

Budapesti Komplex Szakképzési Centrum

Weiss Manfréd Technikum, Szakképző Iskola  
és Kollégium

Összefoglaló dokumentáció:

- Weissesvagyok keretrendszer,
- Pszichológus időpontfoglaló

Témavezető: Kovács László Bálint

Készítette:

- Szász Melinda
- Scháb Tibor
- Zölderdő Zétény

<b>Budapesti Komplex Szakképzési Centrum.....</b>	<b>1</b>
<b>Weiss Manfréd Technikum, Szakképző Iskola.....</b>	<b>1</b>
<b>és Kollégium.....</b>	<b>1</b>
<b>1. Bevezetés.....</b>	<b>4</b>
1.1. Projekt bemutatása.....	4
1.2. Témaválasztás indoklása.....	4
1.3. Célkitűzés.....	5
1.4. Célcsoport.....	5
1.5. Fejlesztői környezet.....	6
1.6. Fejlesztői dokumentáció.....	7
<b>2. Adatszerkezet.....</b>	<b>8</b>
2.1. Adatbázis táblák.....	8
2.1.1. A “login” tábla.....	9
2.1.2. A “naplo” tábla.....	9
2.1.3. A “korabbinev” tábla.....	9
2.1.4. A “user” tábla.....	10
<b>3. Keretrendszer felhasználói felület.....</b>	<b>12</b>
3.1. Frontend Bejelentkezés.....	12
3.1.1. HTML dokumentum deklaráció és fejléc.....	12
3.1.2. Bejelentkezés űrlap.....	13
3.1.3. Űrlap mezők.....	13
3.1.4. Gombok és linkek.....	14
3.2. Frontend Regisztráció.....	15
3.2.1. HTML dokumentum deklaráció és fejléc.....	15
3.2.2. Regisztrációs űrlap.....	15
3.2.3. Űrlap mezők.....	16
3.2.4. Gombok és linkek.....	16
<b>4. Keretrendszer fejlesztői felület.....</b>	<b>17</b>
4.1. Backend bejelentkezés.....	17
4.1.1. Felhasználó lekérdezése.....	17
4.1.2. Hibás adatok kezelése.....	17
4.1.3. Session változók beállítása.....	18
4.1.4. Sikeres bejelentkezés visszajelzése.....	18
4.1.5. Kapcsolat lezárása.....	18
4.2. Backend Regisztráció.....	19
4.2.1. Beérkező adatok ellenőrzése.....	19
4.2.2. Adatbázis kapcsolódás.....	20
4.2.3. Inaktív felhasználók státuszának frissítése.....	20
4.2.4. E-mail egyediségének ellenőrzése.....	20
4.2.5. Felhasználónév ellenőrzése.....	21
4.2.6. Felhasználó adatainak mentése.....	21
4.2.7. Felhasználónév naplózása.....	22

4.2.8. E-mail megerősítő üzenet.....	22
4.2.9. Sikeres regisztráció visszajelzése.....	22
4.2.10. Adatbázis kapcsolat lezárása.....	22
<b>5. Profil oldal.....</b>	<b>23</b>
5.1. Áttekintés.....	23
5.2. SESSION és jogosultságkezelés.....	23
5.3. Form struktúra.....	24
5.3.1. Főbb elemek:.....	24
5.4. Javascript és PHP funkcionalitás.....	26
5.4.1. Mezők változása.....	26
5.4.2. Jelszóváltoztatás.....	27
5.4.3. Weisses email cím.....	28
5.4.4. Felhasználónevek naplózása.....	28
5.5. Sikeres módosítás.....	29
<b>6. Pszichológus időpontfoglaló.....</b>	<b>30</b>
6.1. Felhasználói felület.....	30
6.1.1. Időpont választó modul.....	30
6.1.2. Űrlap Modul.....	30
6.2. Adminisztrációs felület.....	30
6.2.1. Foglalkozáskezelő Dashboard.....	30
6.2.2. Foglalkozáskezelési Funkciók.....	31
6.3. Adatbázis rendszer.....	31
6.3.1. Fő Táblák és Kapcsolataik.....	31
6.4. Funkciók leírása.....	32
6.4.1. Időpont foglalási folyamat.....	32
6.4.2. Adminisztrációs folyamat.....	32
6.5. Speciális funkciók és Testreszabási lehetőségek.....	33
6.5.1. Ügyfél Kommunikációs eszközök.....	33
6.5.2. Speciális beállítások.....	33
6.5.3. Biztonsági rendszerek.....	33
6.6. Technikai specifikációk.....	34
6.6.1. Szerver követelmények.....	34
6.6.2. Telepítési folyamat.....	34
6.7. Karbantartási útmutató.....	35
6.7.1. Rendszeres feladatok.....	35
6.7.2. Hibaelhárítási útmutató.....	35
6.8. Jövőbeli fejlesztések.....	35
6.8.1. Középjövőben tervezett funkciók.....	35
6.8.2. Hosszú távú tervek.....	35

# 1. Bevezetés

## 1.1. Projekt bemutatása

Ez a dokumentáció a pszichológus időpontfoglaló responzív weboldal tervezéséről és fejlesztéséről szól.

- A weboldal célja, hogy megkönnyítse a kliensek számára az időpontfoglalást.
- Tájékoztatást nyújt a pszichológusról, az elérhető szolgáltatásokról és a jövőbeli időpontokról.
- A tervezési és fejlesztési folyamat során kiemelt figyelmet kapott a felhasználói élmény és a biztonság.
- A rendszer célja a pszichológus munkájának támogatása, valamint kényelmes és hatékony megoldások nyújtása a kliensek számára.

## 1.2. Témaválasztás indoklása

- A mentális egészség növekvő fontossága indokolja a weboldal témáját.
- Célunk, hogy egyszerűbbé tegyük a pszichológushoz való időpontfoglalást.
- A digitális technológiák fejlődése miatt a felhasználók elvárják az online

elérhetőséget.

- A weboldal közvetlen kapcsolattartást biztosít a pszichológussal, növelve a bizalmat.
- A cél a mentális egészségügyi ellátás modernizálása és az elérhetőség javítása.

### 1.3. Célkitűzés

- Egy könnyen kezelhető, felhasználóbarát időpontfoglaló rendszer kialakítása.
- Segítségnyújtás a pszichológusi támogatást kereső diákok számára.
- Lehetőség profil létrehozására, amely lehetővé teszi a belépést saját fiókba.
- A felhasználók könnyen hozzáférhetnek az időpontfoglaláshoz és más szolgáltatásokhoz.

### 1.4. Célcsoport

- A weboldalt széleskörű közönség számára terveztük.
- Elsősorban azoknak, akik stressz, szorongás, depresszió vagy egyéb problémák miatt pszichológiai támogatást keresnek.
- Fontos célcsoport a fiatal felnőttek, akik digitális megoldásokat részesítenek előnyben.

- A szülők is hasznos információkhoz juthatnak, hogy gyermeküknek segítséget kérhessenek.
- A weboldal támogatja azokat is, akik már jártak terápiára és folytatni szeretnék személyes fejlődésüket.
- Összességében a cél, hogy mindenki könnyebben és gyorsabban elérhesse a mentális egészségügyi támogatást.

## 1.5. Fejlesztői környezet

- Visual Studio Code: könnyen használható kódszerkesztő a weboldal kialakításához.
- XAMPP: tartalmazza az Apache szerveret, a MySQL-t és a PHP-t, így lehetővé teszi a lokális fejlesztést.
- phpMyAdmin: egyszerűen kezelhető eszköz az adatbázis karbantartására.
- MySQL: hatékonyan kezeli a felhasználói és egyéb adatokat.
- A fejlesztőeszközök ingyenesek és ismertek számunkra, így ideálisak a projekt kivitelezéséhez.

## 1.6. Fejlesztői dokumentáció

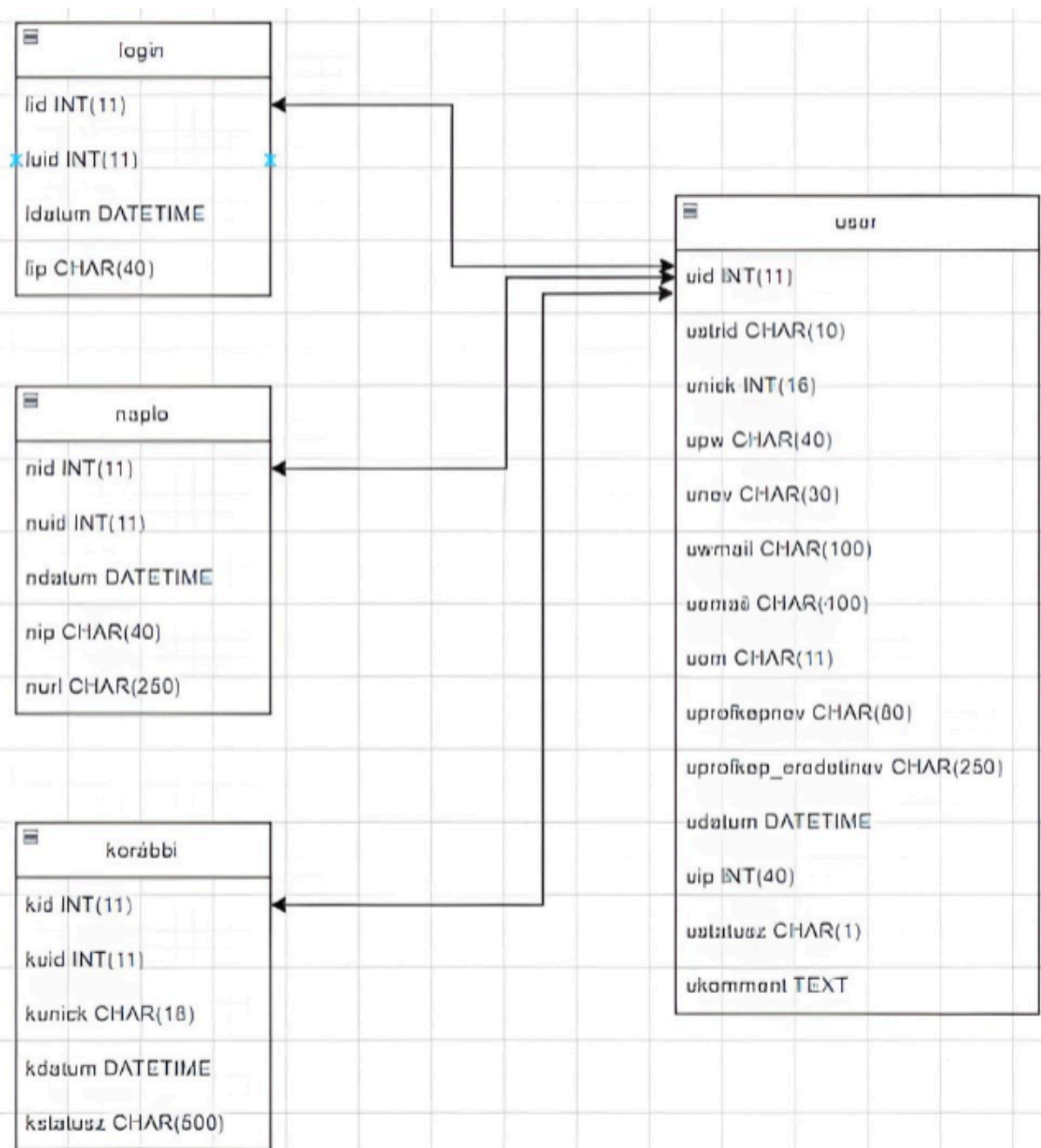
- Domain: A weboldal online eléréséhez domainre volt szükség.
- Az iskola által működtetett weboldalon fut az időpontfoglaló rendszer, Kovács László tanárúr segítségével.
- GitHub: A fejlesztéshez közös GitHub környezetet alakítottunk ki.
- Egy közösen létrehozott email cím segítségével regisztráltunk.
- Létrejött egy pszichologus-projekt nevű repository, amelyben mindannyian résztvevőként szerepelünk.
- A változtatások nyomon követhetők, a fájlok közösen elérhetők.
- Git telepítése a <https://git-scm.com/> oldalról történt.
- Az elkészült projekt – beleértve az adatbázist, kódokat és dokumentációt – megtalálható a következő GitHub linken:

<https://github.com/Z3t3ny/pszichologus-projekt>

## 2. Adatszerkezet

A weboldalunk adatbázisa több táblát tartalmaz. Ezek a táblák strukturáltak és szervezettek, hogy hatékonyan kezeljék és tárolják az adatokat.

### 2.1. Adatbázis táblák





### 2.1.1. A "login" tábla

- lid (login azonosító): Ez a mező tárolja a bejelentkezés azonosítót
- luid (login user azonosító): Ez a mező tárolja a felhasználói bejelentkezés azonosítót
- ldatum (login dátum): Ez a mező tárolja a bejelentkezés dátumát
- lip (login ip): Ez a mező tárolja a bejelentkezés ip címét

login	
lid	INT(11)
luid	INT(11)
ldatum	DATETIME
lip	CHAR(40)

### 2.1.2. A "naplo" tábla

- nid (napló azonosító): Ez a mező tárolja a naplózott azonosítóját
- nuid (napló user azonosító): Ez a mező tárolja a naplózott felhasználó azonosítóját
- ndatum (napló dátum): Ez a mező tárolja a naplózás dátumát
- nurl (napló url): Ez a mező tárolja a naplózott urlt

naplo	
nid	INT(11)
nuid	INT(11)
ndatum	DATETIME
nip	CHAR(40)
nurl	CHAR(250)

### 2.1.3. A "korabbinev" tábla

- kid (korábbi azonosító): Ez a mező tárolja a korábbi azonosítót
- kuid (korábbi user azonosító): Ez a mező tárolja a korábbi felhasználó azonosítót
- kunick (korábbi nickname): Ez a mező tárolja a korábbi becenevet
- kdatum (korábbi dátum): Ez a mező tárolja a korábbi dátumot

kid	INT(11)
kuid	INT(11)
kunick	CHAR(18)
kdatum	DATETIME
kstatusz	CHAR(500)

- kstatusz (korábbi státusz): Ez a mező tárolja a korábbi státuszt

#### 2.1.4. A “user” tábla

- uid (felhasználó azonosító): Ez a mező tárolja a felhasználó
- azonosítót ustrid (): Ez a mező tárolja a
- unick (felhasználó nickname): Ez a mező tárolja a
- felhasználó becenevét amit a regisztrációnál megadott
- upw (felhasználó jelszó) : Ez a mező tárolja a felhasználó jelszavát.
- unev (felhasználó név) : Ez a mező tárolja a felhasználónevét.
- uwmail (felhasználó weiss email) : Ez a mező tárolja a felhasználó weiss-es email címét.
- uemail (felhasználó email) : Ez a mező tárolja a felhasználó egyéb email címét.
- uom (felhasználó OM azonosító) : Ez a mező tárolja a felhasználó OM azonosítóját
- .
- uprofkepnev (felhasználó profilkép név) : Ez a mező tárolja a felhasználó profilképének a nevét.
- uprofkep\_eredetinev (felhasználó profilkép eredetinev) : Ez a mező tárolja a profilkép azon nevét, ahogy azt a felhasználó feltöltötte.
- udatum (felhasználó dátum) : Ez a mező tárolja a felhasználó regisztrálásának időpontját
- ucomment(felhasználó komment): Eme mezőbe az oldal moderátorai tehetnek megjegyzést bizonyos felhasználókra, figyelmeztetés vagy tájékoztatásképpen.
- uip (felhasználó ip) : Ez a mező tárolja a felhasználó ip címét
- ustatusz (felhasználó státusz) : Ez a mező tárolja a felhasználó státuszát.

user			
uid	INT(11)		
ustrid	CHAR(10)		
unick	INT(16)		
upw	CHAR(40)		
unev	CHAR(30)		
uwmail	CHAR(100)		
uemail	CHAR(100)		
uom	CHAR(11)		
uprofkepnev	CHAR(80)		
uprofkep_eredetinev	CHAR(250)		
udatum	DATETIME		
uip	INT(40)		
ustatusz	CHAR(1)		
ukomment	TEXT		

## 3. Keretrendszer felhasználói felület

### 3.1. Frontend Bejelentkezés

#### 3.1.1. HTML dokumentum deklaráció és fejléc

A dokumentum HTML5-ös szabványt használ.

```
<!DOCTYPE html>
```

Meta tagek:

Beállítják a karakterkódolást (UTF-8), a kompatibilitást az Internet Explorer-rel, a mobilnézet támogatását és az oldal leírását.

Cím:

„Bejelentkezés” – ez jelenik meg a böngésző fülén.

CSS:

`pszihologus.css` fájl van csatolva stíluslapként.

```
<head>
  <meta charset='utf-8'>
  <meta http-equiv='X-UA-Compatible' content='IE=edge'>
  <title>Bejelentkezés</title>
  <meta name='description' content='Weiss pszihologus'>
  <meta name='viewport' content='width=device-width, initial-scale=1'>
  <link rel='stylesheet' href='pszihologus.css'>
</head>
```

### 3.1.2. Bejelentkezés űrlap

- A `form` elem POST metódussal küldi az adatokat a `reg_ir.php` fájlba, és az eredményt egy `kisablak` nevű célkeretbe (target frame) tölti be – ez valószínűleg egy `iframe` lehet, amely a HTML más részében van (bár itt nem látható).
- Az űrlap CSS osztálya: `zformok`.

```
<form action='login_ir.php' method=post target='kisablak' class="zformok">
```

### 3.1.3. Űrlap mezők

- Felhasználónév/Email: `name='email'`, kötelező mező.
- Jelszó: `type='password'`, szintén kötelező.
- Az azonosító (`id`) azonos minden mezőnél, ami nem ideális, de jelen esetben a stílus miatt van így (`forminputok`).

```
<div class='loginContainer'>
  <!-- Username -->
  <label for='username'>Felhasználónév/Email</label>
  <input id="forminputok" placeholder='' name='email' required>

  <br><br>

  <!-- Password -->
  <label for='pswrđ'>Jelszó</label>
  <input id="forminputok" type='password' placeholder='' name='pw' required>
```

### 3.1.4. Gombok és linkek

- Belépés gomb: Beküldi az űrlapot.
- Regisztráció link: Másik oldalra visz új felhasználók számára.

```
<!-- Submit button -->
<button type='submit' id='loginformgomb'>Belépés</button>

<!-- Sign up link -->
<p class='register'>Még nincs regisztrálva? <a href='regisztracio'>Regisztráljon itt!</a></p>
<!--<p class='forgotpsd'> <a href='#'>Elfelejtettem a jelszavam!</a></p> -->
```

## 3.2. Frontend Regisztráció

### 3.2.1. HTML dokumentum deklaráció és fejléc

A dokumentum HTML5-ös szabványt használ.

```
<!DOCTYPE html>
```

Meta tagek:

Beállítják a karakterkódolást (UTF-8), a kompatibilitást az Internet Explorer-rel, a mobilnézet támogatását és az oldal leírását.

Cím:

A böngésző fülén megjelenő szöveg: *Bejelentkezés*.

CSS:

A `pszihologus.css` fájl stíluslapként van csatolva az oldalhoz.

```
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Bejelentkezés</title>
  <meta name="description" content="Weiss pszihologus">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="pszihologus.css">
</head>
```

### 3.2.2. Regisztrációs űrlap

- A `form` elem POST metódussal küldi az adatokat a `reg_ir.php` fájlba, és az eredményt egy `kisablak` nevű célkeretbe (target frame) tölti be – ez valószínűleg egy iframe lehet, amely a HTML más részében van (bár itt nem látható).
- Az űrlap CSS osztálya: `zformok`.

```
<div class="regFormContainer">
  <form action='reg_ir.php' method=post target='kisablak' class="zformok">
```

### 3.2.3. Űrlap mezők

- A felhasználónév, email és jelszó megadása kötelező (**required**).

További mezők:

- Email (**type="email"**): Érvényes email formátumot vár.
- Jelszó kétszer (jelszó ellenőrzés miatt).

```
<div class="regContainer">
  <!-- Username -->
  <label for="unick">Felhasználónév</label>
  <input id="forminputok" type="text" placeholder="" name="unick" required>

  <!-- Username -->
  <label for="uemail">Email</label>
  <input id="forminputok" placeholder="" name="uemail" required type='email'>
  <br>

  <br>

  <!-- Password -->
  <label for="upw1">Jelszava</label>
  <input id="forminputok" type="password" placeholder="Legalább 8 karakter" name="upw1" required>

  <!-- Password -->
  <label for="upw2">Jelszava még egyszer</label>
  <input id="forminputok" type="password" placeholder="" name="upw2" required>
```

### 3.2.4. Gombok és linkek

- A regisztráció gomb beküldi az űrlapot.

```
<!-- Submit button -->
<button type="submit" id='regformgomb'>Regisztráció</button>
```

- Egy link azoknak, akik már rendelkeznek fiókkal – átnavigál a **belepes** oldalra.

```
<!-- Login link -->
<p class="register">Már van fiókja? <a href="belepes">Jelentkezzen be itt!</a></p>
```



## 4. Keretrendszer fejlesztői felület

### 4.1. Backend bejelentkezés

#### 4.1.1. Felhasználó lekérdezése

Az SQL lekérdezés több mezőben is keresi a megadott azonosítót (email, nicknév, weblap email).

Csak az aktív státuszú (A), felfüggesztett (F) vagy felsőbbrendű (D) fiókokra korlátozódik.

```
$user = mysqli_fetch_array( mysqli_query( $adb , "  
    SELECT * FROM user  
    WHERE ( uemail='$_POST[email]' OR unick='$_POST[email]' OR uwmail='$_POST[email]')  
    AND    upw   ='$md5pw'  
    AND    ( ustatus='A' OR ustatus='F' OR ustatus='D')  
    " ) ) ;
```

#### 4.1.2. Hibás adatok kezelése

Ha nincs megfelelő találat, JavaScript alert üzenetet küld vissza a felhasználónak:

```
if( $user['uid']=="" )  
    die( "<script> alert('Hibás belépési adatok!') </script>" ) ;
```

### 4.1.3. Session változók beállítása

A bejelentkezés után a legfontosabb felhasználói adatok bekerülnek a munkamenetbe:

```
$_SESSION['uid']    = $user['uid']    ;  
$_SESSION['unick']  = $user['unick']  ;  
$_SESSION['umail']  = $user['umail']  ;  
$_SESSION['unev']   = $user['unev']   ;
```

### 4.1.4. Sikeres bejelentkezés visszajelzése

A sikeres bejelentkezést alerttel jelzi, majd a [kezdolap](#) oldalra navigál:

```
print "<script>  
    alert('Sikeresen bejelentkeztél!')  
    parent.location.href='kezdolap'  
</script>" ;
```

### 4.1.5. Kapcsolat lezárása

Az adatbázis kapcsolat lezárásra kerül:

```
mysqli_close( $adb ) ;
```

## 4.2. Backend Regisztráció

Ez a PHP szkript dolgozza fel a felhasználó regisztrációs adatait (név, e-mail, jelszó stb.), ellenőrzi azok helyességét, majd elmenti őket az adatbázisba. Emellett biztosítja, hogy:

- az e-mail egyedi legyen,
- a felhasználónév ne legyen foglalt,
- a jelszó megfelelő hosszúságú legyen,
- és végül visszajelzést ad a felhasználónak JavaScript alert formájában.

### 4.2.1. Beérkező adatok ellenőrzése

- Ellenőrzi, hogy van-e e-mail megadva. Ha nincs, alert üzenet és a script leáll:

```
if( $_POST['uemail']=="" )  
    die( "<script> alert('Nem adtad meg az e-mail címed!') </script>" ) ;
```

- Jelszó hosszának ellenőrzése:

```
if( strlen($_POST['upw1']) < 8 )  
    die( "<script> alert('A jelszónak min. 8 karakter hosszúságúnak kell lennie!') </script>" ) ;
```

- A két jelszó mező összehasonlítása:

```
if( $_POST['upw1']!= $_POST['upw2'] )  
    die( "<script> alert('Nem egyeznek a jelszavaid!') </script>" ) ;
```

#### 4.2.2. Adatbázis kapcsolódás

Betölti az adatbázis kapcsolatot létrehozó fájlt.

```
include("kapcsolat.php") ;
```

#### 4.2.3. Inaktív felhasználók státuszának frissítése

Azokat a regisztrációkat, amelyek több mint 18 percig E (előregisztrált) státuszban vannak, inaktívvá (L) állítja.

```
mysqli_query($adb, "
    UPDATE user
    SET ustatusz = 'L'
    WHERE ustatusz = 'E'
    AND TIME_TO_SEC(TIMEDIFF(NOW(), udatum)) > 18*60
");
```

#### 4.2.4. E-mail egyediségének ellenőrzése

Megnézi, hogy az adott e-mail cím már regisztrálva van-e aktív, félkész vagy előregisztrált formában:

```
$sorokszama = mysqli_num_rows( mysqli_query( $adb , "
    SELECT * FROM user
    WHERE uemail='$_POST[uemail]'
    AND (ustatusz='A' OR ustatusz='F' OR ustatusz='E')
" ) ) ;
```

#### 4.2.5. Felhasználónév ellenőrzése

- Eltávolítja a szóközöket a felhasználónévből:

```
$_POST["unick"] = str_replace(" ", "", $_POST["unick"]);
```

- Ellenőrzi, hogy a választott név szerepel-e a korábbi nevek táblában – ha igen, akkor nem engedi a regisztrációt:

```
if( mysqli_num_rows( mysqli_query( $adb , "
| | | | | SELECT * FROM korabbinév
| | | | | WHERE kunick='$_POST[unick]'
" ) ) )
die( "<script> alert('Ez a felhasználónév már foglalt!') </script>" ) ;
```

#### 4.2.6. Felhasználó adatainak mentése

- A jelszót titkosítja.
- Létrehoz egy véletlenszerű azonosítót a megerősítéshez.

```
$strid = randomstr(10) ;
```

- Az új felhasználó adatait menti az **user** táblába:

```
mysqli_query( $adb , "
    INSERT INTO user (...)
    VALUES (...)
" ) ;
```

#### 4.2.7. Felhasználónév naplózása

Az új felhasználó választott nevét rögzíti a `korabbinev` táblában.

```
mysqli_query( $adb , "
    INSERT INTO korabbinev (kid , kuid , kunick , kdatum , kstatusz)
    VALUES (NULL , '$userid[0]' , '$_POST[unick]', NOW() , '' )
");
```

#### 4.2.8. E-mail megerősítő üzenet

Megerősítő linket tartalmazó e-mail szöveget:

```
$uzenet = "Kedves szöveg... https://weissesvagyok.hu/registracio-megerosites/" . $strid . "
" ;
```

#### 4.2.9. Sikeres regisztráció visszajelzése

Alerten keresztül visszajelzést ad, majd átirányítja a felhasználót a bejelentkezési oldalra.

```
print "<script>
    alert('Regisztrációd sikeresen megtörtént.')
    parent.location.href='belepes'
</script>" ;
```

#### 4.2.10. Adatbázis kapcsolat lezárása

Lezárja az adatbázis kapcsolatot.

```
mysqli_close( $adb ) ;
```

## 5. Profil oldal

### 5.1. Áttekintés

Ez az oldal egy adminisztrációs felület, ahol a bejelentkezett felhasználók szerkeszthetik profiljukhoz tartozó adataikat, például:

- nevüket,
- email-címüket,
- jelszavukat,
- és profilképüket.

A rendszer PHP-ben készült, MySQL adatbázist használ, és JavaScript segíti az interaktív felhasználói élményt.

### 5.2. SESSION és jogosultságkezelés

Azon felhasználók számára nem elérhető a profil funkció, akik nincsenek bejelentkezve.

Az alábbi kódrészlet biztosítja, hogy a felhasználó csak bejelentkezés után férjen hozzá a profilmódosító űrlaphoz:

```
if(!isset($_SESSION['uid']))  
    die( "<script>  
        alert('Lejárt a munkamanet! Jelentkezz be ismét!')  
        parent.location.href='belepes'  
    </script>" );
```

## 5.3. Form struktúra

A HTML-űrlap `profil_ir.php` fájlba küldi az adatokat, fájlfeltöltéssel együtt. A mezők tartalma az oldal betöltésekor automatikusan rendelkezik a bejelentkezett felhasználó adataival.

### 5.3.1. Főbb elemek:

- HTML form  
Akción: `profil_ir.php`, metódus: `POST`, enctype: `multipart/form-data`  
Célja: felhasználói adatok és fájl feltöltése.
- Mezők:
  - `unick` – Felhasználónév
  - `unev` – Teljes név
  - `uemail` – Email
  - `uwmail` – Iskolai email tag (pl. vezeteknev.keresztnev)

```
<div id="nevekcim"><h2>NEVEK</h2></div>
<div id="nevekinput">
  <input class="szinezendok" id="forminputok" type="text" placeholder="Felhasználónév" name="unick" required value="<?=$_SESSION['unick'];?>" />
  <input class="szinezendok" id="forminputok" type="text" placeholder="Teljes név" name="unev" value="<?=$user['unev'];?>" />
</div>
```

- `uom` – OM azonosító
- `upw` – Jelszó ellenőrzés
- `upw2`, `upw3` – Új jelszó + megerősítés

```
<div id="azonositokcim"><h2>AZONOSÍTÓK</h2></div>
<div id="azonositokinput">
  <input class="szinezendok" id="forminputok" type="text" placeholder="OM azonosító" name="uom" value="<?=$user['uom'];?>" />
  <p>
    <input class="neszinezd" id="forminputok" type="password" name="upw" required placeholder="Jelszava" />
    <button type="button" id="modositogomb">Módosítás</button>
  </p>
  <input id="forminputok" class="ujjelszoinput" type="password" name="upw2" placeholder="Új jelszava" />
  <input id="forminputok" class="ujjelszoinput" type="password" name="upw3" placeholder="Új jelszava még egyszer" />
</div>
</div>
```



- **uprofkep** – Képfeltöltés (JPG, PNG)

```
<input class="file-uploader" type="file" onchange="upload()" accept="image/*" name="uprofkep" />
</div>
```

- **Profilkép megjelenítés**

A felhasználó új képet tölthet fel. Ha már van profilképe, azt megjelenítjük, különben egy alapértelmezett kép látszik.

```
<?php
if($user['uprofkepnev'] != "")
{
    print("<script>
        document.getElementById('a').style.backgroundImage = 'url(/profilkepek/$user[uprofkepnev])'
    </script>
    ");
}
else
{
    print("<script>
        document.getElementById('a').style.backgroundImage = 'url(https://qph.cf2.quoracdn.net/main-qimg-f32f85d21d59a5540948c3bfbce52e68)'
    </script>
    ");
};
?>
```


## 5.4. Javascript és PHP funkcionalitás

### 5.4.1. Mezők változása

- Az oldal csak akkor engedi menteni az adatokat, ha tényleges változás történt – ezt vizuálisan is jelzi, ha változás történik, a mentés gomb (profilformgomb) láthatóvá válik:

```
function ellenorizValtozasokat() {  
  let vanValtozas = false;  
  
  inputok.forEach(input => {  
    const current = input.value;  
    const kezdeti = kezdetiErtekek[input.name];  
  
    const nemErdekes = input.classList.contains('neszinezd');  
  
    if (current !== kezdeti) {  
      if (!nemErdekes) {  
        vanValtozas = true;  
      }  
  
      if (input.classList.contains('szinezendok')) {  
        input.classList.add('valtozott');  
        input.placeholder = "Mentetlen változtatások";  
      }  
    } else {  
      if (input.classList.contains('szinezendok')) {  
        input.classList.remove('valtozott');  
      }  
    }  
  });  
  
  if (vanValtozas) {  
    gomb.classList.add('lathato');  
  } else {  
    gomb.classList.remove('lathato');  
  }  
}  
  
// Eseményfigyelő minden inputra  
form.addEventListener('input', (event) => {  
  if (event.target.tagName === 'INPUT') {  
    ellenorizValtozasokat();  
  }  
});
```

Profil adatlap



NEVEK

KONTAKTOK

@wm-iskola.hu

AZONOSÍTÓK

### 5.4.2. Jelszóváltoztatás

A jelszómezők alapból rejtettek, csak gombnyomásra jelennek meg. A régi jelszó szükséges az új elfogadásához:

```
document.getElementById("modositogomb").addEventListener("click", function () {
    const inputok = document.querySelectorAll(".ujjelszoinput");
    inputok.forEach(input => {
        // Ürítjük, mivel ez az alap érték
        input.value = "";
        input.dispatchEvent(new Event("input", { bubbles: true })); // szükséges!
    });

    const inputok2 = document.querySelectorAll(".neszinezd");
    inputok2.forEach(input => {
        input.value = input.defaultValue; // vagy kezdetiErtekek[input.name], ha globálisan elérhető
        input.dispatchEvent(new Event("input", { bubbles: true })); // szükséges!
    });

    const gomb = document.getElementById("modositogomb");
    let lathato = inputok[0].style.display === "block";

    inputok.forEach(input => {
        input.style.display = lathato ? "none" : "block";
    });

    gomb.textContent = lathato ? "Módosítás" : "X";

});
```

### 5.4.3. Weisses email cím

Iskolánk diákjai az iskolába való beiratkozást követően személyre szabott, "saját" email címet kapnak, Ezek az email címek nem feltétlen szükségesek a regisztrációhoz, avagy az oldalakon aktívnak maradáshoz, mivel:

- a frissen felvett diákok, az email cím kiosztását megelőzően, akár már a gólyatábor folyamán regisztrálhassanak
- a már nem weisses diákok is lehetőséget kapnak a közösségben maradáshoz

```
$weissemail = mysqli_num_rows( mysqli_query( $adb , "  
SELECT * FROM user  
WHERE uwmailtag = '$_POST[uwmail]'  
AND (ustatusz='A' OR ustatusz='F' OR ustatusz='D')  
AND uid != '$_SESSION[uid]'  
" ) ) ;
```

```
if( $weissemail > 0 ) die("<script>alert('Ezzel a weisses email címmel már van megerősített fiók!')</script>");
```

### 5.4.4. Felhasználónevek naplózása

Ha megváltozik a felhasználónév, elmentjük az előzőt a **korabbinev** táblába, így egy adott felhasználóhoz, a már általa eddig összes felhasználónév fog tartozni. Ez nyomon követi a névváltoztatásokat – hasznos visszakereséshez vagy moderációhoz:

```
if($_SESSION['unick'] != $_POST['unick']){  
    mysqli_query( $adb , "  
INSERT INTO korabbinev ( kid , kuid , kunick , kdatum , kstatusz )  
VALUES ( NULL , '$_SESSION[uid]', '$_POST[unick]', NOW() , '' )  
" );  
}  
  
$_SESSION['unick'] = $_POST['unick'];
```

## 5.5. Sikeres módosítás

Ha minden mentés sikeres, a felhasználót visszairányítjuk a profil oldalára, és tájékoztatjuk:

```
print(  
    "<script>  
  
    alert('Az adataidat sikeresen módosítottuk!')  
    parent.location.href='profil'  
  
    </script>");
```

**A(z) weissessvagyok.infy.uk üzenete**

Az adataidat sikeresen módosítottuk!

OK

## 6. Pszichológus időpontfoglaló

### 6.1. Felhasználói felület

#### 6.1.1. Időpont választó modul

- Dátumválasztó:
  - Automatikusan betölti a mai dátumot
  - Letiltja a múltbeli dátumokat (nem választhatóak)
  - Heti nézetben jeleníti meg a napokat
  - Speciális jelölés ünnepnapokra és szabadnapokra
- Időpontválasztó:
  - Dinamikus időblokkok (általában 45-50 perces konzultációs idők)
  - Reális időintervallumok (pl. 8:00-16:00 között)
  - Automatikus frissítés dátumváltáskor
  - Foglalt időpontok vizuális kiemelése (piros szín, "Foglalt" felirat)

#### 6.1.2. Űrlap Modul

- Személyes adatok:
  - Teljes név (kötelező)
  - Email cím (kötelező, formátum ellenőrzéssel)
  - Telefonszám (opcionális, de ajánlott)
  - Megjegyzés mező (speciális kérések, problémaleírás)
- Validációs rendszer:
  - Valós idejű hibajelzés
  - Kötelező mezők jelölése
  - Email formátum ellenőrzés
  - Telefonszám formátum ellenőrzés (nem kötelező)

## 6.2. Adminisztrációs felület

#### 6.2.1. Foglaltságkezelő Dashboard

- Áttekintő táblázat:
  - Összes foglalás listája időrendi sorrendben
  - Színkódolt státuszjelzés (zöld=jóváhagyott, sárga=függőben, piros=elutasított)
  - Részletes oszlopok: név, email, dátum, időpont, státusz, beküldés ideje

- Speciális szűrők:
  - - Dátumtartomány szerinti szűrés
  - - Státusz szerinti szűrés (összes/függőben/elfogadva/elutasítva)
  - - Név vagy email cím alapján keresés
  - - Oldaltörés és lapozás funkció

## 6.2.2. Foglalkoztatási Funkciók

- Jóváhagyás:
  - Azonnali státuszváltoztatás
  - - Opcionális email értesítés küldése
  - - Naptárba automatikus felvétel (ha integrálva van)
- Elutasítás:
  - Indoklás megadása (opcionális)
  - Email értesítés a felhasználónak
  - Archivumba helyezés
- Szerkesztés:
  - Dátum/időpont módosítása
  - Kapcsolattartási adatok frissítése
  - Megjegyzések hozzáadása
- Törlés:
  - Kétlépcsős megerősítés
  - Végleges eltávolítás az adatbázisból
  - Naplózás a műveletről

## 6.3. Adatbázis rendszer

### 6.3.1. Fő Táblák és Kapcsolataik

- Foglalkoztatási tábla (foglalkozások):
  - 15+ mező a részletes adattároláshoz
  - Indexek a gyors kereséshez
  - Kapcsolódás a felhasználói táblához (ha van regisztrációs rendszer)
- Felhasználók tábla (user):
  - Regisztrált ügyfelek adatai
  - Bejelentkezési előzmények
  - Kedvenc időpont preferenciák
- Naplózási tábla (naplo):
  - Admin műveletek nyomon követése
  - Rendszeresemények rögzítése

## 6.4. Funkciók leírása

### 6.4.1. Időpont foglalási folyamat

#### 1. Első lépések:

- Felhasználó kiválasztja a kívánt dátumot
- Rendszer betölti az elérhető időpontokat
- Foglalt időpontok automatikus kiszűrése

#### 2. Adatmegadás:

- Személyes adatok kitöltése
- Validáció valós időben
- Opcionális megjegyzés mező

#### 3. Foglалás elküldése:

- Adatok ellenőrzése
- Időpont foglaltságának utolsó ellenőrzése
- Adatbázisba írás
- Visszajelzés a felhasználónak

#### 4. Feldolgozás:

- Admin értesítés új foglalásról
- Automatikus visszaigazoló email (ha be van állítva)
- Függőben státuszba helyezés

### 6.4.2. Adminisztrációs folyamat

#### 1. Bejelentkezés:

- Többszörös hitelesítés lehetősége
- CAPTCHA védelem
- Bejelentkezési kísérletek naplózása

#### 2. Foglалások kezelése:

- Áttekintő oldal testreszabása
- Gyorskeresés funkciók
- Tömeges műveletek (pl. több foglalás státuszának egyszerre módosítása)

#### 3. Naptárintegráció:

- Google Calendar/Outlook szinkronizálás
- Automatikus esemény létrehozás
- Konfliktusok felismerése

#### 4. Jelentéskészítés:

- Heti/havi statisztikák
- Legnépszerűbb időpontok elemzése
- Ügyfélforgalom vizualizációja



## 6.5. Speciális funkciók és Testreszabási lehetőségek

### 6.5.1. Ügyfél Kommunikációs eszközök

- Automatizált email rendszer:
  - Foglalás visszaigazolás
  - Emlékeztető 24 órával előtte
  - Státuszváltozás értesítés
  - Követő email kampányok
- SMS értesítések:
  - Azonnali visszajelzés
  - Rövid link a foglalás részleteihez
  - Választható SMS nyelv

### 6.5.2. Speciális beállítások

- Munkarend konfiguráció:
  - Munkanapok beállítása
  - Ebédszünet időintervalluma
  - Ünnepnapok kezelése
  - Nyitvatartási idő szezonális változásai
- Időpont típusok:
  - Különböző konzultációs típusok
  - Változó időtartamok
  - Árkülönbségek kezelése

### 6.5.3. Biztonsági rendszerek

- Adatvédelem:
  - GDPR-kompatibilis adatkezelés
  - Automatikus adattörlési szabályzat
  - Titkosított kommunikáció
- Hozzáférés-vezérlés:
  - Többszintű jogosultsági rendszer
  - P-alapú korlátozások
  - Kétfaktoros hitelesítés

## 6.6. Technikai specifikációk

### 6.6.1. Szerver követelmények

- - Minimális konfiguráció:
  - - PHP 7.4+
  - - MySQL 5.7+ vagy MariaDB 10.2+
  - - 1GB RAM
  - - 10GB tárhely
- - Ajánlott konfiguráció:
  - - PHP 8.0+
  - - MySQL 8.0+
  - - 2GB RAM
  - - SSD tárhely

### 6.6.2. Telepítési folyamat

#### 1. Előkészületek:

- Domain és hosting beállítása
- SSL tanúsítvány telepítése
- Adatbázis létrehozása

#### 2. Fájlok telepítése:

- Forrásfájlok feltöltése
- Jogosultságok beállítása
- Konfigurációs fájlok módosítása

#### 3. Adatbázis inicializálás:

- Séma importálása
- Alapértelmezett beállítások
- Tesztadatok létrehozása (opcionális)

#### 4. Véglegesítés:

- Tesztfuttatás
- Biztonsági mentés beállítása
- Monitorozási rendszer aktiválása

## 6.7. Karbantartási útmutató

### 6.7.1. Rendszeres feladatok

- Napi:

- Biztonsági mentések ellenőrzése
- Új foglalások monitorozása
- Email küldések státuszának ellenőrzése

- Heti:

- Adatbázis optimalizálás
- Log fájlok tisztítása
- Rendszerfrissítések alkalmazása

- Havi:

- Teljes rendszermentés
- Biztonsági audit
- Teljesítményelemzés

### 6.7.2. Hibaelhárítási útmutató

- Gyakori problémák:
  - Foglалás nem jelenik meg: Adatbázis kapcsolat ellenőrzése
  - Email nem érkezik meg: SMTP beállítások ellenőrzése
  - Lassú működés: Adatbázis indexek ellenőrzése
- Speciális esetek:
  - Duplikált foglалások
  - Időzóna problémák
  - Karakterkódolási hibák

## 6.8. Jövőbeli fejlesztések

### 6.8.1. Közeljövőben tervezett funkciók

- Online fizetési integráció
- Mobilalkalmazás
- Ügyfélportál

### 6.8.2. Hosszú távú tervek

- Mesterséges intelligencia alapú időpontajánló
- Integráció EÜ rendszerekkel
- Többnyelvű támogatás

