



Fejlesztői dokumentáció

2024

Készítők:

*Ladányi Anna*

*Maász Martina Rita*

*Tózer Kitti*

Tartalomjegyzék

[I. Bevezetés 2](#_Toc164152910)

[1. Cél 2](#_Toc164152911)

[2. Téma ismertetése 2](#_Toc164152912)

[3. Témaválasztás indoklása 2](#_Toc164152913)

[II. Fejlesztői dokumentáció 3](#_Toc164152914)

[1. Felhasznált technológiák 3](#_Toc164152915)

[2. Fejlesztői környezet - Működtetési lépések 3](#_Toc164152916)

[3. Kialakított adatszerkezet 4](#_Toc164152917)

[Felhasználók 4](#_Toc164152918)

[Funkciók 4](#_Toc164152919)

[Jogosultságok 5](#_Toc164152920)

[Adatbázis szerkezeti felépítése és terve 6](#_Toc164152921)

[Adatbázis diagram 7](#_Toc164152922)

[Alapkönyvtárak 8](#_Toc164152923)

[Végpontok 8](#_Toc164152924)

[4. Algoritmusok / kódok 9](#_Toc164152925)

[Osztályok 9](#_Toc164152926)

[Függvények 11](#_Toc164152927)

[AJAX-kérések 12](#_Toc164152928)

[Eseménykezelők 12](#_Toc164152929)

[További funckiók 12](#_Toc164152930)

[5. Felhasználói felület 13](#_Toc164152931)

[6. Fejlesztési lehetőségek 18](#_Toc164152932)

[III. Felhasználói dokumentáció 19](#_Toc164152933)

[IV. Tesztelés 20](#_Toc164152934)

[1. Manuális tesztelés 20](#_Toc164152935)

[2. Terheléses vizsgálat 22](#_Toc164152936)

[3. Környezet 22](#_Toc164152937)

[V. Összegzés 23](#_Toc164152938)

# I. Bevezetés

## 1. Cél

Üdvözöljük Önt a szolgáltatásfoglaló weboldalunkon! Célunk, hogy egy összetartó közösséget teremtsünk, ahol az emberek könnyedén és hatékonyan foglalhatnak különböző szolgáltatásokat. Hiszünk abban, hogy a kis közösségek erejével és az egyszerű foglalási folyamatokkal mindennapi életünk könnyebbé válik. A csapatunk elkötelezett amellett, hogy innovatív megoldásokkal és személyre szabott élményekkel járuljon hozzá mind az ügyfelek, mind a szolgáltatók sikeréhez.

## 2. Téma ismertetése

Weboldalunk célja és fő tevékenységi területe a különböző szolgáltatások könnyű és gyors foglalásának lehetőségének biztosítása. Kiterjedt kínálattal rendelkezünk, amely számos szolgáltatást tartalmaz, beleértve például a kozmetikust, tetoválást, fodrászt, de vízvezetékszerelést vagy akár személyi edzést is és még sok mást. Célunk, hogy egy olyan platformot nyújtsunk, ahol az emberek könnyedén megtalálhatják és foglalhatják azokat a szolgáltatásokat, amelyekre szükségük van, és közvetlen kapcsolatba léphetnek a szolgáltatókkal.

## 3. Témaválasztás indoklása

A szolgáltatásfoglaló weboldal létrehozása mellett azért döntöttünk, mert felismertük az emberek igényét egy olyan platform iránt, amely egyszerre kínál különböző szolgáltatásokat és lehetőséget biztosít a könnyű és gyors foglalásra. Számos esetben az embereknek nehézségeik lehetnek a megfelelő szolgáltatások megtalálásában és a foglalási folyamatok bonyolultságában. Ezért szeretnénk megkönnyíteni az emberek számára ezt a folyamatot, és lehetővé tenni számukra, hogy egyetlen platformon keresztül könnyen és gyorsan foglalhassanak különböző szolgáltatásokat. Ezen kívül hiszünk abban, hogy a közvetlen kapcsolat a szolgáltatókkal és a könnyű elérhetőség növeli mind az ügyfelek, mind a szolgáltatók elégedettségét és sikerét, ezért ezt a szemléletet ötvöztük a weboldalunk működésében.

# II. Fejlesztői dokumentáció

## 1. Felhasznált technológiák

**PHP:** A projekt fő nyelvének PHP-t választottuk, mivel dinamikus webes alkalmazás fejlesztésére kiválóan alkalmas.

**Composer**: Dependenciák kezelésére (PDO) és a osztályok automatikus betöltésére használtuk.

**HTML és Bootstrap:** Az alkalmazás felhasználói felületének kialakításához HTML-t használtunk, amelyet Bootstrap keretrendszerrel kiegészítettünk, hogy reszponzív és modern megjelenést biztosítsunk.

**SQL Server és PHPMyAdmin:** Az adatbázis kezelésére SQL Server-t választottuk, melyet PHPMyAdmin felületen adminisztráltunk, mivel jól integrálható és biztonságos.

**Visual Studio Code:** Fejlesztőkörnyezetként Visual Studio Code-ot használtunk, mivel könnyen testreszabható, támogatja a PHP fejlesztést és rendelkezik számos kiegészítővel.

**Git:** A verziókezeléshez Git-et alkalmaztunk, hogy hatékonyan nyomon követhessük és kezelhessük a projekt változásait és fejlesztéseit.

## 2. Fejlesztői környezet - Működtetési lépések

Szükséges szoftverek:

XAMPP és Visual Studio Code

XAMPP modulok indítása:

Apache és MySQL szolgáltatásokat használtunk a XAMPP vezérlőpultjából

Adatbázis inicializálása:

Feltöltöttük a FindEm.sql adatbázisfájlt a PHPMyAdmin felületén keresztül.

Weboldal futtatása:

A projekt könyvtárát a Visual Studio Code-ban nyitjuk meg.

Elindítjuk a programot a terminálban a következő paranccsal (a válaszott port számával):

php -S localhost:xxxx

Egy kiválasztott webböngésző címsorába beírjuk a következő URL-t:

http://localhost:xxxx

## 3. Kialakított adatszerkezet

### Felhasználók

* Vendég / Nem regisztrált felhasználó

Látják a szolgáltatásokat, de csak regisztráció és bejelentkezés után tudnak foglalni

* Regisztrált felhasználó

### Funkciók

* Regisztráció
* Bejelentkezés
* Saját profil adatok
  + Adatok megjelenítése
  + Adatok szerkesztése
* Saját lefoglalt időpontok
  + Foglalt időpontjaim listája
  + Korábbi foglalások
  + Időpont törlése
* Saját szabad szolgáltatás időpontok
  + Szabad időpontok listája
  + Szabad időpont felvétele
* Szolgáltatás kereső
  + Szolgáltatás lista
  + Szolgáltatás keresés
  + Szolgáltatók listájának megjelenítése
    - Szolgáltató összes szabad időpontjának megjelenítése
    - Szolgáltató kiválasztott szolgáltatásának szabad időpontjainak megjelenítése
* Szolgáltatáshoz tartozó időpontok megjelenítése
* Időpont kereső
  + Időpont lista
  + Időpont keresés
  + Időpont foglalás

### Jogosultságok

|  |  |  |
| --- | --- | --- |
| **Funkciók** | **Vendég** | **Regisztrált felhasználó** |
| Kezdőoldal | X | X |
| Menü | X | X |
| Regisztráció | X |  |
| Bejelentkezés |  | X |
| Szabad időpontok |  | X |
| Saját profil adatok |  | X |
| Saját lefoglalt időpontok |  | X |
| Saját szolgáltatások |  | X |
| Saját szabad szolgáltatás időpontok |  | X |
| Valaki által lefoglalt időpontok |  | X |
| Profil módosítása |  | X |
| Időpont kereső |  | X |

### Adatbázis szerkezeti felépítése és terve

CREATE TABLE appointment (

appointment\_id int(11) NOT NULL,

service\_id int(11) NOT NULL,

status\_id int(11) NOT NULL,

appointment\_date date NOT NULL,

appointment\_time time NOT NULL,

appointment\_duration float NOT NULL,

appointment\_fee int(11) NOT NULL,

PRIMARY KEY (appointment\_id),

KEY status\_id (status\_id),

KEY service\_id (service\_id),

CONSTRAINT fk\_service\_id FOREIGN KEY (service\_id) REFERENCES service (service\_id) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT fk\_status\_id FOREIGN KEY (status\_id) REFERENCES status (status\_id) ON DELETE CASCADE ON UPDATE CASCADE

) ;

CREATE TABLE reservation (

reservation\_id int(11) NOT NULL,

appointment\_id int(11) NOT NULL,

user\_id int(11) NOT NULL,

PRIMARY KEY (reservation\_id),

KEY appointment\_id (appointment\_id),

KEY user\_id (user\_id),

CONSTRAINT fk\_appointment\_id FOREIGN KEY (appointment\_id) REFERENCES appointment (appointment\_id) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT fk\_user\_id FOREIGN KEY (user\_id) REFERENCES user (user\_id) ON DELETE CASCADE ON UPDATE CASCADE

) ;

CREATE TABLE role (

role\_id int(11) NOT NULL,

role\_name varchar(20) NOT NULL,

PRIMARY KEY (role\_id)

) ;

CREATE TABLE service (

service\_id int(11) NOT NULL,

service\_provider\_id int(11) NOT NULL,

service\_category\_id int(11) NOT NULL,

service\_name varchar(30) NOT NULL,

service\_district int(11) NOT NULL,

service\_address varchar(50) NOT NULL,

service\_housenumber varchar(5) NOT NULL,

service\_description tinytext NOT NULL,

PRIMARY KEY (service\_id),

KEY service\_provider\_id (service\_provider\_id),

KEY service\_category\_id (service\_category\_id),

CONSTRAINT fk\_service\_category\_id FOREIGN KEY (service\_category\_id) REFERENCES service\_category (category\_id) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT fk\_service\_provider\_id FOREIGN KEY (service\_provider\_id) REFERENCES user (user\_id) ON DELETE CASCADE ON UPDATE CASCADE

) ;

CREATE TABLE service\_category (

category\_id int(11) NOT NULL,

category\_name varchar(30) NOT NULL,

PRIMARY KEY (category\_id)

) ;

CREATE TABLE status (

status\_id int(11) NOT NULL,

status\_name varchar(20) NOT NULL,

PRIMARY KEY (status\_id)

) ;

CREATE TABLE user (

user\_id int(11) NOT NULL,

role\_id int(11) NOT NULL,

first\_name varchar(30) NOT NULL,

last\_name varchar(30) NOT NULL,

email\_address varchar(50) NOT NULL,

password varchar(20) NOT NULL,

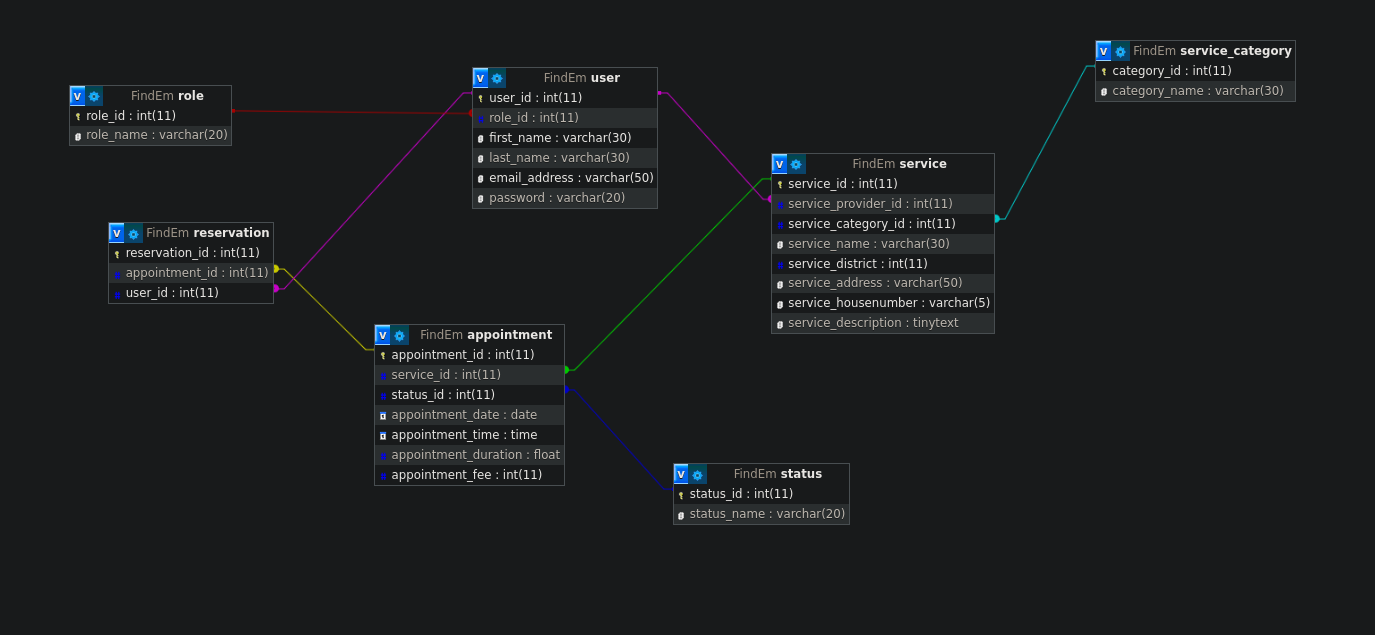
PRIMARY KEY (user\_id),

KEY role\_id (role\_id),

CONSTRAINT fk\_role\_id FOREIGN KEY (role\_id) REFERENCES role (role\_id) ON DELETE CASCADE ON UPDATE CASCADE

) ;

### Adatbázis diagram



### Alapkönyvtárak

* Controllers: A frontend-ről beérezett kéréseket kezelik.
* Models: Két modelt használtunk: User és Service modellek. Ezek végzik az adatbázis műveleteket és adják vissza az információt a controllereknek.
* Views: Itt találhatóak a HTML oldalak, melyeket a Template osztály segítségével jelenítünk meg.
* Helpers: Gyakran használt funkciókat ide tettük (az oldal átirányítás flash üzenetekkel és a megfelelő navbar beállítása a felhasználói szerep függvényében).
* Core: Tartalmazza a DatabaseHandler osztályt, mely létrehozza az adatbázis kapcsolatot PDO segítségével, valamint előre definiált funcciókat kapott hogy a későbbi adatbázis műveletek egyszerűbben kivitelezhetőek legyenek. Itt található még a Router osztály mely a megfelelő url-hez megkeresi a megfelelő controller megfelelő funkcióját. Ide tartozik még a fentebb említett Template osztály is, mely a HTML fájlokat jeleníti meg, melyeknek egy asszociatív lista formályában adhatunk át adatot a backendről.
* Public: Itt tároljuk az applikáció képeit, JavaScript és CSS fájlokat.
* Vendor: Composer által létrehozott mappa. Dependenciákat kezel és a PSR-4 automatikus betöltésért felelős.

### Végpontok

|  |  |  |
| --- | --- | --- |
| **Végpont** | **Metódus** | **Leírás** |
| /about\_us | get | információk megtekintése az oldalról |
| /appointments | get | Szabad időpontok megjelenítése |
| /appointments/reserveAppointment | post | Időpont lefoglalása |
| /appointments/sortCategories | post | Időpontok asziknron szűrése kategóriák szerint |
| /contanct | get | Kontakt információk megjelenítése |
| /data\_protection | get | Adatvédelmi információk megjelenítése |
| /home | get | Főoldal megjelenítése |
| /home/sortCategories | post | Időpontok asziknron szűrése kategóriák szerint |
| /login | get | Bejelntkező oldal megjelenítése |
| /login/userLogin | post | Felhasználói bejelnetkezés |
| /new\_appointment | get | Új időpont feltöltés felület megjelenítése |
| /new\_appointment/addAppointment | post | Új időpont feltöltése |
| /not\_found | get | Nem megfelelő url esetén megjelenő oldal |
| /service\_profile | get | Szolgáltatói profiloldal megjelenítése |
| /service\_profile/logout | post | Szoolgáltatói felhasználó kijelentkezés |
| /service\_profile/updateProfile | get | Szolgáltatói felhasználó profilmódosítási oldalának a megjelenítése |
| /service\_profile/updatePassword | post | Szolgáltatói felhasználó jelszómódosítás |
| /service\_profile/updateService | post | Szolgáltatói felhasználó cégadatainak módosítása |
| /service\_profile/deleteAppointment | post | Le nem foglalt időpont törlése a szolgáltatói profilban |
| /service\_profile/deleteProfile | post | Szolgáltatói fiók törlése |
| /question\_and\_answer | get | Q&A oldal megjelenítése |
| /registartion | get | Regisztrációs oldal megjelenítése |
| /registartion/userRegistartion | post | Regisztráció szolgáltatóknak és szolgáltatás keresőknek is |
| /seeker\_profile | get | Szolgáltatás kereső felhasználói profil megjelenítése |
| /seeker\_profile/logout | post | Kereső felhasználó kijelentkezés |
| /seeker\_profile/updateProfile | get | Kereső felhasználói profil módosító oldal megjelenítése |
| /seeker\_profile/updateProfile | post | Kereső felhasználói jelszómódosítás |
| /seeker\_profile/deleteProfile | post | Kereső felhasználói profil törlése |

## 4. Algoritmusok / kódok

### Osztályok

**HomeController**

Feladata: Ez az osztály a kezdőlaphoz kapcsolódó kérések kezelésére szolgál. Metódusokat tartalmaz a kezdőlap nézetének megjelenítéséhez és a kategóriák rendezéséhez.

Metódus:

sortCategories() - Ez a metódus kezeli az AJAX-kéréseket a időpontok kategóriák szerinti rendezésére. A kérésből lekérdezi a kategória azonosítóját, betölti a megadott kategóriához rendelkezésre álló időpontokat, és JSON-válaszként adja vissza az időpontokat.

**LoginController**

Feladata: Ez az osztály kontrollerként szolgál a bejelentkezéssel kapcsolatos kérések kezelésére. Metódusokat tartalmaz a bejelentkezési nézet megjelenítéséhez és a felhasználói bejelentkezési kísérletek feldolgozásához.

Metódus:

userLogin() - Ez a metódus kezeli a felhasználói bejelentkezési kísérleteket. Érvényesíti a POST kérésen keresztül kapott e-mail címet és jelszót, átirányítja a felhasználót egy üzenettel, ha bármelyik adat hiányzik, majd megpróbálja a felhasználót a User model login() metódusával bejelentkeztetni.

**User**

Feladata: Ez az osztály egy felhasználói modellt jelenít meg. A felhasználói hitelesítéshez kapcsolódó metódusokat tartalmaz, például a bejelentkezési funkciókat.

**Template**

Feladata: Ez az osztály felelős a sablonok/nézetek megjelenítéséért. Betölti a megadott nézetfájlt és átadja az adatokat a megjelenítéshez.

**RegistrationController**

Feladata: Ez az osztály a regisztrációval kapcsolatos kérések kezelésére szolgáló kontroller. Metódusokat tartalmaz a regisztrációs oldal megjelenítéséhez és a felhasználói regisztrációs kísérletek feldolgozásához.

Metódus:

userRegistration() - Ez a metódus kezeli a felhasználói regisztrációs kísérleteket. Érvényesíti a szükséges mezőket mind a normál felhasználók, mind a szolgáltatók esetében. Ezután megkísérli a felhasználó vagy a szolgáltató hozzáadását a felhasználói modell metódusainak segítségével. Siker esetén a felhasználót a kezdőlapra irányítja át, ellenkező esetben a regisztrációs oldalra irányítja vissza egy hibaüzenettel.

**Helper**

Feladata: Ez az osztály különböző segédfunkciókat biztosít, amelyeket az oldalon használnak. Tartalmaz metódusokat a flash üzenetek beállításához, a felhasználók üzenetekkel történő átirányításához és a navigációs adatok beállításához.

**AppointmentsController**

Ez az osztály aa idópontokkal kapcsolatos kérések kezelésére szolgáló vezérlő. Metódusokat tartalmaz az időpontok/hirdetések oldalának megjelenítéséhez.

Metódus:

index() - Ez a metódus az időpontok oldalának megjelenítéséért felelős. Inicializál egy Template objektumot, hogy betöltse az időpontok nézet template-jét (appointments\_view.php), és átadja neki a navigációs adatokat a Helper::setNav() metódus segítségével.

**Service**

Feladata: Ez az osztály egy szolgáltatást jelenít meg, és metódusokat biztosít az adatbázisban lévő, szolgáltatással kapcsolatos adatokkal való interakcióhoz.

Metódusok:

\_\_construct() **-** DatabaseHandler objektum inicializálása az adatbázis-kapcsolat létrehozásához.

loadAvailableAppointments() - A rendelkezésre álló időpontok lekérése az adatbázisból, kiszűrve a már lejárt vagy túl hamar foglalható időpontokat.

loadAvailableAppointmentsForCategory(int $categoryId) - Egy adott kategóriához tartozó elérhető időpontok lekérdezése, ugyanazokat a szűrési feltételeket alkalmazva, mint a loadAvailableAppointments().

getServiceData() - Az aktuális felhasználó szolgáltatási adatainak lekérdezése.

calculateAppointmentTimeInArray(array $appointments) - Kiszámítja a találkozók végidejét, és a megadott tömbben formázza a találkozóidő-adatokat.

getFreeAppointmentsOfProvider() - Az aktuális szolgáltató szabad időpontjainak lekérdezése, kiszűrve a már lejárt vagy túl hamar foglalható időpontokat.

deleteAppointment(int $appointmentId) - Töröl egy időpontot az adatbázisból a megadott időpont-azonosító alapján.

getUpcomingReservationsOfProvider() - Az aktuális szolgáltató közelgő idópontjait kéri le, kiszűrve a már lejárt vagy túl hamar foglalható időpontokat.

getPastReservationsOfProvider() - Az aktuális szolgáltató korábbi foglalásait kéri le.

getReservationsOfUser() - Az aktuális felhasználó közelgő foglalásait kéri le.

getPastReservationsOfUser() - Az aktuális felhasználó korábbi foglalásait kéri le.

addAppointment(array $appointmentData) - Új időpontot ad hozzá az adatbázishoz a megadott időpontadatok alapján. Ha sikeres, átirányítja a felhasználót a szolgáltatás profil oldalára, ellenkező esetben az új időpont oldalra irányítja át egy hibaüzenettel.

### Függvények

**loadProviderForm()**

Funkció: A szolgáltatókkal kapcsolatos űrlapmezők láthatóságának átkapcsolására szolgál.

**sortCategories(categoryId)**

Funkció: AJAX-kérést küld a szervernek a megadott kategóriaazonosító alapján történő időpont-kategóriák rendezéséhez. Frissíti a DOM-ot a rendezett kategóriákkal.

**validateField(field)**

Funkció: Validálja az űrlapmezőket olyan meghatározott kritériumok alapján, mint az e-mail formátum, jelszókövetelmények és szolgáltatóspecifikus mezők.

**showError(field, message)**

Funkció: Megjeleníti az érvénytelen űrlapmezőkre vonatkozó hibaüzeneteket.

**hideError(field)**

Funkció: Elrejti az érvényes űrlapmezők hibaüzeneteit.

**redirectToPage(destinationPage)**

Funkció: Egy késleltetés után irányítja át a felhasználót a megadott oldalra.

**deleteAppointment(appointmentId)**

Funkció: AJAX-kérést küld a szervernek egy adott időpont törlésére. Frissíti a DOM-ot a megmaradt időpontokkal.

**displayProfileDel()**

Funkció: Bekapcsolja a profil törlését megerősítő kérés láthatóságát.

### AJAX-kérések

**sortCategories()**

POST-kérést küld a szervernek a kiválasztott kategória azonosítója alapján történő rendezéshez, kezeli a siker- és hibaválaszokat a DOM frissítéséhez, és hiba esetén figyelmezteti a felhasználót.

**deleteAppointment()**

POST-kérést küld a szervernek egy adott időpont törlésére a megadott időpont-azonosító alapján, frissíti a DOM-ot a megmaradt találkozókkal és hiba esetén figyelmezteti a felhasználót.

### Eseménykezelők

**Document Ready Function**

Érvényesíti az űrlapmezőket a blur eseményekre, kezeli a szolgáltatói jelölőnégyzet-váltási eseményt a szolgáltatóspecifikus űrlapmezők átkapcsolásához és érvényesítés után elküldi a regisztrációs űrlapot.

**Registration Form Submission kezelő**

Érvényesíti az űrlap mezőit a beküldés előtt és figyelmezteti a felhasználót, ha valamelyik kötelező mező üres.

### További funckiók

**Jelszó-kritériumok megjelenítése**

A jelszómezőre kattintva jelszófeltételek megjelenítése.

**Dinamikus űrlapmező érvényesítés**

Dinamikusan érvényesíti az űrlapmezőket, amint a felhasználó interakcióba lép velük.

## 5. Felhasználói felület

A weboldal felhasználói felületéhez HTML, CSS nyelveken kívül a Bootrstrap 5.3.2-es verziójú keretrendszert alkalmaztunk, hogy a felületet reszponzívvá tegyük.

**header\_view**

komponensek:

* meta adatok
* bootstrap linkek

ugrási lehetőségek: -

**guest\_nav\_view**

komponensek:

* rólunk link
* kapcsolat link
* regisztráció link
* bejelentkezés link
* findEM logo

ugrási lehetőségek:

* about\_us
* contact
* registration
* login

**seeker\_nav\_view:**

komponensek:

* szabad időpontok link
* rólunk link kapcsolat link
* kapcsolat link
* saját profil link
* kijelentkezés link

ugrási lehetőségek:

* appointments
* about\_us
* contact
* seeker\_profile
* seeker\_profile/logout

**provider\_nav:**

komponensek:

* szabad időpontok link
* időpont feltöltés link
* rólunk link
* kapcsolat link
* saját profil link
* kijelentkezés link

ugrási lehetőségek:

* appointments
* new\_appointment
* about\_us
* contact
* service\_profile
* service\_profile/logout

**footer\_view**

komponensek:

* kontakt konténer
* az oldal tetejére link
* rólunk link
* GYIK link
* adatvédelem link
* Facebookre ugrás link
* Instagramra ugrás link
* TikTokra ugrás link

ugrási lehetőségek:

* home
* about\_us
* question\_and\_answer
* data\_protection
* facebook.com
* instagram.com
* tiktok.com

**about\_us\_view**

komponensek:

* logo
* rólunk konténer
* carousel

**appointments\_view**

komponensek:

* szolgáltatások logo
* kategória gombok
* felhasználók által feltöltött szabad időpontok megjelenítése szekció

**contact\_view**

komponensek:

* kontakt logo
* Írj nekünk! szekció

**data\_protection\_view**

komponensek:

* adatvédelem logo
* adatkezelési tájékoztató szekció

**home\_view**

komponensek:

* szolgáltatások logo
* lefoglalható szolgáltatások szekció
* szolgáltatás kategória gombok
* carousel
* regisztráció gomb

ugrási lehetőségek:

* registration

**login\_view**

komponensek:

* bejelentkezés logo
* bejelentkezés űrlap
* regisztráció logo
* regisztráció gomb

ugrási lehetőségek:

* registration

**new\_appointment\_view**

komponensek:

* új időpont feltöltése űrlap
* időpont feltöltése gomb

**profile\_provider\_view**

komponensek:

* felhasználói adatok megjelenítése szekció
* felhasználói profil opciói (adatok módosítása, kijelentkezés, profil törlése)
* felhasználó szabad időpontjainak megjelenítése szekció
* új időpont felvitele gomb
* lefoglalt időpontok megjelenítése szekció
* korábban lefoglalt időpontok megjelenése szekció

ugrási lehetőségek:

* provider\_update
* new\_appointment

**profile\_seeker\_view**

komponensek:

* felhasználói adatok megjelenítése szekció
* felhasználói profil opciói szekció   
  (jelszó módosítása, kijelentkezés, profil törlése)
* lefoglalt időpontok megjelenítése szekció
* korábban lefoglalt időpontok megjelenése szekció

ugrási lehetőségek:

* seeker\_update

**provider\_update\_view**

komponensek:

* adatok módosítás űrlap   
  (jelszó módosítása, szolgáltatás adatainak módosítása)

**question\_and\_answer\_view**

komponensek:

* gyakori kérdések és válaszok szekció

**registration\_view**

komponensek:

* regisztráció logo
* regisztációs űrlap
* belejentkezés logo
* bejelentkezés gomb

ugrási lehetőségek:

* login

**seeker\_update\_view**

komponensek:

* adatok módosítás űrlap   
  (jelszó módosítása)

## 6. Fejlesztési lehetőségek

A programot a későbbiekben tovább bővíteni, gazdagítani számos funkcionalitással optimalizációval vagy biztonsági fejlesztések bevezetésével is megtehetjük akár.

A jövőbeli lehetséges és említett fejlesztések közé tartozik az egyes szolgáltatók kedvencnek jelölése a felhasználói profilból, értékelési lehetőségek létrehozása és a szolgáltatási hely regisztrációjának kiválasztása is.

A projekt jövőbeli fejlesztéseiben hangsúlyt kaphatnának még a kiterjesztett felhasználói profilok, amelyek engedélyezésével a felhasználók részletes profiladatokat tudnának megadni, mint például érdeklődési kör, korábbi tevékenységek, preferenciák, így a platform személyre szabottabb ajánlatokat és tartalmakat tudna nyújtani. A felhasználói élmény javítása érdekében interaktív felhasználói útmutatókat és segítségeket vezethetnénk be, hogy azok segítségével könnyebben és hatékonyabban legyen használható az oldal.

Hasznos lenne még a jövőben az úgynevezett teljesítményoptimalizáció, amellyel átfogó felülvizsgálatot végezhetnénk az alkalmazás teljesítményével kapcsolatban, hogy gyorsabb betöltési időket és kevesebb erőforrásfelhasználást érjünk el.

Zárásként, hogy a biztonságra is gondoljunk, a projekt további fejlesztési irányai között szerepelhetne a kétlépcsős azonosítás bevezetése, amely révén egy plusz réteget adhatunk a felhasználók fiókjainak védelméhez, ezzel megnehezítve a jogosulatlan hozzáférést. A szigorúbb adatvédelmi szabályozásoknak való megfelelés érdekében folyamatosan felülvizsgálhatnánk és frissíthetnénk az adatvédelmi irányelveket és az adatvédelmi beállításokat megfelelvén a GDPR szabályoknak.



# III. Felhasználói dokumentáció

Üdvözöljük a FindEM weboldal használatán!

A szolgáltatásfoglaló weboldal egy olyan platform, amely lehetővé teszi számunkra, hogy könnyedén és gyorsan foglalhassunk különböző szolgáltatásokat, mint például kozmetikust, tetoválást, fodrászt, de vízvezetékszerelést vagy akár személyi edzést is és még sok mást.

A weboldal használatához az alábbi rendszerek és eszközök szükségesek:

- Számítógép vagy mobil eszköz

- Internetkapcsolat

- Webböngésző (ajánlott: Google Chrome, Mozilla Firefox, Safari)

A szolgáltatásfoglaló weboldalhoz nem szükséges külön szoftver letöltése és telepítése, csak egy internetkapcsolattal rendelkező eszköz és egy támogatott webböngésző szükséges.

Az oldal elérhető a következő webcímen: findem.hu

Az oldal használatához regisztráció ajánlott. Kattintson a "Regisztráció" gombra, majd töltse ki az adatlapot a szükséges információkkal. A regisztráció sikeres befejezése után jelentkezzen be az e-mail címe és jelszava megadásával.

A foglalás egyszerű és gyors. Keresse meg a kívánt szolgáltatást a kategóriák között, vagy válasszon a főoldalon található legfrissebben feltöltött időpontok közül. Adja meg a szükséges információkat, majd erősítse meg a foglalást.

Ha bármilyen kérdése vagy problémája merülne fel az oldal használata során, kérjük, vegye fel velünk a kapcsolatot a *contact@findem.hu* e-mail címen, amelyet ugyanúgy megtalál a "Rólunk" menüpont alatt.

Kérjük, olvassa el az Adatkezelási Tájékoztatót, mielőtt használná az oldalt. Ezt az "Adatvédelem" menüpont alatt találja.

Reméljük, hogy ez a dokumentáció segítséget nyújt az oldal használatában. Amennyiben további kérdései lennének, forduljon hozzánk bizalommal!



# IV. Tesztelés

## 1. Manuális tesztelés

Első

Tesztelt folyamat:

A regisztrációs űrlap megnyitása, majd a közterület megnevezése és típusa mező kitöltése "Kanizsai Dorottya utca" értékkel.

Tesztelt környezet:

Operációs rendszer: Windows 10

Böngésző: Microsoft Edge version 123.0.2420.81

Tesztelési eredmény:

Hibás viselkedés: Nem kezeli helyesen a szóköz karaktert a mezőben.

Várható viselkedés: A programnak képesnek kell lennie kezelni a szóköz karaktert a mezőben, és nem szabad hibát jeleznie.

Tesztelési megjegyzések:

A mezőnek képesnek kell lennie elfogadni és helyesen kezelni a szóköz karaktert.

Az űrlapnak egyértelműen tájékoztatnia kellene a felhasználót arról, hogy a szóközöket is beleértve kell megadnia a címet.

Tesztelési Javaslat:

Ellenőrizni kell a közterület megnevezése és típusa mező validációját, hogy a szóköz karaktert is elfogadja.

Ellenőrizni kell a felhasználói felületen lévő információkat, hogy meggyőződjön arról, hogy a felhasználók tájékoztatva vannak a szükséges adatbeviteli formátumról.

Második

Tesztelt folyamat:

A regisztrációs űrlap megnyitása és a következő adatok megadása:

Jelszó mezők kitöltése az előírt formátum szerint.

Tesztelt környezet:

Operációs rendszer: Windows 10

Böngésző: Microsoft Edge verzió 123.0.2420.81

Tesztelési eredmény:

Helyes viselkedés: A jelszómező helyesen kezeli a speciális karaktert.

Várható viselkedés: A programnak képesnek kell lennie érzékelni a speciális karaktert a jelszómezőben, és hibát kell jeleznie.

Tesztelési megjegyzések:

A program képes érzékelni a speciális karaktert a jelszómezőben, és hibát jelez. Az űrlap egyértelműen tájékoztatja a felhasználót arról, hogy a jelszót milyen követelmények alapján kell megadnia.

Harmadik

Tesztelt folyamat:

Az oldal bejelentkezési űrlapjának megnyitása és a következő adatok megadása:

E-mail cím

Jelszó

Tesztelt környezet:

Operációs rendszer: Windows 10

Böngésző: Microsoft Edge verzió 123.0.2420.81

Tesztelési eredmények:

Hibás e-mail cím kezelése:

E-mail cím formátum kezelése:

Helyes viselkedés: Az oldal figyelmezteti a felhasználót, ha az e-mail cím formátuma hibás (hiányzó "@" karakter).

Várható viselkedés: Az oldal tájékoztatja a felhasználót, ha az e-mail cím formátuma hibás, és megjelenít egy informatív hibaüzenetet.

Üres mezők kezelése:

Helyes viselkedés: Az oldal figyelmezteti a felhasználót, ha valamelyik mező üresen marad a bejelentkezési űrlapon.

Várható viselkedés: Az oldal tájékoztatja a felhasználót, ha valamelyik mező üresen marad, és megjelenít egy informatív hibaüzenetet.

Jelszó validáció:

Helyes viselkedés: Az oldal figyelmezteti a felhasználót, ha a megadott jelszó nem felel meg a követelményeknek (pl. nincs benne szám, vagy túl rövid).

Várható viselkedés: Az oldal tájékoztatja a felhasználót, ha a megadott jelszó nem felel meg a követelményeknek, és megjelenít egy informatív hibaüzenetet.

További lépések:

Ha a bejelentkezés sikeres volt, az oldal átirányítja a felhasználót a bejelentkezett felhasználó profiljára

Várható viselkedés: Az oldal figyelmezteti a felhasználót, ha a megadott jelszó nem felel meg a követelményeknek, és megjelenít egy informatív hibaüzenetet a megfelelő formátumról.

## 2. Terheléses vizsgálat

A terheléses vizsgálat célja az volt, hogy értékeljük a rendszer teljesítményét és stabilitását nagyobb adatmennyiség mellett.

### 3. Környezet

Url: localhost:8000

Operációs rendszer: Windows 10

Böngésző: Microsoft Edge version 123.0.2420.81

Tesztmenet

Adatok előkészítése: Felkészítettünk egy jelentős mennyiségű tesztadatot az adatbázisban.

Terhelés beállítása: Növeltük a terhelést a rendszeren, például megnöveltük a időpontok számát vagy az adatok beviteli sebességét.

Tesztvégrehajtás: Végrehajtottuk a terheléses tesztet az előre meghatározott paraméterekkel.

Rendszerfigyelés: Folyamatosan figyelemmel kísértük a rendszer teljesítményét és stabilitását a terhelés alatt.

Adatok rögzítése: Mértük a rendszer válaszidejét, a tranzakciók sikerességi arányát és egyéb fontos teljesítmény-mutatókat.

Eredmények

A rendszer válaszideje átlagosan 2 másodperc volt a terheléses teszt során.

A tranzakciók sikerességi aránya 100% volt a terheléses teszt során.

A rendszer teljesítménye stabil maradt, és nem tapasztaltunk jelentős hibákat vagy összeomlásokat.

Következtetések

A terheléses teszt során a rendszer hatékonyan kezelte a nagyobb adatmennyiséget, és stabil teljesítményt nyújtott a terhelés alatt is.

# V. Összegzés

A FindEM weboldal fejlesztése során nagy hangsúlyt fektettünk a könnyű és hatékony foglalási folyamatok biztosítására, valamint arra, hogy egy összetartó közösséget teremtsünk a szolgáltatók és az ügyfelek között. A projekt során számos nehézséggel és érdekességgel találkoztunk, amelyeket röviden összefoglalok:

A fejlesztési folyamat során az emberek igényeinek felismerése és az egyszerű, gyors foglalási folyamatok kialakítása volt a fő motivációja a projektnek.

A felhasznált technológiákról a következőket mondhatjuk el: A PHP-t választottuk fő nyelvként, mivel dinamikus webes alkalmazásokhoz kiválóan alkalmas. Az adatbázis-kezeléshez SQL Server-t és PHPMyAdmin-t használtunk, amelyek jól integrálhatóak és biztonságosak.

Fejlesztői környezetek a Visual Studio Code-ot választottuk, mivel könnyen testreszabható és támogatja a PHP fejlesztést.

Az algoritmusok / kódok résznél az osztályok és függvények részletes leírása mellett AJAX-kéréseket és eseménykezelőket is említük, mivel ezek is szerepeltek a fejlesztés során.

Vegül több verzió után, az adatbázis részletes felépítése és terve is megvalósult, amelynek segítségével könnyen kezelhetővé váltak az adatok.

Amiket nehézségként vagy érdekességként említhetünk, például a validáció és hibakezelés. A felhasználói inputok validálása és a hibák kezelése volt az egyik legnagyobb kihívás, különösen a különböző felhasználói szerepek figyelembevételével. Ezen kívül, a szerveroldali feldolgozás hatékonyságának növelése és a szolgáltatások dinamikusságának biztosítása jelentős figyelmet igényelt.

Fontos volt számunkra viszont a felhasználói élmény optimalizálása, ide értve az oldal gyorsaságának és használhatóságának optimalizálása érdekében folyamatosan finomhangoltuk a felhasználói felületet és a funkciókat.

A GDPR követelményeinek való megfelelés és a felhasználók adatainak biztonságos kezelése leginkább a továbbfejlesztési lehetőségekben fellelhető. A biztonsági fejlesztések közé tartozik a kétlépcsős azonosítás és a GDPR szabályoknak való megfelelés, amelyek további biztonsági réteget adnának az oldalnak.

A jövőbeli fejlesztési lehetőségeknél meg kell említenünk, a felhasználói profilok bővítését, a felhasználói profilok részletesebbé tételét és a személyre szabottabb ajánlatok megjelenítését. Itt egy értékelési rendszer is hasznos lenne, hogy a felhasználók számára lehetőség biztosítsunk arra, hogy értékeljék és véleményezzék a szolgáltatásokat.

A teljesítményoptimalizációt is figyelembe kell vennünk, hasznos lenne az oldal betöltési sebességének és hatékonyságának növelése.

Összességében, a FindEM fejlesztése során arra törekedtünk, hogy egy modern és felhasználóbarát platformot hozzunk létre, amely könnyedén használható és hatékonyan kapcsolja össze a szolgáltatókat és az ügyfeleket.