Bóka Bendegúz

gp6khz

Adatbázis rendszerek programozása

6. TV Guide

http://157.181.167.180/~gp6khz/index.php

ADATBÁZIS FELÉPÍTÉSE

Account tábla

user_id: SERIAL PRIMARY KEY NOT NULL,
username VARCHAR NOT NULL UNIQUE,
email VARCHAR UNIQUE,
password VARCHAR

Actors tábla

actor_id SERIAL PRIMARY KEY NOT NULL,
actor_name VARCHAR

Channels tábla

channel_id SERIAL PRIMARY KEY NOT NULL,
channel_name VARCHAR

My_list tábla

id SERIAL PRIMARY KEY NOT NULL,
user_n VARCHAR NOT NULL,
show_id INTEGER NOT NULL

Shows tábla

show_id SERIAL PRIMARY KEY, NOT NULL,
show_name VARCHAR,
show_img VARCHAR,
show_details VARCHAR,
actor_id INTEGER,
channel_id INTEGER

RELÁCIÓS MODELL

ACCOUNT (user_id,username,email,password)

SHOWS (show_id, show_name, show_img,show_details,actor_id,channel_id)

ACTORS(actor_id, actor_name)

CHANNELS(channel_id,channel_name)

MY_LIST (id, user_n,show_id)

A WEBOLDAL FELÉPÍTÉSE

- **admin.php**
- **query.php**
- **delet.php**
- **delete.php**
- **edit.php**
- **index.php**
- **login.php**
- **register.php**
- **server.php**
- **mylist.php**

A létre hozzás és implemntálás sorrendjében

ELSŐ OLDAL: LOGIN.PHP

Az index oldalon az extra servicek eléréséhez(pl.: my list**) kénytelen be jelentkezni a felhasználó. FORM on keresztűl küldi ,ha hibás a felhasználói név vagy a jelszó nem egyezik az adatbázisban eltároltal akkor a form felet jelzi/ki írja. login hez tartozó kód a server.php-ban:**

```
if (isset($_POST['login_user'])) {
    $username = pg_escape_string($db, $_POST['username']);
    $password = pg_escape_string($db, $_POST['password']);
```

```
    if (empty($username)) {
        array_push($errors, "Username is empty!");
    }
    if (empty($password)) {
        array_push($errors, "Password is empty!");
    }
    if (count($errors) == 0) {
        $password = md5($password);
        $query = "SELECT * FROM account WHERE username='$username' AND
password='$password'";
        $results =  pg_query($db, $query);
        $data = pg_fetch_all($results);
        if (pg_num_rows($results) > 0) {
            while ($row = pg_fetch_assoc($results)) {
                $id = $row['id'];
            }
        }
        if (pg_num_rows($results) == 1) {
            $_SESSION['username'] = $username;
            $_SESSION['id'] = $id;
            $_SESSION['success'] = "Succfully logedin!";
            header('location: index.php');
        } else {
            array_push($errors, "Wrong username or password!");
        }
    }
}
```

MÁSODIK OLDAL: REGISTRATION.PHP

Ha még nem rendelkezik a felhasználó fiókkal, akkor a registration.php oldalon tud létre hozni egyet. Amennyiben a felhasználónév vagy az email cím már foglalt (létezik**) akkor erről tájékoztatja az oldal alert-ön keresztül. registration.php hoz tartozó server.php kód:**

```
if (isset($_POST['reg_user'])) {
    $username = pg_escape_string($db ,$_POST['username']);
    $email = pg_escape_string($db ,$_POST['email']);
    $password_1 = pg_escape_string($db ,$_POST['password_1']);
    $password_2 = pg_escape_string($db ,$_POST['password_2']);
    $usertype = pg_escape_string($db ,$_POST['password_2']);
```

```
    if (empty($username)) {
        array_push($errors, "User name must be set!");
    }
    if (empty($email)) {
        array_push($errors, "Email  name must be set!");
    }
    if (empty($password_1)) {
        array_push($errors, " Password name must be set!");
    }
    if ($password_1 != $password_2) {
        array_push($errors, "Passwords do not match!");
    }
    if (count($errors) === 0) {
    $sql_u = "SELECT * FROM account WHERE username='$username'";
    $sql_e = "SELECT * FROM account WHERE email='$email'";
    $res_u = pg_query($db, $sql_u);
    $res_e = pg_query($db, $sql_e);
    if (pg_num_rows($res_u) > 0) {
        $name_error = "Username is already taken";
        echo '<script type="text/javascript">';
        echo 'alert("Username is already taken");';
        echo 'window.location.href = "registration.php";';
        echo '</script>';
    } else if (pg_num_rows($res_e) > 0) {
        echo '<script type="text/javascript">';
        echo 'alert("Email is already taken");';
        echo 'window.location.href = "registration.php";';
        echo '</script>';
    }else{
            $password = md5($password_1);
            $query = "INSERT INTO account (username, email, password)
                      VALUES('$username', '$email', '$password')";
            pg_query($db, $query);
            $query1 = "SELECT id from account where username='$username'";
            $result = pg_query($db, $query1);
            if (pg_num_rows($result) > 0) {
                while ($row = pg_fetch_assoc($result)) {
                    $id = $row['id'];
                }
            }
```

```
                $_SESSION['username'] = $username;
                $_SESSION['email'] = $email;
                $_SESSION['success'] = "Succesfull login";
                header('location: login.php');

        }
    }
}
```

HARMADIK OLDAL: INDEX.PHP

Itt láthatók a műsorok 3 csatornára és idő szerinti sorendbe, a search mezőben lehet müsor címekre keresnim, amenyiben a felhasználó bejelentkezett akkor a műsorokat hozzá lehet adni egy személyes listához(műsor alján ADD gomb**) ,valamint ki lehet használi az oldalon található Dáma játékot amíg a műsor nem kezdődik.**

NEGYEDIK OLDAL: MY_LIST.PHP

Ittlehet az elmentet műsorokat megtekinteni vagy törölni. a query.php, keresztűl, ahol végignézi hogy a my_list táblában van e és ha van mi van a felhasználó nevéhezz rendelve.(lekérdezések külön lesz feltüntetve**)**

ÖTÖDIK OLDAL: ADMIN.PHP

Ezzen az oldalon lehet az adminnak változtatni a shows tábla tartalmát, mivel nincs role kifejezetten ezért magát a felhasználói nevet elenőrzi az oldal. Itt kifejezetten a hozzá adást lehet meg tenni a törlés és át írás kétt másik oldal végzi el. az oldalon 2 tábla is szereppel a könyebb átláthatóság érdekében és hogy lásuk az eredményét a változtatásoknak.

HATODIK OLDAL: EDIT.PHP

Itt az admin át irhatja a shows tábla adatait. Egy post metodusin kersztül kapja meg az id-t amivel ki keresi az alap adatokat amit be ír egy form-ba, az át írás után elküldi a szervernek ami UPDATE segitségével felűl írja az adatbázis adatait.

```
if(isset($_POST['update_list'])){
    $username = $_SESSION['username'];
    $show_id = pg_escape_string($db,$_POST['show_id']);
    $show_name = pg_escape_string($db,$_POST['show_name']);
    $show_start = pg_escape_string($db,$_POST['show_start']);
    $actor_id = pg_escape_string($db,$_POST['actor_id']);
```

```
    $show_img = pg_escape_string($db,$_POST['show_img']);
    $show_details = pg_escape_string($db,$_POST['show_details']);
    $channel_id = pg_escape_string($db,$_POST['channel_id']);

    if (empty($username)) {
        array_push($errors, "User name is empty!");
    }
    if (empty($show_id)) {
        array_push($errors, "Show id i empty!");
    }

    if (count($errors) == 0){
            $sql = "UPDATE shows SET show_name ='".$show_name ."',
show_details='".$show_details
      ."' , show_start='".$show_start ."', show_img='".$show_img ."',
actor_id='".$actor_id
      ."', channel_id='".$channel_id ."' WHERE show_id = '".$show_id."'";
        pg_query($db, $sql);
        header('location: admin.php');
    }
}
```

HETEDIKOLDAL: DELET.PHP

A müsorok (shows) ból itt lehet torolni az oldal csak egy rövid php oldal amirögtön visza dob az adminramiután végzett felhasználónevet. A jelszavaknak egyeznie kell, különben tájékoztat az oldal hogy nem egyeznek.(kód a delete.php nél lesz feltűntettve**)**

NYOLCADIK OLDAL: DELETE.PHP

Mivel a felhasználó is tud törölni ezért minimális változtatásokkal csináltam a hetedik oldal alapján egy törlést ami a felhasználó listájából törli a nem kívánt müsorokat

```
<?php
$show_id = $_GET['show_id'];
$dbname = "gp6khz";
$conn = pg_connect("host=localhost port=5432 dbname= gp6khz user=gp6khz
password=gp6khz") or die('Cannot log in');
$show_sql = "DELETE FROM my_list WHERE show_id=$show_id";
$show_result = pg_query($show_sql);
```

```
$cmdtuples = pg_affected_rows($show_result);
if (count($errors) == 0) {
    $query = "DELETE FROM my_list WHERE show_id= $show_id";
    pg_query($conn, $query);
    header('location: mylist.php');
}
```

KILENCEDIK OLDAL: SERVER.PHP

A felhasználó/admin formokon keresztűl elküldöt változókat dolgozza fel, első sorban az INSERT és elenőrzésekért felelős, valamint egyes oldalakon elenőrzi hogy afelhasználónak van e jogusultsága amiket még nemtárgyaltunk az a:

3. SHOWS FELVÉTEL:

Megkapja FORM POST-on keresztül a show-nak adatait, majd eltárolja egy változóban. Megnézi hogy a változói NULL-oke , ha igen akkor hibát dob, ha nem akkor felviszi a táblába az adatokat.

```
if (isset($_POST['reg_show'])) {
    $show_name = pg_escape_string($db,$_POST['show_name']);
    $show_start = pg_escape_string($db,$_POST['show_start']);
    $show_details = pg_escape_string($db,$_POST['show_details']);
    $actor_id = pg_escape_string($db,$_POST['actor_id']);
    $channel_id = pg_escape_string($db,$_POST['channel_id']);
    $show_img = pg_escape_string($db,$_POST['show_img']);

    if (empty($show_name)) {
        array_push($errors, "Show name must be added!");
    }
    if (empty($show_start)) {
        array_push($errors, "Starting time must be added!");
    }
    if (empty($show_details)) {        array_push($errors, "Deatails must be added!");
    }
    if (empty($actor_id)) {
        array_push($errors, "Actor must be added!");
    }
    if (empty($channel_id)) {
        array_push($errors, "Channel must be added!");
    }
    if (count($errors) == 0) {
        $query = "INSERT into shows
(show_name,show_start,show_details,actor_id,channel_id,show_img)

values('$show_name','$show_start','$show_details',$actor_id,$channel_id,'$show_img')";
        pg_query($db, $query);
        $_SESSION['show_name'] = $show_name;
        $_SESSION['success'] = "Succes!";
```

```
        header('location: admin.php');
    }
}
```

5. # Személyes műsorlista (MY_LIST) :

Megkapja FORM POST-on keresztül az adatokat a műsorról, mint hidden post. Ellenőrzi hogy a POSTOLT adat nem-e NULL. Amennyiben nem, akkor továbblép. Megnézi az aktuálisan belépett felhasználó USERNME-étt, amelyet úgy SESSION-ből kap meg

```
if(isset($_POST["add_to_list"]) && $_POST["add_to_list"] == "Add"){
    $username = pg_escape_string($db, $_POST['username']);
    $show_id = pg_escape_string($db, $_POST['show_id']);

    if (empty($username)) {
        array_push($errors, "User name is empty!");
    }
    if (empty($show_id)) {
        array_push($errors, "Show id i empty!");
    }
    if (count($errors) == 0) {
        $query = "INSERT INTO my_list (user_n,show_id)  VALUES
('$username',$show_id)";
        pg_query($db, $query);
    }
}
```

TIZENHARMADIK OLDAL: QUERY.PHP

<u>Ebben található az oldalon lévő majdnem összes SELECT.</u>

Minden lekérdezés külön classban van mint function, hogy az adott oldalon mint object tudjak rá hivatkozni , és feltudjam használni for each ciklusban:

1. LEKÉRDEZÉS:

A kifejezet csatorna lekérdezése ebből van három de itt csak ezt mutatom meg mivel csak WHERE en belűl van változás és a class nevében(index.php):

```
class ChannelX{
    protected $conn;
    protected $data = array();
    function __construct()
    {
```

```
        $db = new dbObj();
        $connString =  $db->getConnstring();
        $this->conn = $connString;
    }
    public function getChannel1Shows()
    {
        $sql ="SELECT
s.show_id,s.show_start,s.show_name,s.show_img,s.show_details,a.actor_name FROM shows
AS s JOIN channels AS c ON s.channel_id=c.channel_id JOIN actors AS a ON
a.actor_id=s.actor_id WHERE c.channel_name='BBC' ORDER BY s.show_start ASC;";
        $queryRecords = pg_query($this->conn, $sql) or die("Error during query!");
        $data = pg_fetch_all($queryRecords);
        return $data;
    }
```

2. LEKÉRDEZÉS

A shows tábla összes értéke ösze kapcsolva az actors és a channels táblával (admin.php):

```
class Shows{
    protected $conn;
    protected $data = array();
    function __construct()
    {
        $db = new dbObj();
        $connString =  $db->getConnstring();
        $this->conn = $connString;
    }
    public function getShows()
    {
        $sql ="SELECT
s.show_id,s.show_start,s.show_name,s.show_img,s.show_details,a.actor_name,c.channel_na
me FROM shows AS s JOIN channels AS c ON s.channel_id=c.channel_id JOIN actors AS a ON
a.actor_id=s.actor_id ORDER BY s.show_start ASC;";
        $queryRecords = pg_query($this->conn, $sql) or die("Error during query!");
        $data = pg_fetch_all($queryRecords);
        return $data;
    }
}
```

3. LEKÉRDEZÉS:

Lekéri az adatbázisban lévő összes még nem használt szinészt névvel és id val. ezt egy alllekérdezéssel éri el(admin.php).

```
class NotUsedActors{
    protected $conn;
    protected $data = array();
    function __construct()
    {
        $db = new dbObj();
        $connString =  $db->getConnstring();
        $this->conn = $connString;
    }
    public function getNotUsedActors()
    {
        $sql ="SELECT * FROM actors WHERE actor_id NOT IN(SELECT actor_id FROM shows)
";
        $queryRecords = pg_query($this->conn, $sql) or die("Error during query!");
        $data = pg_fetch_all($queryRecords);
        return $data;
    }
}
```

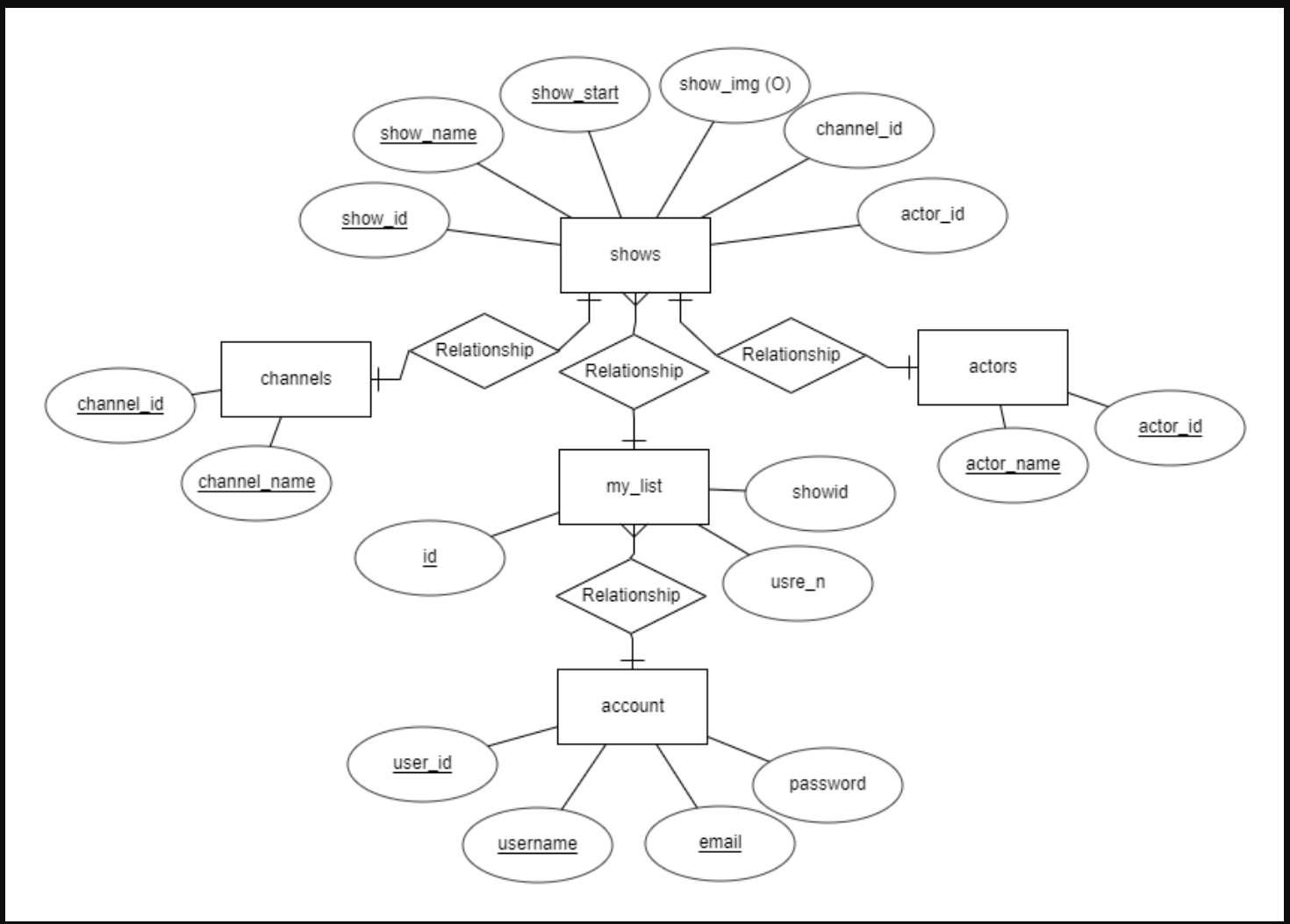
4. LEKÉRDEZÉS

Lekéri a userlist tábla összes értékét ahol a felhasználó neve szerepel a táblában és visza tér az ennek megfelelő műsorokkal(my_list.php).

```
class UserList{
    protected $conn;
    protected $data = array();
    function __construct()
    {
        $db = new dbObj();
        $connString =  $db->getConnstring();
        $this->conn = $connString;
    }
    public function getUserList()
    {
        $sql = "SELECT * FROM my_list AS l JOIN shows AS s ON s.show_id=l.show_id
JOIN account AS u ON u.username=l.user_n ORDER BY s.show_start ASC;";
```

```
        $queryRecords = pg_query($this->conn, $sql) or die("Error during query!");
        $data = pg_fetch_all($queryRecords);
        return $data;
    }
}
```

E-K MODELL



RELÁCIÓS MODELL