**Web-programozás II**

Beadandó feladat

Játékok weboldal

Neumann János Egyetem

*Informatika Tanszék*

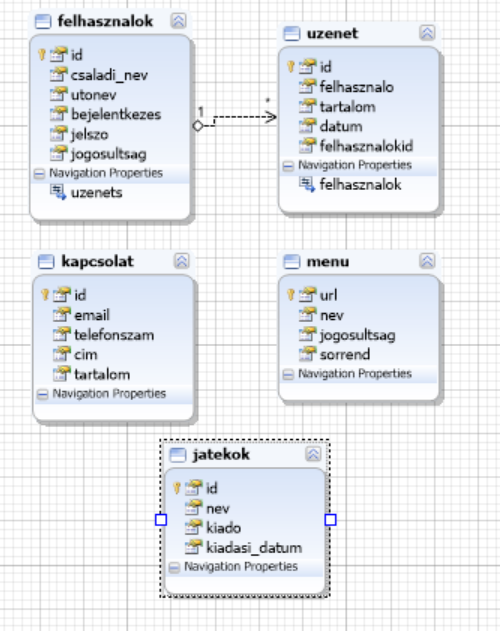
Bárándi Dominik (W93NMV)

Szabó Dániel (Z3IK64)

2023.10.20

Adatbázis - *| Készítette: Bárándi Dominik, Szabó Dániel*

Az adatbázisban 5 tábla van.



A **felhasznalok** táblában tároljuk a felhasználókat.

Az **uzenet** táblában tároljuk a felhasználók által küldött üzeneteket.

A **kapcsolat** táblában tároljuk az adminnak küldött üzeneteket (csak az admin láthatja).

A **menu** táblában az oldalon használt összes url megtalálható.

A **játékok** táblában a játékokról szóló adatokat tároljuk.

1. feladat – *MVC tervezési minta | Készítette: Bárándi Dominik, Szabó Dániel*

index.php tartalma:

//alkalmaz�s gy�k�r k�nyvt�ra a szerveren

define('SERVER\_ROOT', $\_SERVER['DOCUMENT\_ROOT'].'/Beadando/');

//URL c�m az alkalmaz�s gy�ker�hez

define('SITE\_ROOT', 'http://localhost/Beadando/');

// a router.php vez�rl� bet�lt�se

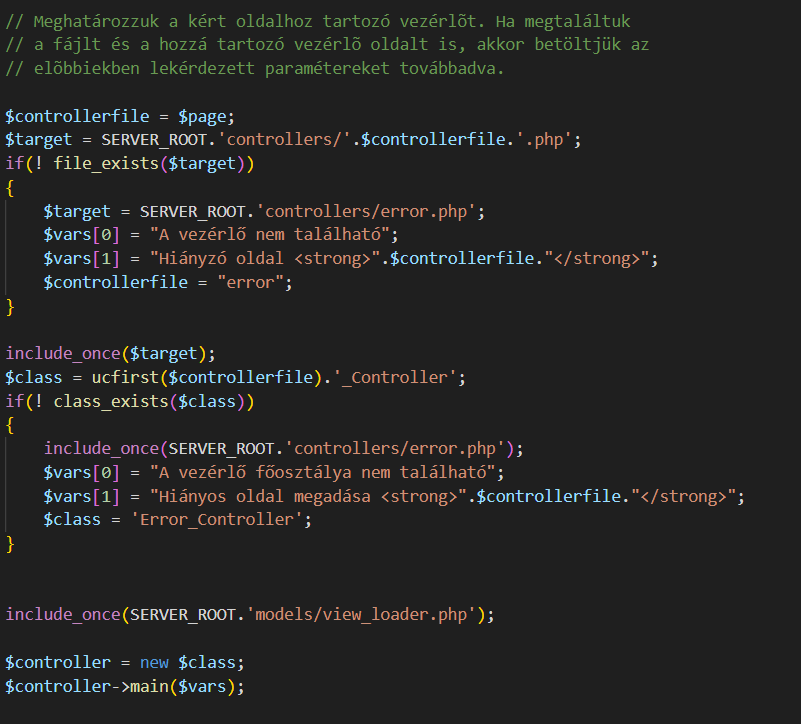
require\_once(SERVER\_ROOT . 'controllers/' . 'router.php');

router.php tartalma:

view bekérése



A hibák kiszűrése (nem létező vezérlő)



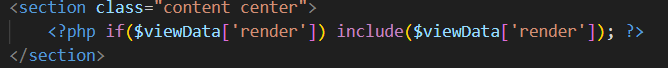
view\_loader modell tartalma:

Azok az adatok amiket átadunk a view-nak

* konstruktorral megadjuk
* assign-al módosíthatjuk és hozzáadhatjuk

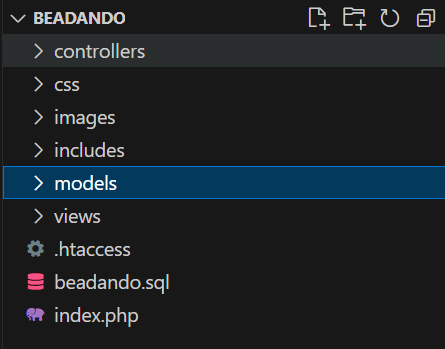


page\_main-be beillesztjük az adatot

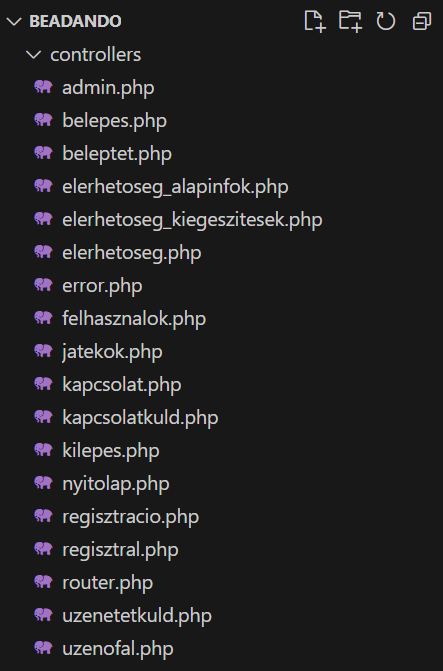


2. feladat - *Mappák rendszer-szerű struktúrája | Készítette: Szabó Dániel*

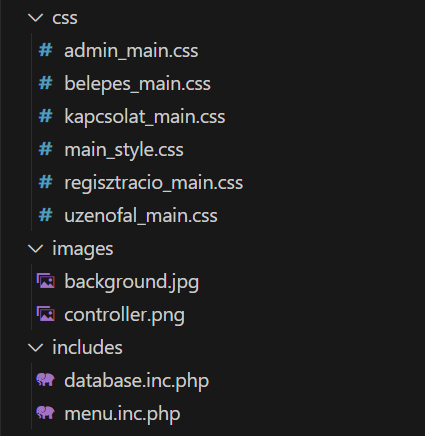
A mappa rendszer



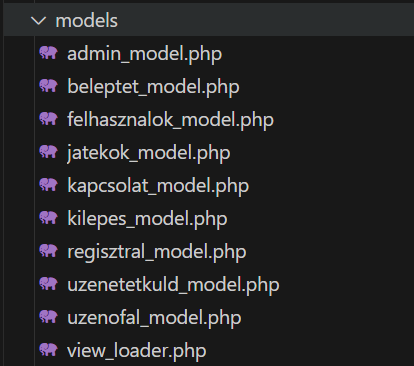
Controllers mappa tartalma



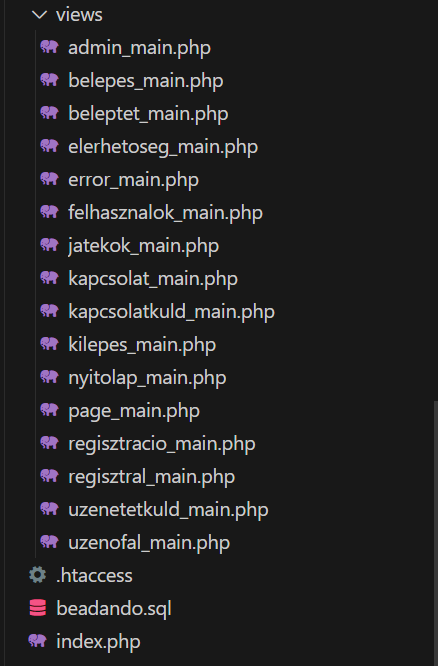
Css-includes-images mappák tartalma



Models mappa tartalma



Views mappa tartalma

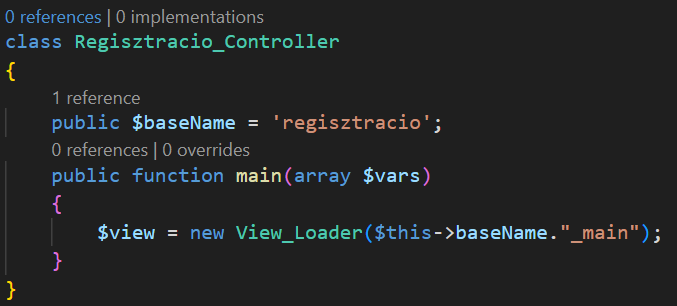


4. feladat – *Regisztráció/belépés | Készítette: Szabó Dániel*

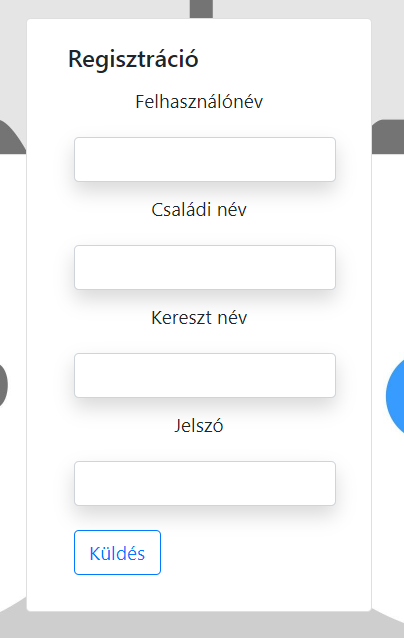
Regisztráció

Ezt az oldalt csak a látogatók láthatják, bejelentkezett felhasználók nem.

regisztracio.php nevű kontrollerrel nyitjuk meg az oldalt (regisztacio\_main.php).



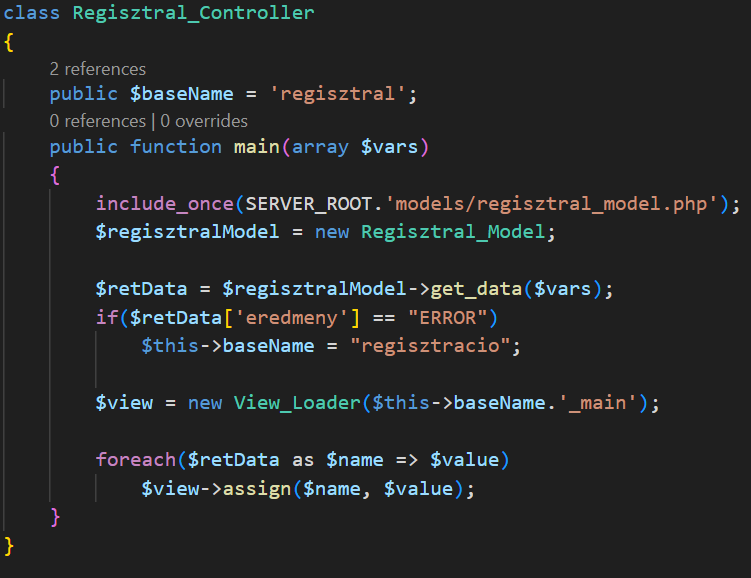
Megjelenés:



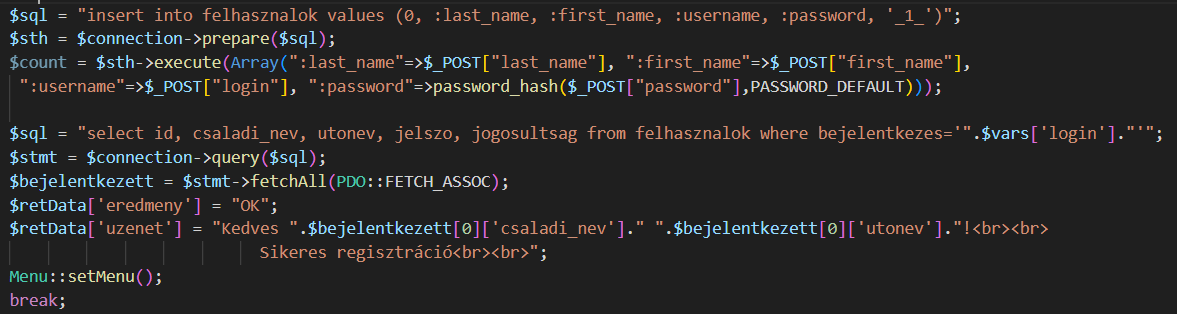
A form kitöltése után regisztral.php kontrollerre küldi az információt, ami a regisztral\_model.php modellel ellenőrzi, hogy létezik-e már ilyen felhasználó.

Abban az esetben, ha nem talál és nincs adatbázis hiba akkor beilleszti a felhasználók táblába sima felhasználó jogosultsággal.

regisztal.php tartalma:



regisztal\_model.php beillesztés része



Sikeres regisztáció után a regisztal\_main.php view kerül betöltésre, ami egy üzenetet tartalmaz.

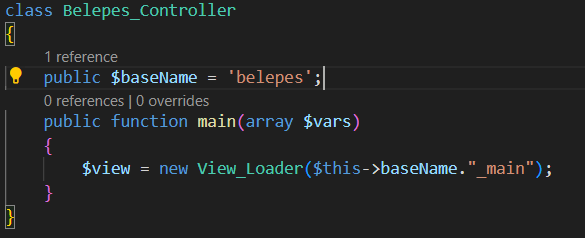
Bejelentkezés

A bejelentkezés hasonlóan működik, mint a regisztráció.

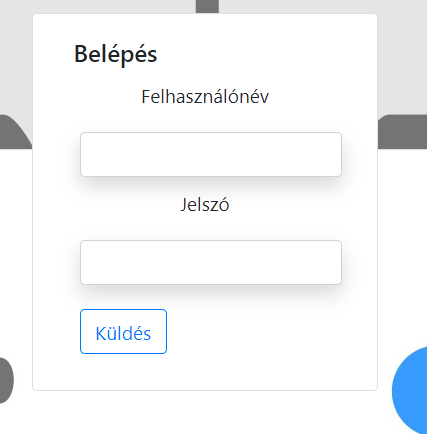
Ezt az oldalt csak a látogatók láthatják, bejelentkezett felhasználók nem.

belepes.php jeleníti meg az oldalt(bepeles\_main.php).

belepes.php tartalma:



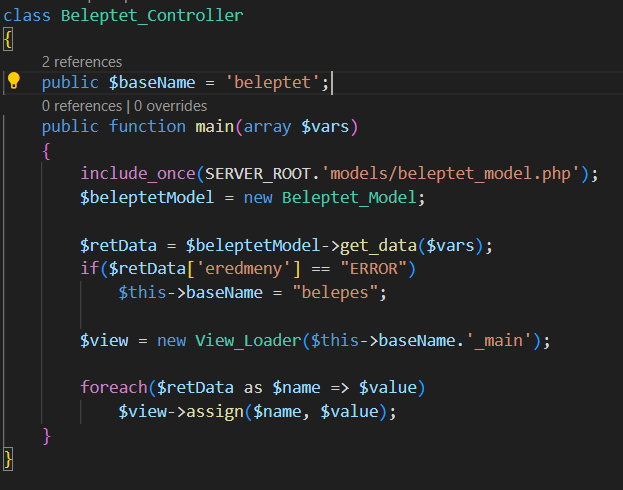
Megjelenés:



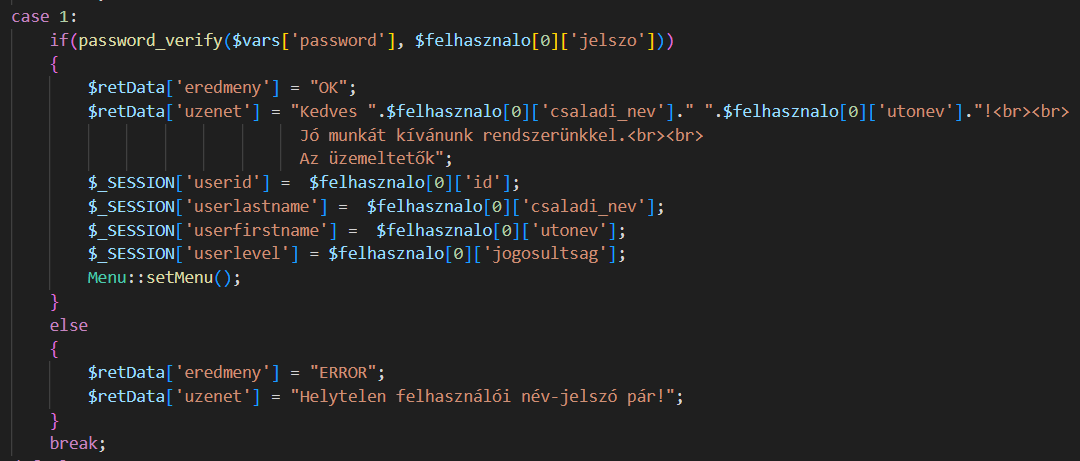
A form kitöltése után beleptet.php kontrollerre küldi az információt, ami a beleptet \_model.php modellel ellenőrzi, hogy létezik-e ilyen felhasználó.

Abban az esetben, ha nincs adatbázis hiba és helyesen adott meg minden adatot akkor belépteti a felhasználót.

beleptet.php tartalma:

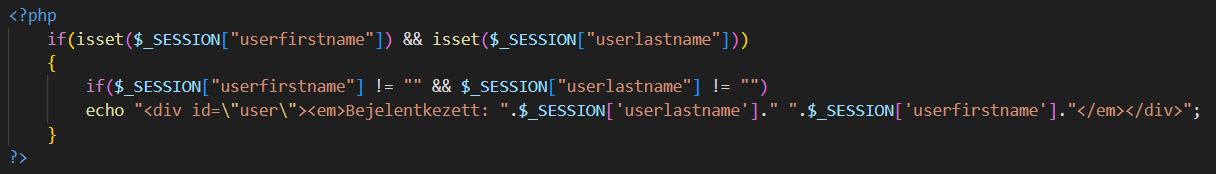


beleptet\_model.php beleptetés része:



A SESSION változókat beállítjuk és ezekre hivatkozunk az oldalakon.

belépett felhasználó kiíratása a page\_main.php-ben:



Megjelenés:

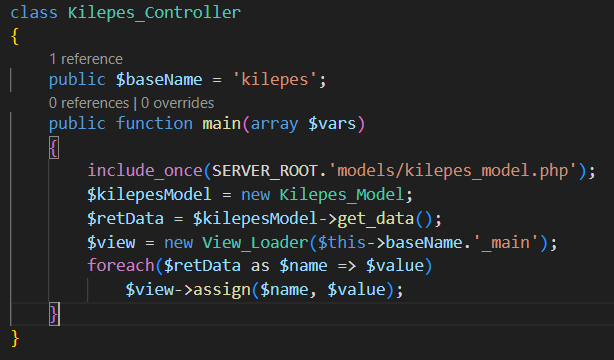


Kilépés

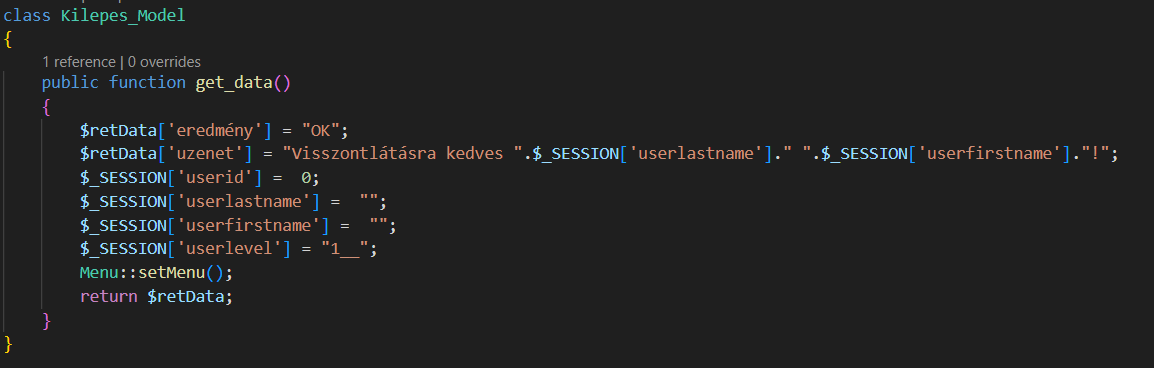
Ezt az oldalt csak a bejelentkezett felhasználók láthatják.

A kilepes.php nevű kontroller a kilepes\_model.php nevű modellel nullázza a SESSION változókat és megnyitja a kilepes\_main.php nevű oldalt az üzenettel.

kilepes.php tartalma:



kilepes\_model.php tartalma:



5. feladat – *Üzenőfal | Készítette: Bárándi Dominik*

Az üzenőfalat csak a bejelentkezett felhasználó vagy az adminisztrátor láthatja. Az üzeneteket adatbázisban tároljuk, és amikor elküldünk egyet automatikusan meg is jelenítjük a legfrissebbet. Elküldött üzenet után visszajelzést írunk ki.

Kártya megjelenítésnél fejlécben az üzenet sorszáma, felhasználónév található. Tartalomban pedig az üzenet alatta pedig a dátum.

*Üzenet preparált beszúrása*

class Uzenetetkuld\_Model

{

    public function get\_data()

    {

        $retData['eredmeny'] = "";

        try {

            $connection = Database::getConnection();

            $sql = "insert into uzenet values (0, :felhasznalo, :tartalom, :datum)";

            $sth = $connection->prepare($sql);

            $count = $sth->execute(Array(":felhasznalo"=>$\_SESSION['userlastname'], ":tartalom"=>$\_POST['textarea'], ":datum"=>date("Y-m-d")));

        }

        catch (PDOException $e) {

                    $retData['eredmeny'] = "ERROR";

                    $retData['uzenet'] = "Adatbázis hiba: ".$e->getMessage()."!";

        }

        return $retData;

    }

}

*Üzenetek megjelenítése dátum szerint legfrissebb*

class Uzenofal\_Model

{

    public function get\_data()

    {

        $retData['eredmeny'] = "OK";

        $retData['rows'] = Array();

        try {

            $connection = Database::getConnection();

            $sql = "select id, felhasznalo, tartalom, datum from uzenet order by datum desc";

            $stmt = $connection->query($sql);

            $retData['rows'] = $stmt->fetchAll(PDO::FETCH\_ASSOC);

        }

        catch (PDOException $e) {

            $retData['eredmeny'] = "ERROR";

            $retData['uzenet'] = "Hiba:<br>".$e->getMessage()."!";

        }

        return $retData;

    }

}

<?php

switch ($viewData['eredmeny']) {

    case "OK":

        echo "<h3>Üzenetek listája:</h3>";

?>

        <div class="message-container">

            <?php foreach ($viewData['rows'] as $row) { ?>

                <div class="card">

                    <div class="card-header">

                        <div class="User"><?php echo $row['id'] . '.   '; ?><?php echo $row['felhasznalo']; ?></div>

                    </div>

                    <div class="card-body">

                        <p class="card-text"><?php echo $row['tartalom']; ?></p>

                        <i><?php echo $row['datum']; ?></i>

                    </div>

                </div>

            <?php } ?>

        </div>

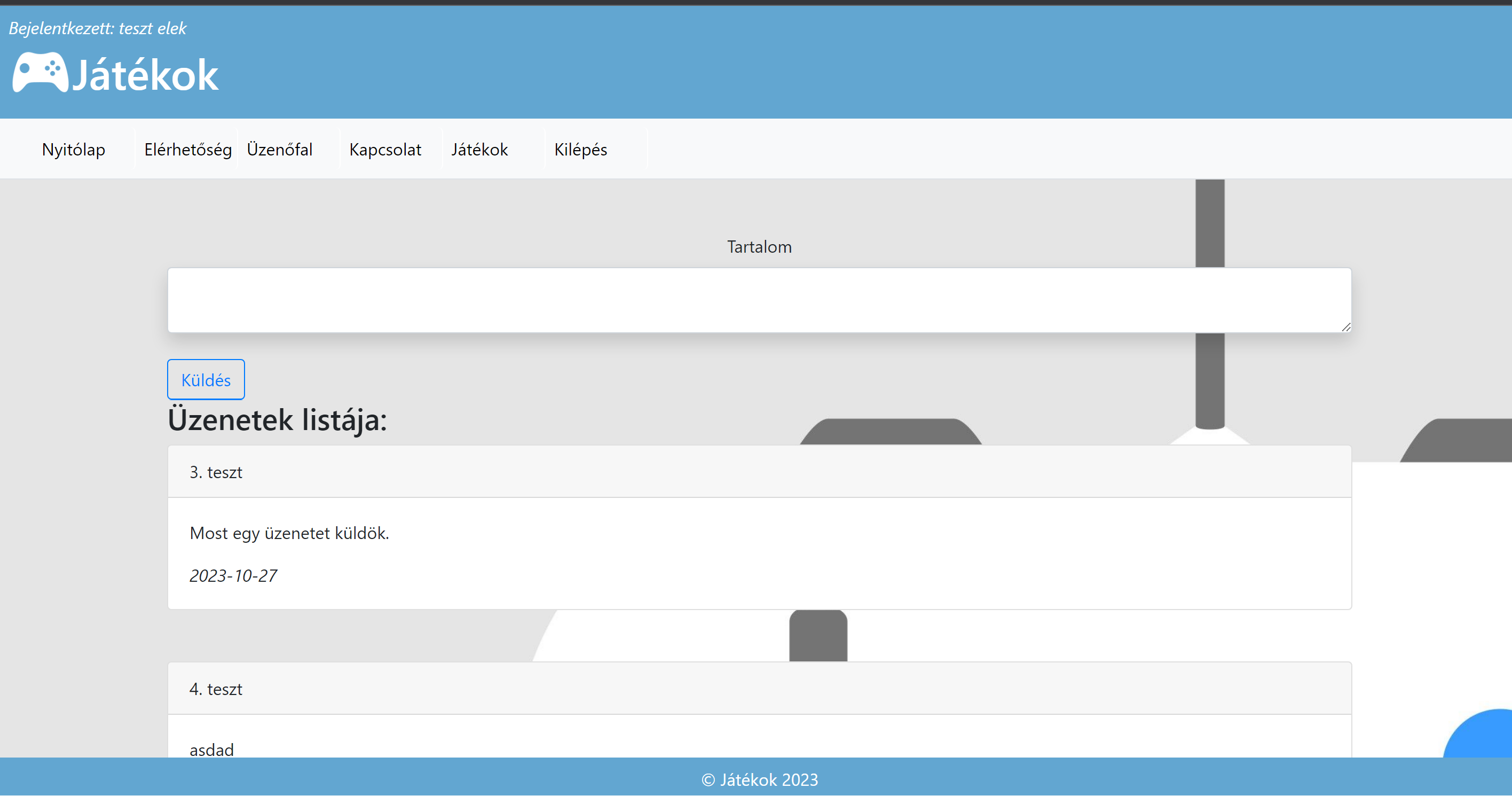
        <?php break;

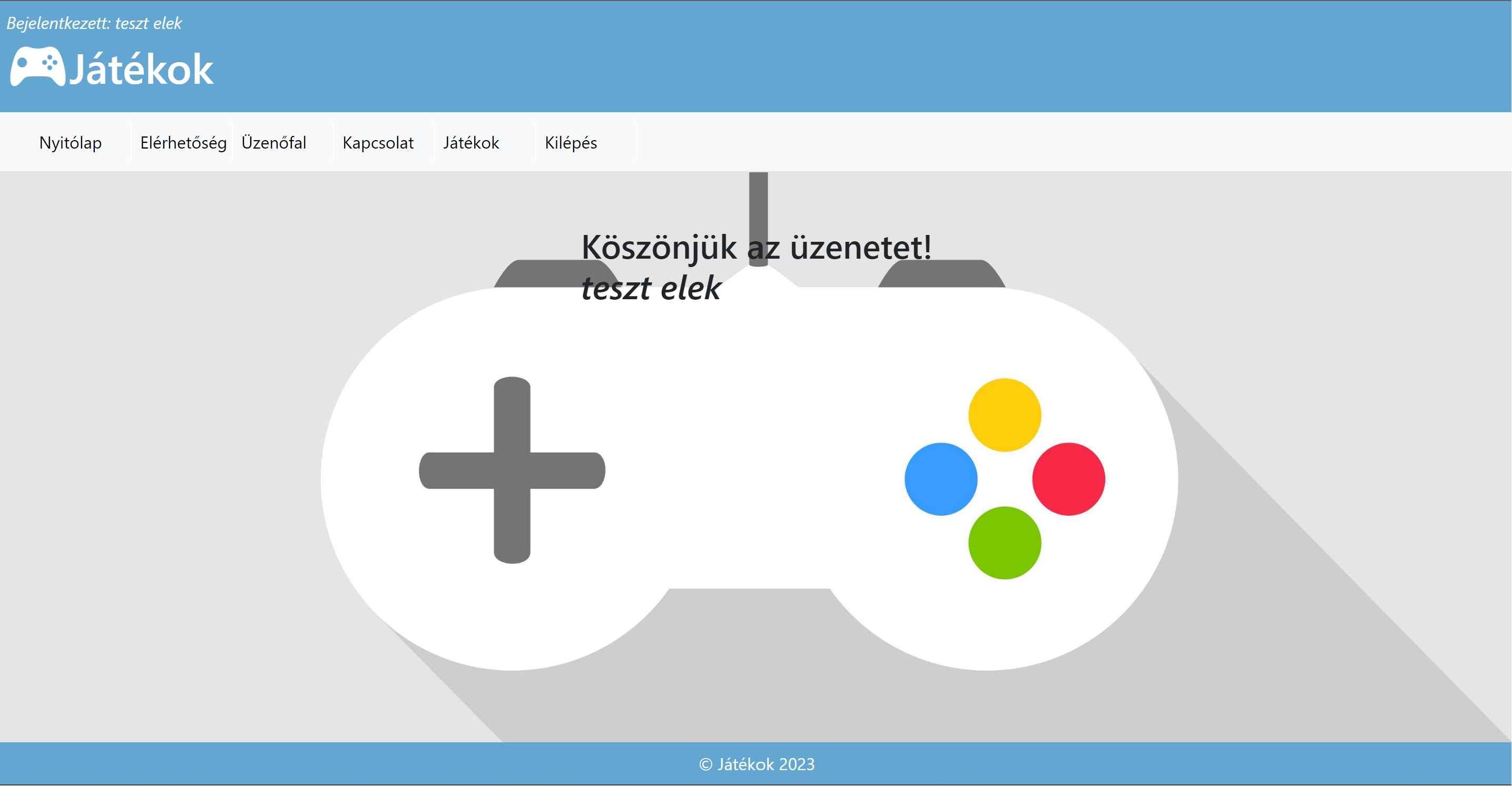
    case "ERROR":

        echo "<p>" . $viewData['uzenet'] . "</p>";

        break;

}





6-7. feladat - *Kapcsolati oldal | Készítette: Bárándi Dominik*

Kapcsolati oldalon egy űrlap segítségével lehet üzenetet küldeni az adminisztrátornak. Az üzeneteket adatbázisban tároljuk, nem jelenítjük meg csak az admin láthatja ezeket. Cím tartalom és alul a telefonszámmal ami nem kötelező.

*Kötelező és az input mezők létrehozása, visszajelzés az elküldött üzenet után*

#Kapcsolati nézet

<div>

    <h1>Kapcsolat</h1>

    <form action="<?= SITE\_ROOT ?>kapcsolatkuld" method="post">

        <div>

            <label for="email" class="w-100 text-center">Email</label>

            <input type="email" name="email" id="email" class="shadow form-control" required>

        </div>

        <div>

            <label for="phone" class="w-100 text-center">Telefonszám</label>

            <input type="text" name="phone" id="phone" class="shadow form-control">

        </div>

        <div>

            <label for="title" class="w-100 text-center">Cím</label>

            <input type="text" name="title" id="title" class="shadow form-control" required>

        </div>

        <div>

            <label for="content" class="w-100 text-center">Tartalom</label>

            <textarea type="textarea" name="textarea" id="textarea" class="shadow form-control" required></textarea>

        </div>

        <input id="Alertbtn" type="submit" class="btn btn-outline-primary" value="Küldés">

    </form>

</div>

# Visszajelzés

<?php

if (!isset($\_SESSION['userlastname']) || empty($\_SESSION['userlastname'])) {

    $\_SESSION['userlastname'] = 'Látogató';

}

?>

<h2>Üzenet elküldve az adminisztrátornak! <div id="user"><em><?= $\_SESSION['userlastname']." ".$\_SESSION['userfirstname'] ?></em></div></h2>

// Kapcsolati modell

class Kapcsolat\_Model

{

    public function get\_data()

    {

        $retData['eredmeny'] = "";

        try {

            $connection = Database::getConnection();

            $sql = "insert into kapcsolat values (0, :email, :telefonszam, :cim,:tartalom)";

            $sth = $connection->prepare($sql);

            $count = $sth->execute(Array(":email"=>$\_POST['email'], ":telefonszam"=>$\_POST['phone'], ":cim"=>$\_POST['title'],":tartalom"=>$\_POST['textarea']));

        }

        catch (PDOException $e) {

                    $retData['eredmeny'] = "ERROR";

                    $retData['uzenet'] = "Adatbázis hiba: ".$e->getMessage()."!";

        }

        return $retData;

    }

}

*Admin modell, ő láthatja a beküldött kapcsolati űrlapokat*

class Admin\_Model

{

    public function get\_data()

    {

        $retData['eredmeny'] = "OK";

        $retData['rows'] = Array();

        try {

            $connection = Database::getConnection();

            $sql = "select id, email, telefonszam, cim, tartalom from kapcsolat order by id";

            $stmt = $connection->query($sql);

            $retData['rows'] = $stmt->fetchAll(PDO::FETCH\_ASSOC);

        }

        catch (PDOException $e) {

            $retData['eredmeny'] = "ERROR";

            $retData['uzenet'] = "Hiba:<br>".$e->getMessage()."!";

        }

        return $retData;

    }

}

