**Weprog\_I\_Beadando: Eletfa-projekt**

Tartalomjegyzék

[Feladat leírása 2](#_Toc102078128)

[A projekt bemutatása 3](#_Toc102078129)

[Gépi és nyelvi igények 3](#_Toc102078130)

[Mappa és fájlszerkezet bemutatása 4](#_Toc102078131)

[Tesztelés 20](#_Toc102078132)

[Felhasználói leírás 20](#_Toc102078133)

[Fejlesztési lehetőségek 20](#_Toc102078134)

[Külső állományok bemutatása 20](#_Toc102078135)

Feladat leírása

A Web-programozás-1 beadandó feladat tetszőlegesen választható nonprofit szervezet weboldalának újra tervezéséhez, az Életfa Környezetvédő Szövetség weboldala került választásra, melynek eredeti változata a <https://eletfa.org.hu> hivatkozáson keresztül tekinthető meg.

A weboldalt Front Controller dinamikus tervezési minta alapján került megvalósításra, vízszintes menü alkalmazásával.

Az alkalmazás főbb funkciói a kép feltöltés, küldött üzenetek mentése adatbázisba, regisztráció és beléptetési rendszer, illetve az adatbázis és azon belül a működéshez szükséges relációs táblák automatizált elkészítése került megvalósításra.

A projekt bemutatása

Gépi és nyelvi igények

|  |  |
| --- | --- |
| Minimum rendszer követelmény: | Apache: 2.4  Mysql: 5.6 (vagyMariaDB: 10.x)  PHP verzió: 5.6  PHPmemory\_limit 256 MB  PHPpost\_max\_size 50 MB  PHPsession.gc\_maxlifetime 1440 s  Szerverinternetkapcsolat: 100 Mbps  Tárhely méret: 250 Mb SSD  pMEM (Fizikai memória) 512 Mb  CPU magok száma 1 |

|  |  |
| --- | --- |
| Ajánlott rendszerkövetelmény: | Apache: 2.4  Mysql: 5.6 verzió,Mariadb: 10.2.21  PHP verzió: 7.3 verzió vagy annál frissebb  PHPmemory\_limit 256 MB  PHPpost\_max\_size 50 MB  PHPsession.gc\_maxlifetime 1440 s  Szerverinternetkapcsolat: 1000 Mbps  Tárhely méret: 250 Mb SSD  pMEM (Fizikai memória) 512 Mb  CPU magok száma 2 |

Mappa és fájlszerkezet bemutatása

* Adatbázis szkriptek // A fájl a működéshez nem szükséges, csupán az sql parancsok másolatát tartalmazza, amivel létre lehet hozni a szükséges adatbázist és a relációs táblákat.
* Dokumentáció // A fájl a működéshez nem szükséges, a program bemutatását és leírását tartalmazza.
* ImageGallery // A kép(ek) feltöltést követően itt tárolódnak.
* Images // Az oldalon felhasznált képeket, ikonokat és logókat tartalmazza
* Includs // Az alkalmazás működéséhez szükséges php szkripteket tartalmazza
  + chechPatterns.php // PHP változók a regisztráció beviteli mezőinek ellenőrzéséhez.

<?php

$checkPattern = '/^([A-Za-z0-9\_\-\.])+\@([A-Za-z0-9\_\-\.])+\.([A-Za-z]{2,4})$/'; //email bevitel reguláris kifejezéséhez.

$checkPatternFName='/^[A-Z-À-Ű-]\d\*[a-z-á-ű-].{1,25}[\S\*]$/'; //vezetéknév reguláris kifejezéséhez

$checkPatternPassword = '/^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*[0-9]).{5,35}$/'; //jelszó reguláris kifejezéséhez

$checkPatternLName = '/^[A-Z-À-Ű-]\d\*[a-z-á-ű-]{1,25}[\S\*]$/'; //keresztnév reguláris kifejezéséhez

$checkPatternZip = '/^(?=.\*[0-9]).{4,5}$/'; //irányítószám reguláris kifejezéséhez

$checkPatternCity = '/^[A-Z-À-Ű-]\d\*[a-z-á-ű-]{1,58}[\S\*]$/'; //város reguláris kifejezéséhez

?>

* + config.inc.ph // A fejlécelemeinek tömbjét, menü elemeinek tömbjét, illetve a kép(ek) feltöltéshez rögzített PHP változókat tartalmazza.

<?php

//////////////////////appearance//////////////////////////

$titleName = array( // oldal címe

'title' => 'Életfa Környezetvédő Szövetség'

);

$header = array( //fejléc elemei

'logo' => 'eletfa-u2.png',

'logiLogo' => 'login.png',

'loginAlt' => 'login',

'logoAlt' => 'eletfa-u2',

'searchLogo' => 'search.png',

'searchAlt'=>'search',

'title' => 'Életfa Környezetvédő Szövetség',

'motto' => ''

);

$footer = array( //lábléc elemei

'copyrights' => 'Copyright '.date("Y"."."),

'company' => 'Életfa Környezetvédő Szövetség'

);

/////////////////////page arrays///////////////////////////

$pgnf = array( //Nemtalálható oldalakhoz szökséges tömbe.

'file' => '404', 'text' => 'A keresett oldal nem található!'

);

$pages = array( // Főmenü oldalainak tömbje

'/' => array('file' => 'introduction', 'text' => 'Bemutatkozás', 'menu' => array(1,1)),

'eticalCodex' => array('file' => 'eticalCodex', 'text' => 'Etikai Kódex', 'menu' => array(1,1)),

'schoolCommunityService' => array('file' => 'schoolCommunityService', 'text' => 'Iskolai Közösségi Szolgálat', 'menu' => array(1,1)),

'publicBenefitReports' => array('file' => 'publicBenefitReports', 'text' => 'Közhasznűsági Beszámolóink', 'menu' => array(1,1)),

'mission' => array('file' => 'mission', 'text' => 'Küldetésünk', 'menu' => array(1,1)),

'strategy' => array('file' => 'strategy', 'text' => 'Stratégia', 'menu' => array(1,1)),

'management' => array('file' => 'management', 'text' => 'Tagjaink, Vezetőség', 'menu' => array(1,1)),

'story' => array('file' => 'story', 'text' => 'Életfa Története', 'menu' => array(1,1)),

'policy' => array('file' => 'policy', 'text' => 'Adatkezelési Szabályzat és Tájékoztató', 'menu' => array(1,1)),

);

$contact = array( //kapcsolat oldalainak tömbje

'con' => array('file' => 'contact', 'text' => 'Lakossági bejelentés', 'menu' => array(1,1)),

'atnblty' => array('file' => 'atnblty', 'text' => 'Elérhetőségek', 'menu' => array(1,1)),

);

$programs = array ( //programok oldalainak tömbje

'EFOP' => array('file' => 'EFOP', 'text' => 'EFOP-5.2.2 - Civil Hálózatok Közép-Európában', 'menu' => array(1,1)),

'CSR' => array('file' => 'CSR', 'text' => 'Társadalmi Szerepvállalás - CSR', 'menu' => array(1,1)),

'=' => array('file' => '=', 'text' => 'Élet=Fa - Közösségi Faültetés', 'menu' => array(1,1)),

);

$imggallery = array ( // képgaléria oldalainak tömbje

'img' => array('file' => 'img', 'text' => 'Képek', 'menu' => array(1,1)),

// 'imgupld' => array('file' => 'imgupld', 'text' => 'Képfeltöltés', 'menu' => array(0,1)),

);

$login = array ( // bejelentkezési oldal tömbje

'lgn' => array('file' => 'lgn', 'text' => 'Bejelentkezés', 'menu' => array(1,1)),

);

$reg = array ( //regisztrációs oldal tömbje

'reg' => array('file' => 'reg', 'text' => 'Regisztráció', 'menu' => array(1,1)),

);

$DataFromDb = array ( // oldal az adatbázisból kinyert adatok mejelenítéséhez

'DataFromDb' => array('file' => 'DataFromDb', 'text' => 'Táblázat', 'menu' => array(0,1)),

);

///image gallery upload/////////////////////////////

$folder = './ImageGallery/'; //Mappa a kép feltöltéshez

$types = array('.jpg','.png'); //képek formai követelménye

$mediaTypes = array('image/jpeg','image/png'); //képek típus követelménye

$datetpl = "Y.m.d. H:i"; // Dátum formai megjelenítéséhez

$imgMaxSize = 500\*1024; //kép maximális méretének meghatározása

?>

* + crypter.php //Regisztráció során a Post metódust helyezi el egy tömbe, amiben a jelzsavakat titkosítja jelenleg sha1 titkosítással.

$psw = array (

'Vezeték név'=>$\_POST['regFirstName'],

'Kereszt név'=>$\_POST['regLasttName'],

'E-mail'=>$\_POST['regEmail'],

'Város'=>$\_POST['regCity'],

'Irányítószám'=>$\_POST['regPostalcode'],

'Cím'=>$\_POST['regAddress'],

'Jelszó'=>sha1($\_POST['regPassword']),

'Jelszó megerősítés'=>sha1($\_POST['regPasswordCh']));

?>

* + lgnpdo.php // A bejelentkezés változóit tartalmazza

<?php

$psw = array (

'Vezeték név'=>$\_POST['regFirstName'],

'Kereszt név'=>$\_POST['regLasttName'],

'E-mail'=>$\_POST['regEmail'],

'Város'=>$\_POST['regCity'],

'Irányítószám'=>$\_POST['regPostalcode'],

'Cím'=>$\_POST['regAddress'],

'Jelszó'=>sha1($\_POST['regPassword']),

'Jelszó megerősítés'=>sha1($\_POST['regPasswordCh']));

?>

* + msgpdo.php

<?php

include './Includes/pdo.ini.php';

try {

/\*”A PDO (PHP Data Object) osztály a PHP-be beépített adatbáziskezelő osztálykönyvtár,

melynek segítségével biztonságos, objektumorientált módon kezelhetjük az adatbázisainkat a PHP-n belül.„\*/

$msgdbcon = new PDO( //PDO adatbázis kapcsolat

"mysql:host=$host;",

$user,

$password);

$msgdbcon->setAttribute(PDO::ATTR\_ERRMODE,PDO::ERRMODE\_EXCEPTION);

$msgdbcon->query('SET NAMES utf8 COLLATE utf8\_hungarian\_ci'); //ékezetes karakterek kezelése

// Üzenetküldés esetén automatikusan létrehozza a szükséges adatbázist amenniyben nem létezik és kapcsolódik az adatbázis kezelőhöz.

$createdb = "CREATE DATABASE IF NOT EXISTS Ganf\_Beadando\_Eletfa;

USE Ganf\_Beadando\_Eletfa;";

$msgdbcon-> exec($createdb);

// Üzenetküldés esetén automatikusan létrehozza a szükséges relációs adattáblázatot amennyiben nem létezik és kapcsolódik az adatbázis kezelőhöz.

$redDborder ="CREATE table IF NOT EXISTS Uzenetek(

ID INT NOT NULL AUTO\_INCREMENT ,

nev VARCHAR(50) NOT NULL ,

email VARCHAR(40) NOT NULL ,

targy VARCHAR(40) NOT NULL ,

uzenet VARCHAR(500) NOT NULL ,

datum VARCHAR(10) NOT NULL,

ido VARCHAR(8) NOT NULL,

PRIMARY KEY (`ID`)

);";

$msgdbcon-> exec($redDborder);

// Amenniyben a POST metódus nem üres értékű, úgy beilleszti az Users Táblázatba az értékeket

if (!empty($\_POST)) {

$redDborder = "INSERT INTO Uzenetek(

nev,

email,

targy,

uzenet,

datum,

ido)

VALUES(?,?,?,?,?,?)"; //Helytartó étékek

}

$time = new DateTime("now", new DateTimeZone('Europe/Budapest')); //A dátum és idő rögzítéséhez előkészített változó, ami az időt Budapesti időzóna szerint rögzíti.

$msgdbcon-> prepare($redDborder)->execute([

$\_POST['name'],

$\_POST['email'],

$\_POST['subject'],

$\_POST['msg'],

date("Y-m-d"),

$time->format('H:i:s')

]);

//echo "Az adatbázishoz kapcsolódás sikeres!". "<br>";

// Az adatbázisban rögzített üzeneteket írja ki táblázat formában rögzítést követően

$msgstmsg = $msgdbcon->query("SELECT \* FROM Uzenetek

ORDER BY ID DESC");

echo

"<table>".

"<tr>

<th>ID</th>

<th>Név</th>

<th>Email cím</th>

<th>Tárgy</th>

<th>Üzenet</th>

<th>Dátum</th>

<th>Idő</th>";

while ($msgRow = $msgstmsg->fetch()) {

echo

"<tr>".

"<td>".

$msgRow ['ID']."</td>".

"<td>".

$msgRow ['nev']." ".

"</td>".

"<td>".

$msgRow ['email'].

"</td>".

"<td>".

$msgRow ['targy'].

"</td>".

"<td>".

$msgRow ['uzenet'].

"</td>".

"<td>".

$msgRow['datum'].

"</td>".

"<td>".

$msgRow['ido'].

"</td>".

"</tr>";

}

echo "

</tr>

</table><br>"

;

} catch (PDOException $e) { //Hibakezelés

echo "Az adatbázishoz nem lehetett csatlakozni: :" . $e->getMessage();

}

?>

* + pdo.ini.php // Az adatbázis kapcsolathoz szükséges változókat tartalmazza (kiszolgáló, adatbázis név, felhasználó, jelszó)
  + regpdo.php

<php

include './Includes/pdo.ini.php'; // Adatbázis kapcsolathoz szükséges változók meghívása

if (isset($\_POST['regFirstName'])

&& isset($\_POST['regLasttName'])

&& isset($\_POST['regEmail'])

&& isset($\_POST['regCity'])

&& isset($\_POST['regPostalcode'])

&& isset($\_POST['regAddress'])

&& isset($\_POST['regPassword'])){

/\*"A PDO (PHP Data Object) osztály a PHP-be beépített adatbáziskezelő osztálykönyvtár,

melynek segítségével biztonságos, objektumorientált módon kezelhetjük az adatbázisainkat a PHP-n belül. "\*/

try {

$dbcon = new PDO( //PDO adatbázis kapcsolat

"mysql:host=$host;",

$user,

$password);

$dbcon->setAttribute(PDO::ATTR\_ERRMODE,PDO::ERRMODE\_EXCEPTION);

$dbcon->query('SET NAMES utf8 COLLATE utf8\_hungarian\_ci'); //ékezetes karakterek kezelése

// Regisztráció esetén automatikusan létrehozza a szükséges adatbázist amenniyben nem létezik és kapcsolódik az adatbázis kezelőhöz.

$createdb = "CREATE DATABASE IF NOT EXISTS Ganf\_Beadando\_Eletfa;

USE Ganf\_Beadando\_Eletfa;";

$dbcon -> exec($createdb);

// Regisztráció esetén automatikusan létrehozza a szükséges relációs adattáblázatot amenniyben nem létezik és kapcsolódik az adatbázis kezelőhöz.

$redDborder ="CREATE TABLE IF NOT EXISTS Ganf\_Beadando\_Eletfa.Users

(ID INT NOT NULL AUTO\_INCREMENT ,

vezetek\_nev VARCHAR(28) NOT NULL ,

kereszt\_nev VARCHAR(27) NOT NULL ,

email VARCHAR(40) NOT NULL ,

varos VARCHAR(60) NOT NULL,

iranyitoszam VARCHAR(5) NOT NULL,

cim VARCHAR(100) NOT NULL,

reg\_jelszo VARCHAR(40) NOT NULL ,

datum VARCHAR(10) NOT NULL,

ido VARCHAR(8) NOT NULL,

PRIMARY KEY (`ID`),

UNIQUE(`reg\_jelszo`,`email`));";

$dbcon-> exec($redDborder);

// Amenniyben a POST metódus nem üres értékű, úgy beilleszti az Users Táblázatba az értékeket

if (!empty($\_POST)) {

$redDborder = "INSERT INTO Users(

vezetek\_nev,

kereszt\_nev,

email,

varos,

iranyitoszam,

cim,

reg\_jelszo,

datum,

ido)

VALUES(?,?,?,?,?,?,?,?,?)";

}

$time = new DateTime("now", new DateTimeZone('Europe/Budapest')); //A dátum és idő rögzítéséhez előkészített változó, ami az időt Budapesti időzóna szerint rögzíti.

// echo "Az adatbázishoz kapcsolódás sikeres!". "<br>";

// A regisztrációs űrlap jelszavának titkosításához az adatbázisba történő mentésnél.

$dbcon-> prepare($redDborder)->execute([

$\_POST['regFirstName'],

$\_POST['regLasttName'],

$\_POST['regEmail'],

$\_POST['regCity'],

$\_POST['regPostalcode'],

$\_POST['regAddress'],

sha1($\_POST['regPassword']),

date("Y-m-d"),

$time->format('H:i:s')

]);

// Az adatbázisban rögzített üzeneteket írja ki táblázat formában rögzítést követően

$regstmsg = $dbcon->query("SELECT \* FROM Users

ORDER BY ID DESC");

echo

"<table>".

"<tr>

<th>ID</th>

<th>Név</th>

<th>Email cím</th>

<th>Cím</th>

<th>Jelszó</th>

<th>Dátum</th>

<th>Idő</th>";

while ($regRow = $regstmsg->fetch()) {

echo

"<tr>".

"<td>".

$regRow['ID']."</td>".

"<td>".

$regRow['vezetek\_nev']." ".

$regRow['kereszt\_nev'].

"</td>".

"<td>".

$regRow['email'].

"</td>".

"<td>".

$regRow['iranyitoszam']." ".

$regRow['varos'].", ".

$regRow['cim'].

"</td>".

"<td>".

$regRow['reg\_jelszo'].

"</td>".

"<td>".

$regRow['datum'].

"</td>".

"<td>".

$regRow['ido'].

"</td>".

"</tr>";

}

echo "

</tr>

</table><br>"

;

} catch (PDOException $e) { //Hibakezelés

echo "Az adatbázishoz nem lehetett csatlakozni: :" . $e->getMessage();

}

}

?>

* Képfeltöltéshez // A kép feltöltéshez tartalmaz feltöltendő képeket 01;02 képek mérete nem megfelelő a feltöltéshez, 03.-as kép feltölthető.
* Scripts // Javascript fájlokat tartalmazó mappa
  + checkForm.js // A beviteli mezők ellenőrzéséhez reguláris kifejezéseit tartalmazza

// Az onload funkció az oldal betöltésekor aktiválódik

var form\_reg = document.getElementById("form\_reg");

if (form\_reg) {

form\_reg.disabled=true;

}

};

window.onload=function(){

var form\_sendmsg = document.getElementById("form\_sendmsg");

if (form\_sendmsg) {

form\_sendmsg.disabled=true;

}

};

oninput=function () { // a funkció bevitel megkezdése esetén aktiválódik

//Globális változók

var ok = true;

var notOk = null;

//Regisztrációs ürlap váltorók

var form\_firstName = document.getElementById("form\_firstName");

var form\_lestName = document.getElementById("form\_lestName");

var form\_regEmail = document.getElementById("form\_regEmail");

var form\_regCity = document.getElementById("form\_regCity");

var form\_regPostalcode = document.getElementById("form\_regPostalcode");

var form\_regAddress = document.getElementById("form\_regAddress");

var form\_regPassword = document.getElementById("form\_regPassword");

var form\_reRegPassword = document.getElementById("form\_reRegPassword");

//Kontakt ürlap változók

var form\_msgtext = document.getElementById("form\_msgtext");

var form\_email = document.getElementById("form\_email");

var form\_sendmsg = document.getElementById("form\_sendmsg");

var form\_subject = document.getElementById("form\_subject");

var form\_name = document.getElementById("form\_name");

//Reguláris kifejezések meghatározása

var checkPattern = /^([A-Za-z0-9\_\-\.])+\@([A-Za-z0-9\_\-\.])+\.([A-Za-z]{2,4})$/;

var checkPatternPassword = /^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*[0-9]).{5,35}$/;

var checkPatternLName = /^[A-Z-À-Ű-]\d\*[a-z-á-ű-]{1,25}[\S\*]$/;

var checkPatternFName = /^[A-Z-À-Ű-]\d\*[a-z-á-ű-].{1,25}[\S\*]$/;

var checkPatternZip = /^(?=.\*[0-9]).{4,5}$/;

var checkPatternCity = /^[A-Z-À-Ű-]\d\*[a-z-á-ű-]{1,58}[\S\*]$/;

var checkPatternAddress = /^([A-Za-z-Á-Űá-ű]{1,20})+([ ])([A-Za-z-Á-Űá-ű \.]{1,20})+([ ])([A-Za-z-Á-Űá-ű-0-9 \.]{1,10})$/;

//Regisztrációs ürlap ellenőrzése

// Ellenőrzi a bevitelt és amenniyben a reguláris kifejezéstől eltér a bevitt adat, úgy a beviteli gombot letiltja és elszinezi a szöveget pirosra.

// Helyes kitöltés esetén a szöveg zöld színű lesz és a beküldés gomb aktiválódik.

if (form\_firstName) {

if (!checkPatternFName.test(form\_firstName.value)) {

ok= false;

form\_firstName.style.color='#f99';

notOk=form\_firstName;

}else{

form\_firstName.style.color ='#9f9'

}

}

if (form\_lestName) {

if (!checkPatternLName.test(form\_lestName.value)) {

ok= false;

form\_lestName.style.color='#f99';

notOk=form\_lestName;

}else{

form\_lestName.style.color ='#9f9'

}

}

if (form\_regEmail) {

if (!checkPattern.test(form\_regEmail.value)) {

ok = false;

form\_regEmail.style.color='#f99';

notOk=form\_regEmail;

}else{

form\_regEmail.style.color ='#9f9'

}

}

if (form\_regCity) {

if (!checkPatternCity.test(form\_regCity.value)) {

ok= false;

form\_regCity.style.color='#f99';

notOk=form\_regCity;

}else{

form\_regCity.style.color ='#9f9'

}

}

if (form\_regPostalcode) {

if (!checkPatternZip.test(form\_regPostalcode.value)) {

ok= false;

form\_regPostalcode.style.color='#f99';

notOk=form\_regPostalcode;

}else{

form\_regPostalcode.style.color ='#9f9'

}

}

if (form\_regAddress) {

if (!checkPatternAddress.test(form\_regAddress.value)) {

ok= false;

form\_regAddress.style.color='#f99';

notOk=form\_regAddress;

}else{

form\_regAddress.style.color ='#9f9'

}

}

if (form\_regPassword) {

if (!checkPatternPassword.test(form\_regPassword.value)) {

ok= false;

form\_regPassword.style.color='#f99';

notOk=form\_regPassword;

}else{

form\_regPassword.style.color ='#9f9'

}

}

if (form\_reRegPassword) {

if (form\_reRegPassword.value != form\_regPassword.value) {

ok= false;

form\_reRegPassword.style.color='#f99';

notOk=form\_reRegPassword;

}else{

form\_reRegPassword.style.color ='#9f9'

}

}

//Kontakt ürlap ellenőrzése

if (form\_name) {

if (!checkPatternFName.test(form\_name.value)) {

ok= false;

form\_name.style.color='#f99';

notOk=form\_name;

}else{

form\_name.style.color ='#9f9'

}

}

if (form\_email) {

if (!checkPattern.test(form\_email.value)) {

ok = false;

form\_email.style.color='#f99';

notOk=form\_email;

}else{

form\_email.style.color ='#9f9'

}

}

if (form\_subject) {

if (form\_subject.value.length==0) {

ok= false;

form\_subject.style.color='#f99';

notOk=form\_subject;

}else{

form\_subject.style.color ='#9f9'

}

}

if (form\_msgtext) {

if (form\_msgtext.value.length==0) {

ok= false;

form\_msgtext.style.color='#f99';

notOk=form\_msgtext;

}else{

form\_msgtext.style.color ='#9f9'

}

}

if (notOk) {

notOk=focus(); //Helytelenül kitöltött mezőre fókuszál

}

if (form\_sendmsg) { //üzenetküldési gomb megjelenítése, amennyiben helyesek az adatok

form\_sendmsg.disabled=!ok;

}

if (form\_reg) { //Regisztrációs gomb megjelenítése, amennyiben helyesek az adatok

form\_reg.disabled=!ok;

}

return ok;

};

* + modal.js // A Regisztráció és bejelentkezés felületek gomb kattintás funkciója.

var modal = document.getElementById('regForm'); //elem azonosító alapján törtőnő meghívás

window.onclick = function(event) { // kattintás funkció, elem elrejtése és megjelenítéséhez szükséges

if (event.target == modal) {

modal.style.display = "none";

}

}

* + videoGalery.js //Bemutatkozó oldalon elhelyezett videó tartalmak váltásához szükséges szkript.

let slideIndex = 1;

showSlides(slideIndex);

function plusSlides(n) {

showSlides(slideIndex += n);

}

function currentSlide(n) {

showSlides(slideIndex = n);

}

function showSlides(n) {

let i;

let slides = document.getElementsByClassName("vidBox");

if (n > slides.length) {slideIndex = 1} //léptetés

if (n < 1) {slideIndex = slides.length} //léptetés

for (i = 0; i < slides.length; i++) {

slides[i].style.display = "none"; // Következő elemek elrejtése

}

slides[slideIndex-1].style.display = "block"; // megjelenítés blokkban

}

* Styles // Az oldal stílus megjelenítését tartalmazza
* Templates // sablon oldalakat tartalmazó mappa
  + Menu
  + main.tpl.php // vízszintes menü sablon
  + Pages // A megjelenítendő oldalak sablonjait tartalmazza
    - 404.tpl.php //Nemtalható oldalak hibaüzenete
    - atnblty.tpl.php // Elérhetőségek és térkép
    - contact.tpl.php // kapcsolat felvétel ürlap
    - DataFromDb.tpl.php //Adatbázi tartalmát megjelenítő oldal. Még fejlesztés alatt
    - eticalCodex.tpl.php // szöveges tartalom
    - img.tpl.php // Képgaléria sablon oldal
    - img.tpl.php // Bemutatkozás, kezdő oldal és videó tartalmak megjelenítése.
    - lgn.tpl.php // Bejelentkezési űrlap
    - management.tpl.php // szöveges tartalom
    - mission.tpl.php // szöveges tartalom
    - policy.tpl.php // szöveges tartalom
    - szöveges tartalom.tpl.php // szöveges tartalom
    - reg.tpl.php // regisztrációs ürlap
    - schoolCommunityService.tpl // szöveges tartalom
    - story.tpl.php // szöveges tartalom, az eredeti oldalra mutató hivatkozásokkal
    - strategy.tpl.php // szöveges tartalom, az eredeti oldalra mutató hivatkozásokkal
    - videoGalery.tpl.php // videók megjelenítése külső szolgáltatótól és mentett tartalomból
  + VisualContent //vizuális tartalmak
    - Video // Tárhelyen tárolt videó
    - map.tpl.php // térkép
  + index.tpl.php // Főoldal sablonja
* Tervezés // A fájl a működéshez nem szükséges, az elkészítendő oldal látványtervét tartalmazza.
* imgupload.tpl.php // A képek feltöltésének űrlapja és php szkriptje

<link rel="stylesheet" href="Styles/styles.css">

<?php

include('./Includes/config.inc.php');

$imgupld = array();

if(isset($\_POST['send'])) {

foreach ($\_FILES as $imgFile) {

if ($imgFile['error']==4);

elseif (!in\_array($imgFile['type'],$mediaTypes))

$imgupld[]="Nem megfelelő típus: ".$imgFile['name'];

elseif ($imgFile['error']==1

or $imgFile['error']==2

or $imgFile['size']>$imgMaxSize)

$imgupld[]="Túl nagy fájl: ".$imgFile['name'];

else {

$upldfldr = $folder.strtolower($imgFile['name']);

if (file\_exists($upldfldr))

$imgupld[]="Már létezik ilyen fájl: ".$imgFile['name'];

else {

move\_uploaded\_file($imgFile['tmp\_name'],$upldfldr);

$imgupld[]="OK: ".$imgFile['name'];

}

}

}

}

if (!empty($imgFile)) {

echo '<ul>'; foreach($imgupld as $u)

echo "<li>$u</li>"; echo '</ul>';

}

?>

<h1>Kép feltöltés a galériába</h1>

<form action="./imgupld.tpl.php" method="POST" enctype="multipart/formdata">

<label>1. kép:

<input type="file" name="#1" required> </label>

<label>2. kép:

<input type="file" name="#2"> </label>

<label>3. kép:

<input type="file" name="#3"> </label>

<input type="submit" name="send">

</form>

* index.php // főoldal

<?php

include('./Includes/config.inc.php');

//különálló választéklista a menühöz, ez a függvény

választja külön a menü elemeit

if ($url = '/') {

$find = $pages['/'];

}elseif ($url = 'con') {

$find = $contact['con'];

}elseif ($url = 'prog') {

$find = $programs['prog'];

}elseif ($url = 'img') {

$find = $imggallery['img'];

}elseif ($url = 'lgn') {

$find = $login['lgn'];

}elseif ($url = 'reg') {

$find = $reg['reg'];

}elseif ($url = 'DataFromDb') {

$find = $DataFromDb['DataFromDb'];

}

//Oldalak kezelése menühöz

if (isset($\_GET['page'])) {

if (isset($pages[$\_GET['page']]) && file\_exists("./Templates/Pages/{$pages[$\_GET['page']]['file']}.tpl.php"))

{$find = $pages[$\_GET['page']];

}elseif (isset($contact[$\_GET['page']]) && file\_exists("./Templates/Pages/{$contact[$\_GET['page']]['file']}.tpl.php"))

{$find = $contact[$\_GET['page']];

}elseif (isset($programs[$\_GET['page']]) && file\_exists("./Templates/Pages/{$programs[$\_GET['page']]['file']}.tpl.php"))

{$find = $programs[$\_GET['page']];

}elseif (isset($imggallery[$\_GET['page']]) && file\_exists("./Templates/Pages/{$imggallery[$\_GET['page']]['file']}.tpl.php"))

{$find = $imggallery[$\_GET['page']];

}elseif (isset($login[$\_GET['page']]) && file\_exists("./Templates/Pages/{$login[$\_GET['page']]['file']}.tpl.php"))

{$find = $login[$\_GET['page']];

}elseif (isset($reg[$\_GET['page']]) && file\_exists("./Templates/Pages/{$reg[$\_GET['page']]['file']}.tpl.php"))

{$find = $reg[$\_GET['page']];

}elseif (isset($DataFromDb[$\_GET['page']]) && file\_exists("./Templates/Pages/{$DataFromDb[$\_GET['page']]['file']}.tpl.php"))

{$find = $DataFromDb[$\_GET['page']];

}else if (isset($extras[$\_GET['page']]) && file\_exists("./Templates/Pages/{$extras[$\_GET['page']]['file']}.tpl.php"))

{$find = $extras[$\_GET['page']];

}

else

{

$find = $pgnf;header("HTTP/1.0 404 Not Found");

}

}

include('./Templates/index.tpl.php');

?>

* msg.php // A beküldött üzenetet jeleníti meg

<?php

include './Includes/chechPatterns.php';

if (empty($\_POST)) {

echo "Nincs üzenet";

}

else {

if (!isset($\_POST['name']) || !preg\_match($checkPatternFName,$\_POST['name'])){

exit("A név hossza nem megfelelő: ".$\_POST['name']);

}

if (!isset($\_POST['email']) || !preg\_match($checkPattern ,$\_POST['email'])){

exit("Hibás e-mail cím: ".$\_POST['email']);

}

if (!isset($\_POST['subject']) || empty($\_POST['subject'])){

exit("Nincs tárgy: ".$\_POST['subject']);

}

if (!isset($\_POST['msg']) || empty($\_POST['msg'])){

exit("Nincs üzenet: ".$\_POST['msg']);

}

echo "Üzenet: ";

echo "<pre>";

var\_dump($\_POST);

echo "</pre>";

}

include './Includes/msgpdo.php';

?>

* README.md // Tájékoztató
* reg.php // A regiszráció adatait jeleníti meg (titkosított jelszóval.)

if (empty($\_POST)) {

echo "Nincs üzenet";

}

else {

if (!isset($\_POST['regFirstName']) || !preg\_match($checkPatternFName,$\_POST['regFirstName'])){

exit("A név hossza nem megfelelő: ".$\_POST['regFirstName']);

}

if (!isset($\_POST['regLasttName']) || !preg\_match($checkPatternLName,$\_POST['regLasttName'])){

exit("A név hossza nem megfelelő: ".$\_POST['regLasttName']);

}

if (!isset($\_POST['regEmail']) || !preg\_match($checkPattern,$\_POST['regEmail'])){

exit("A név hossza nem megfelelő: ".$\_POST['regEmail']);

}

if (!isset($\_POST['regCity']) || !preg\_match($checkPatternCity,$\_POST['regCity'])){

exit("A név hossza nem megfelelő: ".$\_POST['regCity']);

}

if (!isset($\_POST['regPostalcode']) || !preg\_match($checkPatternZip,$\_POST['regPostalcode'])){

exit("A név hossza nem megfelelő: ".$\_POST['regPostalcode']);

}

if (!isset($\_POST['regAddress']) || empty($\_POST['regAddress'])){

exit("A név hossza nem megfelelő: ".$\_POST['regAddress']);

}

if (!isset($\_POST['regPassword']) || !preg\_match($checkPatternPassword,$\_POST['regPassword'])){

exit("A név hossza nem megfelelő: ".$\_POST['regPassword']);

}

echo "Regisztráció: ";

echo "<pre>";

var\_dump ($psw);

echo "</pre>";

}

Tesztelés

Felhasználói leírás

Fejlesztési lehetőségek

Külső állományok bemutatása