásával vagy az energiaforrás csatlakoztatásával lehet elérni. A számítógép bekapcsolásakor hallani fogjuk a ventilátorok zaját és az esetleges rendszerhangokat.

- Operációs rendszer betöltése: Miután bekapcsoltuk a számítógépet, az operációs rendszer betöltődik. Ez lehet például Windows, macOS vagy Linux.
- Böngésző megnyitása: Az operációs rendszer betöltődése után keresd meg a számítógépen a "Böngésző" nevű ikont vagy programot. Ez általában egy színes kerek ikon lehet, amelyet kétszeri kattintással nyithatsz meg.
- Weboldal cím beírása: Amikor a böngésző megnyílik, egy üres lapot fogsz látni. Ebben a lapban meg kell adnod a weboldal címét, amely a programhoz tartozik. Ez a **<https://www.cakipincesor.hu>** .
- Enter lenyomása: Miután beírtad a weboldal címét a böngésző címsorába, nyomd meg az "Enter" billentyűt. Ezután a böngésző elkezd betölteni és megjeleníteni a weboldalt.

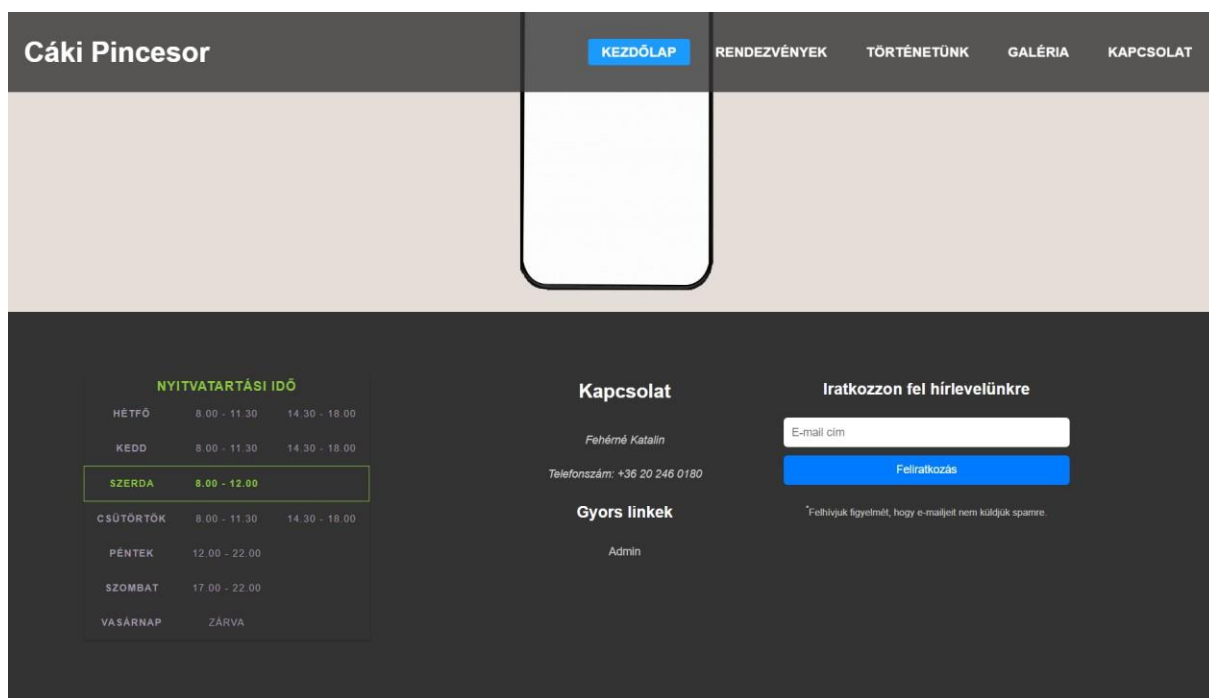
4.4 Program használata



Az oldal betöltődése után a felhasználó egy gyönyörű és hangulatos képpel találkozik, amelyen a cák pincesor látványát láthatja. Az alábbi menüpontokat találjuk:

- Kezdőlap
- Rendezvények
- Történetünk
- Galéria
- Kapcsolat

A Kezdőoldal alján elhelyezett feliratkozási lehetőség révén a látogatók könnyen értesülhetnek a legfrissebb hírekről és eseményekről, és lehetőséget biztosít arra, hogy aktívan részt vehessenek a cák pincesor életében és programjaiban. Emellett az e-mail értesítések segíthetnek abban is, hogy a látogatók időben értesüljenek a közelgő eseményekről, így könnyebben tervezhetik meg látogatásukat és részvételüket. Ezen felül találunk egy nyitvatartási időt ami az aktuális napnak a nyitva tartását jelzi nekünk.



A **Rendezvények** menüpont bemutatja a jelenlegi és közelgő rendezvényeket a cák pincesoron. A látogatók itt tájékozódhatnak a helyszínen tartott különféle eseményekről, mint például borfesztiválok, borkóstolók, koncertek vagy gasztronómiai rendezvények.

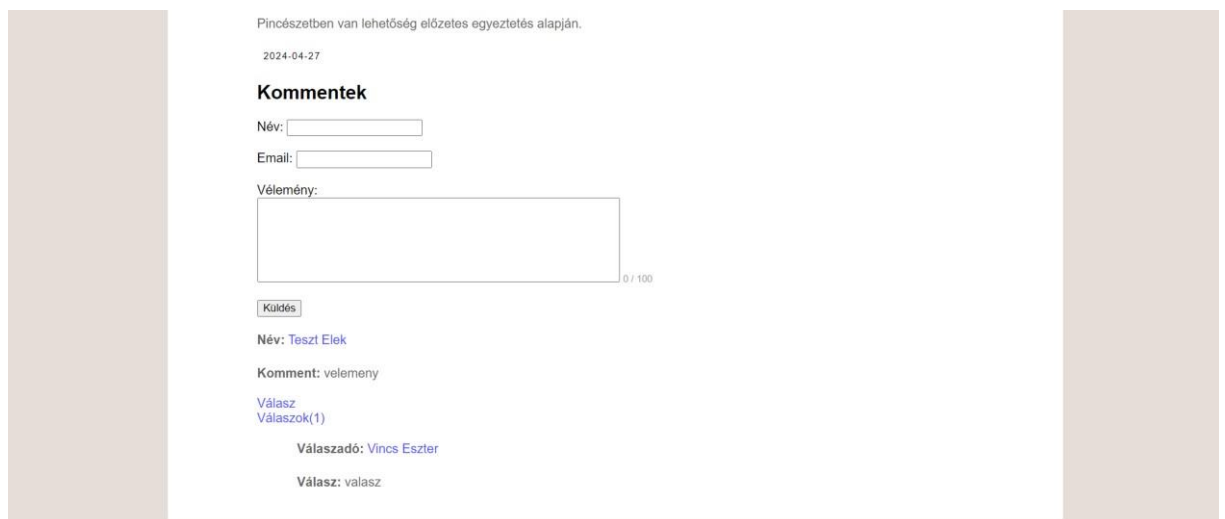


Természetesen, az **Rendezvények** oldalon a "Kommentek és Válaszok" szekció lehetőséget kínál a látogatóknak arra, hogy megosszák véleményüket és élményeiket az adott rendezvényekkel kapcsolatban, valamint lehetőséget nyújt a szervezőknek vagy más látogatóknak, hogy válaszoljanak ezekre a kommentekre.

Kommentek: A látogatók itt hozzászólhatnak és megoszthatják tapasztalataikat az adott rendezvényről. Lehetőség van arra, hogy kifejezzék a rendezvényen szerzett élményeiket, megosszák pozitív vagy negatív visszajelzéseiket. A felhasználók nevére kattintva megjelenik, hogy milyen kommenteket írtak, valamint melyik rendezvényekhez kapcsolódtak ezek a hozzászólások.

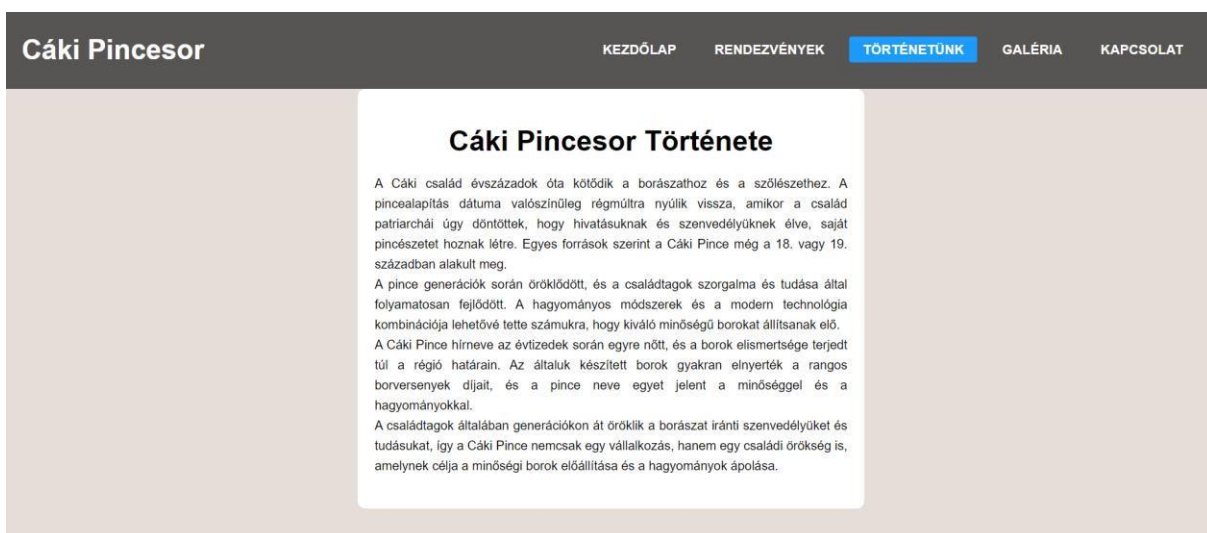
Válaszok: A "Válaszok" rész lehetőséget kínál a szervezőknek vagy más látogatóknak, hogy válaszoljanak a beérkezett kommentekre. Ez segíthet a szervezőknek az esetleges kérdések tisztázásában, az elégedetlen vendégek megnyugtatózásában vagy észrevételeik kezelésében. Ezenkívül a látogatók közötti párbeszéd elősegíti a közösségépítést és az eseményekről kialakuló pozitív tapasztalatok megosztását.

Fontos megjegyezni, hogy a felhasználóknak nem kell regisztrálniuk az oldalon a kommentek vagy válaszok írásához. Ez a rugalmasság és a könnyű hozzáférhetőség elősegítheti a látogatók részvételét és hozzájárulhat az események sikeréhez.

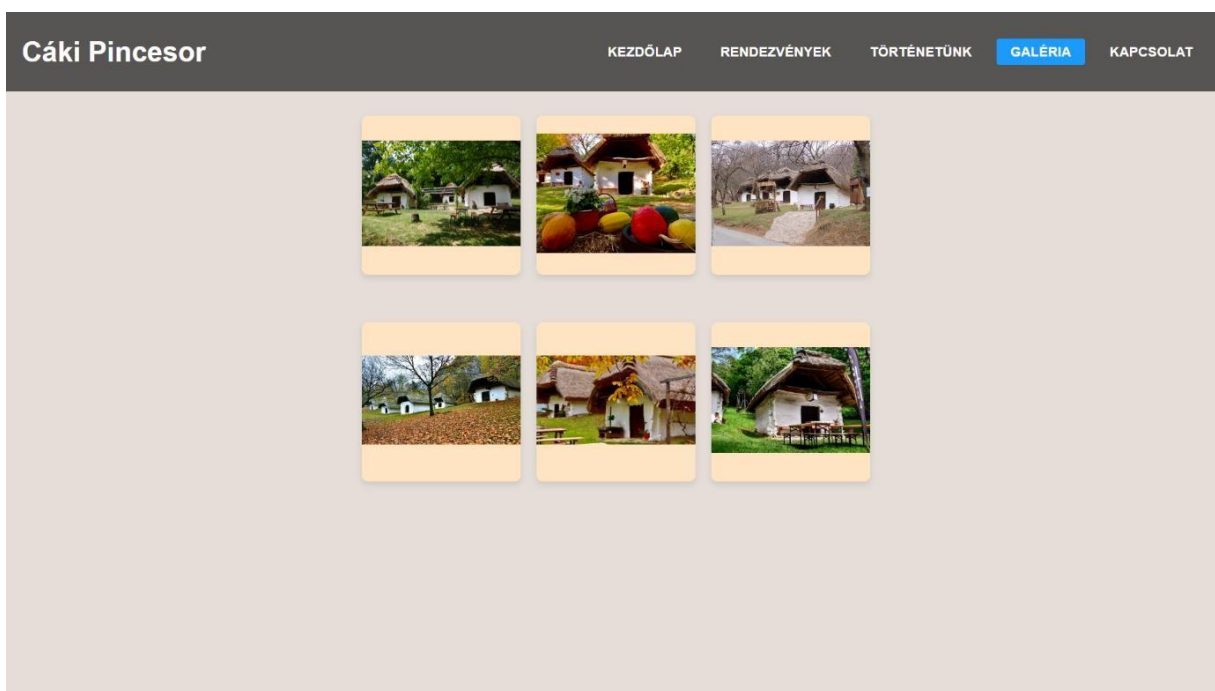


The screenshot shows a web form for comments. At the top, it says "Pincészetben van lehetőség előzetes egyeztetés alapján." followed by the date "2024-04-27". The section is titled "Kommentek". Below this, there are input fields for "Név:" (Name) and "Email:", followed by a larger text area for "Vélemény:". To the right of the text area is a character count "0 / 100". Below the text area is a "Küldés" (Send) button. Underneath the button, it shows "Név: Teszt Elek" and "Komment: vélemény". Below this, there is a "Válasz" (Reply) section with "Válaszok(1)" (1 reply). The reply is from "Válaszadó: Vincs Eszter" and the text is "Válasz: válasz".

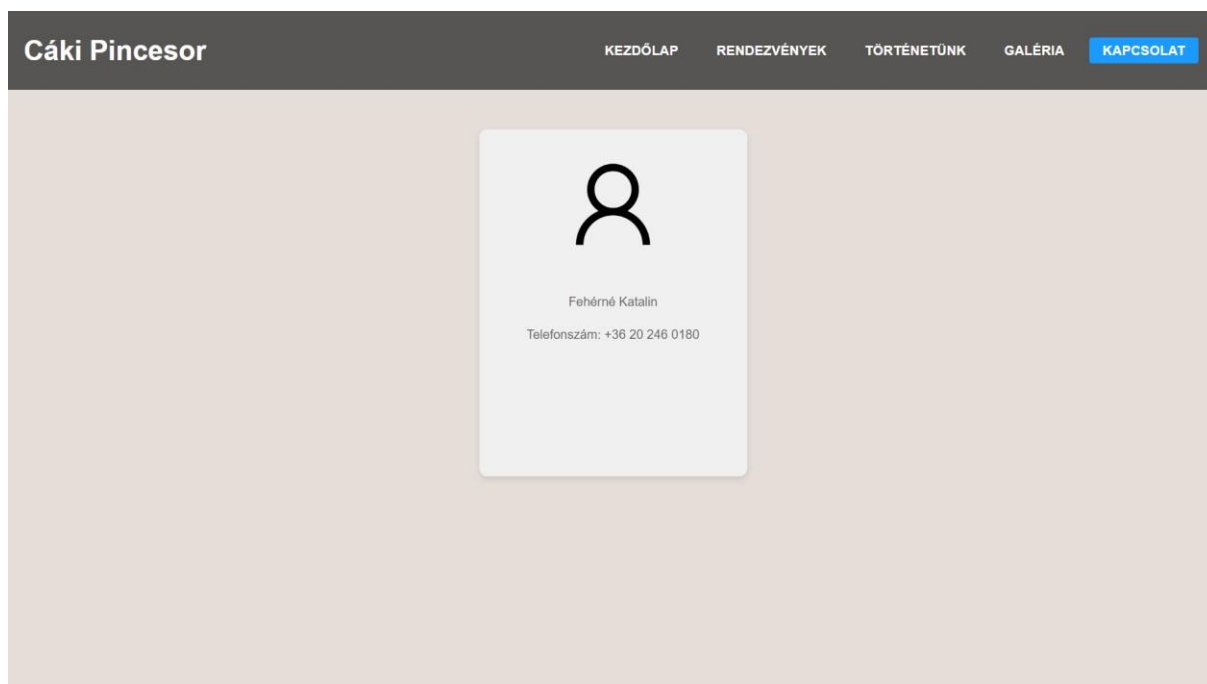
A **Történetünk** menüpont részletesen bemutatja a cági pincesor történetét és hagyományait. A látogatók megtudhatják a pincesor alapításának és fejlődésének történetét, valamint információkat kapnak a helyi borászati tradíciókról és kultúráról.



A **Galéria** menüpont gyűjteményt tartalmaz képekről a cáki pincesorról, a helyi borokról és a turisztikai látnivalókról. A galéria lehetőséget kínál a látogatóknak, hogy közelebbről megismerjék a pincesor hangulatát és szépségét.



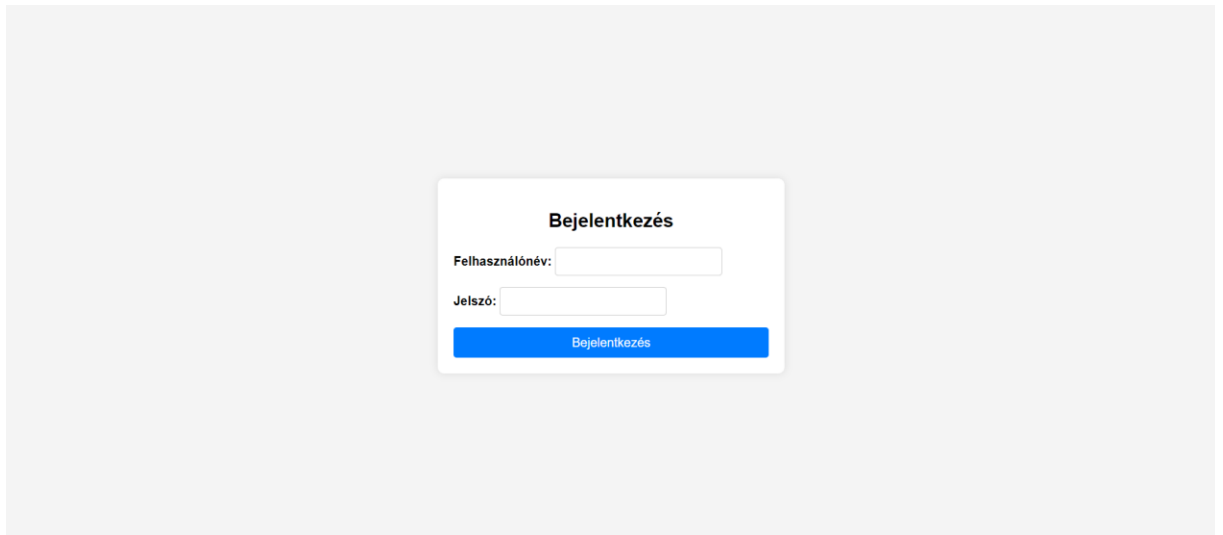
A **Kapcsolat** menüpont lehetőséget nyújt a látogatóknak arra, hogy kapcsolatba lépjenek a cáki pincesor képviselőivel. Itt találhatóak az elérhetőségek, például e-mail címek, telefonszámok amelyeken keresztül a látogatók érdeklődhetnek, visszajelzést adhatnak vagy foglalást tehetnek.



4.5 Admin felület

Az admin felületre való bejelentkezés általában egy biztonságos bejelentkező ablakon vagy űrlapon keresztül történik, mely csak az adminisztrátoroknak elérhető. Az adminisztrátoroknak egyedi felhasználónevet és jelszót kell megadniuk a belépéshez, ami biztosítja a felhasználóazonosítást és az oldal biztonságát.

Emellett lehetőséget ad az adminisztrátoroknak arra, hogy hatékonyan felügyeljék és irányítsák az oldal tartalmát és működését, valamint biztosítsák annak zökkenőmentes és biztonságos működését a felhasználók számára.



Az admin felület számos fontos funkciót kínál az oldal üzemeltetőinek és adminisztrátorainak.

Az alábbi menüpontokkal szolgál:

- Kezdőlap
- Esemény hozzáadása
- Jegyek
- Admin hozzáadása
- Kijelentkezés

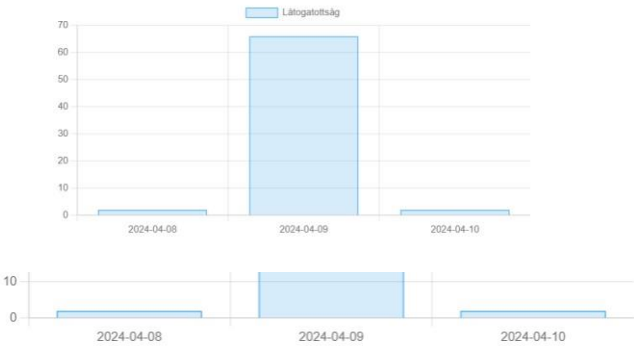
Kezdőlap: Ez az oldal tartalmazza az aktuális rendezvényekkel kapcsolatos legfontosabb információkat, mint például az eladott jegyek számát, a látogatottságot és a kapott kommenteket. Ez az áttekintő oldal lehetővé teszi az adminisztrátoroknak, hogy gyorsan és könnyen nyomon kövessék az oldal teljesítményét és a felhasználók visszajelzéseit

Üdvözöllek, admin!

Látogatottság: 69

ID Esemény neve Ár Darabszám Dátum
Nincs találat a jegyekre.

Oldal neve Látogatók száma
Fiók 2
Kapcsolat 12
Galéria 18
Történetünk 33
Kezdő oldal 61



Legfrissebb kommentek

- [Vér Tamás](#) írta a(z) [Cáki-pince túra és műzeális borok kóstolója](#) eseményhez: 2024-04-09 23:28:19
- [Horváth Ilonka](#) írta a(z) [Cáki-pince túra és műzeális borok kóstolója](#) eseményhez: 2024-04-09 23:25:31
- [Vicc Elek](#) írta a(z) [Cáki-pince túra és műzeális borok kóstolója](#) eseményhez: 2024-04-09 23:23:25
- [frommywindow](#) írta a(z) [Kenyerűnnep a Căki Pincesoron](#) eseményhez: 2024-04-09 23:03:59
- [Zuhany Rózsika](#) írta a(z) [Kenyerűnnep a Căki Pincesoron](#) eseményhez: 2024-04-09 23:01:31

Esemény hozzáadása: Ezen az oldalon az adminisztrátorok új rendezvényeket vagy eseményeket hozhatnak létre az oldalon. Itt adhatják meg az esemény nevét, dátumát, helyszínét, részleteit és más fontos információkat, valamint feltölthetik az eseményhez kapcsolódó képeket vagy videókat.

Új Esemény Létrehozása

Esemény Neve:

Esemény Dátuma:
éééé, hh, nn, □

Leírás:

Kép feltöltése:
Fájl kiválasztása Nincs fájl kiválasztva

Kategória	Ár
Felnőtt	500
Diák	300
Nyugdíjas	300
Családi	1000
Csoportos	300
Mozgássérült (+1 kísérő)	Ingyenes

Létrehozás

Jegyek: Ez az oldal lehetővé teszi az adminisztrátoroknak, hogy hozzáadják az eladott jegyeket az aktuális rendezvényekhez.

Kezdőlap
Esemény hozzáadása
Jegyek
Admin hozzáadása
Kijelentkezés

Jegyek rögzítése

Esemény:
Válassz eseményt
Jegyek darabszáma:
Befolyt összeg:
Mentés

Admin hozzáadása: Ezen az oldalon az adminisztrátorok új adminisztrátorokat adhatnak hozzá az oldalhoz. Ez lehetővé teszi az adminisztrátori jogokkal rendelkező személyek körének bővítését, így könnyebben kezelhetik az oldal tartalmát és felhasználóit.

Kezdőlap
Esemény hozzáadása
Jegyek
Admin hozzáadása
Kijelentkezés

Új admin hozzáadása

Felhasználónév:
Jelszó:
E-mail cím:
Hozzáadás

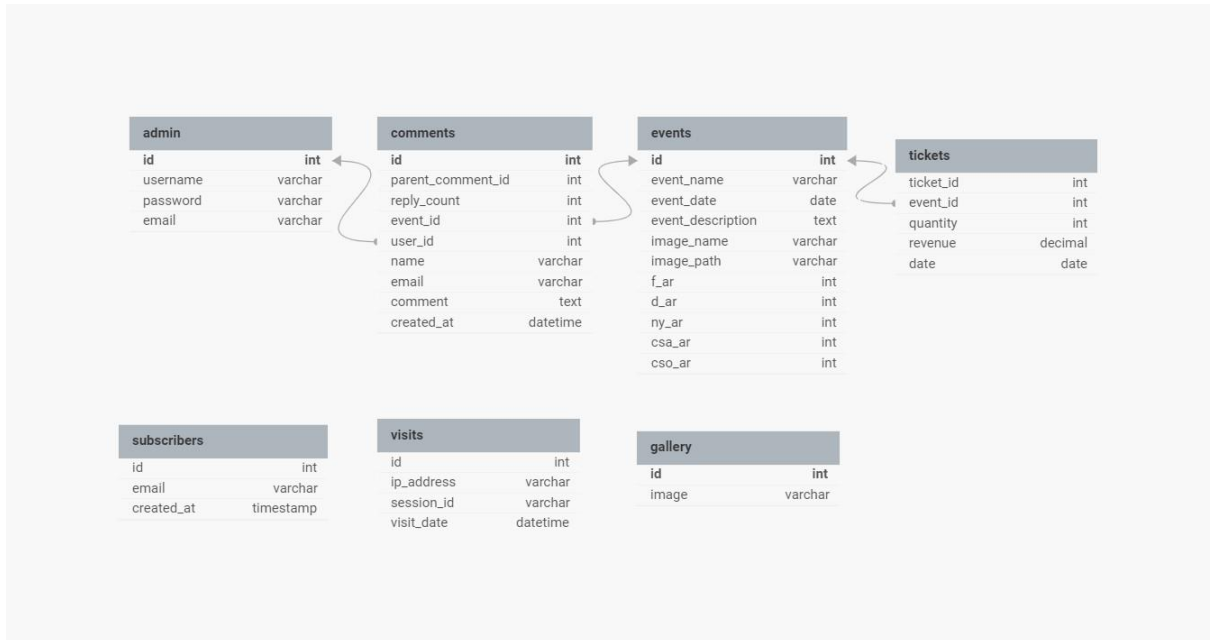
Majd a **Kijelentkezés:** Ez a lehetőség lehetővé teszi az adminisztrátoroknak, hogy biztonságosan kilépjenek az admin felületről, így védelmet biztosítva az illetéktelen hozzáférések ellen.

4.6 Összefoglaló

Remélem tetszett a program és elégedett vele. Ha bármi észrevétele van a programmal kapcsolatban, kérem, vegye fel velünk a kapcsolatot az alábbi email címen: gaborszente14@gmail.com vagy a következő telefon számon: 06 20 936 6766.

5. Fejlesztői dokumentáció

5.1 Adatbázis



5.2 Adatbázis táblák

5.2.1 Admin

Admin táblában az adminisztrátorok adatai vannak el tárolva.

admin	
id	
username	
password	
email	

- **id:** Ez egy egyedi azonosító mező, ami az adminisztrátor azonosítására szolgál. Egy számérték, amely minden egyes adminisztrátorhoz egyedi.
- **username:** Ez a mező tárolja az adminisztrátor felhasználónevét. A felhasználónévnek egyedinek kell lennie minden adminisztrátor esetében.

- **password:** Ez a mező tárolja az adminisztrátor jelszavát. Fontos megjegyezni, hogy az itt tárolt jelszavak hashelve vannak a biztonság érdekében, mint például a `password_hash()` függvény segítségével PHP-ben.
- **email:** Ez a mező tárolja az adminisztrátor e-mail címét. Ez a mező lehetővé teszi az adminisztrátorokkal való kommunikációt és az értesítéseket.

5.2.2 Comments

A **comments** tábla az eseményekhez írt kommenteket és válaszokat tárolja el.

comments
id
parent_comment_id
reply_count
event_id
user_id
name
email
comment
created_at

- **id:** Ez egy egyedi azonosító mező, ami minden egyes kommentnek egyedi azonosítót ad.
- **parent_comment_id:** Ez a mező lehetővé teszi a hierarchikus kommentek kezelését. Ha egy válasz van egy másik kommentre, akkor ennek a mezőnek az értéke az adott komment azonosítója lesz. Ha az adott komment az első szinten van (nincs válasz másik kommentre), akkor ez a mező értéke null lesz.

- **reply_count:** Ez a mező tárolja az adott kommentre érkezett válaszok számát. Ez segíthet az adminisztrátoroknak az interakció mértékének nyomon követésében.
- **event_id:** Ez a mező azonosítja az adott kommenthez kapcsolódó eseményt vagy rendezvényt.
- **user_id:** Ez a mező tárolja a kommentelő felhasználójának azonosítóját. Ha a felhasználó regisztrált felhasználó, akkor ez az azonosító lehet a felhasználó egyedi azonosítója. Ha a felhasználó vendég felhasználó (nem regisztrált), akkor lehetőség van arra, hogy ezt az értéket üresen hagyjuk vagy null értékkel adjuk meg.
- **name:** Ez a mező tárolja a kommentelő nevét.
- **email:** Ez a mező tárolja a kommentelő e-mail címét.
- **comment:** Ez a mező tárolja magát a kommentet, amit a felhasználó megosztott.
- **created_at:** Ez a mező tárolja a komment létrehozásának időpontját, vagyis azt az időpontot, amikor a kommentet beküldték.

5.2.3 Events

Az **events** tábla az oldalon megjelenő események adatainak tárolására szolgál.

events
id
event_name
event_date
event_description
image_name
image_path
f_ar
d_ar
ny_ar
csa_ar
csa_ar

- **id:** Ez egy egyedi azonosító mező, ami minden egyes eseménynek egyedi azonosítót ad.
- **event_name:** Ez a mező tárolja az esemény nevét.
- **event_date:** Ez a mező tárolja az esemény dátumát.
- **event_description:** Ez a mező tárolja az esemény leírását vagy részleteit. Ide tartoznak a programok, a részletes információk, az időpontok és bármilyen más releváns információk az adott eseménnyel kapcsolatban.
- **image_name:** Ez a mező tárolja az eseményhez tartozó kép nevét vagy fájlnevét.
- **image_path:** Ez a mező tárolja az eseményhez tartozó kép elérési útját vagy elérési útvonalát a szerveren.
- **f_ar:** Ez a mező tárolja a "Felnőtt" típusú jegy árát az adott eseményhez.
- **d_ar:** Ez a mező tárolja a "Diák" típusú jegy árát az adott eseményhez.
- **ny_ar:** Ez a mező tárolja a "Nyugdíjas" típusú jegy árát az adott eseményhez.

- **csa_ar:** Ez a mező tárolja a "Családi" típusú jegy árát az adott eseményhez.
- **cso_ar:** Ez a mező tárolja a "Csoportos" típusú jegy árát az adott eseményhez.

5.2.4 Gallery

A **gallery** táblában tárolom el azoknak a képeknek a nevét , amiket megjelenítek a galéria oldalon.

gallery
id
image

- **id:** Ez egy egyedi azonosító mező, ami minden egyes galériaképnek egyedi azonosítót ad.
- **image:** Ez a mező tárolja a galériaképek elérési útját vagy elérési útvonalát a szerveren. A képek általában fájlnévet tartalmaznak, amelyek az adott képet azonosítják a szerveren.

5.2.5 Subscribers

A **Subscribers** tábla a hírlevélre feliratkozott felhasználókat tárolja el.

subscribers
id
email
created_at

- **id:** Ez egy egyedi azonosító mező, ami minden egyes feliratkozott felhasználónak egyedi azonosítót ad.
- **email:** Ez a mező tárolja a feliratkozott felhasználó e-mail címét.
- **created_at:** Ez a mező tárolja a felhasználó feliratkozásának dátumát és időpontját. Alapértelmezetten a jelenlegi időpont kerül beállításra az aktuális időpillanatban.

5.2.6 Tickets

A **Tickets** táblában a jegyek eladási darabszámát és befolyt összegét tárolja el.

tickets
ticket_id
event_id
quantity
revenue
date

- **ticket_id:** Ez egy egyedi azonosító mező, ami minden egyes jegynek egyedi azonosítót ad.
- **event_id:** Ez a mező tárolja az adott jegyhez kapcsolódó esemény azonosítóját. Ez az azonosító összeköti a jegyet az adott eseménnyel az "events" táblában található azonosítóval.
- **quantity:** Ez a mező tárolja a jegyek darabszámát, amelyet egy adott eseményhez rendeltek.

- **revenue:** Ez a mező tárolja a jegyekből származó bevételeket. A bevételek általában tizedes pontossággal vannak megadva, és lehetővé teszik az összegzést vagy a pénzügyi elemzéseket.
- **date:** Ez a mező tárolja a jegyek értékesítési dátumát, vagyis azt az időpontot, amikor a jegyeket eladták.

5.2.7 Visits

A **visits** táblában a látogatók adatait tárolom el.

visits
id
ip_address
session_id
visit_date
page_visited

- **id:** Ez egy egyedi azonosító mező, ami minden egyes látogatásnak egyedi azonosítót ad.
- **ip_address:** Ez a mező tárolja az adott látogató IP-címét. Az IP-cím egyedileg azonosítja az interneten keresztül kapcsolódó eszközt vagy számítógépet.
- **session_id:** Ez a mező tárolja a látogatáshoz kapcsolódó munkamenet azonosítóját. A munkamenet azonosítója lehetővé teszi az adott látogatás összekapcsolását az oldal használata során létrehozott ideiglenes munkamenettel.
- **visit_date:** Ez a mező tárolja a látogatás dátumát és időpontját, vagyis azt az időpontot, amikor a látogató az oldalt meglátogatta.

- **page_visited:** Ez a mező tárolja az oldalon meglátogatott oldal elérési útját vagy elérési útvonalát. Ez lehetővé teszi az adminisztrátorok számára, hogy nyomon kövessék, hogy mely oldalakat látogatták meg a látogatók.

6. Fontosabb kódrészek

submit_comment.php – Kommentelésért felel

Ez a PHP szkript egy szerveroldali kód, amely egy webes űrlap POST kérésének feldolgozására szolgál. A szkript egy komment beküldési folyamatot valósít meg egy adatbázisban. Be mutatom a szkript részeit:

Adatbázis Kapcsolat Beállítása:

A szerver, felhasználónév, jelszó és adatbázis neve beállításra kerül, amelyek lehetővé teszik az adatbázis-szerverhez való csatlakozást.

Adatok Ellenőrzése és Tisztítása:

Ellenőrzi, hogy a kérés POST módszerrel érkezett-e.

A `clean_input` függvény segítségével megtisztítja a felhasználó által megadott adatokat (név, email, komment, esemény azonosító), hogy megelőzze az SQL injekciót és más biztonsági kockázatokat.

Adatbázis Kapcsolat Létrehozása:

Új MySQL-i objektumot hoz létre az adatbázis kapcsolat kezelésére.

Ha a kapcsolat sikertelen, a szkript leáll egy hibaüzenettel.

Felhasználó Ellenőrzése:

Lekérdezi az adatbázist, hogy van-e már a megadott email címmel felhasználó.

Ha van, lekéri a felhasználó azonosítóját.

Ha nincs, generál egy véletlenszerű felhasználó azonosítót a `generateRandomUserId` függvény segítségével.

Komment Beszúrása:

Előkészíti az INSERT SQL parancsot, hogy beszúrja az új kommentet az adatbázisba.

Ha a beszúrás sikeres, átirányítja a felhasználót egy másik oldalra.

Ha nem sikerül, kiír egy hibaüzenetet.

Függvények Definíciója:

A `clean_input` függvény eltávolítja a felesleges szóközöket, a backslash karaktereket és konvertálja a speciális karaktereket HTML entitásokká.

A `generateRandomUserId` függvény egyedi felhasználó azonosítót generál, amíg nem talál olyat, ami még nincs az adatbázisban.

A szkript biztonsági intézkedéseket tartalmaz, mint például az adatok tisztítása és a prepared statements használata, amelyek segítenek megelőzni az SQL injekciót. Az adatok érzékeny természete miatt fontos lenne a jelszavak biztonságos kezelése, például környezeti változókból való tárolással vagy konfigurációs fájlokban, amelyek nem kerülnek verziókezelés alá.

```

1 if ($_SERVER["REQUEST_METHOD"] == "POST") {
2     $name = clean_input($_POST["name"]);
3     $email = clean_input($_POST["email"]);
4     $comment = clean_input($_POST["comment"]);
5     $event_id = clean_input($_POST["event_id"]);
6     $created_at = date('Y-m-d H:i:s');
7     $conn = new mysqli($servername, $username, $password, $dbname);
8     if ($conn->connect_error) {
9         die("Connection failed: " . $conn->connect_error);
10    }
11    $check_user_query = "SELECT user_id FROM comments WHERE email=?";
12    $stmt = $conn->prepare($check_user_query);
13    $stmt->bind_param("s", $email);
14    $stmt->execute();
15    $stmt->store_result();
16    if ($stmt->num_rows > 0) {
17        $stmt->bind_result($user_id);
18        $stmt->fetch();
19    } else {
20        $user_id = generateRandomUserId($conn);
21    }
22    $insert_query = "INSERT INTO comments (user_id, name, email, comment, event_id, created_at) VALUES (?, ?, ?, ?, ?, ?)";
23    $stmt = $conn->prepare($insert_query);
24    $stmt->bind_param("ssssis", $user_id, $name, $email, $comment, $event_id, $created_at);
25
26    if ($stmt->execute()) {
27        header("Location: ../../events.php");
28        exit();
29    } else {
30        echo "Hiba történt a komment beküldése közben: " . $stmt->error;
31    }
32    $stmt->close();
33    $conn->close();
34 } else {
35     echo "Hiba történt az adatok feldolgozása közben.";
36 }
37 function clean_input($data) {
38     $data = trim($data);
39     $data = stripslashes($data);
40     $data = htmlspecialchars($data);
41     return $data;
42 }
43 function generateRandomUserId($conn) {
44     $unique = false;
45     $user_id = '';
46     while (!$unique) {
47         $user_id = bin2hex(random_bytes(16));
48         $check_user_id_query = "SELECT * FROM comments WHERE user_id=?";
49         $stmt = $conn->prepare($check_user_id_query);
50         $stmt->bind_param("s", $user_id);
51         $stmt->execute();
52         $stmt->store_result();
53         if ($stmt->num_rows == 0) {
54             $unique = true;
55         }
56     }
57     return $user_id;
58 }

```

latogato.php - látogatási adatokat rögzítő szkript

Session Kezdeté:

A `session_start()` függvény meghívásával elindít egy új munkamenetet vagy folytat egy meglévőt. A munkamenetek lehetővé teszik az adatok tárolását és azok hozzáférhetőségét a weboldal összes oldalán.

Látogatás Ellenőrzése:

Ellenőrzi, hogy a felhasználó látogatása már rögzítésre került-e a munkamenetben (`$_SESSION['visited']`). Ha még nem, akkor folytatódik a feldolgozás.

Látogatás Adatainak Gyűjtése:

A `session_id()` függvény segítségével lekéri a munkamenet azonosítót.

A `date()` függvény segítségével rögzíti a látogatás dátumát és időpontját.

A `basename()` és `$_SERVER['PHP_SELF']` segítségével meghatározza, hogy melyik oldalt látogatta meg a felhasználó.

A `$_SERVER['REMOTE_ADDR']` segítségével lekéri a felhasználó IP címét.

Adatbázis Ellenőrzés:

Egy SQL lekérdezést hajt végre, hogy ellenőrizze, van-e már rögzítve a látogatás az adatbázisban a munkamenet azonosító alapján.

Új Látogatás Rögzítése:

Ha az adott munkamenet azonosítóval még nem történt látogatás rögzítés, akkor egy új bejegyzést illeszt be az adatbázisba az IP cím, munkamenet azonosító, látogatás dátuma és a meglátogatott oldal információival.

Munkamenet Frissítése:

Ha már rögzítésre került a látogatás a munkamenetben, akkor egy másik ellenőrzést hajt végre, hogy az adott oldal látogatása rögzítve lett-e már az adott munkamenet azonosítóval.

Ha nem, akkor hozzáad egy új bejegyzést az adatbázisba hasonló adatokkal, mint az előző pontban.

Ez a kód segít nyomon követni a felhasználók viselkedését az oldalon, beleértve azt is, hogy mely oldalakat látogatták meg és mikor. Ez hasznos lehet a weboldal forgalmának elemzésére, a felhasználói élmény javítására, vagy akár biztonsági célokra is.

```
1 <?php
2 session_start();
3 require_once 'admin/assets/php/conn.php';
4 if (empty($_SESSION['visited'])) {
5     $session_id = session_id();
6     $visit_date = date('Y-m-d H:i:s');
7     $page_visited = basename($_SERVER['PHP_SELF']);
8     $ip_address = $_SERVER['REMOTE_ADDR'];
9     $sql_check = "SELECT * FROM visits WHERE session_id = ?";
10    $stmt_check = $conn->prepare($sql_check);
11    $stmt_check->bind_param("s", $session_id);
12    $stmt_check->execute();
13    $result = $stmt_check->get_result();
14    if ($result->num_rows == 0) {
15        $sql_insert = "INSERT INTO visits (ip_address, session_id, visit_date, page_visited) VALUES (?, ?, ?, ?)";
16        $stmt_insert = $conn->prepare($sql_insert);
17        $stmt_insert->bind_param("ssss", $ip_address, $session_id, $visit_date, $page_visited);
18        $stmt_insert->execute();
19        $_SESSION['visited'] = true;
20    }
21 }
22 else {
23     $session_id = session_id();
24     $visit_date = date('Y-m-d H:i:s');
25     $page_visited = basename($_SERVER['PHP_SELF']);
26     $ip_address = $_SERVER['REMOTE_ADDR'];
27     $sql_check_page = "SELECT * FROM visits WHERE session_id = ? AND page_visited = ?";
28     $stmt_check_page = $conn->prepare($sql_check_page);
29     $stmt_check_page->bind_param("ss", $session_id, $page_visited);
30     $stmt_check_page->execute();
31     $result_page = $stmt_check_page->get_result();
32     if ($result_page->num_rows == 0) {
33         $sql_insert_page = "INSERT INTO visits (ip_address, session_id, visit_date, page_visited) VALUES (?, ?, ?, ?)";
34         $stmt_insert_page = $conn->prepare($sql_insert_page);
35         $stmt_insert_page->bind_param("ssss", $ip_address, $session_id, $visit_date, $page_visited);
36         $stmt_insert_page->execute();
37     }
38 }
39 ?>
```

index.php - Esemény leírásának rövidítés

Az \$event_description változóba elmenti az esemény leírását, amelyet valószínűleg egy adatbázisból vagy más forrásból kapott.

Az \$max_chars változóba elmenti a maximális karakterhosszt, amelyet a rövidített leírásnak el kell érnie.

Ellenőrzi, hogy az \$event_description hossza meghaladja-e a \$max_chars értéket.

Ha az \$event_description hossza meghaladja a \$max_chars értéket, akkor a szöveget szavakra bontja a explode() függvény segítségével.

Egy ciklus segítségével végig iterál a szavakon, és hozzáadja őket a \$display_text változóhoz, amíg a karakterek száma el nem éri a \$max_chars értéket.

A rövidített leírást a \$display_text változó tartalmával és egy "..." jellel kiírja.

Ha az \$event_description hossza nem haladja meg a \$max_chars értéket, akkor az eredeti leírást kiírja.

Ez a kód hasznos, ha egy hosszabb szöveget szeretnénk röviden megjeleníteni, például egy esemény leírását. A kód meghatározza a maximális karakterhosszt, és ha az eredeti szöveg meghaladja ezt a hosszt, akkor csak az első szavakat jeleníti meg, majd hozzáadja a "..." jelet a végéhez. Ha az eredeti szöveg rövidebb, akkor azt teljes egészében megjeleníti.

```
1  <?php
2      $event_description = $row["event_description"];
3      $max_chars = 250;
4      if (strlen($event_description) > $max_chars) {
5          $words = explode(' ', $event_description);
6          $display_text = '';
7          $char_count = 0;
8          foreach ($words as $word) {
9              if ($char_count + strlen($word) > $max_chars) {
10                 break;
11             }
12             $display_text .= $word . ' ';
13             $char_count += strlen($word) + 1;
14         }
15         echo $display_text . '...';
16     } else {
17         echo $event_description;
18     }
19  ?>
```

site_add.php – Esemény létrehozása

```
1 if ($_SERVER["REQUEST_METHOD"] == "POST") {
2     if (!empty($_POST['event_name']) && !empty($_POST['event_date']) && !empty($_POST['event_description']) && !empty($_FILES["event_image"]["name"])) {
3         $event_name = $_POST['event_name'];
4         $event_date = $_POST['event_date'];
5         $event_description = $_POST['event_description'];
6         $image_name = $_FILES["event_image"]["name"];
7         $image_path = "uploads/" . basename($_FILES["event_image"]["name"]);
8         $target_dir = "../uploads/";
9         $target_file = $target_dir . basename($_FILES["event_image"]["name"]);
10        move_uploaded_file($_FILES["event_image"]["tmp_name"], $target_file);
11        $adult_price = $_POST['adult_price'];
12        $student_price = $_POST['student_price'];
13        $pensioner_price = $_POST['pensioner_price'];
14        $family_price = $_POST['family_price'];
15        $group_price = $_POST['group_price'];
16        $sql = "INSERT INTO events (event_name, event_date, event_description, image_name, image_path, f_ar, d_ar, ny_ar, csa_ar, cso_ar)
17        VALUES ('$event_name', '$event_date', '$event_description', '$image_name', '$image_path', '$adult_price', '$student_price', '$pensioner_price', '$family_price', '$group_price')";
18        if ($conn->query($sql) === TRUE) {
19            $event_id = $conn->insert_id;
20            header("Location: ../events.php?event=$event_id");
21            exit;
22        }
23        else {
24            echo "Hiba történt az adatbázisban: " . $conn->error;
25        }
26    }
27    else {
28        echo "Kérem, töltsse ki mindkét mezőt!";
29    }
}
```

Ez a kód űrlap adatokat fogad és azokat elmenti az adatbázisban. Az alábbiakban részletesen kifejttem a kód működését:

A kód első sorában egy feltételt ellenőriz, hogy a kérés metódusa "POST" legyen. Ez azt jelenti, hogy csak akkor fut le a kód, ha az űrlapot elküldték a "POST" metódussal. Ezután ellenőrzi, hogy az űrlap mezői (event_name, event_date, event_description, event_image) nem üresek-e. Ha mindegyik mező ki van töltve, akkor a következő lépések következnek. A következő sorokban a kód elmenti a POST adatokat változóba, például az event_name-et az \$_POST['event_name'] értékébe. Ezután a kód létrehoz egy elérési utat és fájlnévet az event_image fájlnak, majd a move_uploaded_file() függvény segítségével áthelyezi a feltöltött fájlt a célkönyvtárba.

Ezután a kód elmenti az árakat (adult_price, student_price, pensioner_price, family_price, group_price) a megfelelő változóba.

Ezután a kód egy SQL lekérdezést hajt végre, amely az adatokat beszúrja az "events" táblába az adatbázisban. Ha a lekérdezés sikeres volt (azaz az adatok sikeresen beszúródtak az adatbázisba), akkor a kód elmenti az új esemény azonosítóját (event_id), majd a header() függvény segítségével átirányítja a felhasználót az esemény részleteinek oldalára. Ha a lekérdezés nem volt sikeres, akkor a kód kiírja a hibaüzenetet az adatbázis hibájáról. Ha az űrlap mezői nem voltak kitöltve, akkor a kód kiírja a "Kérem, töltsse ki mindkét mezőt!" üzenetet.

7. Fejlesztési lehetőségek

Online borrendelés és szállítás lehetősége

Ennek révén a távolabbról érkező látogatók is könnyen beszerezhetnék termékeket. Ez növelhetné az eladásokat.

Interaktív térkép a pincészetek és borkóstoló helyszínek feltüntetésével

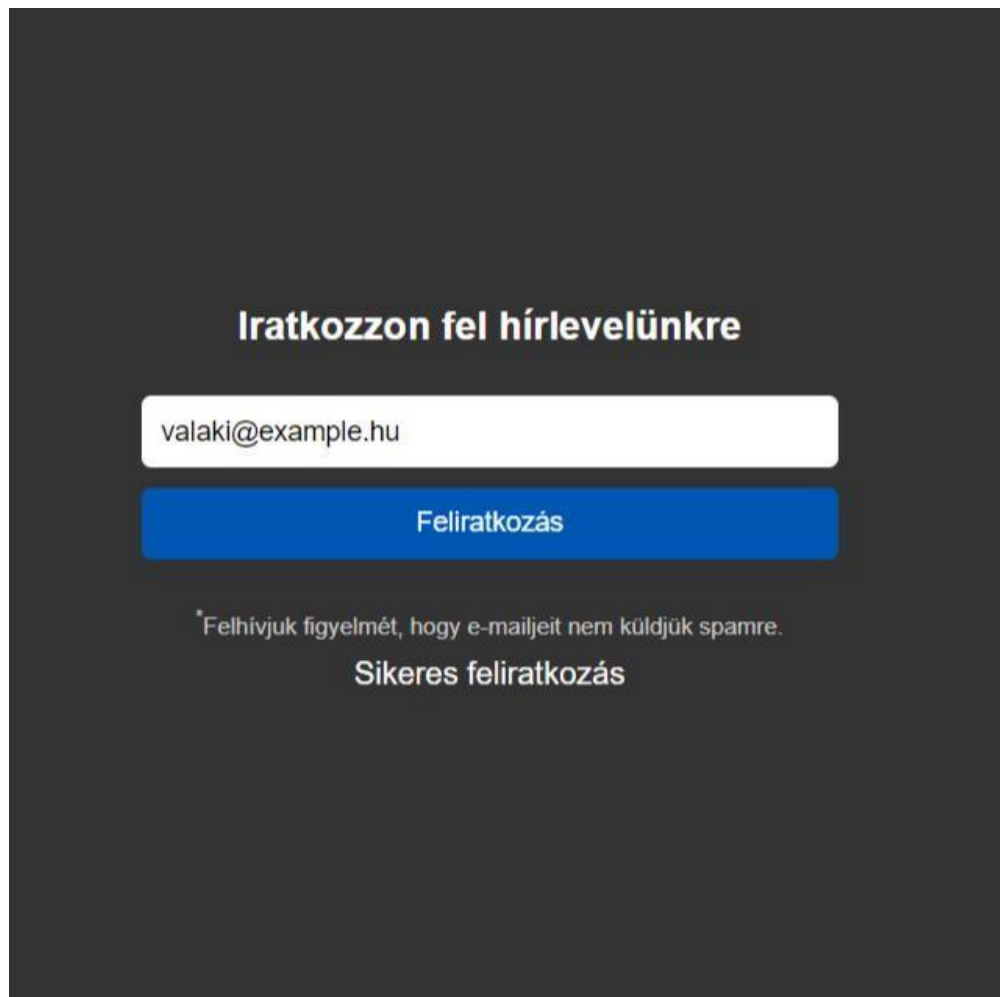
Ez segíthetné a látogatókat a különböző helyszínek megtalálásában és megközelítésében. A térképre rá kattintva további információkat lehetne megjeleníteni egy-egy helyszínről.

Közösségi média integráció (Facebook, Instagram)

Ennek segítségével a tartalmak megoszthatóvá válnának, nagyobb közönséget lehetne elérni az ott jelen lévő követők révén.

8. Tesztdokumentáció

Az e-mail feliratkozáskor fontos, hogy a felhasználók kapjanak visszaigazolást arról, hogy sikeresen feliratkoztak, ezáltal megerősítve a feliratkozásukat és bizalmukat az oldalban.



The image shows a dark-themed user interface for a newsletter subscription confirmation. At the top, the text "Iratkozzon fel hírlevelünkre" is displayed in white. Below this is a white input field containing the email address "valaki@example.hu". Underneath the input field is a prominent blue button with the white text "Feliratkozás". Below the button, a small line of text reads: "*Felhívjuk figyelmét, hogy e-mailjeit nem küldjük spamre." Finally, the text "Sikeres feliratkozás" is shown in a larger white font, indicating the successful completion of the subscription process.

Amennyiben már létezik az email:

Iratkozzon fel hírlevelünkre

valaki@example.hu

Feliratkozás

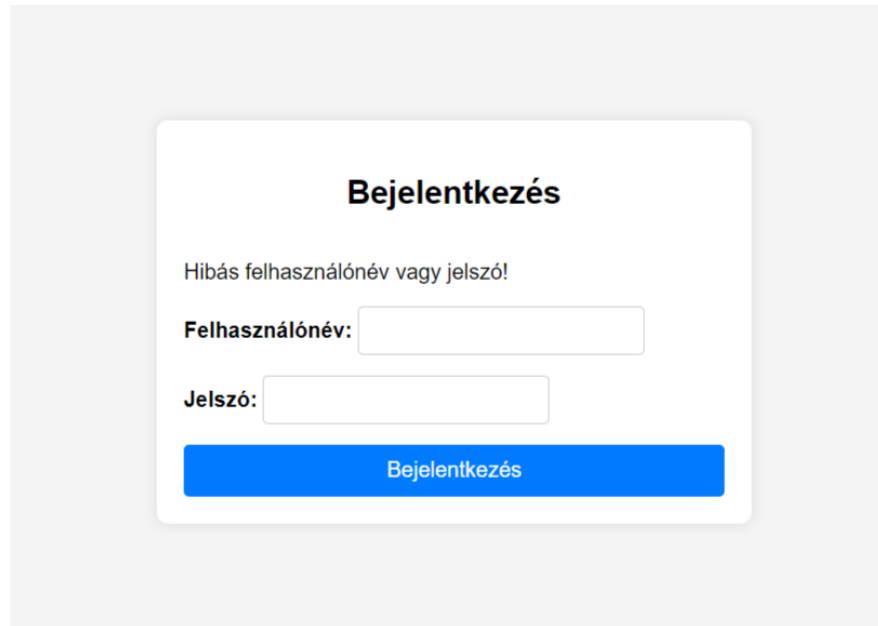
*Felhívjuk figyelmét, hogy e-mailjeit nem küldjük spamre.

Az email már fel van iratkozva

Az alábbi forráskódot a submit.php-ban találjuk:

```
1  if ($result->num_rows > 0) {
2      $message = "Az email már fel van iratkozva";
3  } else {
4      $sql = "INSERT INTO subscribers (email) VALUES ('$email')";
5      if ($conn->query($sql) === TRUE) {
6          $message = "Sikeres feliratkozás";
7      } else {
8          $message = "Hiba történt a feliratkozás közben: " . $conn->error;
9      }
10 }
```

Az admin felületen, ha be szeretnénk jelentkezni és hibás felhasználónév vagy jelszó próbálunk be lépni akkor az alábbi hibaüzenetet kapjuk.



Bejelentkezés

Hibás felhasználónév vagy jelszó!

Felhasználónév:

Jelszó:

Bejelentkezés

Az alábbi forráskódot az index.php-ban találjuk:

```
<body>
  <div class="container">
    <form class="login-form" method="POST" action="login.php">
      <h2>Bejelentkezés</h2>
      <?php
        if(isset($_SESSION['error'])) {
          echo "<p class='error'>{$_SESSION['error']}</p>";
          unset($_SESSION['error']);
        }
      ?>
      <div class="input-group">
        <label for="username">Felhasználónév:</label>
        <input type="text" id="username" name="username" required>
      </div>
      <div class="input-group">
        <label for="password">Jelszó:</label>
        <input type="password" id="password" name="password" required>
      </div>
      <button type="submit">Bejelentkezés</button>
    </form>
  </div>
</body>
```

9. Irodalomjegyzék

- <https://www.w3schools.com>
Azért keresgéltem a W3Schools oldalán, hogy segítségül vegyem a CSS dizájn kialakítását és a színek kezelését.
- <https://php.net/>
A PHP hivatalos oldalát, a php.net-et használtam arra, hogy megértsem a PHP kódok szintaxisát.
- <https://www.codecademy.com/>
A Codecademy platformját választottam arra, hogy elmélyüljek a PHP elemek tanulásában.
- <https://chat.openai.com/>
A ChatGPT-t alkalmaztam a JavaScript kód elkészítésében, valamint néhány PHP kódrészlet javításában.

A záródolgozat forráskódja és a dokumentáció az alábbi linken érhető el:

<https://github.com/gaborhh/cakipincesor>