

Firestarter Akadémia

Vizsgaremek

„Alakítsd át az életed velünk.”

Tartalomjegyzék

Tartalomjegyzék

Előszó.....	3
Fejlesztői dokumentáció.....	3
Szoftver – Hardver	4
Az oldal tesztelése Cypress segítségével	6
Az oldalak fejlesztése	8
A főoldal / index.php	10
Regisztráció/Bejelentkezés	13
Az oktatói kezelői felület.....	17
Functions.php.....	19
Adminisztrátori Felület	20
Személyes profil.....	21
Összefoglalás.....	22
Források.....	23
Készítők.....	24

Előszó

...

Fejlesztői dokumentáció

...

Üdvözljük a Firestarter Akadémia Weboldal projekt dokumentációjában. Ez a projekt egy modern, átfogó platformot biztosít a Firestarter Akadémia számára, amelynek célja, hogy összekösse az oktatókat és a fejlődni vágyó egyéneket különféle önismereti, üzleti és kommunikációs képzéseken keresztül.

A rendszer PHP alapú backendre és MySQL adatbázisra épül, lehetővé téve a felhasználók számára a zökkenőmentes regisztrációt (hagyományos módon vagy Google OAuth segítségével), bejelentkezést, valamint a szolgáltatások (mint például Life Coaching, Business Coaching, Stresszkezelési Workshopok, Karriertervezés és Kommunikációs Tréningek) böngészését és azokra történő időpontfoglalást egy interaktív naptáron keresztül.

A platform rezponzív kialakítással, sötét és világos téma támogatással, valamint animációkkal gazdagított felhasználói felülettel rendelkezik. Emellett robusztus adminisztrációs és oktatói felületeket is kínál a felhasználók, jogosultságok, szolgáltatások és foglalások hatékony kezeléséhez.

Ez a dokumentáció célja, hogy részletes betekintést nyújtson a projekt felépítésébe, funkcióiba és technikai megvalósításába, segítve ezzel a fejlesztőket és az érdeklődőket a rendszer megértésében és használatában.

Szoftver – Hardver

Hardver – Egy személyi számítógépen végeztük a munkánkat.

Specifikációk:

- CPU –Ryzen 3 Pro 5350G
- RAM - 8Gb
- ROM – Kingston SSD 480Gb

Operációs rendszer - Microsoft Windows 11 professional x64.

Szoftver – A kódot mindketten **Visual Studio Code** keretrendszer segítségével írtuk (mind a frontendet mind a backendet).

Adatbázis - MySQL

A projekt adatbázis-kezelő rendszereként a **MySQL**-t választottuk. Ez a széles körben elterjedt, megbízható adatbázis-kezelő megfelelt a projekt követelményeinek, és már rendelkezettünk a használatához szükséges tapasztalattal.

Az adatbázis adminisztrációját és a táblák kezelését a **phpMyAdmin** webes felület segítségével végeztük, ami egyszerűsítette az adatszerkezet kialakítását és a kezdeti adatok feltöltését.

Frontend - HTML, CSS, JavaScript

A felhasználói felület (frontend) megvalósításához a webfejlesztés alaptechnológiáit használtuk: **HTML5**, **CSS3** és **JavaScript**.

- A **HTML** biztosítja a tartalom strukturális vázát.

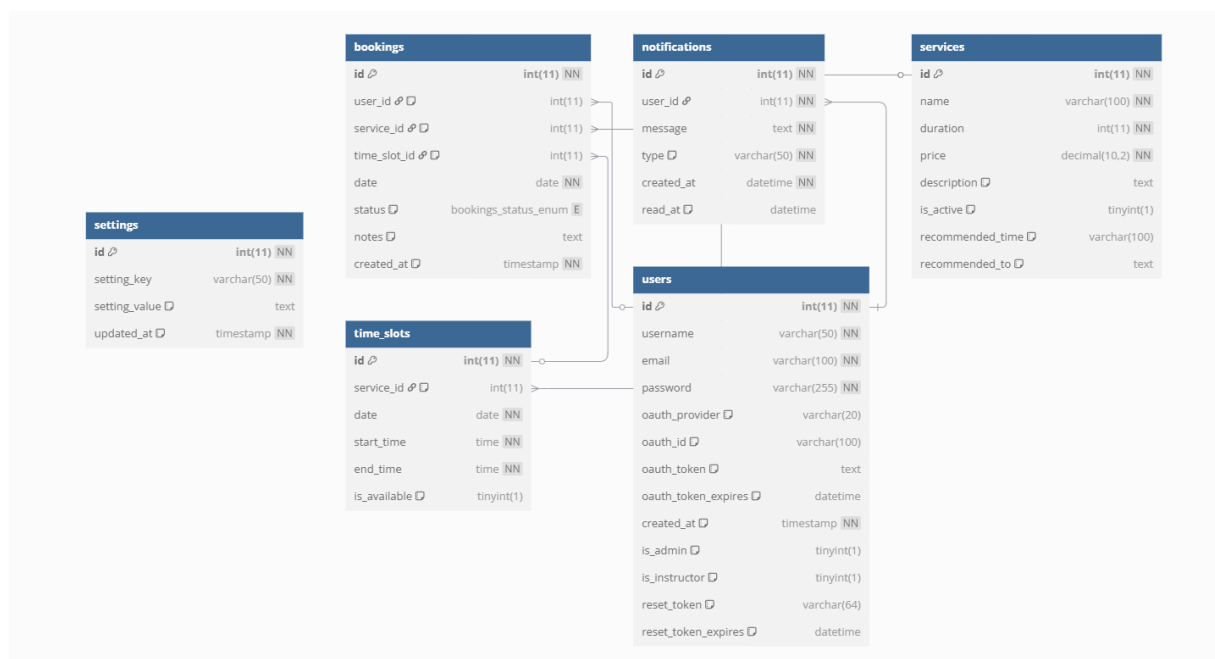
- A **CSS** felelős a vizuális megjelenésért, a dizájnért és a reszponzív kialakításért (különböző képernyőméretekhez való alkalmazkodás).
- A **JavaScriptet** a dinamikus funkciók, felhasználói interakciók (pl. űrlapvalidáció, animációk) és az aszinkron adatkérések megvalósítására alkalmaztuk.

Ez a technológiai kombináció lehetővé tette egy modern, felhasználóbarát és gyorsan fejleszthető felület kialakítását.

Backend - PHP

A backend futtatásához a **PHP**-t választottuk. Népszerűsége, széles körű tárhely-támogatottsága és a rendelkezésre álló kiterjesztések miatt jó választásnak bizonyult.

Az adatbázis-műveletekhez a **PDO (PHP Data Objects)** kiterjesztést használtuk. Ez egységes és biztonságos módszert kínál a MySQL adatbázissal való kommunikációra. A PHP szorosan integrálható a frontend technológiákkal, lehetővé téve a szerver által feldolgozott adatok dinamikus megjelenítését a felhasználó számára.



Az oldal tesztelése Cypress segítségével

```
Running: test.cy.js (1 of 1)

User Registration Test
  ✓ should register a new account (4741ms)

Login Tests
  ✓ should log in as regular user (1460ms)
  ✓ should log in as instructor (1482ms)

Instructor Panel Test
  ✓ should create a service (2544ms)
  ✓ should delete the service (589ms)

Admin Panel Test
  ✓ should give the test account instructor permissions (844ms)
  ✓ should take away the test account instructor permissions (592ms)
  ✓ should delete the account (440ms)

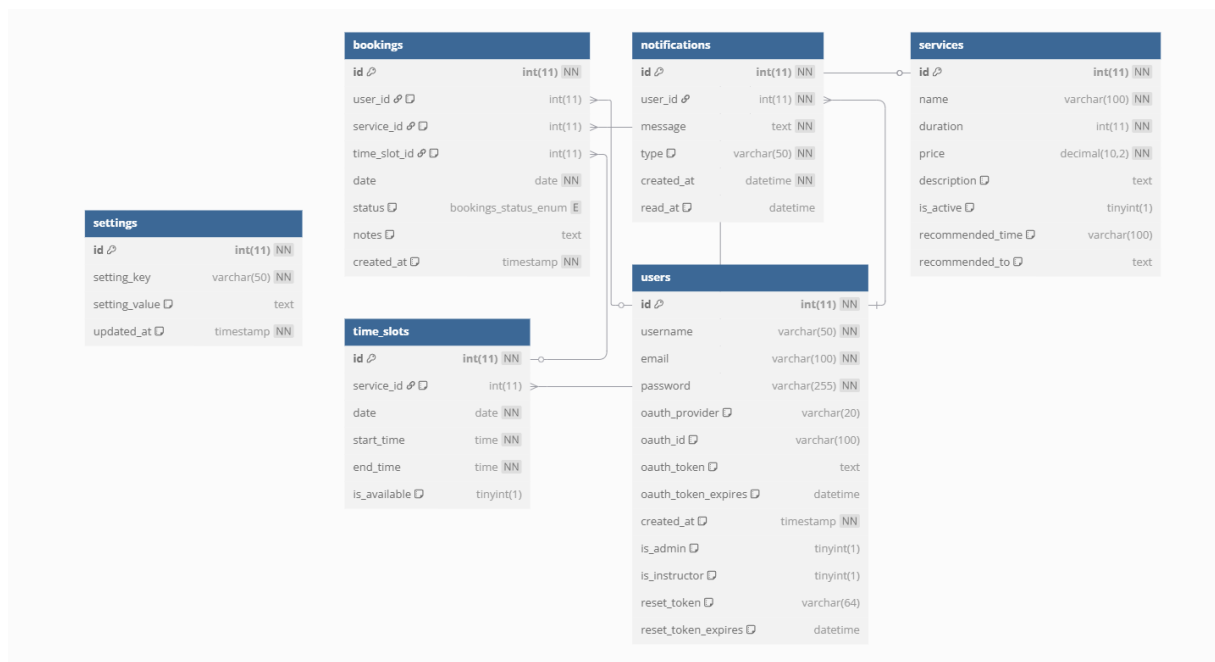
8 passing (18s)
```

Az oldal teszteléséhez a **Cypress** nevű nyílt forráskódú, JavaScript alapú eszközt használtuk. A tesztelés során a következő fő funkciókat ellenőriztük:

1. Új fiók regisztrációja
2. Belépés regisztrált fiókkal
3. Belépés oktatói fiókkal
4. Új szolgáltatás létrehozása
5. Az új szolgáltatás törlése
6. A teszt fiók oktatói jogosultság adása
7. Oktatói jogosultság elvonása a teszt fióktól
8. Teszt fiók törlése

A tesztek lefuttatása

1. Győződjön meg róla, hogy a weboldal fut és a [Node.js](#) telepítve van.
 2. Lépjen a projekt gyökérkönyvtárába a terminálban.
 3. Futtassa az „npm install” parancsot a szükséges csomagok (beleértve a Cypress-t) telepítéséhez.
 4. Nyissa meg a Cypress interaktív tesztfutatót az „npx cypress open” paranccsal. Innen kiválaszthatja és futtathatja a teszteket.
- 4.5 Alternatívként futtassa le az összes tesztet a háttérben az „npx cypress run” paranccsal.



Az oldalak fejlesztése

Oldalak: Az első lépés a szükséges oldalak azonosítása volt. A csapatunk összegyűjtötte, hogy milyen oldalakat szeretnénk, például:

Főoldal

Képzések

Rólunk

Bejelentkezés/Regisztráció

Profil oldal

Admin panel

Funkciók: Minden oldalhoz hozzárendeltük a szükséges funkciókat, mint például:

Felhasználói regisztráció és bejelentkezés

Időpontfoglalás

Szolgáltatások kezelése

Adatbázis Tervezése

Adatbázis Szerkezet: A következő lépés a teljes adatbázis megtervezése volt. A csapatunk átbeszélte, hogy milyen táblákra van szükség, például:

users: Felhasználói adatok

services: Szolgáltatások

bookings: Foglalások

time_slots: Időpontok

Kapcsolatok: Megvitattuk a táblák közötti kapcsolatokat, például, hogy a bookings tábla hogyan kapcsolódik a users és services táblákhoz.

```
<script>
  document.body.setAttribute('data-theme', 'dark');
  AOS.init({
    duration: 1000,
    once: true
  });

  function toggleTheme() {
    const body = document.body;
    const button = document.querySelector('.theme-switch');
    const modeText = button.querySelector('.mode-text');

    if (body.getAttribute('data-theme') === 'dark') {
      body.removeAttribute('data-theme');
      modeText.textContent = '☀';
      localStorage.setItem('theme', 'light');
    } else {
      body.setAttribute('data-theme', 'dark');
      modeText.textContent = '🌙';
      localStorage.setItem('theme', 'dark');
    }
  }

  window.addEventListener('DOMContentLoaded', () => {
    const savedTheme = localStorage.getItem('theme');
    const button = document.querySelector('.theme-switch');
    const modeText = button.querySelector('.mode-text');

    if (savedTheme === 'dark') {
      document.body.setAttribute('data-theme', 'dark');
      modeText.textContent = '🌙';
    }
  });
</script>
```

Ez a javascript rész minden oldalon megtalálható sötét vagy fehér mód kiválasztásához.

A főoldal / index.php

Adatbázis-kapcsolat és munkamenet:

Ez a rész:

- Elindítja a PHP munkamenetet
- Kapcsolódik az adatbázishoz
- Ellenőrzi a felhasználó bejelentkezését
- Lekérdezi a szolgáltatásokat az adatbázisból

```
<?php
session_start();

require_once 'db_config.php';
require_once 'functions.php';

$user = isLoggedIn($pdo);
$services = getServices($pdo);
?>
```

Fejléc és navigáció:

A fejléc rész:

- Megjeleníti a logót
- Navigációs linkeket tartalmaz
- Dinamikusan változik a felhasználó jogosultságai alapján
- Belépési/Regisztrációs lehetőséget nyújt nem bejelentkezett felhasználóknak
- Témaváltó gombot tartalmaz



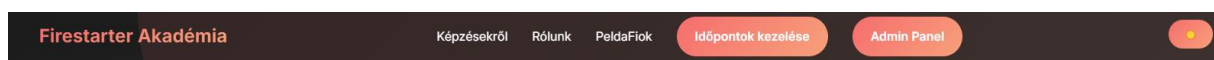
```

<body>
  <header>
    <div class="header-content">
      <div class="logo">Firestarter Akadémia</div>
      <nav>
        <a href="./kepzeseink.php" id="kepzesekek">Képzésekről</a>
        <a href="./rolunk.html" id="rolunk">Rólunk</a>
        <?php if (!$user) {
          echo '<a class="login-button" href="./login.php">Belépés/Regisztráció</a>';
        } else {
          echo '<a href="./profile_page.php">' . htmlspecialchars($user['username']) . '</a>';
        }
        if (isInstructor($user)) {
          echo '<a href="foglalas.php" class="login-button">Időpontok kezelése</a>';
        }
        if (isAdmin($user)) {
          echo '<a href="admin.php" class="login-button">Admin Panel</a>';
        }
      ?>
      <button class="theme-switch" onclick="toggleTheme()">
        <span class="mode-text">☀</span>
      </button>
    </nav>
  </div>
</header>

```

Ez a rész:

- Ellenőrzi hogy bevan-e lépve
- Rendszer ellenőrzi az oktatói és admin jogosultságot



Főszekció (hero section):

A főszekció:

- Feltűnő, figyelemfelkeltő bemutató terület
- Rövid bemutatkozás
- Animációt használ a tartalom megjelenítésére (AOS library)
- Tartalmazza a kapcsolattartási információt

```

<section class="hero">
  <div class="hero-content" data-aos="fade-up">
    <h1>Alakítsd át az életed velünk</h1>
    <p>Szakértő segítség az önfejlesztésben, karrierépítésben és kapcsolataid fejlesztésében</p>
    <div class="contact-info">
      <p>Kapcsolat: +36 70 631 3311</p>
    </div>
  </div>
</section>

```


```

<section class="services">
  <?php
  $delay = 100;
  foreach ($services as $service):
    $imageUrl = "../pictures_from_training_courses/" . $service['name'];
  >>
  <div class="service-card" data-aos="fade-up" data-aos-delay="<?php echo $delay; ?>">
    ">
    <h3><?php echo htmlspecialchars($service['name']); ?></h3>
    <p><?php echo htmlspecialchars($service['description']); ?></p>
    <a href="/kepzeseink.php#<?php echo $service['id']; ?>" class="login-button">Részletek</a>
    <a href="/idopont.php#<?php echo $service['id']; ?>" class="login-button">Jelentkezés</a>
  </div>
  <?php
  $delay += 100;
  endforeach;
  >>
</section>

```

Ez a rész:


- Jeleníti meg az adatbázisban lévő szolgáltatásokat



Business Coaching Tréning

Fejleszd vállalkozói képességeidet és válj vezetővé a szakmádban! Ez a képzés segít abban, hogy jobb döntéseket hozz, hatékonyabban kommunikálj és felkészültebben építsd a vállalkozásodat.


Részletek Jelentkezés



Hatékony Kommunikáció és Konfliktuskezelés

Sajátítsd el azokat a kommunikációs technikákat, amelyek segítenek a személyes és szakmai kapcsolataid fejlesztésében. A képzés során gyakorlati példákon keresztül tanulhatsz meg a konstruktív konfliktuskezelést és az aszertív kommunikációt.


Részletek Jelentkezés



Karriertervezés és Énmárka Építés

Ez a program segít a személyes és szakmai énmárka kialakításában, valamint karriered tudatos tervezésében. Tanuld meg, hogyan alakíthatasz ki egyedi és erős énmárkát, amely segíti a céljaid elérését.


Részletek Jelentkezés



Life Coaching Képzés

Fejleszd önismereted és találj rá a belső erőforrásaidra! Ebben a képzésben segítünk felismerni, hogyan érheted el a legjobbat önmagadból, és támogatunk abban, hogy magabiztosabban lépj az önmegvalósítás útjára.


Részletek Jelentkezés



Mindfulness és Produktivitás Program

Fedezd fel, hogyan növelheted a koncentrációd és produktívod a tudatos jelenlét gyakorlásával. A program ötvözi a mindfulness technikákat a modern időgazdálkodási módszerekkel, hogy segítsen egyensúlyt teremteni az élet különböző területei között.

Részletek Jelentkezés



Stresszkezelés és Reziliencia Workshop

Ismerd meg, hogyan kezelheted hatékonyabban a mindennapi stresszt, és hogyan építheted ki a belső erőforrásaidat, hogy könnyebben alkalmazkodj a kihívásokhoz. Ez a workshop gyakorlati tanácsokkal és technikákkal lát el.


Részletek Jelentkezés

Regisztráció/Bejelentkezés

Regisztráció

Regisztráció

Vagy regisztrálj



Üdvözljük!

Ha már van fiókja, jelentkezzen be és fedezze fel szolgáltatásainkat!

Bejelentkezés

Regisztráció:

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST['username'];
    $email = $_POST['email'];
    $password = $_POST['password'];
    if (empty($username) || empty($email) || empty($password)) {
        $message = "A mezők kitöltése kötelező.";
    } elseif (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $message = "Helytelen e-mail formátum.";
    } else {
        $stmt = $pdo->prepare("SELECT * FROM users WHERE username = ? OR email = ?");
        $stmt->execute([$username, $email]);

        if ($stmt->rowCount() > 0) {
            $message = "Ez a felhasználó név vagy e-mail már használatban van.";
        } else {
            $hashed_password = password_hash($password, PASSWORD_DEFAULT);

            $stmt = $pdo->prepare("INSERT INTO users (username, email, password) VALUES (?, ?, ?)");

            try {
                $stmt->execute([$username, $email, $hashed_password]);
                sendEmail($email, 'Regisztráció', 'Sikeres regisztráció!');
                $message = "Regisztráció sikeres!";
            } catch (PDOException $e) {
                $message = "Regisztráció sikertelen: " . $e->getMessage();
            }
        }
    }
}
```

Ez a rész:

- Ellenőrzi hogy a kérés POST methóddal történt
- Ha igen akkor validálja a 'username', 'email' és a 'password' változókat
- Ha minden oké akkor megnézi hogy létezik-e a táblába egy SELECT SQL paranccsal
- Ha nem létezik akkor 'kódolja' a jelszót és egy INSERT INTO SQL paranccsal elkészíti a fiókot

Bejelentkezés


Email cím

Jelszó

[Elfelejtett jelszó?](#)

Bejelentkezés

Vagy jelentkezz be



Még nincs fiókod? [Regisztrálj most!](#)

Bejelentkezés:

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $email = $_POST['email'];
    $password = $_POST['password'];

    if (empty($email) || empty($password)) {
        $message = "A mezők kitöltése kötelező.";
    } else {
        $stmt = $pdo->prepare("SELECT * FROM users WHERE email = ?");
        $stmt->execute([$email]);
        $user = $stmt->fetch();

        if ($user && password_verify($password, $user['password'])) {
            $_SESSION['user_id'] = $user['id'];
            $_SESSION['username'] = $user['username'];

            $token = bin2hex(random_bytes(32));
            $expires = time() + 86400;

            $stmt = $pdo->prepare("INSERT INTO auth_tokens (user_id, token, expires_at) VALUES (?, ?, ?)");
            $stmt->execute([$user['id'], $token, date('Y-m-d H:i:s', $expires)]);

            setcookie('auth_token', $token, [
                'expires' => $expires,
                'path' => '/',
                'httponly' => true,
                'secure' => true,
                'samesite' => 'Lax'
            ]);

            $_SESSION['auth_token'] = $token;

            header("Location: index.php");
            exit();
        } else {
            $message = "Helytelen e-mail vagy jelszó.";
        }
    }
}
```

Ez a rész:

- Ellenőrzi hogy a kérés POST módszerrel történt
- Ha igen akkor validálja az 'email' és a 'password' változókat
- Ha minden oké akkor megnézi hogy létezik-e a táblába egy SELECT SQL paranccsal
- A password_verify ellenőrzi hogy egyezik a jelszó
- Készít egy „token”-t amit INSERT INTO SQL paranccsal behelyez az adatbázisba és ezt a cookie-ba is behelyezi setcookie-val
- Ezután visszatér az index.php-hoz.

The image shows a dark-themed user interface for a password reset. A central light gray rounded rectangle contains the following elements: a title 'Elfelejtett jelszó' in bold black text, an instruction 'Add meg az email címed, és küldünk egy jelszót.' in smaller black text, a text input field with the placeholder 'Email cím', a prominent orange button labeled 'Jelszó visszaállítása', and a link 'Vissza a bejelentkezéshez' in orange text.

Elfelejtett Jelszó:

Ez a rész:

- Ellenőrzi hogy a kérés POST módszerrel történt
- Ha megtalálja az adatbázisban a megadott e-mailel regisztrált fiókot akkor csinál egy új véletlenszerű jelszót, berakja az adatbázisba és elküldi az e-mailbe.
- Ha nem létezik a megadott e-mailel regisztrált fiók akkor vissza küld egy hiba üzenetet.

Az oktatói kezelői felület

Szolgáltatások kezelése:

Szolgáltatások kezelése

Szolgáltatás neve

Szolgáltatás leírása

Ajánlott időtartam

Kiknek ajánlott

Időtartam (perc)

Ár

Fájl kiválasztása Nincs fájl kiválasztva

Hozzáadás

Név	Időtartam	Ár	Műveletek
Hatékony Kommunikáció és Konfliktuskezelés	60 perc	6 590 Ft	<div>Szerkesztés</div> <div>Törölés</div>
Life Coaching Képzés	60 perc	5 990 Ft	<div>Szerkesztés</div> <div>Törölés</div>

Ez a rész:

- Kezeli az új szolgáltatások hozzáadását, szerkesztését, törlését

```
if (!isset($_POST['name'], $_POST['duration'], $_POST['price'], $_POST['image'], $_POST['description'], $_POST['recommended_time'], $_POST['recommended_to'])) {  
    $image = $_FILES['image'];  
    $name = filter_input(INPUT_POST, 'name', FILTER_SANITIZE_STRING);  
    $duration = filter_input(INPUT_POST, 'duration', FILTER_SANITIZE_NUMBER_INT);  
    $price = filter_input(INPUT_POST, 'price', FILTER_SANITIZE_NUMBER_INT);  
    $description = filter_input(INPUT_POST, 'description', FILTER_SANITIZE_STRING);  
    $recommended_time = filter_input(INPUT_POST, 'recommended_time', FILTER_SANITIZE_STRING);  
    $recommended_to = filter_input(INPUT_POST, 'recommended_to', FILTER_SANITIZE_STRING);  
  
    if (addService($pdo, $user, $name, $duration, $price, $image, $description, $recommended_time, $recommended_to)) {  
        $_SESSION['success'] = "Szolgáltatás sikeresen hozzáadva!";  
    } else {  
        $_SESSION['error'] = "Hiba történt a szolgáltatás hozzáadásakor.";  
    }  
    header('Location: #alert');  
    exit();  
}
```

```
case 'delete':  
    if (!isset($_POST['id'])) {  
        throw new Exception('Hiányzó azonosító.');    }  
    $success = deleteService($pdo, $_POST['id']);  
    echo json_encode(['success' => $success, 'message' => 'Szolgáltatás törölve.']);  
    break;
```

Időpontok kezelése:

Időpontok kezelése

Hatékony Kommunikáció és Konfliktuskezelés

2025-04-03

--:--

--:--

Időpont hozzáadása

Ez a rész:

- Kezeli az új időpontok hozzáadását.

```
    } elseif (isset($_POST['service_id'], $_POST['date'], $_POST['start_time'], $_POST['end_time'])) {
        $service_id = filter_input(INPUT_POST, 'service_id', FILTER_SANITIZE_NUMBER_INT);
        $date = filter_input(INPUT_POST, 'date', FILTER_SANITIZE_STRING);
        $start_time = filter_input(INPUT_POST, 'start_time', FILTER_SANITIZE_STRING);
        $end_time = filter_input(INPUT_POST, 'end_time', FILTER_SANITIZE_STRING);

        $settings = getSettings($pdo);
        $work_start = strtotime("1970-01-01 " . $settings['work_day_start']);
        $work_end = strtotime("1970-01-01 " . $settings['work_day_end']);
        $slot_start = strtotime("1970-01-01 " . $start_time);
        $slot_end = strtotime("1970-01-01 " . $end_time);

        if ($slot_start < $work_start || $slot_end > $work_end) {
            $_SESSION['error'] = "Az időpont a munkaidőn kívül esik! (Munkaidő: {$settings['work_day_start']} - {$settings['work_day_end']});";
            header('Location: #alert');
            exit();
        }

        try {
            if (addTimeSlot($pdo, $service_id, $date, $start_time, $end_time)) {
                $_SESSION['success'] = "Időpont sikeresen hozzáadva!";
            }
        } catch (Exception $e) {
            $_SESSION['error'] = $e->getMessage();
        }
        header('Location: #alert');
        exit();
    }
}
```

Foglalások					
Dátum	Időpont	Kliens	Szolgáltatás	Státusz	Műveletek
2025-04-05	12:30 - 13:30	Teszt	Hatékony Kommunikáció és Konfliktuskezelés	Függőben	Jóváhagyás Lemondás

Ez a rész:

- Mutatja a lefoglalt időpontokat és a státuszt
- Kezeli a jóváhagyást és lemondást

Functions.php

Ebben a PHP-ban van mind az összes backend funkció.

Például:

- isLoggedIn -> Ellenőrzi hogy a felhasználó bevan jelentkezve
- getServices -> Az adatbázisban elmentett szolgáltatásokat jeleníti meg
- getUserBookings -> Az adatbázisban a felhasználónak a mentett foglalásának megjelenítésért felel
-
- get/edit/deleteService -> Megjelenít, módosít, törli a kiválasztott szolgáltatást
- getAvailableTimeSlots -> Megjeleníti a szabad időpontokat
- addTimeSlot -> Hozzáadja az okató által kiválasztott időpontot
- getBookings -> Megjeleníti a lefoglalt időpont részleteit
- createBooking -> Létrehozza a foglalást a felhasználó által megadott adatokkal
- updateBookingStatus -> Frissíti a foglalás státuszát attól függően hogy az okató elfogadta a foglalást
—A felhasználó is tudja módosítani a lemondás funkcióval—
- sendEmail -> Kezeli az email küldési folyamatokat

```
function getUserBookings($pdo, $user_id) {  
    $stmt = $pdo->prepare("  
        SELECT  
            b.*,  
            s.name as service_name,  
            t.start_time,  
            t.end_time,  
            t.date,  
            DATE_FORMAT(t.date, '%Y-%m-%d') as formatted_date,  
            DATE_FORMAT(t.start_time, '%H:%i') as formatted_start_time,  
            DATE_FORMAT(t.end_time, '%H:%i') as formatted_end_time  
        FROM bookings b  
        JOIN services s ON b.service_id = s.id  
        JOIN time_slots t ON b.time_slot_id = t.id  
        WHERE b.user_id = ?  
        ORDER BY t.date DESC, t.start_time ASC  
    ");  
    $stmt->execute([$user_id]);  
    return $stmt->fetchAll();  
}
```

Adminisztrátori Felület

Adminisztrátori Felület						
Keresés felhasználónév vagy email alapján...						
ID	Felhasználónév	Email	Admin	Oktató	Műveletek	
1	a	a@a.com	Igen	Igen	Törlés	Jelszó módosítás
2	a2	a2@gmail.com	Nem	Nem	Törlés	Jelszó módosítás
8	firestarter	firestarterakademia@gmail.com	Nem	Nem	Törlés	Jelszó módosítás
9	2025 Projekt	projekt2025teszt@gmail.com	Nem	Nem	Törlés	Jelszó módosítás
10	kiss.janos	kissjanos10@gmail.com	Nem	Igen	Törlés	Jelszó módosítás
11	nagy.anna	nagyanna11@gmail.com	Nem	Igen	Törlés	Jelszó módosítás
12	kovacs.peter	kovacspeter12@gmail.com	Nem	Igen	Törlés	Jelszó módosítás

Ez a rész:

- **Csak Admin jogosultsággal megtekinthető**
- Megjeleníti az összes regisztrált felhasználó adatait
- Megjeleníti/módosítja a felhasználók jogosultságait
- A törlés gomb segítségével törölhető a regisztrált fiók
- Folyamatosan frissülő keresővel lehet keresni a fiókok között

Személyes profil

Profil Adatok

Kurzusaim

Profil Adatok

Felhasználói Adatok

Felhasználónév: a

Email: a@a.com

Email Módosítása

Új email cím

Email frissítése

Jelszó Módosítása

Jelenlegi jelszó

Új jelszó

Új jelszó megerősítése

Jelszó módosítása

Ezen az oldalon:

- Megtekinthető a felhasználó adatai (username-email)
- Módosítható az email cím és jelszó
- Megtekinthetőek a lefoglalt kurzusok

Kurzusaim

Foglalásaim

Még nincs foglalásod.

Összefoglalás

A projekt célja egy modern, felhasználóbarát online platform létrehozása volt, amely összeköti a Firestarter Akadémia oktatóit a fejlődni vágyó egyénekkal, különféle önismereti, üzleti és kommunikációs képzéseken keresztül.

A megvalósított rendszer lehetővé teszi a felhasználók számára a zökkenőmentes regisztrációt (hagyományos úton vagy Google fiókkal), a bejelentkezést, a kínált szolgáltatások (pl. coachingok, workshopok, tréningek) böngészését, valamint az időpontfoglalást egy interaktív naptár segítségével. A felhasználói élményt reszponzív kialakítás, választható sötét/világos téma és finom animációk gazdagítják. A rendszer elkülönített, funkciókban gazdag felületeket biztosít az általános felhasználók, az oktatók (szolgáltatások, időpontok, foglalások kezelése) és az adminisztrátorok (teljeskörű rendszerfelügyelet) számára.

A fejlesztés során PHP alapú backendet, MySQL adatbázist, valamint HTML, CSS és JavaScript technológiákat alkalmaztunk a frontendhez. A rendszer működését Cypress segítségével automatizált tesztekkel ellenőriztük.

Ez a dokumentáció részletes betekintést nyújt a projekt teljes életciklusába, a tervezéstől a megvalósításon át a tesztelésig. Különálló fejezetek foglalkoznak a fejlesztői szempontokkal (használt technológiák, rendszerarchitektúra, adatbázis séma, kulcsfontosságú kódrészletek magyarázata) és a felhasználói kézikönyvvel (amely lépésről-lépésre útmutatókkal segíti a platform használatát minden felhasználói szerepkör számára). Ezenkívül a dokumentum javaslatokat tartalmaz a lehetséges jövőbeli fejlesztésekre és listázza a felhasznált külső forrásokat is.

Összességében a Firestarter Akadémia weboldal egy sikeresen megvalósított vizsgaremek projekt, amely egy működőképes, modern és könnyen használható platformot eredményezett, kielégítve a kezdeti célkitűzéseket és biztosítva egy stabil alapot a további bővítésekhez.

Források

...

AOS Library:

<https://cdnjs.com/libraries/aos>

<https://michalsnik.github.io/aos/>

Óktatók képei:

<https://thispersondoesnotexist.com/>

Inter font:

<https://rsms.me/inter/>

PHPMailer:

<https://github.com/PHPMailer/PHPMailer>

Google API PHP

<https://github.com/googleapis/google-api-php-client>

Készítők

Balogh Richárd, Takács Botond