VISOKA TEHNIČKA ŠKOLA STRUKOVNIH STUDIJA - SUBOTICA
SZABADKAI MŰSZAKI SZAKFŐISKOLA
SUBOTICA TECH – COLLEGE OF APPLIED SCIENCES

MULTIPLATFORM WEBOLDAL KÉSZÍTÉSE A SZABADKAI VÁROSI MÚZEUM SZÁMÁRA

Hallgató

MALATENSZKI DÁVID

Mentor

DR. ZLATKO ČOVIĆ

Szabadka, 2015

Előszó

A dolgozat témája a szabadkai Városi Múzeum új responsive honlapja, amely a múzeum teljes adatbázisát tartalmazza, illetve ennek a kezelése az admin felület segítségével.

A dolgozatban nagy hangsúly lett fektetve a mai elvárásokra, ezért készült el responsive formában, illetve ami talán még ennél is fontosabb, az az admin felület, ahol az adatbázis teljes körű kezelése lehetséges.

Tartalom

Ábrajegyzék

Forráskód jegyzék

A dolgozat témája

- Bootstrap keretrendszer alkalmazása, amely lehetővé teszi szinte az összes képernyőfelbontásra való design optimalizálást
- Közös munka a Városi Múzeummal
- A következő technológiák alkalmazása: HTML5, CSS3, Javascript, jQuery, PHP, MySQL, Ajax, responsive webdesign
- A platform használjon biztonsági technológiákat
- Fejlett adminisztrációs oldal
- Dinamikus weboldal

1. Bevezető

A weboldal összetett feladatot hivatott ellátni. A hétköznapi felhasználók oldaláról vizsgálva a dolgot, szükséges, hogy a weboldal elnyerje a tetszésüket. Fontos, hogy az eligazodás egyszerű legyen, szép, letisztult design, elegendő tartalom, illetve többnyelvűség.

Másik megközelítésből, ha a rendszergazda szemszögéből vizsgáljuk az oldalt, akkor számára fontos egy teljes körű admin felület, ahol lehetősége van feltöltésre, szerkesztésre, kiállítások és szekciók vezetésére. Egy másik probléma a rossz szándékú látogatók, akik nem a múzeum javát szolgálják. Ellenük kell egy biztonságos belépési felület.

1.1. A probléma leírása

A múzeumnak szüksége van egy adatbázis kezelő felületre, illetve egy weboldalra a látogatók számára:

- Teljes responsive megjelenés, a különböző képernyőfelbontások miatt
- Egy felület, ahol fel lehet venni a kapcsolatot a múzeummal, gyorsan, egyszerűen
- Többnyelvű tartalom elérhetősége
- Kellemes megjelenés
- Biztonságos, többlépcsős regisztrációs és belépési oldal a kezelőknek
- Teljes körű adatbázis feltöltési lehetőségek (kép, video, dokumentum, hanganyag)
- Az adatbázisba felkerült adatok teljes körű szerkesztési lehetősége
- Tárgyak látogatottságát számláló funkció
- Dinamikusan létrehozható szekciók, alszekciók és a hozzájuk tartozó tárgyak vezetése
- Tárgyak könnyű besorolása a kiállításokhoz

1.2. A dolgozat célja

A dolgozat egyik része a látogatók irányába közölt weboldal, ahol gyorsan, egyszerűen tudnak böngészni a múzeumban található kiállítások között, és ezekről bővebb információt kaphatnak leírások, képek, dokumentumok, hanganyagok, és videók formájában.

A másik rész, az adminisztrálás. A múzeum teljes adatbázisa kerül idővel feltöltésre, illetve szükséges a kiállítási szekciók kezelése is, gyors, egyszerű, dinamikus módszerrel.

Egy tárgy feltöltésekor szükséges olyan adatbázis struktúrát használni, ami több ezer sor esetén is gyors válaszidőt kínál. Ennek eredményeként egy olyan adatbázis lett létrehozva, amely megfelelő szinten lett normalizálva, külön választva a kiállítás elemeinek részeit.

Fontos figyelembe venni a kéretlen felhasználókat is. Ezek kiküszöbölésére egy regisztrációs és bejelentkezési rész lett létrehozva. A regisztrációnál szükséges az email visszaigazolása, illetve egy felsőbb, fő admin engedélye is.

1.3. Módszerek

A dolgozat elkészítése és a funkciók használatósága érdekében több technológiát is használtam:

1.3.1. HTML

A Hypertext Markup Language (HTML, hiperszöveges leírónyelv) egy programozási előírás, amely szabályozza, hogy miként kell megírni a weblapokat ahhoz, hogy a számítógépek, illetve egyéb okos készülékek a helyes és várt módon jelenítsék meg az oldalt.

A hiperszöveg egyszerű, formázatlan szöveg, amely a weblap tartalmát hordozza, valamit programozási információk, amelyek a weblap megjelenítéséhez és más weblapok kapcsolásához szükséges.

A leírókód elárulja a szövegről vagy egyéb weblaptartalomról, hogy: először is azonosítja, hogy milyen szerkezeti felépítést igényel a tartalom. Megadja, hogy bizonyos szavak, címsorok, listák vagy bekezdések legyenek.

1.3.2. CSS

A CSS egy rövidítés, az angol Cascading Style Sheets (rangsorolt stíluslapok) kifejezés rövidítése, ami nem más, mint egy másik programozási előírás. A CSS stílus néven ismert szabályok segítségével határozza meg a vizuális megjelenítést.

1.3.3. PHP

A PHP kifejezetten az internetre kifejlesztett, szerveroldali szkriptnyelv. A HTML oldalakba az oldal egyes megnyitásakor lefutó PHP kódot ágyazhatunk. A PHP kód értelmezése a webszerveren történik, ami a látogató által megtekinthető HTML-t hoz létre. Ez a programnyelv egy nyílt forráskódú projekt, ami azt jelenti, hogy bárki hozzáférhet a forráskódhoz, és ingyenesen használhatja, módosíthatja, illetve terjesztheti azt. A PHP kódok nagy része megírható úgy, hogy operációs rendszerektől és webszerverektől függetlenül futtatható legyen.

1.3.4. MySQL

A MySQL egy nagyon gyors, stabil, relációs adatbázis-kezelő rendszer. Az adatbázis lehetővé teszi az adatok hatékony tárolását, keresését, rendezését és kinyerését. A MySQL kiszolgáló az adatokhoz való hozzáférést szabályozva biztosítja, hogy egyidejűleg többen is használhassák az adatokat, gyorsabb hozzáférést kínál hozzájuk, és garantálja, hogy csak a jogosult felhasználók szerezhetnek hozzáférést. Ezért a MySQL több felhasználós, többszálú kiszolgáló. Strukturált lekérdező nyelvet (Structured Query Language - SQL), a szabványos adatbázis-kezelő nyelvet használja.

1.3.5. **JavaScript**

Napjainkban majdnem minden weboldal tartalmaz JavaScript-et, egy olyan programozási nyelvet, amely a látogató böngészőjében fut.A JavaScript, mint neve is mutatja, egy script nyelv. A JavaScript-et HTML lapokba lehet beágyazni, a lappal együtt töltődnek le, majd a böngésző értelmezi és futtatja azokat. Hasonlít a Java-ra, de ugyanakkor sokkal kisebb, egyszerűbb, korlátozottabb működésű is annál.

1.3.6. **jQuery**

A jQuery népszerű JavaScript könyvtár, mely a HTML kód, és a kliensoldali JavaScript közötti kapcsolatot hangsúlyozza.Célja az, hogy amennyire csak lehetséges, leválassza a JavaScript kódot a HTML-ből, és különböző eseményvezérlőkön, és azonosítókon keresztül kommunikáljon a weblap HTML elemeivel.

1.3.7. **Ajax**

Az Ajax betűszó az Asynchronous JavaScript and XML (aszinkron JavaScript és XML) kifejezés rövidítése. Ahhoz, hogy tökéletesen megértsük az Ajax működését és megvalósítását, értenünk kell az egyes összetevőit is. Még ha csak szinkron kérelmeket indítunk is, vagy a JSON-t, esetleg egy másik átviteli módszert használunk, az Ajax alapjainak ismerte csak segíthet a fejlesztésben.

Az Ajax egyre népszerűbb és felkapottabb, ezért könnyű elfelejteni, mit is jelent valójában és mi az, amihez semmi köze. Az Ajax nem más, mint egy hihetetlenül hasznos eszköz arra, hogy közvetlenül JavaScript kódokból kommunikáljunk a kiszolgálóval – nem több ennél, még ha a használata a webalkalmazások fejlesztésében korábban felfedezetlen területekre nyit is ajtót.

1.3.8. **Bootstrap**

Ezt a CSS keretrendszert a Twitternél fejlesztette ki Marc Otto és Jacob Thornton, amit először csak saját használatra szántak. A Bootstrap keretrendszert egy olyan library-nek készítették a Twitternél, ami egy egységes képet ad a közösségi oldaluknak. Több olyan HTML-re és CSS-re épülő előre definiált komponenst tartalmaz, amit gyakran használunk honlapjainkon. Ilyenek például a gombok, űrlapok, menük, tabok, lenyíló menük és még sok-sok hasznos elemet találhatunk a hivatalos honlapon.

Minden eleme jól dokumentált, könnyen értelmezhető és használatával sok időt megspórolhatunk, ha egy új honlapot hozunk létre. Már meglévő rendszerünket, vagy akár csak az adminisztrációs felületet is átírni Bootstrap kompatibilisra, nagyobb kezdeti költségekkel jár, viszont hosszabb távon megéri használni. A mai elvárásoknak megfelelően támogatja a HTML5 újításait, illetve a CSS3 is alapértelmezett a rendszerben. A Bootstrap segítségével természetesen Responsive oldalakat is létrehozhatunk, hiszen támogatja a 12 rácsos megjelenítést.

2. Szakirodalmi áttekintés

A dolgozat legfontosabb részei PHP, CSS és MySQL webtechnológiákra épül. Ezek a technológiák nyílt forráskódú projektek, ami azt jelenti, hogy ingyenesen használhatók.

- A PHP hivatalos weboldala a http://www.php.net, ezen az oldalon minden dokumentáció megtalálható, illetve a teljes nyelvre ad egy kezelési útmutatót.
- A CSS használatának tanulmányozására egy kitűnő oldal a http://www.w3schools.com/css/

  Az előzőhöz hasonlóan itt is minden megtalálható, továbbá a beépített felületnek köszönhetően azonnal ki is tudjuk próbálni a kódsorokat.
- A MySQL hivatalos weboldala a http://www.mysql.com, ahol minden fontos információ megtalálható

Egyéb technológiák, amelyek szintén nyílt forráskódúak:

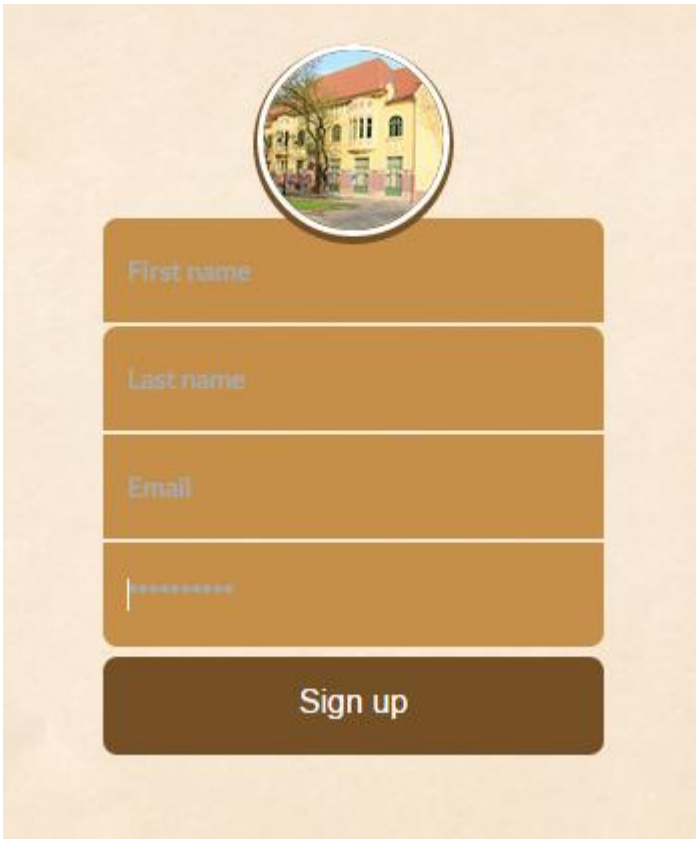
- jQuery : http://jquery.com.
- Bootstrap : http://getbootstrap.com
- Drag and Drop : http://draggabilly.desandro.com/

3. Problémamegoldás

3.1. Adminisztrációs felület

3.1.1. Regisztráció az admin felületre

Az adminisztrációs felületre való belépés előtt regisztrálnunk kell magunk. Itt alap adatokat kell megadni. Keresztnév, vezetéknév, email cím, és jelszó.



1. ábra Regisztráció

A sikeres regisztrációhoz meg kell adnunk minden értéket. Miután kitöltésre került, ellenőrzésre kerül, hogy bevittünk-e minden adatot, illetve hogy az email cím eleget tesz minden elvárásnak, ami azt tükrözi, hogy valós a cím.

Az adatok küldésénél ellenőrzésre kerül az SQL injection használata, ami a legtipikusabb code injection támadás. Ez azt jelenti, hogy a relációs adatbázissal kapcsolatot tartó programnyelvben rakjuk össze szöveges formában az SQL kódot. Ha ezt a lépést kihagynánk, akkor a támadó kárt tudna tenni az adatbázisban, illetve egyszerűen be tudna jelentkezni is.

```
$u_email =
mysqli_real_escape_string($mysqli,htmlspecialchars(trim($_POST['u_email'])));
```

1. forráskód SQL injection kivédése

Regisztráció esetén elsősorban ellenőrzésre kerül, hogy megtalálható e már az email cím az adatbázisban. Ha már szerepel, akkor a hibaüzenettel a felhasználóval tudatjuk ezt a problémát. Továbbá eleget kell tennie az elvárásoknak is, mégpedig, hogy helyes az email cím formátuma.

```
if($_POST['u_email'] == '')
     {
        $_SESSION['error']['u_email'] = "Kötelező megadni az email címet.";
     }
   else
     {
      if(preg_match("/^([a-zA-Z0-9])+([a-zA-Z0-9._-])*@([a-zA-Z0-9_-])+([a-zA-
Z0-9._-]+)+$/", $_POST['u_email']))
        {
         $u_email= $_POST['u_email'];
         $sql1 = "SELECT * FROM user WHERE u_email = '$u_email'";
         $result1 = mysqli_query($mysqli,$sql1) or die(mysqli_error($mysqli));
         if (mysqli_num_rows($result1) > 0)
            {
              $_SESSION['error']['u_email'] = "Az email cím foglalt.";
            }
        }
      else
        {
          $_SESSION['error']['u_email'] = "Az email cím nem megfelelő.";
        }
     }
```

2. forráskód Regisztráció – email ellenőrzés

Fontos még a jelszavak védelme is. Kódoláshoz MD5-ös titkosítást használtam.

```
$u_password   =
md5(mysqli_real_escape_string($mysqli,htmlspecialchars(trim($_POST['u_password
'])))));
```

3. forráskód md5 kódolás

Az adatok helyes megadása után egy aktivációs email üzenetet kap. Erre kattintva egy email érkezik a fő adminnak, és ha ő is jóvá hagyja a regisztrációt, csak akkor tud a felhasználó bejelentkezni az oldalra.

Az email küldés PHP kóddal történik. Miután megkapta a felhasználó az emailt, és kattintott a benne található linkre, egy a lentihez hasonló email kerül kiküldésre, a fő admin számára. Ha ő is kattint, akkor a bejelentkezés lehetséges.

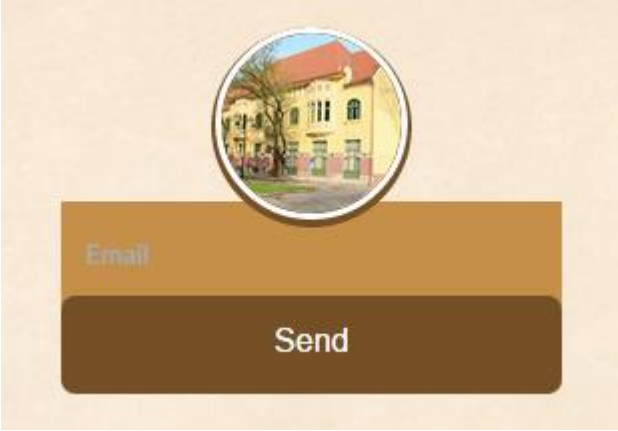
```
     $headers = "MIME-Version: 1.0" . "\r\n";
     $headers .= "Content-type:text/html;charset=UTF-8" . "\r\n";
     $headers .= 'From: '.$admin. "\r\n".
                 'Reply-To: '.$admin."\r\n" .
                 'X-Mailer: PHP/' . phpversion();
     $subject = "Felhasználó visszaigozálsa ".$u_first_name." ".$u_last_name."
részére" . "\r\n";
     $message = "Kérlek kattints a linkre az email címed megeősítéséhez:" .
"\r\n";
     $message .=
"http://www.kmcssz.org/museum/admin/confirm.php?passkey=$com_code" . "\r\n";
     $message .= "További szép napot" . "\r\n";

     $sentmail = mail($u_email,$subject,$message,$headers);
```

4. forráskód Email küldés

3.1.2. Elfelejtett jelszó az admin felületen

Előfordulhat, hogy a regisztrált felhasználók megfeledkeznek a jelszavukról. Ebben az esetben az elfelejtett jelszó funkció alatt tudnak kérni egy emailt, amiben a linkre kattintva meg tudják adni az új jelszavuk.



2. ábra Elfelejtett jelszó

Ilyenkor az adatbázisban ellenőrzésre kerül, hogy létezik e már a megadott email cím, és ha igen, akkor kerül elküldésre az email.

3.1.3. Bejelentkezés az admin felületre

A helyes felhasználónév és jelszó beírása után bejelentkezhetünk és így az oldal összes funkcióját elérhetjük. Bejelentkezéskor szintén sózás és titkosítás hajtódik végre, és ezután ellenőrzi az egyezés feltételét, a sikeres beléptetés érdekében. Ha teljesülnek a feltételek, a főbb adatokat átadásra kerülnek egy session változóba, ami egészen kijelentkezésig élni fog. Segítségével nem kell minden oldalon bejelentkezgetni, a rendszer végig tudja azonosítani a felhasználót.

3. ábra Bejelentkezés

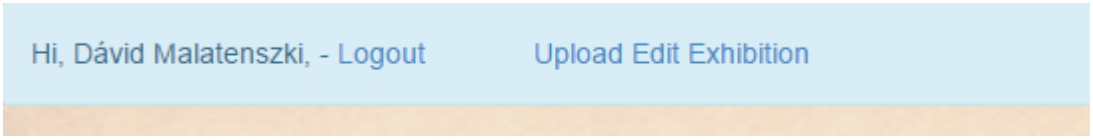
Bejelentkezéskor a $_SESSION[‘user_name’]; megkapja a kereszt és a vezetéknév összefűzött értékét. Ha ez megtalálható, akkor beenged bennünket az oldalra, üdvözöl, illetve megjeleníti a lehetőségeinket. A kijelentkezésre kattintva töröljük a munkafolyamatot, és további munkához ismét be kell jelentkeznünk.

```
session_start();
 if($_SESSION['user_name'] == '')
 {
  header("Location: index.php");
  exit;
 }
 echo "<div class='alert alert-info'><a href=\"#\" class=\"close\" data-
dismiss=\"alert\" aria-label=\"close\">&times;</a>
Hi, ".$_SESSION['user_name']." - <a href=\"logout.php\">Logout</a>  <a
style=\" margin-left: 50px;\"href=\"upload.php\">Upload</a> <a
href=\"edit.php\">Edit</a> <a href=\"exh.php\">Exhibition</a> </div>";
```

5. forráskód Munkamenet ellenőrzés

3.1.4. Admin felület

Bejelentkezés után egy hiperlinkekkel ellátott oldal jelenik meg.



4. ábra Admin menü

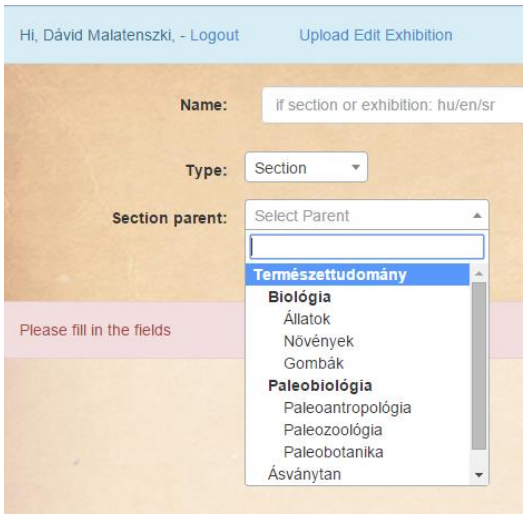
A Logout mint ahogy már említettem, a munkamenetet törli, és kijelentkeztet az oldalról, és átirányít a főoldalra, ahol ismét a bejelentkezés menüpont látható.

```
unset($_SESSION['user_name']);
header('Location: index.php');
```

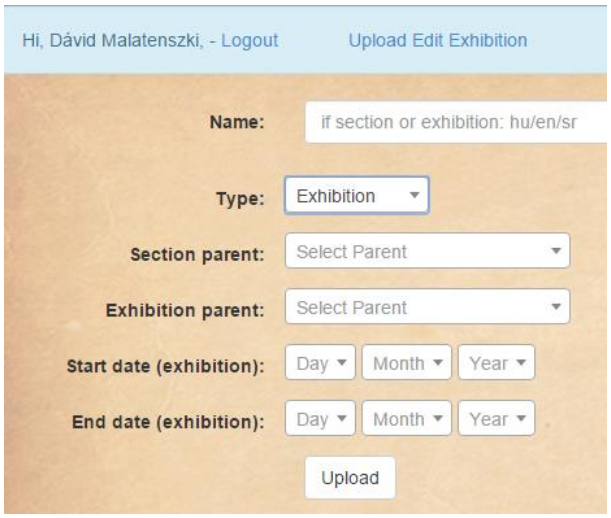
6. forráskód Kijelentkezés

3.1.4.1. Feltöltés: Szekció és kiállítás

Feltöltés során 3 típust tudunk választani. Szekció, Kiállítás és tárgy. A szekció feltöltésénél meg kell adnunk a nevet, 3 különböző nyelven, illetve a Szülő kategóriáját. Ha főkategóriának szeretnénk, akkor azt a részt üresen kell hagynunk. A PHP függvény a szülők fa struktúráját automatikusan generálja le, figyelve a content order táblát, ami leírja, hogy melyik gyereknek melyik a szülője. Kiállítás feltöltésénél is hasonló a folyamat, azzal ellentétben, hogy itt meg kell adnunk a kiállítás élettartamát.



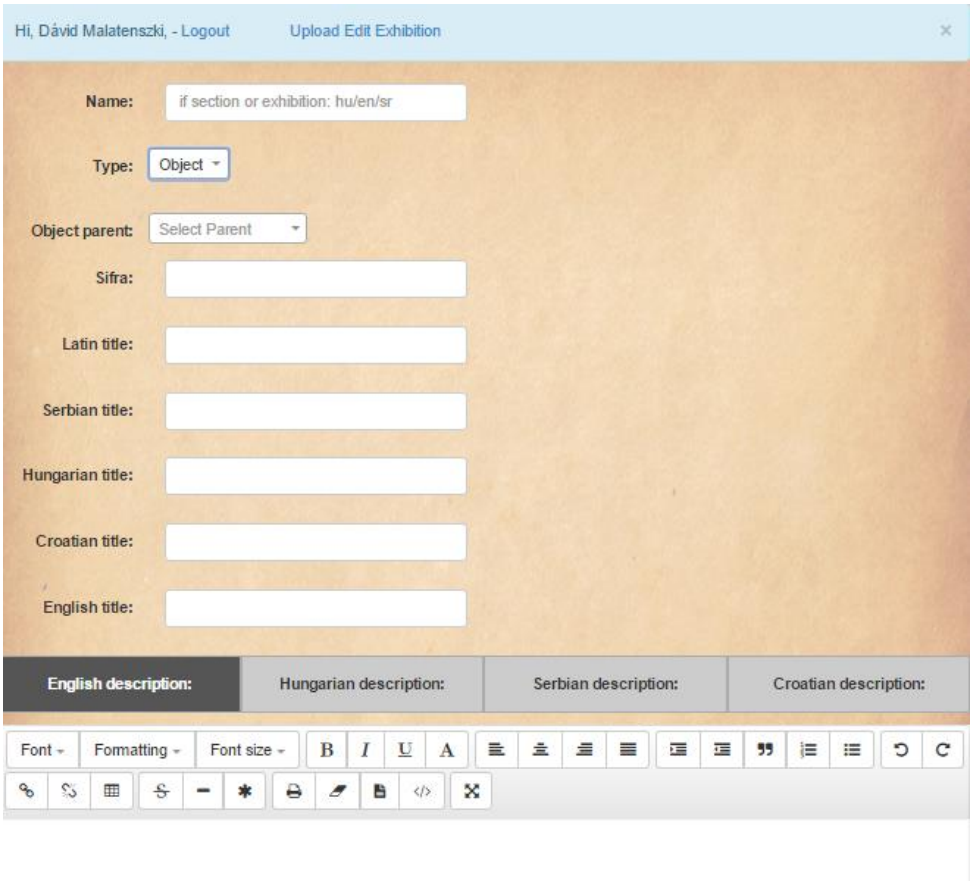
5. ábra Szekció feltöltés



6. ábra Tárgy feltöltés [1]

3.1.4.2. Feltöltés: Tárgy

Tárgyak feltöltésénél az adatok sokkal bőségesebbek. Szükséges megadnunk a tárgy nevét, sifráját, a latin, magyar, szerb, horvát és angol címek és leírások legalább egyikét.



7. ábra Tárgy feltöltés[2]

Az oldalba beépítésre került egy szövegszerkesztő felület, amely hasonlít a Word szövegszerkesztőre. Fontossága abban merül ki, hogy a felvitt szövegeket a későbbiekben előre megformázva szeretnénk megjeleníteni, így ezeket így kell az adatbázisban is eltárolnunk.

8. ábra Tárgy feltöltés[3]

Ezután a multimédia tartalmakat tudjuk megadni. A videó feltöltéséhez a youtube linkeket kell megadnunk. Több videó esetén pontosvesszővel elválasztva.

Képeket, dokumentumokat, és hanganyagokat egy előre definiált típus tartományból kell kiválasztanunk, ami igazából tartalmazza az összes, lehetséges típust.

```
$allowed_exts = array("gif", "jpeg", "jpg", "png", "GIF", "JPEG", "JPG",
"PNG");
$allowed_doc_exts = array("doc", "docx", "pdf", "xls", "xlsx", "ppt",
"pptx","zip", "rar","DOC", "DOCX", "PDF", "XLS", "XLSX", "PPT", "PPTX","ZIP",
"RAR");
$allowed_sound_exts = array("mp3", "wav", "wma","MP3");
```

7. forráskód Kép, Hang, Dokumentum ellenőrzés

Fájlok feltöltéskor a files/img, files/doc, files/sound mappába kerülnek a tárhelyen, típustól megfelelően. Képek esetén minden képről készül egy maximális 200px magas és 300p széles kép is, megtartva az arányokat. Ezekre akkor van szükségünk, ha egy oldalon többet meg szeretnénk egyszerre jeleníteni, de ezzel nem szeretnénk az oldal betöltési idejét lényegesen meghosszabbítani. Több file feltöltése esetén tallózáskor a CTRL billentyűt nyomva ki tudunk több elemet is választani.

Adatbázisba való eltároláskor a teljes útvonal eltárolásra kerül. A file átnevezésbe beletartozik, hogy a név elé bekerül a tárgy sifrája is, ami kiküszöböli, hogy két egyforma nevű file lesz, illetve jobban átlátható az adatbázis is.

```
...
$file_name = toASCII($_FILES['img']['name'][$i]);
$img_path =
"files/img/".$c_sifra."_".mysqli_real_escape_string($mysqli,stripslashes(trim(
$file_name)));
...
```

8. forráskód File átnevezés, ellenőrzés

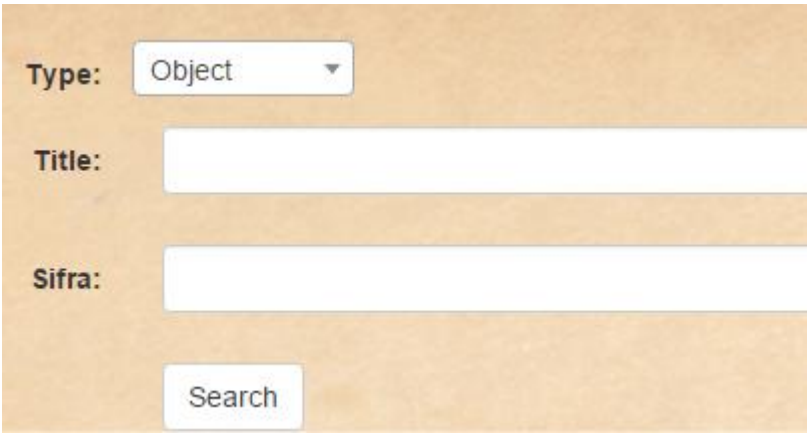
Ékezetes és különleges karaktereket tartalmazó filenevek esetén egy PHP kód ezeket a karaktereket átalakítja a számukra megfelelő párra, és így kerül feltöltésre.

```
function toASCII($str) {
$unwanted_array = array('ö'=>'o','ø'=>'o', 'ù'=>'u', 'ú'=>'u', 'û'=>'u',
'ý'=>'y', 'þ'=>'b', 'ÿ'=>'y' .....  );
return strtr($str, $unwanted_array);
  }
```

9. forráskód Különleges karakterek cseréje

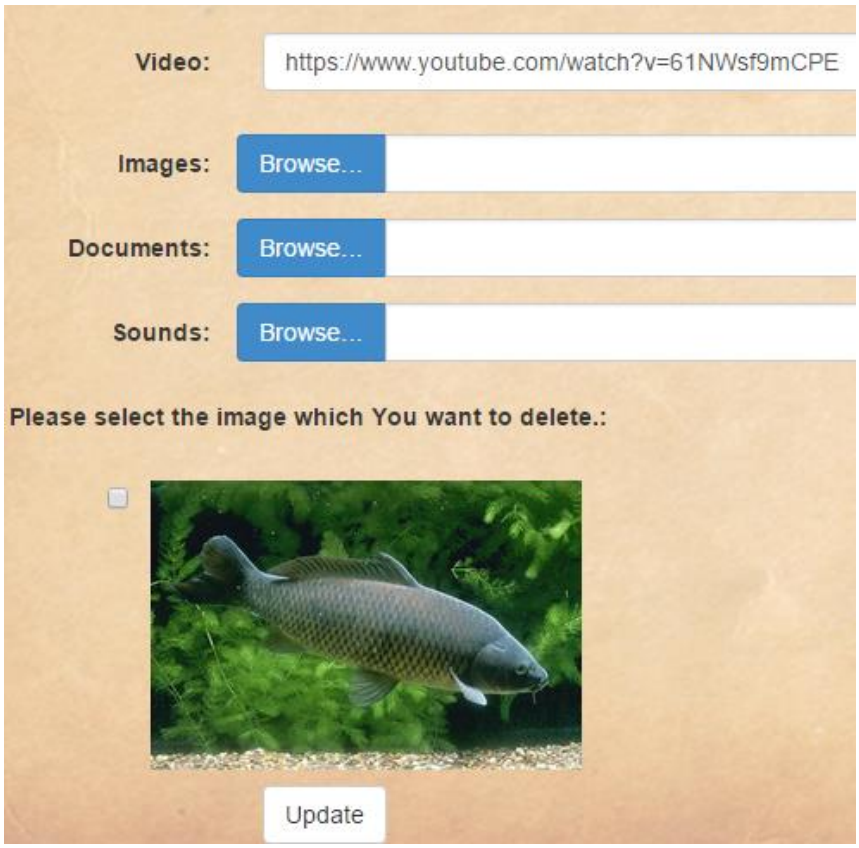
Ha adatellenőrzés során hibát talál a kód, illetve egy kötelező részt nem töltöttünk ki, akkor hibaüzenettel visszairányít az oldalra.

3.1.4.3. Szerkesztés



9. ábra Szerkesztés

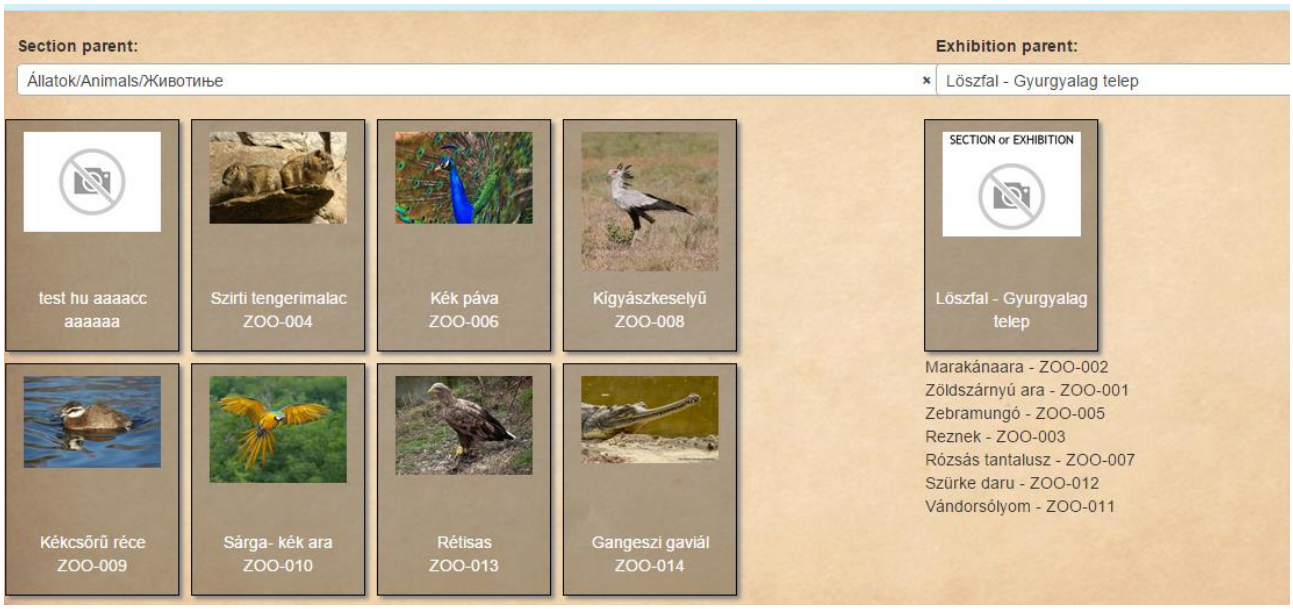
Szerkesztéskor ki kell választanunk, hogy mit szeretnénk szerkeszteni. Ha kiválasztottuk, akkor meg kell adnunk a szerkeszteni kívánt sor sifráját, illetve nevét. Ezután a feltöltéshez hasonló felület jelenik meg, azzal ellentétben, hogy itt már a mezőkben benne lesznek az elemek eddig meglévő adatai. Az oldal alján kilistázódnak a képek, dokumentumok és hanganyagok is. Ezeket kiválasztva véglegesen törölni tudjuk őket az adatbázisból és a webtárhelyről is. Ezek mellett természetesen újakat is tudunk feltölteni.



10. ábra Tárgy szerkesztés

3.1.4.4. Tárgyak hozzárendelése a kiállításokhoz

Tárgyakat kiállításhoz egyszerűen Drag and Drop módszerre tudunk hozzárendelni. A képernyő bal oldalán kilistázódik a kiválasztott szekció összes tárgya. A jobb oldali részben a kiállítások között tudunk navigálni, illetve egy alap információt látunk arról, hogy az épp kiválasztott kiállításban melyik tárgyakat tettük már bele.

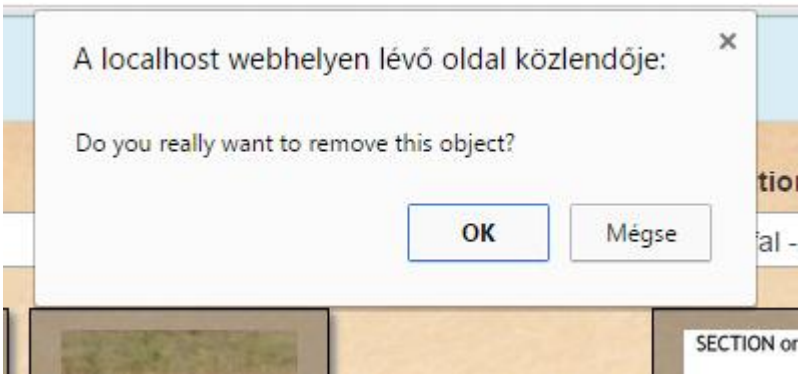


11. ábra Kiállítás hozzárendelés[1]

Amikor a képernyő bal oldaláról egy tárgyat áthúzunk a jobb oldalra, és beletesszük a kiállításba, akkor az a baloldalról eltűnik, a jobb oldalon pedig a kiállítás alatt megjelenik.

12. ábra Kiállítás hozzárendelés[2]

Ezután ha a kiállításhoz hozzárendelt tárgyak közül valamelyiket szeretnénk törölni, hogy át tudjuk tenni másik kiállításba, akkor nincs más dolgunk, mint az adatai mellett jobbra található x-re kattintunk.



13. ábra Kiállítás hozzárendelés törlése

Ilyenkor egy visszaigazolást váró üzenet jelenik meg, hogy valóban szeretnénk-e törölni a tárgy hozzárendelését a kiállításhoz. Igen válasz esetén töröljük, és újra megjelenik a baloldalon.

Ezen az oldalon az adatok Ajax-ot használva jelennek meg, illetve frissülnek. Egyéb technológia használata nem célszerű, mivel az oldal állandóan frissítené önmagát.

```
function getExh() {
        var value2 = $("#exhibition").val();
              if (value2=="") {
                    document.getElementById("exh").innerHTML="";
                    return;
              }
              if (window.XMLHttpRequest) {
                    // code for IE7+, Firefox, Chrome, Opera, Safari
                    xmlhttp=new XMLHttpRequest();
              } else { // code for IE6, IE5
                    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
              }
              xmlhttp.onreadystatechange=function() {
              if (xmlhttp.readyState==4 && xmlhttp.status==200) {

document.getElementById("exh").innerHTML=xmlhttp.responseText;
                initializeDragDrop();
              }
              }
              xmlhttp.open("GET","ajax.php?val="+value2,true);
              xmlhttp.send();
        }
```

10. forráskód Ajax

A Drag and Drop működéséhez Draggabilly-t használtam. Ennél fontos megadnunk a div-et, amit meg tudunk fogni, illetve meg kell adnunk, hogy hova tudjuk letenni őket. Amikor átraktuk a tárgyakat a kiállításokba, akkor a mentés gombra kell kattintanunk, hogy elmentődjön adatbázisba a mozgatás.

3.2. Látogatói weboldal

Az oldalt meglátogatva egy kezdőlap tárul elénk. Itt megtalálható a kiállítás főbb pontjai, illetve rövid leírás az oldalról, felül a navigációs felület, alul pedig a fontosabb információk. Az oldal 3 részre bontható: Header, Content és Footer. A Header és Footer nem változik, a Contet pedig minden oldalon más. A Header tartalmazza a <head> elemeket, illetve a navigációs részt. A Footer az alsó részen található információkat. A Content változik, annak függvényében, hogy jelen pillanatban melyik oldalon tartózkodunk.

14. ábra Látogatói oldal

3.2.1. Iratkozz fel

A Footerben található input mező arra szolgál, hogy a látogatók fel tudjanak iratkozni a múzeum hírlevelére.

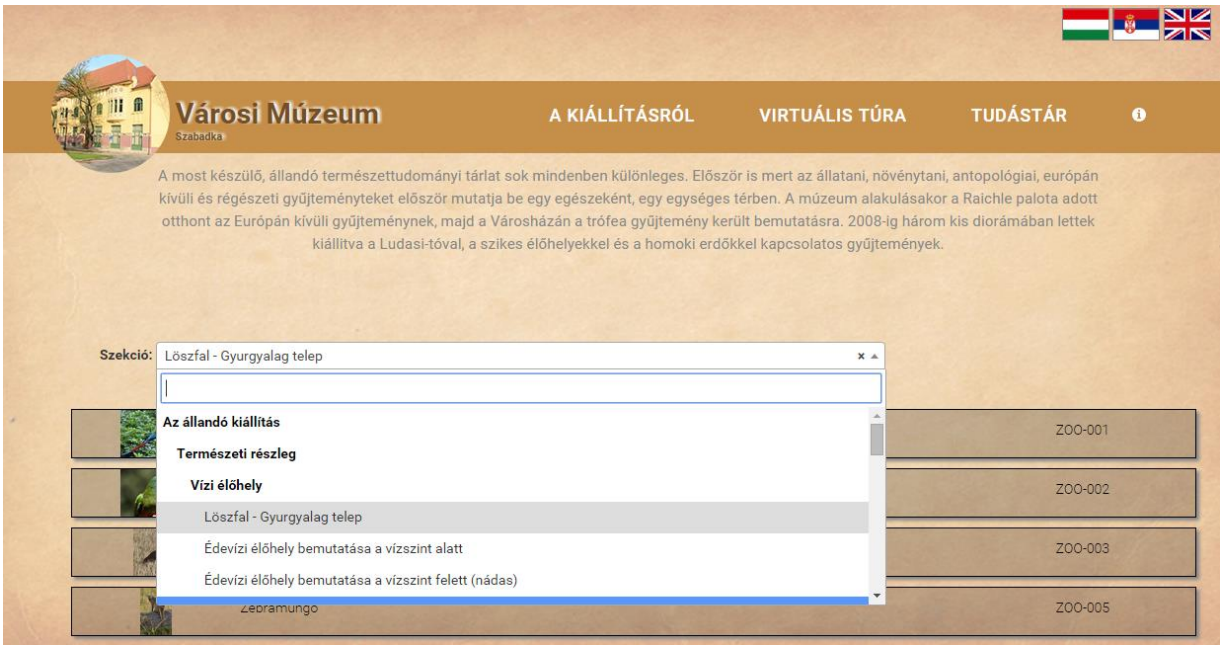
A mező szabályos kitöltése után ellenőrzésre kerül, hogy a bevitt érték megtalálható e már a feliratkozókat tartalmazó adattáblában. Ha nem, akkor bekerül, és később fel tudjuk használni.

```
$sub_email = mysqli_real_escape_string($mysqli,
htmlspecialchars(trim($_POST['sub_email'])));
$sql = "INSERT INTO subscribers (email)
SELECT * FROM (SELECT '$sub_email') AS tmp
WHERE NOT EXISTS (
SELECT email FROM subscribers WHERE email = '$sub_email') LIMIT 1";
```

11. forráskód Felíratkozás ellenőrzése

3.2.2. A kiállításról, Tudástár

A navigációs modulban a Kiállítások menüpontra kattintva egy Select segítségével ki tudjuk listázni az általunk kiválasztott kiállításhoz tartozó tárgyakat.



15. ábra Tudástár

Új kiállítás feltöltése esetén a rendszergazdának nem kell a SELECT inputot frissítenie, mivel ez is adatbázisból olvasódik ki. A kiválasztott kiállításra kattintva meghívódik a függvény, amely Ajax segítségével betölti az oldalon definiált helyre az adatbázisból azokat a tárgyakat, amelynek kiállítási szülője a kiválasztott szülő.

A Tudástár is lényegében ezeket a feladatokat látja el, azzal ellentétben, hogy ott nem a kiállítások, hanem a múzeum szekciói között tud keresni.

A kilistázás után a számunkra szimpatikus tárgyra kattintva egy új oldalra irányít át bennünket a rendszer, mégpedig az Object.php-ra.

3.2.3. **Tárgyak – object.php**

Jóformán a tárgyhoz tartozó minden információt megkapunk, ha a tárgyra kattintunk. A nyelv kezelő kódnak köszönhetően címként mindig a kiválasztott nyelvű cím fog megjelenni, épp úgy, mint a leírásnál is.

A leírás az előzőekben említett editor segítségével előre lett formázva. Abban az esetben, ha például angol nyelven nézzük az oldalt, viszont az adatbázisban nincs meg az angol nyelvű leírás, illetve cím, akkor a rendszer automatikusan az alapértelmezett –magyar- nyelvet jeleníti meg.

16. ábra Tárgyak bővebb információi

Ha a tárgyhoz tartozik hang, dokumentum, videó illetve kép anyag, akkor ezek automatikus megjelennek az oldalon. A dokumentumokat le tudjuk tölteni, videó hang és képanyagot pedig megtekinthetjük, illetve meghallgathatjuk. Ha valamihez nem tartozik multimédia anyag, akkor csak a szöveg részek jelennek meg.

```
<?php
(!empty($files['documents'])){
 $i=1;
 echo "<p>A hozzá tartozó dokumentumok:</p>";
 foreach ($files['documents'] as &$key) {
      echo "<a href='".$key."' target='_blank'>".$t."_document[".$i."]</a><br
/>";
     $i++;
 }
}
?>
```

12. forráskód File kimutatása

A tárgy szöveg alapú adatait egy tömbben tárolom, amelyet a következő SQL kód lefutása után kapok meg:

```
$sql_object = "SELECT c.c_name, c.c_sifra, c.c_counter,
                      t.t_hu, t.t_hr, t.t_en, t.t_latin, t.t_sr,
                      de.de_en, de.de_hu, de.de_sr, de.de_hr,
                      v.v_src
               FROM title t, content c, description de, video v
                WHERE c.c_id = '$c_id' AND
                     t.c_id = '$c_id' AND
                     de.c_id = '$c_id'AND
                     v.c_id = '$c_id'";
```

13. forráskód Adatbázis lekérdezés

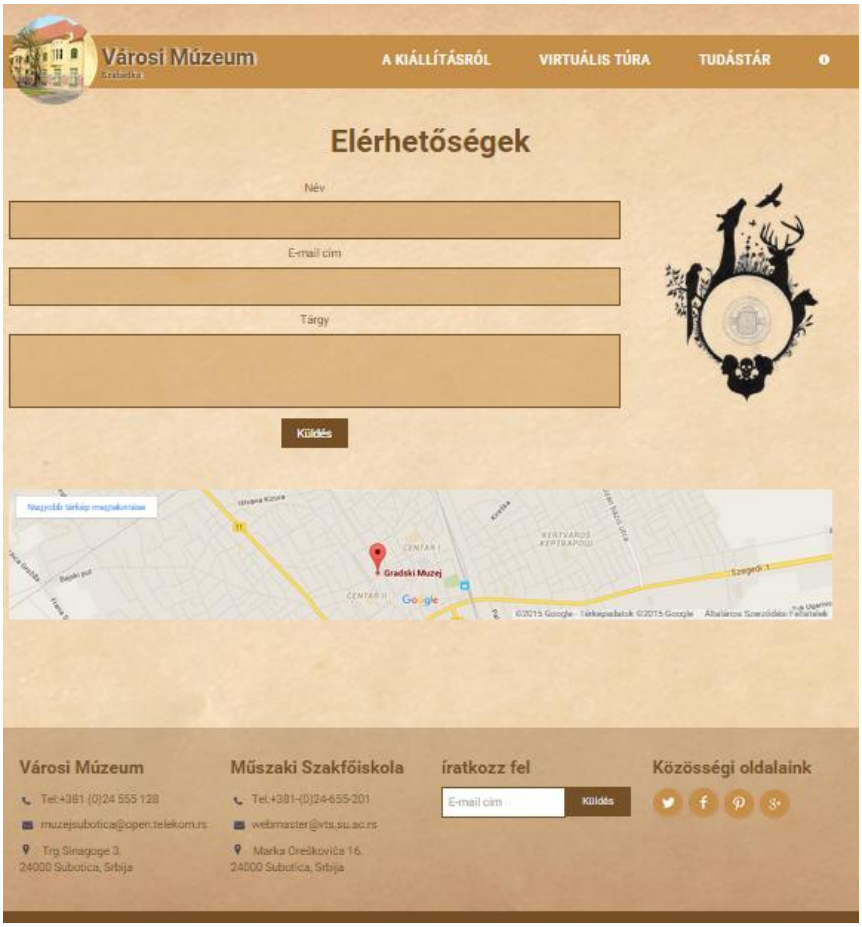
Ha a tárgyhoz tartozik videó anyag is, akkor azt ellenőrzöm, hogy tartalmaz-e pontos vesszőt. Ha igen, akkor tudni kell róla, hogy több linket tartalmaz, vagyis szét kell bontani, és egy tömbben tárolni.

```
$files['videos'] =  $object['v_src'];
if(strpos($object['v_src'], ";"))
  $files['videos']= explode(";", $object['v_src']);
unset($object['v_src']);
```

14. forráskód String szétdarabolás

3.2.4. **Információk**

Az információk fülre kattintva egy email küldéshez hasonló felület jelenik meg. Itt a kért adatokat megadva azonnal tudunk üzenetet küldeni a múzeum számára. Továbbá itt található még itt egy beágyazott Google Maps térkép is.



17. ábra Elérhetőségek

Ha a mezőket nem töltöttük ki, akkor egy hibaüzenetet kapunk, és az email nem lesz kézbesítve.

```
if(!isset($_POST['username']) ||
        !isset($_POST['useremail']) ||
        !isset($_POST['usermsg'])) {
        died('Sajnáljuk, de probléma lépett fel az ürlap kitöltésekor.');
        }
```

15. forráskód Hibaüzenet

Ha kitöltöttük a mezőket, akkor ellenőrzésre kerül a helyességük, és a valódiságuk.

```
$username =
mysqli_real_escape_string($mysqli,htmlspecialchars(trim($_POST['username'])));
$usermsg =
mysqli_real_escape_string($mysqli,htmlspecialchars(trim($_POST['usermsg'])));
$useremail =
mysqli_real_escape_string($mysqli,htmlspecialchars(trim($_POST['useremail'])))
;
```

16. forráskód Email küldés ellenőrzése

Az email helyessége is ellenőrzésre kerül, a következő módon.

```
$email_exp = '/^[A-Za-z0-9._%-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}$/';
```

17. forráskód Email ellenőrzése

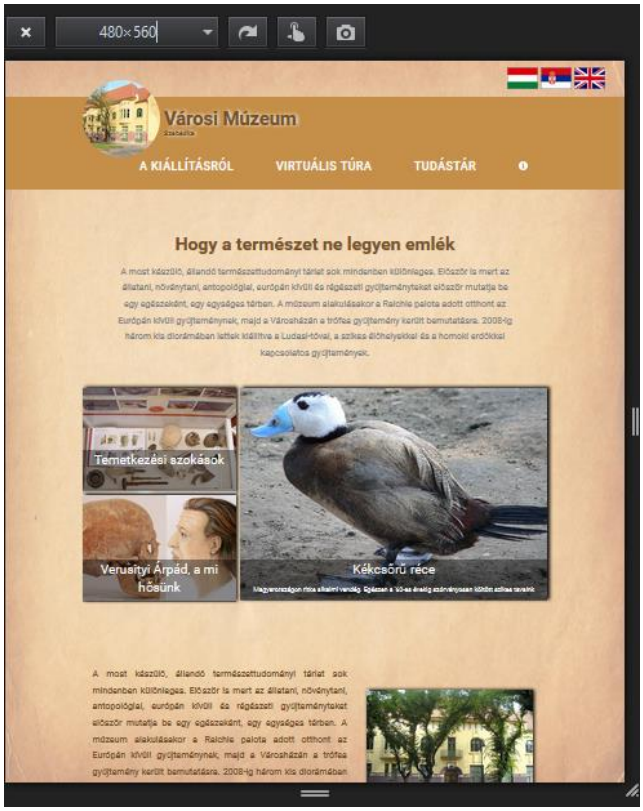
Ezután ha minden rendben van, akkor elküldésre kerül az email a megadott értékekkel, egy előre meghatározott email címre, valamint lefut egy keresés az adatbázisban, az email címre. Ha még nem szerepel a táblában, akkor bekerül, hogy a későbbiekben szükség szerint fel tudjuk használni.

3.3. CSS

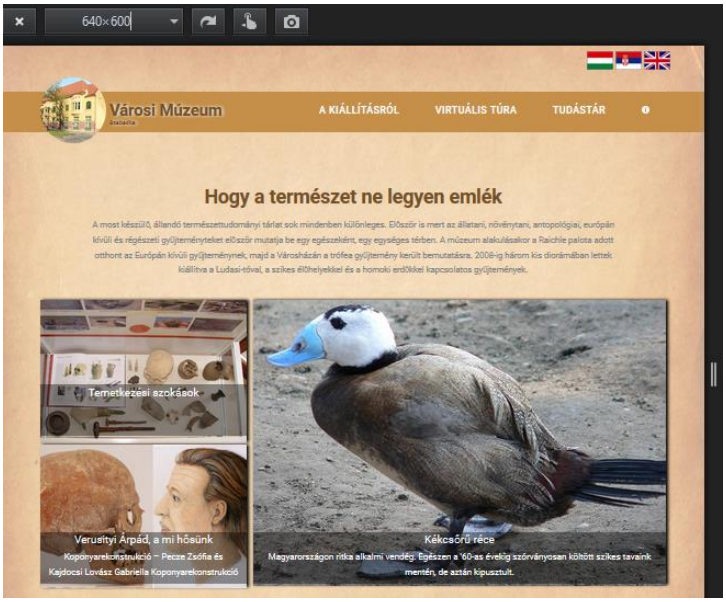
Az oldal teljes mértékben responsive. 1024 pixel alatt összesen 6 mediaqurie-t használ a lehető legpontosabb design érdekében.

A következő néhány kép a különböző képernyőfelbontásokat ábrázolja.

18. ábra 320px szélesség



19. ábra 480px szélesség



20. ábra Kiállítás 640px szélesség

A mediaqurie használata a következő módon történik. Meghatározunk egy maximális szélességet, és ami beletartozik ebbe a határba, arra érvényes lesz az itt megírt CSS kód.

```
@media(max-width:1024px){
  ...
}
@media(max-width:640px){
  ...
}
@media(max-width:320px){
  ...
}
```

18. forráskód Mediaqurie

3.4. Nyelvválasztás

Az oldal nyelvi csomagokat tartalmaz. Egy PHP kód kiválasztja, hogy a feltételek alapján melyik nyelvi csomagot kell, hogy használja.

```
if (isset($_GET['lng']) && !empty($_GET['lng']))
    {
        $nyelv = $_GET['lng'];
        setcookie("Lang", $nyelv);
        $nyelv = $_COOKIE['Lang'];
        $lang = parse_ini_file("lang/".$nyelv.".ini");
        header("Refresh:0; url=".$_SERVER["PHP_SELF"]."");
    }
else
    {
        if(empty($_COOKIE["Lang"]) OR !isset($_COOKIE["Lang"]))
        {
            $nyelv = "hu";
            setcookie("Lang", $nyelv);
            $lang = parse_ini_file("lang/".$nyelv.".ini");
        }
        else
        {
            $nyelv = $_COOKIE["Lang"];
            //setcookie("Lang", $nyelv);
            $lang = parse_ini_file("lang/".$nyelv.".ini");
        }
    }
```

19. forráskód Nyelvi csomag választó

Ha első alkalomkor lépünk be az oldalra, akkor magyar nyelven fog megjelenni. Ezután ha kiválasztunk egy másik nyelvet, akkor az eltárolódik egy sütiben.

Következő alkalomkor mikor betöltésre kerül egy oldal, akkor ismét ellenőrzésre kerül, hogy létezik-e már ez a süti. Ha igen, akkor megnézi, hogy melyik, és azt a nyelvi csomagot tölti be. Ha az URL-be bekerül egy Get-tel küldött adat, ami lng néven fut, és nem üres, akkor a süti tartalma felülíródik, és bekerül az új érték. Ezután már ezt fogja használni. Lekéri a jelenlegi tartózkodási helyünket, levágja a végéről ?lng=en végződést, és frissíti az oldalt, hogy ne legyen ott a címsorban ez az adat.

A nyelvi csomag a következő:

```
Hungary = "Magyar"
English = "Angol"
About_Us = "Rólunk"
Logo1 = "Városi Múzeum"
Logo2 = "Szabadka"
Address1 = "Zsinagóga tér 3.,"
Address2 = "24000 Szabadka, Szerbia"
```

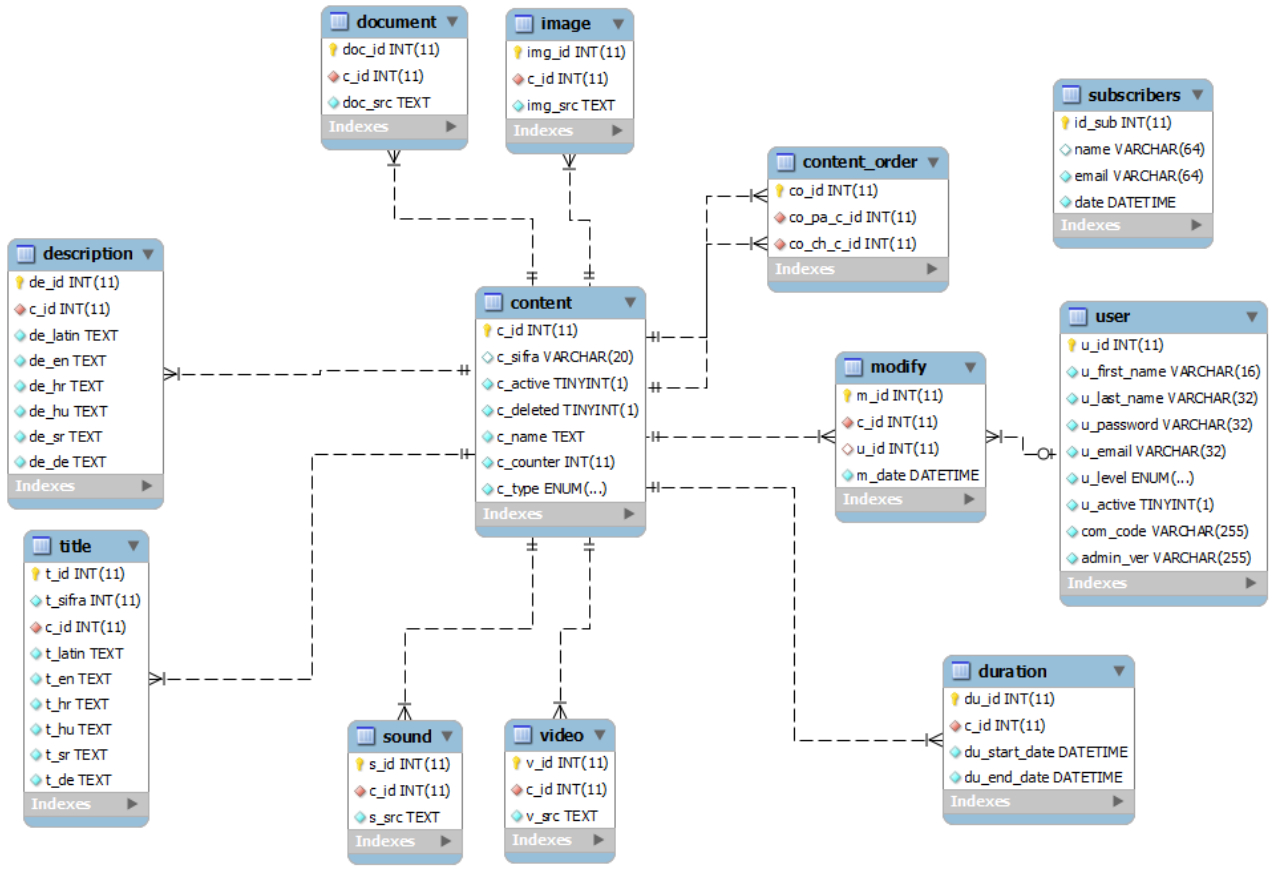
20. forráskód Nyelvi csomag

A PHP kódban a következő részt használva tudjuk text-ként beírni a nyelvi csomag értékét. Az Address1 értéke „Zsinagóga tér 3.,” így a megjelenített szöveg is ez less.

```
<?= $lang['Address1']; ?>
```

21. forráskód Nyelvi csomag használata

3.5. Adatbázis

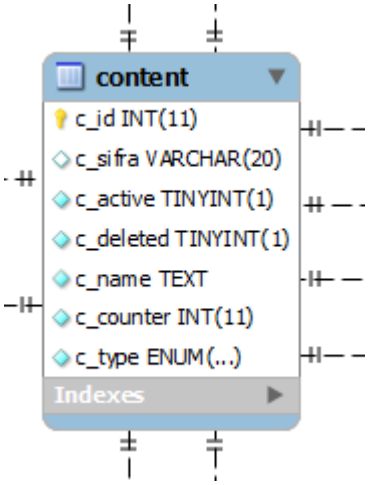


21. ábra Adatbázis táblák

Az adatbázis 12 táblából áll, ebből 11 összefüggő a maradék egy pedig tartalmazza a feliratkozókat.

3.5.1. Content tábla

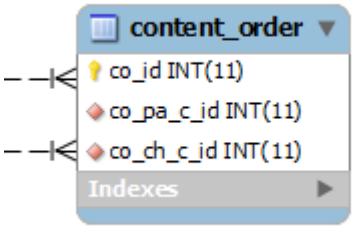
Ez a tábla az elemek csomópontja. Ő adja meg a fő ID-ket, amelyeken keresztül össze vannak kötve a táblák. Emellett ebben van még a sifra, név, számláló, típus, illetve a törlést jelző indikátor is.



22. ábra content tábla

3.5.2. Content Order tábla

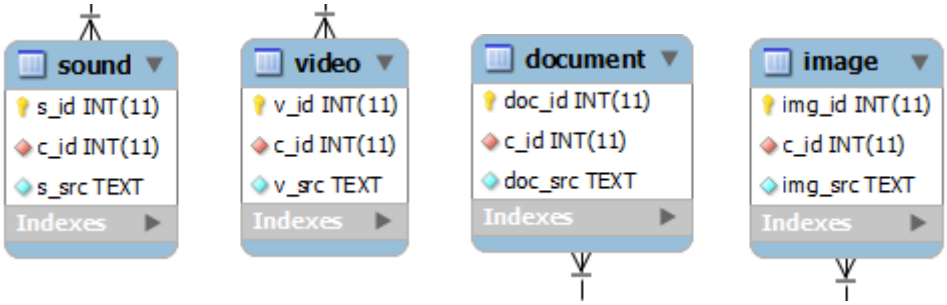
A content order tábla mutatja meg, hogy egy bizonyos elemnek, ki a szülője. Itt lehetséges az az eset, hogy egy szekciónak a szülője is szekció, vagyis akkor ő egy alszekció. Másik eset ha egy tárgyat veszünk figyelembe, akkor itt láthatjuk meg azt, hogy az a bizonyos tárgy melyik szekció eleme, illetve ha be lett sorolva kiállításhoz is, akkor melyik kiállítás eleme.



23. ábra content order tábla

3.5.3. Sound, Video, Document, Image tábla

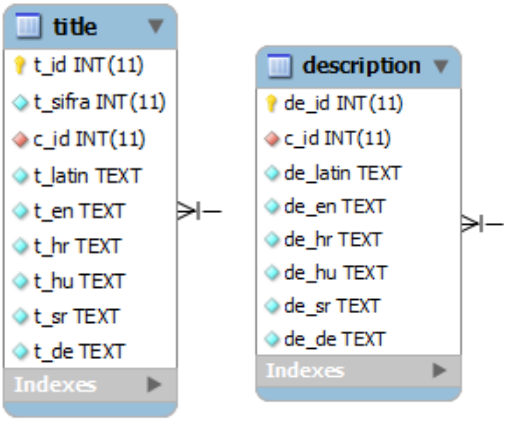
Ezek a táblák tartalmazzák a tárgyakhoz tartozó multimédia anyagokat. A c_id kapcsolja össze a tartalmat, a szülővel. Rendszerint egy saját id-vel, elérési útvonallal, és a c_id-vel rendelkezik.



24. ábra multimedia táblák

3.5.4. Title, Description tábla

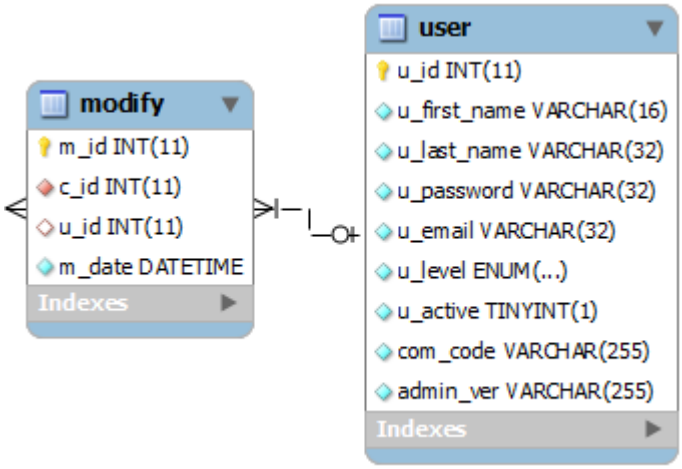
Ezek a táblák tartalmazzák a tárgyak címét és leírását több nyelven. Az előzőekhez hasonlóan itt is a c_id az összekötő elem.



25. ábra Cím és Leírás táblák

3.5.5. Modify, User tábla

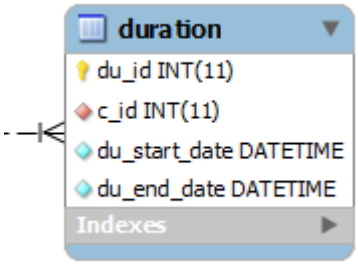
Ezek a táblák tartalmazzák a regisztrált szerkesztők adatait. Felhasználó szinteknél két különböző szintet tudunk megkülönböztetni. Van a fő admin, aki engedélyezni tudja a regisztrációt és szekciót, kiállítást tud létrehozni, illetve az egyszerű szerkesztő, aki tud létrehozni tárgyakat.



26. ábra Felhasználói táblák

3.5.6. Duration

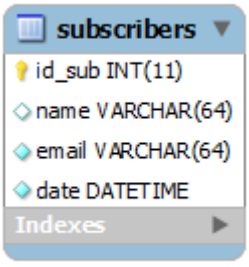
Az exhibition, vagyis kiállítás típusú elemeknek a futási idejét határozza meg, vagyis hogy egy adott kiállítás mettől meddig tekinthető meg.



27. ábra Élettartam tábla

3.5.7. Subscribers

Ez a tábla tartalmazza a feliratkozók adatait, illetve akik emailt küldtek a weboldal segítségével.



28. ábra Feliratkozók tábla

4. Eredmények

A probléma sikeresen megoldódott, amihez hozzátartozik a múzeumi kollégák munkája is, mivel az adatbázishoz szükséges hang, kép és dokumentum anyagokat ők biztosították.

A többféle programnyelv használata lehetővé tette, hogy egy stabil, gyors, szép weboldal készüljön el.

A munka elvégzése során akadtak kisebb problémák, de mivel a programozásban semmi sem lehetetlen, csak tudni kell lekódolni, így ezek megoldódtak.

4.1. Következtetések

A dolgozat tervezésében nagyban hozzájárultak a múzeum munkatársai, akikkel közösen, többszöri egyeztetés után állt össze a végleges weboldal.

5. Összefoglaló

Izrada multiplatformskog web sajta za potrebe Gradskog Muzeja u Subotici

Kreirana web stranica je multiplatformska. Sa aspekta korisnika važna je jednostavna navigacija, čist i dopadljiv dizajn, odgovarajući sadržaj i višejezičnost.

Stranica poseduje i administrativni deo koji administratorima daje mogućnost potpune kontrole nad sadržajem web sajta. Web sajt je realizovan za potrebe Gradskog muzeja u Subotici.

Multiplatform weboldal készítése a Szabadkai Városi Múzeum számára

A létrehozott weboldal multiplatform. A felhasználó szempontjából fontos az egyszerű navigáció, a tiszta és vonzó design, egyszerű navigáció, megfelelő tartalom és a többnyelvűség.

Az oldal rendelkezik adminisztrációs résszel is, amivel a rendszergazda teljes ellenőrzés alatt tarthatja a honlap tartalmát. A honlap a Szabadkái Városi Múzeum igényei alapján készült el.

Die Erstellung einer Multiplattform-Webseite für das Stadtmuseum in Subotica

Die erstellte Webseite ist Multiplattform. Aus der Sicht des Benutzers ,die einfache Navigation, sauberes und attraktives Design , geeigneter Inhalte und Mehrsprachigkeit ist sehr wichtig.

Die Seite hat auch einen administrativen Teil, mit dem Administratoren die volle Kontrolle über den Inhalt der Website haben. Die Website wurde für die Bedürfnisse des Stadtmuseums in Subotica realisiert.

Creation of multi-platform web site for the City Museum in Subotica

The created website is Multiplatform. From the user's point of view it is important, the easy navigation, clean and attractive design, appropriate content and multilingualism.

The site also has administrative part, the administrators have full control over the content of the website. The website was created for the need of the City Museum in Subotica..

6. Irodalom

1. Lynn Beighley, Michael Morrison (2009). Agyhullám: PHP és MySQL
2. Josua Eichorn (2008). Az AJAX alapjai
3. Cody Lindley (2010). jQuery receptek

7. Önéletrajz

Keresztnév: Dávid

Vezetéknév: Malatenszki

Született: 1993.02.02

Születési hely: Zenta

Lakcím: 24410 Horgos, Branka Radičevića 2, Szerbia

Tel: +38162/1697297

email: malatenszki.david@gmail.com

Tanulmányok:

2000 – 2008 – Október 10 Általános Iskola, Horgos

2008 – 2012 – Szárity János Műszaki Iskola, Szabadka (Számítógép elektrotechnika)

2012 – 2015 – Szabadkai Műszaki Szakfőiskola, Szabadka (Internet és elektronikus ügyvitel)

2015 – Óbudai Egyetem (Informatika mérnöktanár mesterképzés)

8. Megjegyzések