

# Jeszenszky Soma

*d44ijw*

Adatbázis rendszerek programozása

30. Állatorvos

<http://157.181.167.180/~d44ijw/index.php>

## Adatbázis felépítése

### User tábla:

userid SERIAL PRIMARY KEY NOT NULL,

username VARCHAR,

email VARCHAR,

phonenummer VARCHAR,

password VARCHAR,

isadmin INTEGER NOT NULL

### Owner tábla:

ownerid SERIAL PRIMARY KEY NOT NULL,

ownername VARCHAR,

owneremail VARCHAR,  
ownerphonenumber VARCHAR,  
address VARCHAR

#### Patient tábla:

patientid SERIAL PRIMARY KEY NOT NULL,  
ownerid INTEGER NOT NULL,  
petname VARCHAR,  
species VARCHAR,  
breed VARCHAR,  
birthday DATE

#### Treatment tábla:

treatmentid SERIAL PRIMARY KEY NOT NULL,  
patientid INTEGER NOT NULL,  
diagnostic VARCHAR,  
treatment VARCHAR,  
medicines VARCHAR

#### Relációs Modell

User(userid, username, email, phonenumber, password, isadmin)

Owner(ownerid, ownername, owneremail, ownerphonenumber, address)

Patient(patientid, ownerid, petname, species, breed, birthday)

Treatment(treatmentid , patientid, diagnostic, treatment, medicines)

#### A weboldal felépítése:

addowner.php

addpatient.php

addtreatment.php

admin.php

datapresentation.php

editowner.php

editpatient.php

index.php

listowners.php

listpets.php

listtreatments.php

registration.php

server.php

## Első oldal:

### 1.registration.php

A felhasználót az admin tudja regisztrálni, mivel ez egy orvosi app, ezért nem tettem mindenki számára elérhetővé, a kód ellenőrzi hogy a felhasználónév az email és a mobil szám már benne van e az adatbázisban és egy script jelzi a hibát, admint csak puttyon keresztül lehet létrehozni. A hozzá tartozó server.phpban lévő kód

```
14  ✓ if (isset($_POST['registrateuser'])) {
15
16      $username = pg_escape_string($_POST['username']);
17      $email = pg_escape_string($_POST['email']);
18      $phonenumner = pg_escape_string($_POST['phonenumner']);
19      $password_1 = pg_escape_string($_POST['password']);
20  ✓  $passwordagain = pg_escape_string($_POST['passwordagain']);
21
22
23      // Nem lehetnek üresek a rublikák
24  ✓  if(empty($username)){
25      |      array_push($errors, "Username can't be empty!");
26      |  }
27  ✓  if(empty($email)){
28      |      array_push($errors, "Email can't be empty!");
29      |  }
30  ✓  if(empty($phonenumner)){
31      |      array_push($errors, "Phonenumner can't be empty!");
32      |  }
33  ✓  if(empty($password_1)){
34      |      array_push($errors, "Password can't be empty!");
35      |  }
36
37  ✓  if($password_1 != $passwordagain){
38      |      array_push($errors, "Passwords don't matches!");
39      |  }
40  }
```

```
//Ha nincs hiba, megnézzük h benne van e már az adatbázisban a felhasználónév és a jelszó
if (count($errors) === 0) {
    $sql_u = "SELECT * FROM user_table WHERE username='$username'";
    $res_u = pg_query($db, $sql_u);
    $sql_e = "SELECT * FROM user_table WHERE email='$email'";
    $res_e = pg_query($db, $sql_e);
    $sql_p = "SELECT * FROM user_table WHERE username='$phonenummer'";
    $res_p = pg_query($db, $sql_p);
    if (pg_num_rows($res_u) > 0) {
        $name_error = "This username is already exist";
        echo '<script type="text/javascript">';
        echo 'alert("This username is already exist");';
        echo 'window.location.href = "registration.php";';
        echo '</script>';
    }else if (pg_num_rows($res_e) > 0) {
        $name_error = "This email is already exist!";
        echo '<script type="text/javascript">';
        echo 'alert("This email is already exist!");';
        echo 'window.location.href = "registration.php";';
        echo '</script>';
    }else if (pg_num_rows($res_p) > 0) {
        $name_error = "This phonenummer is already exist";
        echo '<script type="text/javascript">';
        echo 'alert("This phonenummer is already exist");';
        echo 'window.location.href = "registration.php";';
        echo '</script>';
    }else{
```

Aktiválja a

```
}else{
    if (count($errors) === 0) {
        $password = md5($password_1);
        $query = "INSERT INTO user_table (username, email, phonenummer, password, isadmin)
        VALUES('$username', '$email', '$phonenummer', '$password', 2)";
        pg_query($db, $query);

        $query1 = "SELECT id from users where username='$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['id'] = $id;
        $_SESSION['success'] = "Login successful!";

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

Aktiválja a Wind

## 2. Login (index.php)

A kezdő oldal bejelentkezéssel kezd, amivel az orvos tud belépni, hogy használja az alkalmazást.

Az oldalhoz tartozó kód a server.php-n:

```
if (isset($_POST['loginuser'])){
    $username = pg_escape_string($db, $_POST['username']);
    $password1 = pg_escape_string($db, $_POST['password']);

    if(empty($username)){
        array_push($errors, "Username can't be empty!");
    }
    if(empty($email)){
        array_push($errors, "Email can't be empty!");
    }

    if (count($errors) == 0){
        $password = md5($password1);
        $query = "SELECT * FROM user_table WHERE username='$username' AND password='$password'";

        $results = pg_query($db, $query);
        $data = pg_fetch_all($results);
        if (pg_num_rows($result) > 0) {
            while ($row = pg_fetch_assoc($result)) {
                $id = $row['id'];
            }
        }

        if (pg_num_rows($results) == 1) {
            $_SESSION['username'] = $username;
            $_SESSION['id'] = $id;
            $_SESSION['success'] = "Login successful!";

            header('location: listowners.php');
        }else{
            array_push($errors, "Wrong username or password");
        }
    }
}
```

Aktiválja a Wi  
Aktiválja a Window

### 3. addowner.php

Itt tud az orvos felvenni új gazdát, ellenőrzi hogy a megadott email cím és telefonszám foglalt e már és utána hozzá adja az adatbázishoz. Az oldalhoz tartozó kód a server.php-n:

```
if (isset($_POST['addowner'])) {

    $ownername = pg_escape_string($_POST['ownername']);
    $owneremail = pg_escape_string($_POST['owneremail']);
    $ownerphonenumber = pg_escape_string($_POST['ownerphonenumber']);
    $address = pg_escape_string($_POST['address']);

    if (empty($ownername)) {
        array_push($errors, "Owner name can't be empty");
    }
    if (empty($owneremail)) {
        array_push($errors, "Email address can't be empty");
    }
    if (empty($ownerphonenumber)) {
        array_push($errors, "Phone number can't be empty");
    }
    if (empty($address)) {
        array_push($errors, "Address cant be empty");
    }
}
```

```
if (count($errors) === 0) {

    $sql_e = "SELECT * FROM owner_table WHERE owneremail='$owneremail'";
    $res_e = pg_query($db, $sql_e);
    $sql_p = "SELECT * FROM owner_table WHERE ownerphonenumber='$ownerphonenumber'";
    $res_p = pg_query($db, $sql_p);
    if (pg_num_rows($res_e) > 0) {

        $name_error = "This email is alrely exist!";
        echo '<script type="text/javascript">';
        echo 'alert("This email is alrely exist!");';
        echo 'window.location.href = "addowner.php";';
        echo '</script>';
    } else if (pg_num_rows($res_p) > 0) {

        $name_error = "This phonenumber is alrely exist";
        echo '<script type="text/javascript">';
        echo 'alert("This phonenumber is alrely exist!");';
        echo 'window.location.href = "addowner.php";';
        echo '</script>';
    } else {
        if (count($errors) == 0) {
            $query = "INSERT INTO owner_table (ownername, owneremail, ownerphonenumber, address)
VALUES('$ownername', '$owneremail', '$ownerphonenumber', '$address')";
            pg_query($db, $query);

            $_SESSION['success'] = "Owner added!";
            header("location: listowners.php");
        }
    }
}
```

#### 4. addpatient.php

Itt tud az orvos felvenni új háziállatot a gazdához, itt a program csak azt figyeli, hogy nem lehetne üresek a mezők. Az oldalhoz tartozó kód a server.php-n:

```
if (isset($_POST['addpatient'])) {

    $ownerid = intval(pg_escape_string($_POST['ownerid']));
    $petname = pg_escape_string($_POST['petname']);
    $species = pg_escape_string($_POST['species']);
    $breed = pg_escape_string($_POST['breed']);
    $birthday = pg_escape_string($_POST['birthday']);

    if (empty($petname)) {
        array_push($errors, "Pet name can't be empty");
    }
    if (empty($ownerid)) {
        array_push($errors, "Ownerid can't be empty");
    }
    if (empty($species)) {
        array_push($errors, "Species can't be empty");
    }
    if (empty($breed)) {
        array_push($errors, "Breed can't be empty");
    }
    if (empty($birthday)) {
        array_push($errors, "Birthday can't be empty");
    }

    if (count($errors) === 0) {
        $query = "INSERT INTO patient_table (ownerid, petname, species, breed, birthday)
VALUES('$ownerid', '$petname', '$species', '$breed', '$birthday')";
        pg_query($db, $query);

        $_SESSION['success'] = "Pet added!";

        header("location: listpets.php?id=$ownerid");
    }
}
```

## 5. addtreatment.php

Itt tud az orvos felvenni a betegnek diagnózist, kezelés írni, és gyógyszereket felírni, a gyógyszer és a kezelés is lehet null, mivel nem feltétlenül kell azokat a mezőket kitölteni az orvosnak ezért engedélyezve van. Az oldalhoz tartozó kód a server.php-n:

```
if (isset($_POST['addtreatment'])) {  
  
    $ownerid = intval(pg_escape_string($_POST['ownerid']));  
    $patientid = pg_escape_string($_POST['patientid']);  
    $diagnostic = pg_escape_string($_POST['diagnostic']);  
    $treatment = pg_escape_string($_POST['treatment']);  
    $medicines = pg_escape_string($_POST['medicines']);  
  
    if (empty($diagnostic)) {  
        array_push($errors, "Diagnostic can't be empty");  
    }  
  
    if (count($errors) === 0) {  
        $query = "INSERT INTO treatment_table (patientid, diagnostic, treatment, medicines)  
VALUES('$patientid', '$diagnostic','$treatment', '$medicines')";  
        pg_query($db, $query);  
  
        $_SESSION['success'] = "Treatment added!";  
  
        header("location: listpets.php?id=$ownerid");  
    }  
}
```

## 6.editowner.php

Ezen az oldalon lehet szerkeszteni a tulajdonosnak az adatait ha esetleg valamit elrontott volna az orvos vagy esetleg költözés vagy telefonszám csere történt volna. Megkapja az eredeti adatokat, amiket hiddenbe tárol, tehát elég csak átírni az oldalon a megváltoztatni kívánt adatot és csak az fog módosulni. Az oldalhoz tartozó kód a server.php-n:



```

if (isset($_POST['editowner'])) {
    $ownerid = pg_escape_string($_POST['ownerid']);
    $ownername = pg_escape_string($_POST['ownername']);
    $owneremail = pg_escape_string($_POST['owneremail']);
    $ownerphonenumber = pg_escape_string($_POST['ownerphonenumber']);
    $address = pg_escape_string($_POST['address']);

    $or_ownername = pg_escape_string($_POST['or_ownername']);
    $or_owneremail = pg_escape_string($_POST['or_owneremail']);
    $or_ownerphonenumber = pg_escape_string($_POST['or_ownerphonenumber']);
    $or_address = pg_escape_string($_POST['or_address']);

    $queryownern = empty($ownername) ? $or_ownername : $ownername;
    $queryownere = empty($owneremail) ? $or_owneremail : $owneremail;
    $queryownerp = empty($ownerphonenumber) ? $or_ownerphonenumber : $ownerphonenumber;
    $queryownera = empty($address) ? $or_address : $address;

    $query = "UPDATE owner_table SET ownername = '$queryownern',
owneremail = '$queryownere' ,
ownerphonenumber = '$queryownerp' ,
address = '$queryownera'
WHERE ownerid = $ownerid" ;
    pg_query($db, $query);
    $_SESSION['success'] = "success !!!";

    header('location:listowners.php ');
}

```

## 7. editpatient.php

Ezen az oldalon lehet a kiskedvenc adatait módosítani, ha szükségesnek érzi az orvos.

az oldalhoz tartozó server.php kód:

```

if (isset($_POST['editpatient'])) {
    $ownerid = pg_escape_string($_POST['ownerid']);
    $patientid = pg_escape_string($_POST['patientid']);
    $petname = pg_escape_string($_POST['petname']);
    $species = pg_escape_string($_POST['species']);
    $breed = pg_escape_string($_POST['breed']);
    $birthday = pg_escape_string($_POST['birthday']);

    $or_petname = pg_escape_string($_POST['or_petname']);
    $or_species = pg_escape_string($_POST['or_species']);
    $or_breed = pg_escape_string($_POST['or_breed']);
    $or_birthday = pg_escape_string($_POST['or_birthday']);

    $querypatientn = empty($petname) ? $or_petname : $petname;
    $querypatients = empty($species) ? $or_species : $species;
    $querypatientbr = empty($breed) ? $or_breed : $breed;
    $querypatientbi = empty($birthday) ? $or_birthday : $birthday;

    $query = "UPDATE patient_table SET patientname = '$querypatientn',
species = '$querypatients' ,
breed = '$querypatientbr' ,
birthday = '$querypatientbi'
WHERE patientid = $patientid" ;
    pg_query($db, $query);
    $_SESSION['success'] = "success !!!";

    header('location:listowners.php ');
}

```

## 8. listowners.php

Ezen az oldalon kilistázza az orvosnak az adatbázisban lévő tulajdonosokat, akiknél lehet választani gombokkal hogy szeretnék módosítani az adatokon vagy új kedvencet hozzáadni, esetleg kilistázni már meglévő kedvenceit egy adott felhasználónak és az adott felhasználónak az állataihoz tartozó kezelések listázása is innen érhető el. Lehet névre keresni.

## 9. listpets.php

Az oldalra GET-n keresztül érkezik az ownerid, és ennek a segítségével kerül kilistázásra az adott tulajdonos kedvenceinek a listája.

## 10. listtreatment.php

Ezen az oldalra GET Kéréssel érkezik meg az ownerid, és a patientid, ezek segítségével kerül kilistázásra a tulajdonos kedvenceinek az eddigi kezeléseit csoportosítva a kedvenceihez.

## 11. server.php

Minden oldal elején autentikációval kezdődik és a rendszer ellenőrzi, hogy be van-e lépve a felhasználó, ha nem akkor átirányítja az index.php-ra ahol belépés után tudja használni az oldalt.

### 1. regisztráció

Poston keresztül megkapja az oldal a felhasználónevet, az email címet és a telefonszámát az orvosnak, ezeket ellenőrzi, hogy szerepelnek-e az adatbázisban, és ellenőrzi, hogy a jelszó és a jelszó újra megegyezik-e. Hiba esetén visszadob a registration.php-ra egy hibaüzenettel amiről javascript gondoskodik, ha minden adat stimmel és a hibát számoló tömb 0 hosszú, felviszi a felhasználót az adatbázisba és belépteti az alkalmazásba.

### 2. login

Poston keresztül megkapja a felhasználónevet és a jelszót, amiket ellenőriz, hogy megfelelnek-e az adatbázisban szereplőkkel ha sikeres a belépés, akkor tulajdonosok listázása oldalra irányítja az orvost.

### 3. addowner

Poston megkapja a tulajdonos nevét, telefonszámát, emailcímét, és lakcímét, ellenőrzi hogy az email cím és a telefonszám szerepel-e már az adatbázisban, ha igen egy hibaüzenettel visszadob az oldalra, ha minden megfelel akkor felveszi az adatbázisba. Átirányítva az orvos a páciensek listázásához.

### 4. addpatient

Poston keresztül megkapja az ownerid-t, az állat nevét, fajtát, és fajtaját, és születésnapját, megnézi, hogy nem-e üresek a kapott rublikák és felviszi az adatbázisba a kapott adatokat. Siker esetén átirányít a tulajdonoshoz tartozó kisállat kilistázó oldalra.

### 5. addtreatment

Poston keresztül megkapja a patientid-t az ownerid-t, és a diagnózist, kezelést és gyógyszereket. A Diagnózis nem maradhat üresen. Ezután felviszi az adatbázisba és átirányít a kiskedvencek kilistázásához.

### 6. editowner

Poston keresztül megkapja az eredeti adatokat, és megkapja a beírt adatokat, ha nincs benne adat, akkor vissza írja a régi adatot az adatbázisba, ha jött új adat azt kicseréli az adatbázisban.

7. editpatient

Poston keresztül megkapja az eredeti adatokat, és megkapja a beírt adatokat, ha nincs benne adat, akkor vissza írja a régi adatot az adatbázisba, ha jött új adat azt kicseréli az adatbázisban.

12. admin.php

Ezen az oldalon az admin látja a felhasználókat, és különböző statisztikákat az oldalról. Kinek hány állata van, melyik állatnak mennyi kezelése van, kinek van kutyája vagy macskája, kiknek kell diétáznia. Innen tudja elérni a registratiónt ahonnan új usert tud felvinni.

13. datapresentation.php

Itt található majdnem az összes SELECT. Minden lekérdezés külön classban van, mint function, így az oldalon objectként tudok rá hivatkozni, és foreach ciklussal be tudom járni.

## datapresentation.php

### 1.Lekérdezés

Elkérjük a user adatait az adatbázisból.

```
class User
{
    protected $conn;
    protected $data = array();
    function __construct()
    {
        $db = new dbObj();
        $connString = $db->getConnstring();
        $this->conn = $connString;
    }

    public function getUser()
    {
        $id = $_SESSION['id'];
        $sql = "SELECT * FROM user_table where id = '$id'";
        $queryRecords = pg_query($this->conn, $sql) or die("problem, with getting the user");
        $cmdtuples = pg_affected_rows($queryRecords);
        $row = pg_fetch_assoc($queryRecords);
        return $row;
    }
}
```

## 2. Lekérdezés

Elkérjük az összes user adatot.

```
public function getAllUser()
{
    $sql = "SELECT * FROM user_table WHERE isadmin=2;";
    $queryRecords = pg_query($this->conn, $sql) or die("error, with getting users");
    $data = pg_fetch_all($queryRecords);
    return $data;
}
```

## 3. Lekérdezés

Elkéri egy tulajdonos adatait.

```
public function getOwner(int $ownerid)
{
    $sql = "SELECT * FROM owner_table where ownerid =$ownerid;";
    $queryRecords = pg_query($this->conn, $sql) or die("error, with getting owner");
    $cmdtuples = pg_affected_rows($queryRecords);
    $row = pg_fetch_assoc($queryRecords);
    return $row;
}
```

## 4. Lekérdezés

Elkéri az össze gazdának az adatát.

```
public function getAllOwner()
{
    $sql = "SELECT * FROM owner_table order by ownerid desc;";
    $queryRecords = pg_query($this->conn, $sql) or die("error, with getting owner");
    $data = pg_fetch_all($queryRecords);
    return $data;
}
```

ss Pet

## 5. Lekérdezés

Elkéri egy kisállatnak az adatait.

```
public function getPet(int $patientid)
{
    $sql = "SELECT * FROM patient_table WHERE patientid = $patientid;";
    $queryRecords = pg_query($this->conn, $sql) or die("error, with getting owner");
    $cmdtuples = pg_affected_rows($queryRecords);
    $row = pg_fetch_assoc($queryRecords);
    return $row;
}
```

## 6. Lekérdezés

Elkéri az összes kisállat adatait.

```
public function getAllPetOfTheOwner(int $ownerid)
{
    $sql = "SELECT * FROM patient_table WHERE ownerid = $ownerid;";
    $queryRecords = pg_query($this->conn, $sql) or die("error, with getting owner");
    $data = pg_fetch_all($queryRecords);
    return $data;
}
```

## 7. Lekérdezés

Elkéri az összes kezelést egy adott kisállatnak a patient id-ja segítségével.

```
public function getAllTreatment(int $patientid)
{
    $sql = "SELECT * FROM treatment_table where patientid = $patientid;";
    $queryRecords = pg_query($this->conn, $sql) or die("error, with getting data");
    $data = pg_fetch_all($queryRecords);
    return $data;
}
```

## 8. Lekérdezés

Elkéri a tulajdonos neveket és megmondja kinek mennyi kisállata van bent.

```
public function countOwnersPatients()
{
    $sql = "SELECT ownername, COUNT(*) FROM owner_table ot
join patient_table pt on ot.ownerid = pt.ownerid
GROUP BY ownername
order by ownername asc;";
    $queryRecords = pg_query($this->conn, $sql) or die("error, with getting data");
    $data = pg_fetch_all($queryRecords);
    return $data;
}
```

## 9. Lekérdezés

Elkéri az állatok neveit és hogy mennyi kezelésük van.

```
public function countPatientsTreatments()
{
    $sql = "SELECT petname , COUNT(*) FROM patient_table pt
JOIN treatment_table tt on pt.patientid = tt.patientid
group by petname order by petname asc;";
    $queryRecords = pg_query($this->conn, $sql) or die("error, with getting data");
    $data = pg_fetch_all($queryRecords);
    return $data;
}
```

## 10. Lekérdezés

Lekéri kinek az a kezelése, hogy diétáznia kell.

```
public function getAllPatientWithDietTreatment()
{
    $sql = "SELECT petname, treatment FROM patient_table pt
    JOIN treatment_table tt on pt.patientid = tt.patientid where treatment = 'Diet'; ";
    $queryRecords = pg_query($this->conn, $sql) or die("error, with getting data");
    $data = pg_fetch_all($queryRecords);
    return $data;
}
```

## 11. Lekérdezés

Elkéri azokat akiknek vagy kutya vagy macskája van.

```
public function getOwnersWhosHaveDogAndCat()
{
    $sql = "SELECT distinct ownername FROM owner_table ot
    join patient_table pt on ot.ownerid = pt.ownerid where pt.species='Dog' or pt.species='Cat';";
    $queryRecords = pg_query($this->conn, $sql) or die("error, with getting data");
    $data = pg_fetch_all($queryRecords);
    return $data;
}
```

## EK Modell

