

# Fejlesztői és Felhasználói Dokumentáció



Készítette: [Katona Valentin](#)

local php version: 8.2.4

online php version: 7.4.33

# Tartalomjegyzék

## Tartalom:

Általános leírás .....	3
Fájlok és funkciók.....	4
Adatbázis .....	5
Biztonság .....	6
Frontend .....	7
Használati útmutató .....	8

# Általános leírás

Ez az alkalmazás egy egyszerű autentikációs middleware CRUD (Create, Read, Update, Delete) rendszer, amely lehetővé teszi a felhasználók számára a:

- 1.bejelentkezést
- 2.regisztrációt
- 3.jelszó és felhasználónév módosítását
- 4.valamint fiók törlését.

## A feladat:

Készíteni egy autentikációs middlewaret, valamint bejelentkező oldalt, minimális front-endel. A php fájlokat levédeni a middlewarrel, aminek hatására csak akkor fut le a PHP további része ha a látogató bejelentkezett.

Követelmények:

- ☒ Nem használni előre megírt frameworkot (Laravel, CakePHP, Symfony, stb.)
- ☒ A feladatot vanilla PHP 7.4+ elvégezni.
- ☒ A felhasználókat MySQL adatbázisban tárolni.
- ☒ public.php – olyan fájl amit bejelentkezés nélkül is megtekinthető.
- ☒ protected.php – olyan fájl aminek a tartalmát csak bejelentkezés után látjuk.
- ☒ login.php fájl – amin be tudunk jelentkezni + regisztrálni.
- + user.php – ahol szerkeszthetjük az adatokat bejelentkezés után.
- + index.php – főoldal/leírás.

# Fájlok és funkciók

## Mappa -és fájl struktúra:

 Feladat


|

|  Assets

| |  favicon.ico

| |  logo\_black.png

| |  logo\_white.png

| |  functions.php ————— |

| |  index.php |

| |  login.php |

| |  protected.php |

| |  public.php |

| |  styles.css |

| |  user.php |

| |  documentation.pdf |

| |  database\_dump.sql |

V

- **checkAccess()**: Ellenőrzi, hogy a felhasználó be van-e jelentkezve.
- **connectDB()**: Kapcsolódik az adatbázishoz.
- **login()**: Bejelentkezteti a felhasználót.
- **logout()**: Kijelentkezteti a felhasználót.
- **register()**: Regisztrál egy új felhasználót.
- **changeUsername()**: Megváltoztatja a felhasználónevet.
- **changePassword()**: Megváltoztatja a jelszót.
- **deleteAccount()**: Törli a felhasználói fiókot.
- **validateChangeUsername()**: Validálja az új felhasználónevet.
- **validateChangePassword()**: Validálja az új jelszót.

**index.php**: A főoldalt tartalmazza.

**login.php**: A bejelentkezési és regisztrációs oldalt tartalmazza.

**protected.php**: Egy privát oldal, amely csak bejelentkezett felhasználók számára elérhető.

**user.php**: Lehetővé teszi a felhasználó számára a fiókadatainak módosítását bejelentkezés után.

**public.php**: Egy nyilvános oldal.

# Adatbázis

## MYSQL Táblázat struktúra:

Column	Type	Constraints
id	INT(11) AUTO_INCREMENT	PRIMARY KEY
username	VARCHAR(255)	
password	VARCHAR(255)	

Adatbázis neve: **feladat\_db**

Tábla: **users**

Oszlopok: **id, username, password**

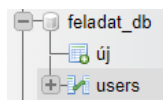
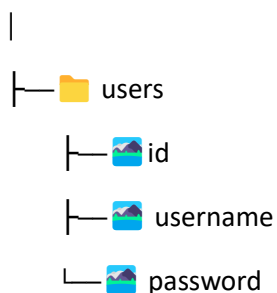
	V	

Itt találhatóak a beregisztrált felhasználók, csak ők láthaják a protected/user oldalakat.

id jelöli a felhasználó számát, username a felhasználó nevét, password a felhasználó titkosított jelszavát.

## MYSQL vizuális struktúra:

feladat\_db



Fontos funkció:

id	username	password
1	user1	password1
2	user2	password2
7		\$hashed_password-al:
9	János Koppány	\$2y\$10\$ihRMt9XKNDc9HEiGwrtUBejjTnTHwRmizlqgLfX9WV...

Létrehozás:

```
CREATE DATABASE feladat_db;
USE feladat_db;
CREATE TABLE users (
  id INT(11) NOT NULL AUTO_INCREMENT,
  username VARCHAR(255) NOT NULL,
  password VARCHAR(255) NOT NULL,
  PRIMARY KEY (id)
);
```

# Biztonság

- A jelszavakat hash-elve tároljuk az adatbázisban.
- A kód preventív módszereket használ az SQL injekció ellen (pl. prepared statements).
- A CSRF támadások elleni védelem érdekében CSRF token-t használunk.
- XSS cross-site-scripting elleni védelemért htmlspecialchars() használata.
- Minden egyéb hibaüzenetet kilogolunk.

# Frontend

Az alkalmazás egyszerű frontendet használ.

A CSS és a JavaScript kódok a megfelelő fájlokban találhatóak.

A stílusokat a styles.css fájlban találhatjuk az alábbi témákra bontva:

1. BODY
2. SHAPES
3. CARD
4. BUTTON
5. NAVBAR
6. MOBILE NAV

# Használati útmutató

1. Fájlok letöltése és kicsomagolása
2. Adatbázis csatlakoztatása
3. Php futtatása böngészőben
4. Belépés

## 1.

Töltsük le a fájlokat a github repositoryból [ITT](#).

Majd csomagoljuk ki őket egy mappába.

## 2.

XAMPP, Laragon, Mac esetén MAMP vagy egyéb localhoston futtatható phpmyadmin indítása.

Mysql adatbázis belépést követően létrehozuk az adatbázist **feladat\_db** néven.

Az adatbázis létrehozását követően beimportáljuk a **users** táblát a kicsomagolt fájlok közül.

## 3.

A kicsomagolt fájlknál elindítjuk a CMD-t, majd **php -S localhost:8000** paranccsal futtatjuk a php-t.

A böngészőbe beírva a következő címen elérjük az oldalt: **localhost:8000** .

## 4.

Beléphetünk az alábbi felhasználónévvel és jelszóval:

Username: **Admin**

Password: **Admin1+**

vagy regisztrálhatunk saját felhasználót.



