FindEM

Fejlesztői dokumentáció

2024

Készítők:

Ladányi Anna

Maász Martina Rita

Tózer Kitti

Tartalomjegyzék

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, CSS és Bootstrap: Az alkalmazás felhasználói felületének kialakításához HTML-t használtunk, amelyet saját CSS stíluslappal és Bootstrap keretrendszerrel egészítettünk ki, 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.

Canva: Képek és grafikák generálásához, például GIF-ekhez és logókhoz is használtuk a Canva platformot.

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 választott 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**
  A szolgáltatások csak egy részét látják és csak kevés információt.
- **Szolgáltatás kereső**
  Regisztráció és bejelentkezés után meg tudja nézni a feltöltött időpontokat és foglalni is tud. Profiljában látja az aktiv és a korábbi foglalásait.
- **Szolgáltató**
  Regisztráció és bejelentkezés után tud feltölteni időpontokat, meg tudja nézni az összes hirdetést de foglalni nem tud. Profiljában törölni tudja a még nem foglalt időpontokat valamint meg tudja nézni az aktív és a korábbi foglalásokat a szolgáltatásánál.

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

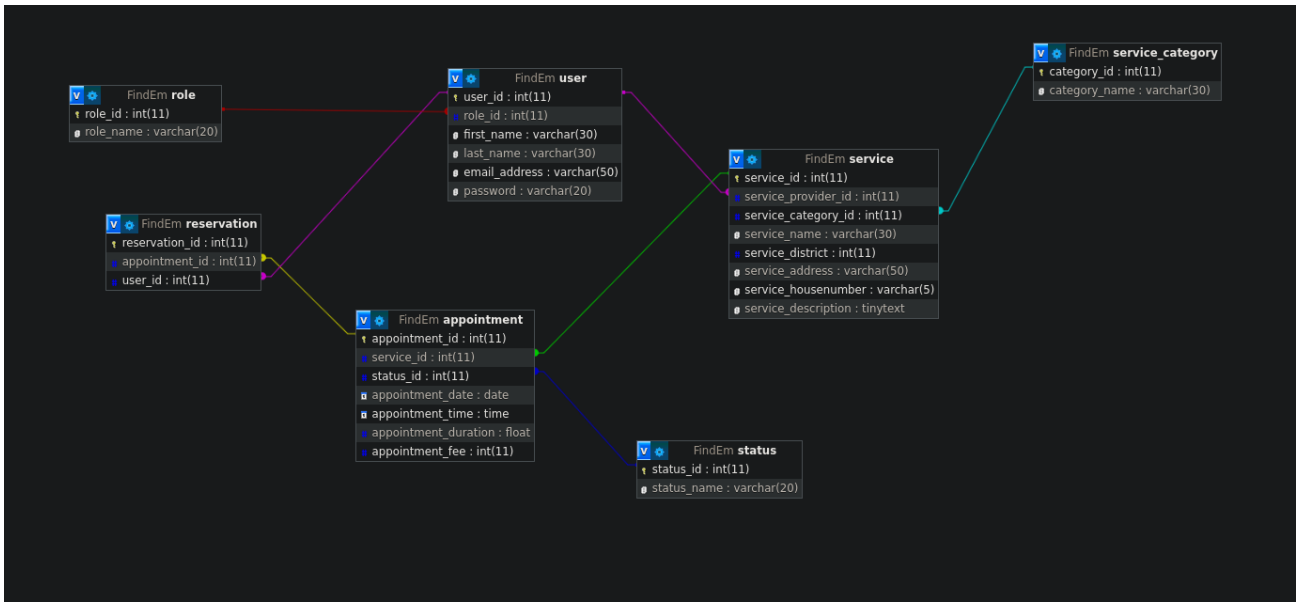
Funkciók	Vendég	Regisztrált kereső	Regisztrált szolgáltató
Kezdőoldal	X	X	X
Menü	X	X	X
Regisztráció	X		
Bejelentkezés		X	X
Szabad időpontok		X	X
Saját profil adatok		X	X
Saját lefoglalt időpontok		X	X
Új időpont feltöltése			X
Profil módosítása		X	X
Időpont kereső		X	X
Kijelentkezés		X	X
Saját szolgáltatások			X
Saját szabad szolgáltatás időpontok			X
Valaki által lefoglalt időpontok			X
Időpont törlése			X
Profil törlése		X	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 aszinkron szűrése kategóriák szerint
/contanct	get	Kontakt információk megjelenítése
/data_protection	get	Adatvédelmi információk megjelenítése
/home	get	Főoldal megjelenítése
/home/sortCategories	post	Időpontok aszinkron szűrése kategóriák szerint
/login	get	Bejelentkező 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	Szolgáltatói 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
/registration	get	Regisztrációs oldal megjelenítése
/registration/userRegistration	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 időpontokkal 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. A képek a canva.com weboldal segítségével készültek.

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)

not_found_view

komponensek:

- not found logo
- az oldal nem található szekció
- vissza a főoldalra gomb

ugrási lehetőségek:

- home

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 Adatkezelési 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.