Csábító Tea Party

Tervdokumentáció

Web Projekt

|  |  |
| --- | --- |
| **Készítették** | **Feladatkör** |
| Szabó Ferenc | Frontend, HTML |
| Katona Zsolt | Backend, PHP |
| Nagy Erik | Adatbázis, dokumentáció |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Dokumentum azonosító: | L-K-GINFBAL-WEBPROG1-1-LA02 |
| Verzió: | 1.3 |
| Dátum: | 2024. 04. 11. |

|  |
| --- |
| **Lektorálta** |
| Nagy Erik |

**©** Minden jog fenntartva. Az alábbi dokumentációban meghatározott szoftver tervek, és egyéb hozzátartozó elemek a „**Csábító Tea Party”** szellemi tulajdona.

Az anyag másolása, sokszorosítása, felhasználása kizárólagosan a „Csábító Tea Party” csapata engedélyével lehetséges!

Tartalomjegyzék

[**1.** **Bevezetés** 4](#_Toc163743765)

[**2.** **Fejlesztési környezet** 4](#_Toc163743766)

[*2.1* *A web szerver: APACHE és XAMPP* 4](#_Toc163743767)

[*2.2* *Verziókövetés: GitHub* 5](#_Toc163743768)

[*2.3* *Fájl átvitel: FTP* 5](#_Toc163743769)

[*2.4* *A Webtárhely szolgáltató* 5](#_Toc163743770)

[*2.5* *Használt technológiák* 6](#_Toc163743771)

[**3** **Design** 7](#_Toc163743772)

[*3.1* *Általános kinézet* 7](#_Toc163743773)

[*3.2* *Reszponzív megjelenés* 7](#_Toc163743774)

[**4.** **Fejlesztési keret: Frontend, Backend és Adatkezelés** 7](#_Toc163743775)

[*4.1* *Frontend* 7](#_Toc163743776)

[*4.2* *Backend* 8](#_Toc163743777)

[*4.3* *Adatkezelés* 11](#_Toc163743778)

# **Bevezetés**

Jelen tervdokumentáció a „Webprogramozas-1” elnevezésű tantárgy, a **Csábító Tea Party** csapataáltal fejlesztés alatt álló számítástechnikai projekt, **weboldal** fejlesztői környezetét, koncepcióját, design elemeinek tervezett megvalósítását és fejlesztési tevékenységeit ismerteti.

# **Fejlesztési környezet**

## *A web szerver: APACHE és XAMPP*

Az **Apache** **HTTP Server**, röviden Apache, egy nyílt forráskódú web szerver szoftver, amely a világ legnépszerűbb web szervere. Az Apache széles körben ismert a rugalmasságáról, megbízhatóságáról, és széleskörű operációs rendszer támogatásáról. Képes kiszolgálni mind statikus, mind dinamikus weboldalakat egyaránt, támogatja a legtöbb modern programozási nyelvet és webes technológiát, így PHP-t, Perl-t, Python-t, Ruby-t, és többet.

A **XAMPP** egy ingyenes és nyílt forrású cross-platform web szerver megoldáscsomag, amelyet kifejezetten fejlesztési célokra terveztek. A XAMPP az Apache web szervert, MariaDB adatbázis-kezelőt, PHP és Perl programozási nyelvek támogatását integrálja egy könnyen telepíthető csomagban, ami rendkívül hasznos eszközzé teszi a fejlesztők számára. A XAMPP segítségével a fejlesztők gyorsan létrehozhatnak egy helyi fejlesztési környezetet a számítógépükön, amely lehetővé teszi számukra a weboldalak és webalkalmazások fejlesztését és tesztelését anélkül, hogy szükség lenne egy élő szerverre. Ez különösen előnyös a fejlesztés korai szakaszaiban, amikor gyakoriak a változtatások, és fontos a gyors iteráció.

A „Webprogramozás-1” tantárgyhoz tartozó projekt feladat fejlesztése során csapatunk igénybe veszi a XAMPP által nyújtott lehetőségeket, ez biztosítja a fejlesztési munka zökkenőmentességét és a projekt sikeres megvalósítását. XAMPP integrálása növeli a fejlesztési folyamat hatékonyságát, lehetővé téve a csapat számára, hogy kizárólag a fejlesztésre összpontosítson miközben a környezet karbantartása személyre szabottan egyszerű és gördülékeny marad.

## *Verziókövetés: GitHub*

A **GitHub**, mint vezető szoftverfejlesztési platform, lehetővé teszi a csapat számára, hogy hatékonyan kezelje a kódbázist, nyomon követhesse a változásokat, és elősegítse a kollaboratív munkát azáltal, hogy egy központosított helyen tárolja a projekt fájljait és dokumentációját. Az integrált verziókezelési rendszer előnyei közé tartozik a fejlesztési folyamatok nagyfokú automatizálása, a visszaállíthatóság, valamint a különböző fejlesztési ágak (branches) kezelése, ami lehetővé teszi a csapat számára, hogy párhuzamosan dolgozzon különböző funkciókon. A GitHub használatával a Csábító Tea Party csapata kihasználhatja a kódmegosztás és a közösségi hozzájárulás előnyeit is, így gyorsítva a fejlesztési ciklusokat és javítva a projekt átláthatóságát és nyomon követhetőségét.

## *Fájl átvitel: FTP*

A „Csábító Tea Party” csapata az **FTP** (File Transfer Protocol) használja fejlesztési fájlok és anyagok interneten keresztüli átvitelére a web szerverre. Az FTP egy hagyományos hálózati protokoll, amely lehetővé teszi fájlok távoli szerverek és kliensek közötti átvitelét. Ennek a technológiának a használata kritikus fontosságú a webfejlesztési projektekben, mivel biztosítja a fejlesztők számára, hogy gyorsan és hatékonyan frissíthessék a weboldal tartalmát, feltölthessék az új kódverziókat, vagy módosíthassák a meglévő fájlokat a szerveren.

Az FTP használatának előnyei közé tartozik a nagyfokú kompatibilitás számos operációs rendszerrel és fejlesztői környezettel, valamint a felhasználóbarát interfészek, amelyek megkönnyítik a fájlok kezelését és átvitelét. Továbbá, az FTP támogatja a kötegelt fájlátvitelt is, lehetővé téve több fájl egyidejű átvitelét, ami jelentős időmegtakarítást jelent a nagyobb projektek esetén.

## *A Webtárhely szolgáltató*

A Csábító Tea Party csapata a Nethely.hu-t választotta webtárhely szolgáltatóként, amely különböző webtárhely csomagokat kínál, beleértve az ingyenes tárhely lehetőséget is. A Nethely.hu szolgáltatásai közé tartoznak az osztott tárhelyek, pro tárhelyek és prémium tárhelyek, amelyek különböző méretű tárhelyet és támogatást nyújtanak a felhasználók számára.

Az ingyenes tárhely csomag reklámmentes, és tartalmaz egyedi adminisztrációs felületet, CMS telepítőt többek között WordPress, Joomla, Drupal rendszerekhez, valamint adatbázis és levelezési lehetőségeket. Az ingyenes tárhely mellett a Nethely.hu ingyenes domain nevet is biztosít bizonyos feltételek mellett, többek között .nhely.hu és .webtelek.hu végződésekkel. Ezen felül lehetőség van saját domain név használatára is az ingyenes tárhely csomagban​.

A Nethely.hu kínálatában szereplő tárhely csomagok különböző technológiai előnyöket kínálnak, mint például NVMe SSD tárolás, AMD EPYC processzorok, napi biztonsági mentések, Let's Encrypt HTTPS tanúsítványok, és PHP 5.6 - 8.3 támogatás. Az ügyfélszolgálat munkanapokon telefonon és e-mailben is elérhető, biztosítva ezzel a gyors és hatékony támogatást a felhasználóknak.

## *Használt technológiák*

A Csábító Tea Party csapata a webfejlesztés során számos modern és bevált technológiát alkalmaz, hogy biztosítsák a webalkalmazásuk gyors, biztonságos és felhasználóbarát működését:

**HTML (HyperText Markup Language)**: Az internet alapját képező nyelv, amely a weboldalak struktúráját adja meg. Az HTML5 a legújabb verziója, amely további szemantikai elemeket és jobb multimedia támogatást kínál, ezáltal elősegíti a reszponzív webdesign kialakítását.

**PHP (Hypertext Preprocessor)**: Egy szerveroldali szkriptnyelv, amelyet kifejezetten webfejlesztéshez terveztek. Képes dinamikus tartalmak generálására, adatbázis-kezelésre, munkamenet-kezelésre, és számos egyéb funkcióra, amelyek interaktív weboldalak készítéséhez szükségesek.

**MySQL**: Az egyik legnépszerűbb nyílt forráskódú relációs adatbázis-kezelő rendszer, amelyet különösen webalkalmazások adattárolására használnak. PHP-val együttműködve erőteljes adatkezelési képességeket biztosít.

**Bootstrap**: Egy ingyenes és nyílt forrású front-end keretrendszer, amely elősegíti a gyors és egyszerű reszponzív webdesign kialakítását. Készen álló HTML és CSS sablonokat, valamint JavaScript komponenseket kínál, amelyekkel könnyedén lehet készíteni felhasználóbarát és esztétikailag vonzó weboldalakat.

**Reszponzív webdesign**: Az a tervezési és fejlesztési módszertan, amelynek célja, hogy a weboldalak minden eszközön - asztali számítógéptől kezdve a mobiltelefonokig - jól nézzenek ki és jól működjenek. A Bootstrap keretrendszer kiválóan támogatja a reszponzív webdesign kialakítását.

**JavaScript**: Egy kliensoldali szkriptnyelv, amely interaktívabbá teszi a weboldalakat. JavaScript keretrendszerek, mint például Angular, React, vagy Vue.js, tovább fokozzák a webfejlesztési projektjeik dinamikus és felhasználóközpontú jellegét.

# **Design**

## *Általános kinézet*

Az kinézet megtervezésekor prioritást élvezett az eredeti kutyamenhely weboldalának lényegi vonásainak megőrzése. A modernizáció jegyében beágyaztuk a Google térképet és YouTube videókat, amelyek lehetővé teszik a látogatók számára, hogy mélyebb betekintést nyerjenek a menhely munkájába és helyszínébe. A menürendszert is modernizáltuk, hogy az intuitívabb és felhasználóbarátabb legyen, miközben fenntartjuk a weboldal hagyományos navigációs sémáit. Ezek a változtatások együttesen hozzájárulnak ahhoz, hogy a weboldal korszerű, mégis ismerős maradjon a visszatérő látogatók számára.

## *Reszponzív megjelenés*

A reszponzív megjelenés kialakításánál a Bootstrap és a CSS kulcsfontosságú eszközök. A projektben ezen eszközök használatára fektettük a hangsúlyt. A Bootstrap grid rendszere lehetővé teszi a fejlesztők számára, hogy rugalmasan és hatékonyan alakítsák ki a weboldalak elrendezését, adaptálva azokat különböző képernyőméretekhez. Ezáltal a weboldalak könnyen olvashatóak és kezelhetőek lesznek minden eszközön. A CSS segítségével pedig tovább finomítható a megjelenés, például a szövegek, színek és margók testre szabásával, ami egyedi és vonzó design-t eredményez. Ezek az eszközök együtt biztosítják, hogy a weboldal ne csak esztétikailag mutasson jól, hanem funkcionálisan is reszponzív legyen.

# **Fejlesztési keret: Frontend, Backend és Adatkezelés**

## *Frontend*

**Feladata:** A weboldal elsődleges célja és feladata, hogy bemutassa a szekszárdi kutyamenhelyet, információkat nyújtson az örökbefogadható kutyákról, továbbá tájékoztassa az érdeklődőket az örökbefogadás folyamatáról és támogatási lehetőségekről.

**Felépítése:** A weboldal struktúrája a Bootstrap grid rendszerére épül, amely lehetővé teszi az oldalak dinamikus és reszponzív elrendezését. Az oldalak 12 egyenlő szélességű oszlopra vannak osztva, amelyek a tartalmakat és a vizuális elemeket egyaránt hatékonyan kezelik. Az egyedi stílusok és az oldal specifikus igényeinek megfelelően kialakított dizájn elemek a CSS fájlon keresztül kerültek implementálásra.

**Működése:** Az oldal működését a felhasználói interakciók és a vizuális elemek harmonikus összehangolására terveztük. Különös figyelmet fordítottunk a felhasználói felület intuitív és felhasználóbarát kialakítására, hogy a látogatók könnyen tájékozódhassanak. Az egyedi és friss stílusjegyekkel gazdagított dizájn célja, hogy egyedi vizuális élményt nyújtson, miközben hű marad az eredeti dizájnhoz.

Ezek a szempontok biztosítják, hogy a weboldal nemcsak informatív és hasznos legyen a látogatók számára, hanem egyben esztétikailag is vonzó, ösztönözve őket az oldalon való hosszabb időtöltésre és az interakcióra.

A Bootstrap elemek kiválóan alkalmazkodnak a különböző képernyőméretekhez. Kisebb képernyőkön, mint például mobiltelefonokon, a navigációs menü egy kattintható ikonná zsugorodik, amelyet a felhasználók kibonthatnak. Az oldal elemei, beleértve a szövegeket és képeket is, egymás alá rendeződnek, biztosítva ezzel a könnyű olvashatóságot és navigációt. A képek mérete is automatikusan igazodik, így mindig a lehető legjobban kihasználják a rendelkezésre álló teret, miközben fenntartják a vizuális minőséget.

## *Backend*

**Feladata:** A backend a weboldal szerveroldali logikájának kivitelezéséért felelős, amely közvetlenül nem látható a felhasználók számára. Ennek a rétegnek a feladata az adatkezelés, logika megvalósítása, és a kliens oldali kérések feldolgozása. Ebben az esetben a PHP nyelvet használtuk, amely egy széles körben alkalmazott szerveroldali scriptnyelv, lehetővé téve dinamikus weboldalak létrehozását kezdetleges objektum orientált programozással.

**Felépítése:** A backend rendszerünk egy egyszerűsített **MVC** (Modell-Nézet-Vezérlő) struktúrára épül, ahol a front controller (router) a központi eleme. Az ***index.php*** file szolgál a kérések fogadására és továbbítására, létrehozva egy egységes folyamatot, ahol minden kérés áthalad. Az ***.htaccess*** szabályok segítségével felhasználóbarát URL-eket használunk, javítva a felhasználói élményt és a weboldal **SEO**-ját. Az ***ApplicationController*** osztály példányosítja és kezeli a különböző kontrollereket, mint a ***RouterController***, ***RequestController***, ***ResponseController***, valamint funkcionális osztályokat, mint az ***AuthController***, ***ImageController***, ***ContactController*** a felhasználói interakciók kezelésére.

**Működése:**

***RouterController:***

A működés központjában a ***RouterController*** áll, amely a kérések útválasztásáért felelős.

* A ***get()*** és ***post()*** metódusokkal a különböző útvonalak és a hozzájuk tartozó callback függvények kerülnek definiálásra.
* A ***resolve()*** metódus minden kérésre reagál, megkeresve és meghívva a megfelelő függvényeket vagy eljárásokat.
* A ***renderLayout()*** metódus az oldal alap sablonját („layoutját”) importálja be.
* A ***renderOnlyView($view)*** metódus a paraméterben kapott értékhez tartozó tartalmat tölti be.
* A ***renderView($view)*** metódus meghívja a megjelenítő függvényeket, és a layoutban elhelyezett marker *({{content}}*) helyére beilleszti a ***renderOnlyView()*** tartalmát.

***RequestController*** és ***ResponseController:***

A *RequestController* és *ResponseController*osztályok a kérések és válaszok kezeléséért felelősek, mint például az útvonalak kinyerése, kérés típusának meghatározása, státuszkódok beállítása, és az átirányítások kezelése.

* *RequestController*: Kérésekkel kapcsolatos függvények/eljárások osztálya.
  + ***getPath()*** metódus az URL-ből lekéri a „/” utáni részt.
  + ***getMethod()*** metódus visszaadja a kérés típusát.
* ResponseController:
  + Speciális válaszokkal kapcsolatos függvények/eljárások osztálya.
  + ***setHttpStatus(int $code)*** metódus a státuszkód visszaküldésért felelős. (200,400,404...)
  + ***redirect($url)*** metódus felel az oldal átirányításokért, az $url-ben megadott címre.

***AuthController:***

Az *AuthController* a felhasználók regisztrációját, bejelentkezését, és kijelentkezését kezeli, biztonságos jelszókezeléssel és session kezeléssel.

1. A weboldalon egy alapszintű felhasználó kezelés került megvalósításra. A felhasználóknak lehetősége van a weboldalra regisztrálni majd azt követően be- és kijelentkezni.
2. Bejelentkezést követően lehetőségük van a galériába fényképet feltölteni. A regisztrációhoz felhasználónevet, vezetéknevet, keresztnevet és jelszót kér az oldal. Bejelentkezni a felhasználónév, jelszó párossal lehetséges.
3. A regisztrációnál megadott adatok **MySQL** adatbázisban kerülnek tárolásra. A jelszó a PHP által natívan támogatott **bcrypt** kódolással kerül eltárolásra.
4. Bejelentkezéskor az oldal egy session-ban tárolja el az adatokat, így a böngésző bezárását követően nem szükséges újra bejelentkezni az oldalra.
5. Regisztráció vagy bejelentkezés során történt hiba esetén megfelelő hibakezelés történik, és a felhasználót értesítjük a fellépő hibáról.

Az ***ApplicatonController*** példányosítja az ***AuthController*** osztályt. Az ***index.php***-ben előre definiáltak szerint a /register, /login és /logout POST kéréseket az ***AuthController*** egyes eljárásai kezelik le. A POST kérések a frontend oldalon elkészített HTML űrlapok beküldése során keletkeznek. Az ***AuthController*** beimportálja és példányosítja a ***DatabaseController***-t, amely felelős a ***MySQL*** kapcsolatért.

* ***register()*** metódus /register POST kérés esetén kerül meghívásra, ellenőrzi a kapott és várt adatok hiánytalanságát, a jelszavak egyezőségét, illetve hogy a regisztrálni kívánt felhasználónév nem-e foglalt. Amennyiben minden ellenőrzés sikeres volt létrehozza a felhasználót az adatbázisban.
* **login()** metódus /login POST kérés esetén kerül meghívásra, ellenőrzi a kapott és várt adatok hiánytalanságát, a jelszavak egyezőségét valamint a felhasználónév létezését az adatbázisban. Sikeres ellenőrzések esetén létrehozza a megfelelő session-t a felhasználónak.
* **logout()** metódus törli az összes sessiont, ezzel kijelentkeztetve a felhasználót.
* **passwordCompare()** metódus összeveti a felhasználó által megadott jelszó kódolt értékét az adatbázisban tárolttal. Igaz/hamis értékekkel tér vissza.
* **hashPassword($pw)** metódus létrehozza bcrypt hasht a *$pw* változóban kapott értékből, majd visszatér a kódolt szöveggel.
* **getUserData($username, $conn)** metódus lekérdezi az adatbázisból a *$username* paraméterből kapott felhasználót. Találat esetén visszatér a felhasználó adataival, egyéb esetben ’hamis’ értékkel tér vissza.

***ContactController*** *és* ***ImageController:***

A *ContactController* és *ImageController* osztályok a kapcsolatfelvételi űrlapok és képfeltöltési funkciók implementációjáért felelősek, biztosítva az adatok adatbázisba való mentését és a képek dinamikus kezelését.

A beérkezett kapcsolat felvételi üzenetek az oldal által használt **MySQL** adatbázisba kerülnek mentésre, regisztrált és nem regisztrált felhasználók egyaránt kitölthetik az űrlapot.

Az ***ApplicatonController*** példányosítja az ***ContactController*** osztályt. Az ***index.php***-ben előre definiáltak szerint a /sendcontactform POST kéréseket az ***ContactController*** send() eljárása kezeli le.

A ***ContactController*** beimportálja és példányosítja a ***DatabaseController***-t, amely felelős a **MySQL** kapcsolatért.

A fényképeket a weboldal egy kijelölt mappából (Content/Images/Gallery), dinamikusan tölti be. Kizárólag regisztrált felhasználók tölthetnek fel képeket. (Fájl méret < 5MB, JPG,PNG,GIF)

Az ***ApplicatonController*** példányosítja az ***ImageController*** osztályt. Az ***index.php***-ben előre definiáltak szerint a /uploadimg POST kéréseket az ***ImageController*** upload() eljárása kezeli le.

* *ContactController:*
  + ***send()*** metódus a kapott és elvárt adatok hiányosságának ellenőrzéséért felel, illetve exportál az adatbázisba.
  + ***getMessage()*** metódus felel az üzenetek lekérdezéséért az adatbázisból.
* *ImageController:*
  + ***upload()***metódus a beérkezett adatokat ellenőrzi (van-e csatolmány, csatolmány mérete, kiterjesztése), majd feltölti azt a megadott mappába.

A weboldalunk jelenlegi állapotban már egy hatékony és felhasználóbarát platformot nyújt, amely PHP alapokon és egy kezdetleges MVC modellen keresztül valósítja meg a felhasználói igényeket és interakciókat. Azonban, mint minden technológiai projekt, a weboldalunk is folyamatos fejlődés alatt áll, és számos lehetőséget kínál további fejlesztésre és optimalizálásra.

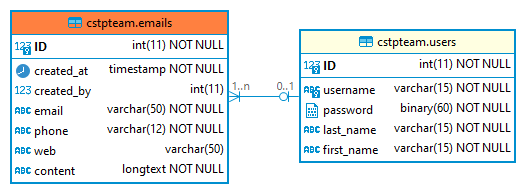
A jövőbeli fejlesztési lehetőségek magukban foglalják a teljes MVC modell kiépítését, fejlettebb hitelesítési eljárások, mint az Authentication middleware és JWT Token autentikáció bevezetését, valamint a hibakezelés és felhasználói felület optimalizálását design komponensek újrahasznosításával és AJAX kérések alkalmazásával, amelyek révén a weboldal dinamikusabbá és felhasználóbarátabbá válik, beleértve a "Maradjon bejelentkezve" funkciót is COOKIE alapú login segítségével.

## *Adatkezelés*

Az adatkezelés során a nethely.hu által szolgáltatott **MySQL** alapú adatbázist használjuk, melynek részletei az alábbi 3 tagolt pontra bonthatók.

**Feladata:** Az adatbázis elsődleges feladata egy olyan megbízható háttérszolgáltatás biztosítása, amely lehetővé teszi a felhasználók regisztrációját és bejelentkezését, valamint azonosításukat az űrlapok kitöltésekor. Emellett támogatja a különböző jogosultságok kezelését, így a bejelentkezett felhasználók képfeltöltési jogosultságaikat is.

**Felépítése:** Az adatbázis pontos felépítése az alábbi ER diagrammon tekinthető meg:



Az adatbázismodell magában foglal egy felhasználók táblát, amely az alapvető felhasználói adatokat, mint például felhasználónév és jelszó tárolja, valamint egy külön email táblát az űrlapokkal kapcsolatos adatok kezelésére. Ez a struktúra lehetővé teszi az egyszerű felhasználói műveletek, mint a regisztráció és bejelentkezés, valamint az űrlapokon keresztüli kommunikáció kezelését.

**Működése:** A *cstpteam.users* tábla az alapvető felhasználói információkat tárolja. Az I**D** mező az elsődleges kulcsként szolgál, amely egyedi azonosítót biztosít minden felhasználónak. A **username** egy rövid karakterlánc (varchar) típus, ami legfeljebb 15 karakter hosszúságú felhasználónevet tárol. A **password** mező egy 60 bájtos binary típusú mező, amely a bcrypt algoritmus által kódolt (hash) jelszavakat tárolja. A **first\_name** és **last\_name** mezők a felhasználók kereszt- és vezetéknevét tárolják szintén varchar(15) típusú mezőkben.

A *cstpteam.emails* tábla tartalmazza az email kommunikációval kapcsolatos adatokat. Az **ID** itt is elsődleges kulcs. A **created\_at** mező egy időbélyeget tárol, amely jelzi az üzenet létrehozásának időpontját. A **created\_by** mező egy idegen kulcs, ami a *users* tábla egy felhasználójának **ID**-jére hivatkozik, így jelölve, hogy melyik felhasználó hozta létre az adott emailt. Az **email** mező az email címeket tárolja varchar(50) formátumban, a **phone** mező a telefonszámokat varchar(12), a **web** mező pedig webcímeket varchar(50) formátumban. Végül a content mező egy longtext típusú mező, amely a kommunikáció tartalmát tárolja, így akár nagyon hosszú szövegek kezelésére is alkalmas.