**SZABADKAI MŰSZAKI SZAKFŐISKOLA**

**SUBOTICA TECH**



**SZEMINÁRIUMI MUNKA -**

**Web programozás**

**Cím:**

**Webes rendszer rendezvényszervezéshez**

**Hallgatók: Horváth Kávai Márk és Vörös Viktor**

**Indexek: 26221058 és 26221100**

**Irányzat: Informatika**

**Tanár: Dr. Zlatko Čović**

Tartalom

[Követelmények 2](#_Toc138816826)

[Adatbázis diagram 2](#_Toc138816827)

[Adatbázis kódja 3](#_Toc138816828)

[SQL magyarázat: 9](#_Toc138816829)

[Technikai részletek: 11](#_Toc138816830)

[Használt technológiák és keretrendszerek: 11](#_Toc138816831)

[Technológiai követelmények: 11](#_Toc138816832)

[Telepítés és konfiguráció: 11](#_Toc138816833)

[Kódstruktúra (összesítés a kódokból): 12](#_Toc138816834)

[Hibaüzenetek és Üzenetek: 12](#_Toc138816835)

[Biztonság: 12](#_Toc138816836)

[Használati útmutató: 12](#_Toc138816837)

[Fő funkciók és műveletek bemutatása: 12](#_Toc138816838)

[Lépésről lépésre útmutató: 12](#_Toc138816839)

[Tesztelés: 13](#_Toc138816840)

[A Log In PHP teljes elemzése: 13](#_Toc138816841)

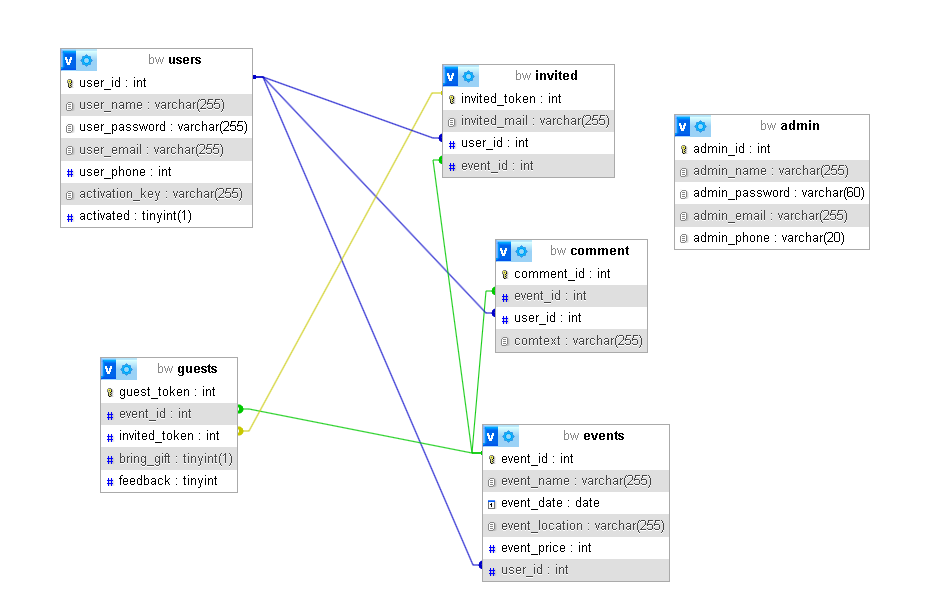
# Követelmények

A projekt célja egy webes rendszer létrehozása rendezvényszervezéshez. Az alkalmazás a

hozzáférés négy szintjét különbözteti meg: vendég, regisztrált / bejelentkezett felhasználó,

meghívott és rendszergazda.

# Adatbázis diagram



## Adatbázis kódja

-- phpMyAdmin SQL Dump

-- version 5.2.1

-- https://www.phpmyadmin.net/

--

-- Host: 127.0.0.1

-- Generation Time: Jun 28, 2023 at 02:37 AM

-- Server version: 10.4.28-MariaDB

-- PHP Version: 8.2.4

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

START TRANSACTION;

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- Database: `events`

--

-- --------------------------------------------------------

--

-- Table structure for table `admin`

--

CREATE TABLE `admin` (

  `admin\_id` int(11) NOT NULL,

  `admin\_name` varchar(255) NOT NULL,

  `admin\_password` varchar(60) NOT NULL,

  `admin\_email` varchar(255) NOT NULL,

  `admin\_phone` varchar(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

--

-- Dumping data for table `admin`

--

INSERT INTO `admin` (`admin\_id`, `admin\_name`, `admin\_password`, `admin\_email`, `admin\_phone`) VALUES

(0, 'vikica', '12345678', 'viktor2xx1@gmail.com', '0635678342'),

(1, 'hkmark', '12345678', 'hkmark2002@gmail.com', '0637596344'),

(6, 'chole', '$2y$10$oMZntwOHDjAWpO5okrtsAOLnqCRgYQbzXBFHwssXytXa7lKwUorpy', 'chole@vts.su.ac.rs', '12313123');

-- --------------------------------------------------------

--

-- Table structure for table `comment`

--

CREATE TABLE `comment` (

  `comment\_id` int(11) NOT NULL,

  `event\_id` int(11) NOT NULL,

  `user\_id` int(11) NOT NULL,

  `comtext` varchar(255) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

-- --------------------------------------------------------

--

-- Table structure for table `events`

--

CREATE TABLE `events` (

  `event\_id` int(11) NOT NULL,

  `event\_name` varchar(255) NOT NULL,

  `event\_date` date NOT NULL,

  `event\_location` varchar(255) NOT NULL,

  `event\_price` int(11) NOT NULL,

  `user\_id` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

--

-- Dumping data for table `events`

--

INSERT INTO `events` (`event\_id`, `event\_name`, `event\_date`, `event\_location`, `event\_price`, `user\_id`) VALUES

(7, 'SzakDolgozat', '2024-06-15', 'Szabadka', 3000, 35),

(8, 'Diplomalas', '2023-07-01', 'Cantavir', 123, 35),

(9, 'Stat vizsga', '2023-06-30', 'Szabadkan a VTS-n', 480, 37);

-- --------------------------------------------------------

--

-- Table structure for table `guests`

--

CREATE TABLE `guests` (

  `guest\_token` int(11) NOT NULL,

  `event\_id` int(11) DEFAULT NULL,

  `invited\_token` int(11) DEFAULT NULL,

  `bring\_gift` tinyint(1) NOT NULL DEFAULT 0,

  `feedback` tinyint(3) NOT NULL DEFAULT 0

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

-- --------------------------------------------------------

--

-- Table structure for table `invited`

--

CREATE TABLE `invited` (

  `invited\_token` int(11) NOT NULL,

  `invited\_mail` varchar(255) NOT NULL,

  `user\_id` int(11) NOT NULL,

  `event\_id` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

--

-- Dumping data for table `invited`

--

INSERT INTO `invited` (`invited\_token`, `invited\_mail`, `user\_id`, `event\_id`) VALUES

(1, 'viktor20010105@gmail.com', 37, 9),

(11, 'hkmark2002@gmail.com', 35, 7),

(5591, 'hkmark2002@gmail.com', 35, 8);

-- --------------------------------------------------------

--

-- Table structure for table `users`

--

CREATE TABLE `users` (

  `user\_id` int(11) NOT NULL,

  `user\_name` varchar(255) NOT NULL,

  `user\_password` varchar(255) NOT NULL,

  `user\_email` varchar(255) NOT NULL,

  `user\_phone` int(15) NOT NULL,

  `activation\_key` varchar(255) NOT NULL,

  `activated` tinyint(1) DEFAULT 0

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

--

-- Dumping data for table `users`

--

INSERT INTO `users` (`user\_id`, `user\_name`, `user\_password`, `user\_email`, `user\_phone`, `activation\_key`, `activated`) VALUES

(35, 'mark', '$2y$10$o5QqTFm5x5JQV.q3Ion/7eUzNHMtrdVyyJmA9zy1IPeSeOUI5gKp6', 'markhorvathkavai@gmail.com', 637596344, '0lzK2xV3cC8DwqzhDLaQcCKJPFewhu0C', 1),

(37, 'Vikica', '$2y$10$xP7Z0eGECs93XxEagVCIPeWsgoBYME1e/ysfuHyTrZxKEbfeZxhxC', 'viktor2xx1@gmail.com', 638821856, 'ivL6F7fUL6VOH5zloAyViuSHRVLqYf8f', 1);

--

-- Indexes for dumped tables

--

--

-- Indexes for table `admin`

--

ALTER TABLE `admin`

  ADD PRIMARY KEY (`admin\_id`);

--

-- Indexes for table `comment`

--

ALTER TABLE `comment`

  ADD PRIMARY KEY (`comment\_id`),

  ADD KEY `event\_id` (`event\_id`),

  ADD KEY `user\_id` (`user\_id`);

--

-- Indexes for table `events`

--

ALTER TABLE `events`

  ADD PRIMARY KEY (`event\_id`),

  ADD KEY `user\_id` (`user\_id`);

--

-- Indexes for table `guests`

--

ALTER TABLE `guests`

  ADD PRIMARY KEY (`guest\_token`),

  ADD KEY `event\_id` (`event\_id`),

  ADD KEY `invited\_token` (`invited\_token`);

--

-- Indexes for table `invited`

--

ALTER TABLE `invited`

  ADD PRIMARY KEY (`invited\_token`),

  ADD KEY `event\_id` (`event\_id`),

  ADD KEY `user\_id` (`user\_id`);

--

-- Indexes for table `users`

--

ALTER TABLE `users`

  ADD PRIMARY KEY (`user\_id`);

--

-- AUTO\_INCREMENT for dumped tables

--

--

-- AUTO\_INCREMENT for table `admin`

--

ALTER TABLE `admin`

  MODIFY `admin\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=8;

--

-- AUTO\_INCREMENT for table `comment`

--

ALTER TABLE `comment`

  MODIFY `comment\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=29;

--

-- AUTO\_INCREMENT for table `events`

--

ALTER TABLE `events`

  MODIFY `event\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=13;

--

-- AUTO\_INCREMENT for table `guests`

--

ALTER TABLE `guests`

  MODIFY `guest\_token` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=7;

--

-- AUTO\_INCREMENT for table `invited`

--

ALTER TABLE `invited`

  MODIFY `invited\_token` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=5592;

--

-- AUTO\_INCREMENT for table `users`

--

ALTER TABLE `users`

  MODIFY `user\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=38;

--

-- Constraints for dumped tables

--

--

-- Constraints for table `comment`

--

ALTER TABLE `comment`

  ADD CONSTRAINT `comment\_ibfk\_1` FOREIGN KEY (`event\_id`) REFERENCES `events` (`event\_id`) ON DELETE CASCADE,

  ADD CONSTRAINT `comment\_ibfk\_2` FOREIGN KEY (`user\_id`) REFERENCES `users` (`user\_id`) ON DELETE CASCADE;

--

-- Constraints for table `events`

--

ALTER TABLE `events`

  ADD CONSTRAINT `events\_ibfk\_1` FOREIGN KEY (`user\_id`) REFERENCES `users` (`user\_id`) ON DELETE CASCADE;

--

-- Constraints for table `guests`

--

ALTER TABLE `guests`

  ADD CONSTRAINT `guests\_ibfk\_1` FOREIGN KEY (`event\_id`) REFERENCES `events` (`event\_id`) ON DELETE CASCADE,

  ADD CONSTRAINT `guests\_ibfk\_3` FOREIGN KEY (`invited\_token`) REFERENCES `invited` (`invited\_token`) ON DELETE CASCADE;

--

-- Constraints for table `invited`

--

ALTER TABLE `invited`

  ADD CONSTRAINT `invited\_ibfk\_1` FOREIGN KEY (`event\_id`) REFERENCES `events` (`event\_id`) ON DELETE CASCADE,

  ADD CONSTRAINT `invited\_ibfk\_2` FOREIGN KEY (`user\_id`) REFERENCES `users` (`user\_id`) ON DELETE CASCADE;

COMMIT;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

# SQL magyarázat:

A fenti SQL egy adatbázis sémát tartalmaz, amely néhány táblát definiál a events nevű adatbázisban. Az adatbázis sémában szereplő táblák a következők:

admin tábla: Ez a tábla az adminisztrátorok adatait tárolja, akik a rendszerhez kapcsolódnak. Az alábbi oszlopokat tartalmazza:

admin\_id: Adminisztrátor azonosítója (egész szám)

admin\_name: Adminisztrátor neve (karakterlánc)

admin\_password: Adminisztrátor jelszava (karakterlánc)

admin\_email: Adminisztrátor e-mail címe (karakterlánc)

admin\_phone: Adminisztrátor telefonszáma (karakterlánc)

comment tábla: Ez a tábla az eseményekhez tartozó kommenteket tárolja. Az alábbi oszlopokat tartalmazza:

comment\_id: Komment azonosítója (egész szám)

event\_id: Az esemény azonosítója (egész szám)

user\_id: Felhasználó azonosítója (egész szám)

comtext: Komment szövege (karakterlánc)

events tábla: Ez a tábla az események adatait tárolja. Az alábbi oszlopokat tartalmazza:

event\_id: Az esemény azonosítója (egész szám)

event\_name: Az esemény neve (karakterlánc)

event\_date: Az esemény dátuma (dátum)

event\_location: Az esemény helyszíne (karakterlánc)

event\_price: Az esemény ára (egész szám)

user\_id: Felhasználó azonosítója (egész szám)

guests tábla: Ez a tábla az eseményekhez tartozó vendégeket tárolja. Az alábbi oszlopokat tartalmazza:

guest\_token: Vendég azonosítója (egész szám)

event\_id: Az esemény azonosítója (egész szám)

invited\_token: Meghívott azonosítója (egész szám)

bring\_gift: Ajándékot hoz-e a vendég (boole érték, 0 vagy 1)

feedback: Visszajelzés a vendégről (egész szám)

invited tábla: Ez a tábla az eseményekhez meghívottakat tárolja. Az alábbi oszlopokat tartalmazza:

invited\_token: Meghívott azonosítója (egész szám)

invited\_mail: Meghívott e-mail címe (karakterlánc)

user\_id: Felhasználó azonosítója (egész szám)

event\_id: Az esemény azonosítója (egész szám)

users tábla: Ez a tábla a felhasználók adatait tárolja. Az alábbi oszlopokat tartalmazza:

user\_id: Felhasználó azonosítója (egész szám)

user\_name: Felhasználó neve (karakterlánc)

user\_password: Felhasználó jelszava (karakterlánc)

user\_email: Felhasználó e-mail címe (karakterlánc)

user\_phone: Felhasználó telefonszáma (egész szám)

activation\_key: Aktiválási kulcs (karakterlánc)

activated: Felhasználó aktiválási állapota (boole érték, 0 vagy 1)

A táblák közötti kapcsolatok az alábbiak szerint vannak definiálva:

A comment tábla az events és a users táblákkal kapcsolatban áll külső kulcsok segítségével.

Az events tábla a users táblával kapcsolatban áll külső kulcs segítségével.

A guests tábla az events és az invited táblákkal kapcsolatban áll külső kulcsok segítségével.

Az invited tábla az events és a users táblákkal kapcsolatban áll külső kulcsok segítségével.

# Technikai részletek:

## Használt technológiák és keretrendszerek:

A projekt keretein belül használtunk: HTML, CSS, JavaScript, AJAX , JSON, BootStrap, PHP és MySQL.

# Technológiai követelmények:

PHP 5.6 vagy újabb

MySQL adatbázis

PHPMailer könyvtár (verzió 6.0 vagy újabb)

# Telepítés és konfiguráció:

Telepítsd és konfiguráld a szükséges környezetet a fenti technológiai követelmények alapján.

Hozz létre egy MySQL adatbázist és frissítsd a db\_config.php fájlt a megfelelő adatbázis kapcsolati információkkal.

Másold be a PHPMailer könyvtárat a projekt könyvtárába, és frissítsd a szükséges fájlneveket a kódban.

# Kódstruktúra (összesítés a kódokból):

A kódok három fő részre oszthatók: inicializáció és konfiguráció, adatfeldolgozás és adatbázis műveletek, e-mail küldése.

Inicializáció és konfiguráció: A szükséges fájlok importálása és az adatbázis kapcsolat beállítása történik.

Adatfeldolgozás és adatbázis műveletek: A regisztrációs adatok feldolgozása, ellenőrzése és az adatok adatbázisba mentése történik itt.

E-mail küldése: Az e-mail küldése a regisztrált felhasználónak vagy a meghivott felhasznalónak.

# Hibaüzenetek és Üzenetek:

Az oldalakon felmerülő hibákról és egyéb üzenetekről a következőképpen tájékoztatja a felhasználót:

A hibák tömbje (errors) tartalmazza a különböző hibaüzeneteket, amelyek megjelennek az egyes oldalakon , űrlapokon.

Az aktiválási link sikeres elküldése vagy az e-mail küldési hiba üzenete jelenik meg a felhasználónak.

# Biztonság:

A jelszavak biztonságos tárolása érdekében a kódokban a password\_hash függvényt használjuk a jelszavak hash-elve történő tárolásához. Ez növeli a jelszavak biztonságát, mivel a felhasználó jelszavát nem tárolják tisztán a rendszerben.

# Használati útmutató:

## Fő funkciók és műveletek bemutatása:

Regisztráció, Bejelentkezés, Események létrehozása, módosítása, törlése. Üzenetek írása az egyes eseményekhez. Meghívók készítése az eseményekre és ezen meghívók e-mailben való elküldése a meghívott személy számára. Bejelentkezett személy adatait módosíthatja.

## Lépésről lépésre útmutató:

Az oldalt megtekinthetjük úgy is ha nem vagyunk regisztrálva viszont így nem hozhatunk létre eseményt. Tehát az oldal teljeskörű használatához szükségünk van be jelentkezni az oldalra vagy regisztrálni amennyiben nincs még profilunk. Ha regisztrálni szeretnék az oldalon szükségünk van egy valódi e-mail címre amire megkapjuk az aktiváló kódunkat amivel majd be tudjuk aktiválni a profilunkat és így már be is tudunk jelentkezni az oldalra.

A Home Page-en láthassuk a felhasználók által létrehozott eseményeket.

Az Events fül alatt hozhatunk létre saját eseményt, továbbá itt is van lehetőségünk törölni és módosítani is.

Profil fül alatt megtekinthetjük saját adatainkat. Itt is törölhetjük a profilunk amennyiben már nem akarunk regisztráltak lenni az oldalon.

About Us fülnél az oldal készítőiről leírás.

# Tesztelés:

Tesztelés során sok problémába ütköztünk most ezek leírása következik.

A kódokban helyes hibakezelés és kivételek használata látható. Például, a try-catch blokkban a kód ellenőrzi a PDO hibákat és az Exception objektumokat, és megjeleníti a megfelelő hibaüzenetet a felhasználónak.

# A Log In PHP teljes elemzése:

**session\_start()**

Ez a függvény elindítja a munkamenetet a session változók használatához.

**ini\_set('display\_errors', 'On')**

Ezzel a beállítással engedélyezed a hibaüzenetek megjelenítését.

**error\_reporting(E\_ALL)**

Ez a beállítás arra utasítja a PHP-t, hogy minden hibát és figyelmeztetést jelenítsen meg.

**$errors = array()**

Egy üres tömb, amelyet a későbbiekben hibák tárolására használnak.

**if (isset($\_SESSION['loggedin']) && $\_SESSION['loggedin'] === true)**

Ellenőrzi, hogy a felhasználó már be van jelentkezve vagy sem. Ha igen, átirányítja a felhasználót a user\_home.php oldalra.

**if ($\_SERVER['REQUEST\_METHOD'] === 'POST')**

Ellenőrzi, hogy a kérés metódusa POST-e vagy sem. Ezt azért teszi, hogy csak akkor folytassa a bejelentkezési folyamatot, ha a felhasználó elküldte az űrlapot.

**include("db\_config1.php")**

Az adatbázis konfigurációs fájl beillesztése, amely tartalmazza az adatbázis kapcsolati információkat.

**$user\_email = $\_POST['user\_email']**

A felhasználó által megadott e-mail cím lekérdezése a POST kérésből.

**$user\_password = $\_POST['user\_password']**

A felhasználó által megadott jelszó lekérdezése a POST kérésből.

Létrehoz egy PDO objektumot az adatbáziskapcsolat létrehozásához. Az adatbázis kapcsolati információk a db\_config1.php fájlból származnak.

**$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION)**

Beállítja a PDO hibamódját kivételek generálására, amikor hiba történik.

**$stmt = $pdo->prepare("SELECT \* FROM users WHERE user\_email = :user\_email")**

Előkészíti a lekérdezést, amely ellenőrzi, hogy van-e olyan felhasználó az adatbázisban, akinek az e-mail címe megegyezik a megadott e-mail címmel.

**$stmt->bindParam(':user\_email', $user\_email)**

A $user\_email változóhoz köti a lekérdezés paraméterét.

**$stmt->execute()**

Végrehajtja a lekérdezést.

**$row = $stmt->fetch(PDO::FETCH\_ASSOC)**

Lekérdezi az eredményhalmaz első sorát és elmenti a $row változóban.

**if ($row)**

Ellenőrzi, hogy van-e találat az adatbázisban a megadott e-mail címre.

**$user\_id = $row['user\_id']**

Elmenti a felhasználó azonosítóját a $user\_id változóban.

**$user\_name = $row['user\_name']**

Elmenti a felhasználó nevét a $user\_name változóban.

**$storedPassword = $row['user\_password']**

Elmenti a felhasználó jelszavát a $storedPassword változóban. Ez a jelszó hashelve van az adatbázisban.

**$activated = $row['activated']**

Elmenti a felhasználó aktivációs állapotát a $activated változóban.

**if ($activated == 1)**

Ellenőrzi, hogy a felhasználó fiókja aktiválva van-e.

**if (password\_verify($user\_password, $storedPassword))**

Ellenőrzi, hogy a megadott jelszó helyes-e a tárolt jelszóval összehasonlítva.

**$\_SESSION['loggedin'] = true**

Beállítja a loggedin session változót true-ra, jelezve, hogy a felhasználó bejelentkezett.

**$\_SESSION['user\_id'] = $user\_id**

Elmenti a felhasználó azonosítóját a user\_id session változóba.

**$\_SESSION['user\_name'] = $user\_name**

Elmenti a felhasználó nevét a user\_name session változóba.

**$\_SESSION['user\_email'] = $user\_email**

Elmenti a felhasználó e-mail címét a user\_email session változóba.

**$\_SESSION['user\_password'] = $user\_password**

Elmenti a felhasználó jelszavát a user\_password session változóba. Fontos megjegyezni, hogy a jelszavakat általában nem tárolják a session változókban biztonsági okokból. Ez csak egy példa a kódhoz.

**header('Location: user\_home.php')**

Átirányítja a felhasználót a user\_home.php oldalra.

**$errors['user\_password'] = "Incorrect email or password!"**

Hibaüzenetet állít be, ha az e-mail cím vagy jelszó nem megfelelő.

**$errors['general'] = "Your account is not activated yet. Please check your email for the activation link."**

Hibaüzenetet állít be, ha a felhasználó fiókja még nincs aktiválva.

**$errors['user\_email'] = "Incorrect email or password!"**

Hibaüzenetet állít be, ha az e-mail cím vagy jelszó nem megfelelő.

**$\_SESSION['errors'] = $errors**

Elmenti a hibákat a errors session változóba, hogy azokat megjeleníthesse az űrlapon.

Ezután következik az HTML kód, amely a bejelentkező űrlapot tartalmazza.