Debreceni Egyetem

Informatika Kar

Webes alkalmazásfejlesztés szabadon választott témában

Témavezető:
Dr. Kuki Attila
Egyetemi adjunktus

Készítette:
Kerecsen Gergő
Gazdaságinformatikus

Debrecen
2009

II. Bevezetés

A dolgozatom témája egy általam fejlesztett webes rendszer. Segítségével adataink állandóan a rendelkezésünkre állhatnak. Az ötlet magja abban keresendő, hogy szükség lehet személyes adatainkra akkor is, ha éppen nem a saját számítógépünknél ülünk. Manapság az internetet rengeteg dologra felhasználhatjuk. Akár mint egy külső tárhelyet használhatjuk, és a rajta lévő adatokkal szinte teljes szabadsággal bánhatunk. Megnőtt az olyan oldalak száma, amelyek a felhasználók számára ingyenes tárhelyet biztosítanak. Az általam fejlesztett oldal is ezek közé tartozik, azzal a kivétellel, hogy a tárhely mellé egy olyan környezetett is kap a felhasználó, ami segítségével az adatait könnyebben kezelni tudja, illetve segítséget nyújt a feladatainak elvégzésében. A dinamikus weboldalak térnyerése az interneten már nem tekinthető elenyészőnek. Egyre nagyobb kereslet van az olyan oldalak irányába, amik képesek interaktív módon, dinamikusan reagálni a felhasználó parancsaira. Az egyszerű HTML nyelven íródott weblapok, már nem képesek lépést tartani a modern technológiákkal. Ezek ugyanis már adatbázis alapon működnek és egyszerű, gyors szerkeszthető tárolást és elérhetőséget biztosítanak. Az oldal legnagyobb előnye, hogy adataink, amik lehetnek képek, zenék, könyvjelzők, vagy szöveges jegyzetek, bárhonnan elérhetőek, elég egy internet kapcsolat és egy böngésző. Emellett az oldal jelszóval védve van, így más, nem férhet hozzá a saját adatainkhoz. Az oldal működéséhez szerver oldalon szükség van egy web szerverre, ami PHP kódokat képes futtatni, illetve egy MYSQL adatbázisra. Az oldal funkcióiért a PHP kód felelős, ami legenerálja az oldal HTML kódját, illetve a szerver-kliens közötti adatforgalmat irányítja. A MYSQL adatbázisban tároljuk a felhasználókat és a jelszavukat. A dolgozatomban továbbiakban a kód és a szolgáltatások alapos ismertetésével foglalkozom,illetve az oldal elkészítéséhez elengedhetetlenül szükséges nyelvek, függvények és definíciók leírásával.

III. Oldal keretrendszere

Ez a rendszer végzi a felhasználó azonosítását, illetve hogy illetéktelen behatolók ne férhessenek hozzá az oldalhoz. A rendszer sütiket használ ahhoz, hogy a munka menet teljes egészében a felhasználó azonosítva legyen. Két modul tartozik a rendszerhez az index.php és a login.php.

III.1. Index.php

Az index.php kódban indul el a munkamenet, más néven session. Ez a globális tömb fogja tartalmazni a felhasználó adatait. Ezen modul fogja irányítani azt, mikor milyen modul fusson le. Vizsgálja a bejelentkezés állapotát vagy, hogy létező modult próbálunk-e meghívni. Az eredménytől függően különböző oldalakra irányít minket.
Kódmagyarázat: Kódot azzal kezdjünk, hogy elindítunk egy munkamenetet a session_start() függvénnyel. Ezután egy globális változó $IN_PHP értékét igazra állítjuk. Ezt az értéket minden modulban vizsgáljuk. Célja, hogy csak a keretrendszeren belül tudjunk php kódot futtatni. Így növelve az oldal biztonságát. Első feltételnél

```
if($_SESSION['delcookie'] === true && $_SESSION['remember']!==true)
 {
        setcookie('admin','',0,"",$settings['hostname']);
        setcookie('pass','',0,"",$settings['hostname']);
        $_COOKIE['admin']='';
        $_SESSION['delcookie']=false;
 }
```

azt vizsgáljuk, hogy kell-e törölnünk a sütiket. Ez a kijelentkezésnél fontos. Ha törölnünk kell a sütiket a munkamenet szerint és nincs bepipálva a jelszó és a felhasználó név megjegyzése, akkor, a sütiket kitöröljük és ez felér egy kijelentkezéssel.
A következő feltételnél

```
if($_SESSION['remember']===true)
  {
setcookie('admin',$_SESSION['remember_u'],time()+mktime(0,0,0,0,7,0),"",$settings['hostname']);
setcookie('pass',$_SESSION['remember_p'],time()+mktime(0,0,0,0,7,0),"",$settings['hostname']);
          $_SESSION['remember']=false;
  }
```

azt vizsgáljuk, hogy a felhasználó megszeretné-e jegyeztetni a felhasználó nevét és a jelszavát rendszerrel. Ha igen, akkor a rendszer sütiket úgy állítja be, hogy azok tovább megmaradjanak meg mint egyéb esetben.

A következő feltételben

```
if($_COOKIE['admin'] != "" && $_SESSION['admin']=="")
  {
   $_POST['admin']=$_COOKIE['admin'];
        $_POST['password']=$_COOKIE['pass'];
        $_GET['action']='login';
}
```

azt vizsgáljuk, hogy a felhasználó név és a jelszó mezőből kapott-e adatot a login.php-tól, s ha igen, akkor beállítja a sütiket ezekre az adatokra.

A kód második felét azzal kezdjük, hogy meghívjuk egyszer a _mysql.php-t és a _settings.php-t. A „_mysql.php” teremt kapcsolatot az sql szerverrel. A _settings.php-ba pedig az oldal hostja található. A következő rész a bejelentkezés állapotát vizsgálja és attól függően, hogy bevagyunk-e jelentkezve vagy sem, különböző oldalakra irányít minket.

Első feltétel, ha a munkamenetbe felhasználó nevének értéke nulla, akkor átirányít minket a login.php ba.

```
if($_SESSION['admin']==""){include('login.php');}
```

Ha modul változó az urlbe üres, akkor automatikusan a main.php-ba irányít minket a kód.

```
if($_GET['module']==""){include('main.php');}
```

Ha nem üres a modul értéke és létezik az adott modul, akkor hozzáfűz még egy „php” stringet és meghívja azt.

```
if(file_exists($_GET['module'].".php"))
        {include($_GET['module'].".php");}
```

Végül, ha nem létezik a modul, akkor a „Nem létező modul” feliratot fogjuk látni az oldalon.

```
echo("Nem létező modul");
```

Az index.php lefutásával megtörtént a munkamenet elíndítása és a aktuális felhasználót a számára megfelelő oldalra irányítottuk.

III.2. Login.php

Ez felelős a felhasználó bejelentkeztetéséért. Ellenőrzi, hogy létezik-e a felhasználó által megadott felhasználói név és helyes-e a hozzá megadott jelszó.

Kódmagyarázat: Első mozzanat, hogy ellenőrizzük, hogy nem kaptuk-e meg utasításként a kijelentkezést. (Urlben az action értéke logout):

```
if($_GET['action']=="logout")
 {
  $_SESSION['delcookie'] = true;
  $_SESSION['admin']='';
  die('<meta http-equiv="refresh" content="0;URL='.$settings['hostname'].'/admin">');
}
```

Ha megkaptuk, akkor a munkamenetnek jelezzük, hogy törölni kell a sütiket, a $_SESSION['delcookie'] értékét igazra állítjuk. A felhasználó nevet kitöröljük és az oldalt frissítjük. Ezután a kijelentkezés folyamata az index.php kódba folytatódik.

A kód következő része akkor fog lefutni, ha beírtunk valamit a login oldalon található felhasználónévhez tartozó beviteli mezőbe.

Először lekérjünk sql adatbázisból a beírt felhasználói névhez tartozó jelszót, ami MD5ben van titkosítva.

```
$q = mysql_query("SELECT pass FROM admins WHERE name='".mysql_real_escape_string($_POST['admin'])."'",$sql) or die('Query hiba: '.mysql_error());
  $row = mysql_fetch_array($q,MYSQL_ASSOC);
```

Tehát ha a jelszó mező nem üres és megegyezik a beírt jelszó md5 tel titkosított kódja a sqlből lekért kóddal, akkor a bejelentkezés megtörténik, a munkamenet felhasználói értéke a bejelentkezett felhasználó nevére módosul.

```
if( ($_POST['password'] != "") && ($row['pass']==md5($_POST['password'])))
   {
        $_SESSION['admin']=$_POST['admin'];
 }
```

Ha a „automatikus belépés” mező be volt pipálva akkor, a jelszót és a felhasználó nevet eltároljuk a munkamenetben, ami utólag majd a sütiben is el lesz tárolva. Így lehetséges az automatikus belépés, nincs szükség újra beírni a felhasználói nevet és a jelszót.

```
if($_POST['remember'] == 'on')
          {
           $_SESSION['remember']=true;
           $_SESSION['remember_u']=$_POST['admin'];
           $_SESSION['remember_p']=md5($_POST['password']);
          }
```

Abban az esetben, ha a beírt jelszó nem megfelelő vagy nem létezik ilyen felhasználó, akkor még ellenőrizzük, hogy a munkamenetbe van-e bejelentkezett felhasználó ha igen akkor kitöröljük a sütijét. Végezetül hibaüzenetet adunk.

```
if($_COOKIE['admin'] != "")
          {
           $_SESSION['delcookie'] = true;
          } else  echo('Hibás felhasználónév és/vagy jelszó!');
```

IV. Oldal szolgáltatásai

IV.1 Könyvjelzők kezelése:

Az oldal internetes linkek „könyvjelzők” tárolását biztosítja. A könyvjelzők azt a célt szolgálják, hogy a böngészés során fontosnak tartott, vagy kedvenc oldalainkat egy kattintással elérhetővé tegyük. A hosszú Url címeket nem kell megjegyeznünk, hanem elmenthetjük az oldal adatbázisába. Az oldalon található menüsorban láthatóak a „könyvjelzők” menüpont. A legördülő menüpontban található még a könyvjelzők feltöltése almenüpont és a könyvjelzők törlése almenüpont. A könyvjelzők oldalon egy listát láthatunk saját könyvjelzőinkkel. A linkekre kattintva az adott oldal új lapon jelenik meg.
A könyvjelzők feltöltése oldalon egy kitölthető beviteli mezőt láthatunk, illetve egy fájl feltöltésére alkalmas mezőt. Az első beviteli mező arra szolgál, hogy a menteni kívánt cím urljét ide írjuk be. Ez a mező egyszerre csak 1 db címet tud értelmezni. A másik beviteli mezőve, a firefox böngészőnk által létrehozott könyvjelző mentést tölthetjük fel a szerverre, ami ezután értelmezi és a benne található urleket a könyvjelzőnk listájába importálja. Három modul kapcsolódik a szolgáltatáshoz a bookmarks.php, upbookmarks.php és a konyvjelzotorles.php. A kódok azzal kezdődnek általánosan, hogy meghívják a head.php-ét és a body1.php ét a kódba ami a fejléc és a oldal testének első részét tartalmazza, majd a kód végén body2.php meghívva lezárja az oldalt.

1.a bookmarks.php

A kód feladata az aktuális felhasználó könyvjelzőinek ki listázása az oldalra.
Működése: A könyvjelzők listája egy txt fájlban van letárolva szerver oldalon. A kód ezt a txt fájlt éri el ftp kapcsolat segítségével. Amiből kiolvassa a benne található adatokat majd html tagek között kiírja. (2.ábra)

Kód magyarázat: A kód azonosítja a felhasználót a $_SESSION tömb segítségével. Ami a felhasználóról tartalmaz adatokat.
$felhasznalo=$_SESSION['admin'];

Erre azért van szükség, mivel minden felhasználónak külön mappája van a szerveren, így lehetséges, hogy a felhasználó a saját mappájában lévő könyvjelző fájlokhoz hozzáférjen. A azonosítás után változókba kerül a Ftp hozzáféréshez szükséges adatok és a cél fájl elérési útvonala. A kód következő részében a már meglévő adatokkal csatlakozunk ftp keresztül szerverhez. Ellenőrizzük, hogy létezik-e az cél fájl, ha létezik, megnyitjuk és kiolvassuk belőle a sorokat egy tömbbe.

```
if (file_exists ("ftp://".$ftp_user_name.":".$ftp_user_pass."@".$ftp_server.$source)){
$f=fopen("ftp://".$ftp_user_name.":".$ftp_user_pass."@".$ftp_server.$source,"r");
while(!feof($f))
  {
        $jelzok[$i]=fgets($f);
        $i++;
       }
}
```

Miután, a olvasás elér a fájl végére, kilép a ciklusból. Ekkor, tömbünk tartalmazza a bookmarks.txt összes sorát. Amit a kód, következő részében Html tagek közöt ki echózzuk a tömbünket bejárva.

```
foreach ($jelzok as $i) {
                echo '<tr>';
                echo '<td>';
     echo "<a href=".$i." target=_blank >".$i."</a>";
                echo '</td></tr>';
}
```

Így az oldalon linként lesz látható és a linkre kattintva egy új oldalon jelenik meg a link által mutatott oldal.

1.b upbookmarks.php

A kód feladata a könyvjelzők mentése az aktuális felhasználó könyvjelző fájllába. Működése: A könyvjelzők listája egy txt fájlban van letárolva szerver oldalon. A felhasználói

felületen beírt adatok mentését végzi ebbe a txt fájlba. Illetve a feltöltött JSON fájlt értelmezi és menti az aktuális felhasználó bookmarks.txt fájllába.
Kód magyarázat: A kód két féle módon futhat le, attól függően, hogy a felhasználó egy url akar feltölteni vagy a Firefox könyvjelző mentését akarja feltölteni. Ezt egy IF es szerkezet felügyeli.

„Uploadbookmark” ág:

if(isset($_POST['uploadbookmark']))

A felhasználó 1 db url akar menteni, rákattint a „könyvjelző mentése”gombra, ekkor a FORM egy Input mező tartalmát fogja elküldeni a cél oldalnak, ami ebbe az esetbe ugyanaz az oldal. A mező típusa text(szöveg mező). A kód először értékül adja a megkapott input mező értékét egy tömb nulladik elemének.

$tomb[0]=$_POST["bookmark"];

Megtörténik az adatok lekérése az aktuális munkamenetről a felhasználó azonosítása végett és a saját bookmarks.txt fájl az elérési útvonalának meghatározása miatt. Az FTP kapcsolathoz szükséges adatok változókba tárolása után létrehozunk egy ideiglenes fájl ami arra szolgál, hogy a php kód ebbe tudjon írni olvasni, míg a kód teljesen le nem fut.

$random = rand(10000, 9999999999999);
$target = fopen("tmp/book.txt" . $random, "w");

Ftp kapcsolat létrehozása után ellenőrizzük, hogy létezik-e az adott felhasználónál a bookmarks.txt, ha létezik megnyitjuk és ugyanazt a tömböt amit a input mező tartalmának rögzítésre használtunk, az 1. indextől kezdve töltjük fel a már bookmarks.txt be lévő adatokkal.

```
if (file_exists ("ftp://".$ftp_user_name.":".$ftp_user_pass."@".$ftp_server.$source)){
$f=fopen("ftp://".$ftp_user_name.":".$ftp_user_pass."@".$ftp_server.$source,"r");
while(!feof($f))
  {
            $tomb[$i] = fgets($f);
```

```
                $i++;}
```

Az így kapott tömböt végig járva beleírjuk tartalmát a ideiglenes fájlunkba.

```
while ($i < $count){
If (substr($tomb[$i],0,4)=="http") // csak a httpvel kezdődő tömböket írja az ideiglenes fájlba
{
 fwrite($target,$tomb[$i]);
 }
 $i++;}
```

Ezután az ideiglenes fájlt újra megnyitva annak tartalmát beleírjuk a már végleges bookmarks.txt fájlba Ftp kapcsolaton keresztül. A végleges fájlba így bekerül az első sorba a felhasználó által az inputmezőbe beírt URL.

```
$handle = fopen("tmp/book.txt". $random, 'r')
ftp_fput($conn, $source, $handle, FTP_BINARY)
```

„uploadjson” ág:

isset($_POST['uploadjson'])

A felhasználó Json fájl elérésének megadása után, a „fájl mentése” gombra kattintva ebben az ágba lévő php kód kerül futtatásra. A FORM ebben az esetben az oldalnak egy Fájl típusú adatot küld. Első esetben ellenőrzi, hogy az adott fájl kiterjesztése megfelelő-e az adott követelményeknek. Ezt úgy teszi, hogy a $_FILES tömböt bejárva, ami a küldött fájl adatait tartalmazza kikeresi azt a mezőt ahol a fájl neve található. Utána a tömb értékét stringként kezeli és explode függvény segítségével szétvágja „pont” karakter mentén. A srting végét nézve ellenőrizheti, hogy az utolsó „pont” karakter után lévő string, ami egyben a kiterjesztés egyezik-e az előre definiált kiterjesztések valamelyikével.

```
$engedfajl = array("json");
 foreach ($_FILES as $file) {
   if ($file['tmp_name'] > '') {
     if (!in_array(end(explode(".",
```

```
        strtolower($file['name']))),
        $engedfajl)) {
     die($file['name'].' érvénytelen fáljtipus!<br/>'.
      '<a href="javascript:history.go(-1);">'.
      '&lt;&lt Vissza</a>');
    }
  }
}
```

Ha nem egyezik, akkor a php kód nem fut tovább és hiba üzenet jelenik meg. Ha jó a kiterjesztés a php kód tovább fut és megnyitja a FORM által küldött fájl. A fájlt egy $tarolo nevű változóba menti.

```
$file = fopen($_FILES['uploadedfile']['tmp_name'], "r") or die("nem sikerült megnyitni a fájlt");
while(!feof($file))
  {
        $tarolo.=fgets($file);
       }
```

Mivel a Json fájlból csak az URL re van szükségünk, ezért szükséges néhány módosítást alkalmazni a nekünk lényeges adat kiszűréséhez. Először is a $tarolo stringet tömbe vágjuk szét "uri":" karakterek mentén így a tömb elemei mind az urlel fognak kezdődni, s már csak le kel vágni a végéről a számunkra felesleges adatokat.

```
$tomb=(explode('"uri":"',$tarolo));
```

A következő ciklus ezt a célt szolgálja. Kikeresi az első idéző jelet a tömb elemében és megjegyzi hányadik pozícióba találta. A első idéző jelig tart az url, így utána már számunkra szükségtelen adatok vannak. Ezért a tömbünk elemébe található stringet levágjuk és értékül adjuk ugyanannak a tömb elemnek.

```
while ($i < $count){
 $pos = strpos($tomb[$i], '"');
 $tomb[$i] = substr($tomb[$i],0,$pos);
```

```
 $tomb[$i].="\r\n";
 $i++;
 $pos=0;
}
```

Ezt végre hajtjuk az összes tömb elemen, s ha lefutott a ciklus, akkor a tömbünk elemei már csak az url-t fogják tartalmazni. Ezzel megtörtént a szűrés és kezdődhet a fájlba mentés. Azonosítjuk a felhasználót, megadjuk a ftp adatokat és az elérési útvonalat. Kapcsolódunk FTP keresztül a szerverhez. A létrehozunk egy ideiglenes fájlt, ellenőrizzük, hogy létezik-e a bookmarks.txt. Kiolvassuk az adatokat és az előbb használt tömb végére elkezdjük menteni az olvasott adatokat. Ha a fájl végére értünk, megkapjuk a tömböt, ami már tartalmazza a friss illetve a régi adatokat is.

```
if (file_exists ("ftp://".$ftp_user_name.":".$ftp_user_pass."@".$ftp_server.$source)){
$f=fopen("ftp://".$ftp_user_name.":".$ftp_user_pass."@".$ftp_server.$source,"r");
while(!feof($f))
  {
      $tomb[$i] = fgets($f); // A szűrésnél használt tömb végére beolvassuk a régi URLeket
      $i++;
      }
```

Nincs más dolgunk már, hogy beleírjuk a ideiglenes fájlba. Ellenőrzésként még figyeljük, hogy csak olyan urlek szerepeljenek, amik érvényesek (http:// kezdet), hibás sorok kiszűrése.

```
If (substr($tomb[$i],0,4)=="http"){
 fwrite($target,$tomb[$i]);
 }
 $i++;}
```

Miután megvagyunk az ideiglenes fájlal, megnyitjuk a végleges fájlt és beleírjuk az ideiglenes fájl tartalmát.

```
$handle = fopen("tmp/book.txt". $random, 'r')
ftp_fput($conn, $source, $handle, FTP_BINARY)
```

Ezzel a Json fájlba található urlek mentésre kerültek a Bookmarks.txt fájlba.

1.c konyvjelzotorles.php

A kód feladata a könyvjelzők törlése bookmarks.txt-ből. Felsorolja a benne található könyvjelzőket és a könyvjelző mellet található pipát kijelölve törlésre küldhetjük.
A kód működése: Beolvassa a bookmarks.txt fájl tartalmát, kilistázza, a törlés gombra kattintva csak azokat írja vissza a fájlba, amik mellet nem volt pipa.
Kód magyarázat: Azonosítjuk a felhasználót, megadjuk bookmars.txt elérését, Ftp kapcsolathoz szükséges adatokat adunk meg. Ideiglenes fájl hozunk létre. Egy tömbbe beolvassuk a már létező Bookmarkst.txt tartalmát, ha létezik.

```
if (file_exists ("ftp://".$ftp_user_name.":".$ftp_user_pass."@".$ftp_server.$source)){
$f=fopen("ftp://".$ftp_user_name.":".$ftp_user_pass."@".$ftp_server.$source,"r");
while(!feof($f))
  {
            $darabonyvj[] = (fgets($f));
            $i++;
      }
```

Az ezután következő kód a „könyvjelzők törlése” gomb után lép életbe.

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

Ezután a kód ellenőrzi, hogy létezik az ideiglenes fájl, illetve bookmarks.txt, ha nem, hiba üzenetet ír ki és kilép a kód. Ha minden rendben, a kód megnyitja a ideiglenes fájlt és egy ciklussal bejárva a $konyvj tömböt beleírja a ideiglenes fájlba azt a tömb elemet amihez tartozó checkbox nem volt kipipálva, egyébként a tömb elem tartalmát törli. Ha ciklus végig ment a tömbön, az ideiglenes fájl fogja tartalmazni az összes menteni kívánt elemet. A ideiglenes fájl tartalmát beleírjuk bookmarks.txt be.

```
for ($szamcheckbox = 0; $szamcheckbox <= $_POST['hossz'] ; $szamcheckbox++) {
                        if ($_POST[$szamcheckbox] != 'on') {
                              fwrite($handle, $darabonyvj[$szamcheckbox]);
```

```
                }
else
{
                        $darabonyvj[$szamcheckbox] = "";
                }
}
```

A HTLM oldalon a FORM-ot létrehozva és a benne kilistázott könyvjelzők mellé. A php generál egy checkboxot, aminek az értéke $darab, ez az változó folyamatosan nő. Ez a változó utolsó értéke is el lesz küldve az oldalnak hossz néven, amiből kiszámolható a $szamcheckbox változó, illetve mettől meddig menyjen a ciklusba.

```
$darab = 0;
foreach ($darabonyvj as $value) {
    if ($value != "") {
            echo "<tr onMouseOver=\"this.bgColor = '#E6E8FA'\" onMouseOut =\"this.bgColor = '#FFFFFF'\"><td>" . $value . "</td><td align=center><input type='checkbox' name='$darab'></td></tr>\n";
            $darab++;
        }
```

A formba rejtve van egy input mező, ennek az értéke lesz $darab-1 és ezzel küldjük el változót.

```
<input type="hidden" name="hossz" value="<?php echo $darab - 1; ?>">
```

IV.2. Képek

Az oldal szolgáltatásai között szerepel a képek kezelése. Az oldal felhasználói felülete segítségével képeket tölthetünk fel az oldalra, amit megtekinthetünk a képek menüpont alatt. Továbbá megtalálható a képek feltöltése és képek törlése almenüpont az oldalon. Három modul kapcsolódik a szolgáltatáshoz a pictures.php, a keptorles.php és a uppictures.php.

2.a pictures.php

Ez a modul felelős a felhasználó személyes képeinek megjelenítéséért. A képekre kattintva a kép megtekinthető teljes méretében egy új lapon. (3.ábra)
Működése: a kód futása közben csatlakozunk ftp keresztül a szerverrel és a felhasználó képek mappájának tartalmát kilistázzuk. Ezt a listát felhasználva a Php kód, html tageket generál az egyes képek elérési útvonala köré és így lesz látható a kép az oldalon.
Kódmagyarázat: Elsőként azonosítjuk, melyik felhasználóról van szó, hogy meghatározhassuk annak képek mappáját. Ezután megadjuk az Ftp kapcsolathoz szükséges adatokat és a mappa elérési útvonalát.

```
$ftp_root ="ftp://".$ftp_user_name.":".$ftp_user_pass."@".$ftp_server.$source;
 $handle = opendir($ftp_root);
```

Az opendir függvénnyel megnyitjuk az a célmappánkat. A readdir függvény és egy ciklus kombinációja segítségével végig megyünk a mappa tartalmán miközben a $fájl változó felveszi a mappában található fájlok nevét. Ha a fálj nevéhez még hozzáfűzzük a mappa teljes elérési útvonalát, akkor megkapjuk az adott kép teljes elérési útvonalát a szerveren, amit a html tag-ek között echózzuk.

```
while ($fajl = readdir($handle))
 {

            If(0 ==fmod($szamlalo, 4)){
    echo '<tr>';
            }
            echo '<td class="header">';
    echo '<a href="'.$eleres.$fajl.'" target=_blank ><img src="'.$eleres.$fajl.'"
width="150" height="100"/></a>';
            echo '</td>';
```

```
            If(3 ==fmod($szamlalo, 4)){
    echo '</tr>';
            }
            $szamlalo++;
 }
```

Az oldal forráskódjában már csak az fog látszani, hogy a kép elérési útvonala a <img> tag-ek között található „scr” értékként. Azt hogy a oldalon egy sorba csak meghatározott számú kép jelenjen meg, azt úgy érjük el, hogy a cikluson belül egy IF es szerkezet van ami egy számláló értékét vizsgálja függvény segítségével, ha értéke 4el osztva nulla maradékot dob az azt jelenti hogy a új sort kezdünk ”<tr>” beszúrásával, ha 4 gyel osztva 3 maradékod dob akkor a sort lezárjuk „</tr>” beszúrásával. A ciklus addig fut, míg a célmappánkba található képek végére nem ér. Az utolsó mozzanat, hogy lezárjuk a nyitott kapcsolatokat.

closedir($ftp_root);

2.b uppictures.php

A modul a képek feltöltéséért felelős. A felhasználói felületen láthatunk egy tallózható input mezőt illetve egy gombot. Az inputmezőre kattintva böngészhetünk saját gépünkön, s ha megtaláltuk a feltöltésre szánt képet, megnyitás gomb után a kép feltöltése gombra kattintva feltölthetjük szerverre a képünket.

Működés: Az oldal megkapja feltöltésre szánt képet, aminek ellenőrzi a kiterjesztését, majd az aktuális felhasználó képek mappájába másolja azt.

Kódmagyarázat: A kép feltöltése gombra kattintva a szerver letölti a felhasználó gépéről a képet.

if(isset($_POST['uppictures'])){

A oldal a megkapott fájlnak ellenőrzi a kiterjesztését a könyvjelzők feltöltésénél használt módon.

```
$engedfajl = array("jpg","bmp","png");
  foreach ($_FILES as $file) {
    if ($file['tmp_name'] > '') {
      if (!in_array(end(explode(".",
          strtolower($file['name']))),
```

```
        $engedfajl)) {
     die($file['name'].' érvénytelen fáljtipus!<br/>'.
      '<a href="javascript:history.go(-1);">'.
      '&lt;&lt Vissza</a>');
    }}}
```

Ezután megnyitjuk a fájl illetve a fájl nevét is külön változóba tesszük.

```
$fp = fopen($_FILES['uploadedfile']['tmp_name'], "r");
$nev=$_FILES['uploadedfile']['name'];
```

Azonosítjuk a felhasználót, megadjuk az FTP kapcsolathoz szükséges adatokat és kapcsolódunk. A képet feltöltjük a felhasználó képek könyvtárába. A feltöltés sikerességét vagy sikertelenségét kiíratjuk az oldalra.

```
$conn = ftp_connect($ftp_server) or die("Nem tudtam az FTP-re felkapcsolódni.");
ftp_login($conn,$ftp_user_name,$ftp_user_pass);
ftp_pasv( $conn, true );
if (ftp_fput($conn, $ftp_root.$nev, $fp, FTP_BINARY)) {
   echo "Sikeres feltöltöttünk $nev\n";
} else {
   echo "Probléma $nev feltöltésében\n";
}
```

2.c keptorles.php

A modul segítségével törölni tudjuk a képeket a szerverről. A felhasználói felületen megtekinthetjük a képeket mellette lévő checkbox-ot kipipálva, s majd a „képek törlése” gombra kattintva törölhetjük.

Kódmagyarázat: Az oldalon képek listájának előállításához szükségünk van a FTP kapcsolathoz. Ehhez kérjük be elsőként az adatokat. A felhasználót azonosítjuk. Megadjuk a cél mappánt elérési útvonalát a már azonosított felhasználói név segítségével. Létrehozzuk az FTP kapcsolatot, ami a törléshez szükséges. Html oldalon kilistázzuk a cél mappánk tartalmát.

Mellé a checkboxokat generálunk. A checkboxok neve fogja majd megadni, hogy melyik képről van éppen szó. A neve egy szám, ami folyamatosan növekszik nullától.

```
while ($fajl = readdir($handle))
 {
    echo "<tr onMouseOver=\"this.bgColor = '#E6E8FA'\" onMouseOut =\"this.bgColor = '#FFFFFF'\">";
                echo "<td>";
                echo '<img src="'.$eleres.$fajl.'"  width="150" height="100"/>';
                echo "</td><td align=center><input type='checkbox' name='$k'></td></tr>\n";

                $k++;
 }
```

Amikor a FORM elküldi az adatokat a fogadó oldalnak, ami ebben az esetben ugyanaz az oldal, a php kód ismét végig megy a képek mappa tartalmán, közbe vizsgálja, hogy nem-e ahhoz a képhez ért oda, amit ép, törölni kell. Ha a kép sorszáma egyezik azzal a checkbox nevével, ami be van pipálva akkor a kép törölve lesz. Miután a ciklus végig ment a mappa tartalmán és törölte azokat a képeket, melyeket kellet, az oldal frissül és visszatérünk a kiinduló helyzetbe.

```
if(isset($_POST['generate'])){
                        $szamlalo=0;
                        while ($fajl = readdir($handle))
        {
                        if ($_POST[$szamlalo] == 'on') {
                         if (ftp_delete($conn, $Leleres.$fajl)) {
                                echo "$fajl törlése sikeres\n";
                                } else {
                                echo "Nem tudtam törölni $fajl\n";
                                }
```

```
                    }
                    $szamlalo++;
                    }
                    echo '<meta http-equiv="refresh"
content="0;URL='.$settings['hostname'].'/szakdolgozat/?module=keptorles">';
}
```

IV.3. Zene lejátszó

Az oldal szolgáltatása között szerepel a mp3 formátumú fájlok lejátszása. Saját zene fájlainkat tölthetjük fel. A zene oldalon meghallgathatjuk őket, majd törölhetjük is. A zene menüpont alatt megtalálhatjuk a mp3 feltöltése és a mp3 törlése menüpontot. Három modul kapcsolódik a szolgáltatáshoz player.php, upmusic.php,zenetorles.php.

3.a Player.php

A oldal meghív egy mp3 lejátszót és paramétereket ad meg a lejátszónak, ami segítségével a lejátszó azonosítani tudja mely mp3 számok tartoznak a aktuális felhasználóhoz. Az oldal a http://flash-mp3-player.net/ oldalról ingyenesen letölthető programot használja. (1.ábra)
Kódmagyarázat: Itt is a felhasználó azonosításával kezdjük illetve az ftp kapcsolathoz szükséges adatok beírásával. Ftp kapcsolatra azért van szükségünk hogy létre tudjunk hozni egy listát a felhasználó zenéiről, melyből utána egy lejátszási listát készítsünk.

```
$felhasznalo=$_SESSION['admin'];
$source = "/szakdolgozat/".$felhasznalo."/music/"; //konkrét esetben
$ftp_root ="ftp://".$ftp_user_name.":".$ftp_user_pass."@".$ftp_server.$source;
$handle = opendir($ftp_root);
```

Az egész úgy történik, hogy megnyitjuk a célmappánkat, amin egy ciklus végig megy és egy string-be fűzi az ott talált fájlok nevét. Majd még az elejére fűzi a zene mappa teljes elérési útvonalát, így megkapjuk a fájl teljes elérési útvonalát majd, ezt folytatva minden a zenemappában található fájl nevét elérését felfűzzük ebbe az egy stringbe, a különböző fájl

elérések közé még beillesztünk egy speciális karaktert, hogy a lejátszó értelmezni tudja a lejátszási listát.

```
$playlist.='"mp3=';
while ($fajl = readdir($handle)){
 $playlist.=$eleres.$fajl;
 $playlist.='|';
 }
$playlist = substr($playlist,0,strlen($playlist)-1);
$playlist.='&amp;width=600&amp;height=500"';
$playlist=iconv('ISO-8859-2', 'UTF-8', $playlist);
```

Html kóddal meghívjuk a lejátszó objektumot majd paramétereit is megadjuk, szélességét, magasságát, színét. Ezután az előre elkészített lejátszási listát $playlist változót echozuk ki php segítségével a hmtl kódba. Végezetül lezárjuk a még élő kapcsolatokat.

3.b upmusic.php

A modul, felelős a mp3 fájlok feltöltéséért. Tallózás gombra kattintva böngészhetünk saját számítógépünkön, majd ha a mp3 feltöltése gombra kattintunk a kiválasztott mp3 feltöltésre kerül a szerverre.
Kódmagyarázat: Az elején deklarálunk egy új függvényt amit arra fogunk használni hogy a mp3 zenék magyar karaktereit átalakítsuk angolra, hogy a karakter kódolási problémákat kiküszöböljük.

```
function normaliza ($string){
   $a = 'öÖüÜóÓőŐúÚéÉáÁűŰíÍ';
   $b = 'oouuoooouueeaauuii';
   $string = utf8_decode($string);
   $string = strtr($string, utf8_decode($a), $b);
   $string = ($string);
   return utf8_encode($string);}
```

A mp3 feltöltése gombra kattintva a szerver letölti a felhasználó gépéről a zenét.

if(isset($_POST['upmusic']))

Az oldal a megkapott fájlnak ellenőrzi a kiterjesztését a könyvjelzők feltöltésénél használt módon.

```
$engedfajl = array("mp3");
  foreach ($_FILES as $file) {
    if ($file['tmp_name'] > '') {
      if (!in_array(end(explode(".",
          strtolower($file['name']))),
          $engedfajl)) {
       die($file['name'].' érvénytelen fáljtipus!<br/>'.
        '<a href="javascript:history.go(-1);">'.
        '&lt;&lt Vissza</a>');
}}}
```

Ezek után, megnyitjuk a fájlt, hogy ftp kapcsolatnál tudjuk használni, majd a nevét is változóba mentjük. A fájl nevére alkalmazzuk „normaliza” függvényt. Aminek hatására a nevében található hosszú vagy nagy és hosszú ékezetek rövidre és kicsire módosulnak. Megadjuk az ftp kapcsolathoz szükséges adatokat és létrehozzuk a kapcsolatot. Ezek után nincs más dolgunk csak feltölteni az adott fájl az új névvel a szerverre.

```
if (ftp_fput($conn, $ftp_root.$nev, $fp, FTP_BINARY)) {
   echo "Sikeres feltöltöttünk $nev\n";
} else {
   echo "Probléma $nev feltöltésében\n";
}
```

3.c zenetorles.php

Az általunk már nem használt mp3 fájlokat törölhetjük a szerverről. Ebben nyújt segítséget a zenetorles.php modul. A szerveren lévő zenefájlainkat felsorolva fogjuk látni. A nevük mellet található ellenőrző dobozt kipipálva majd a mp3 törlése gombra kattintva végleg megszabadulhatunk fájlainktól.

Kódmagyarázat: Azonosítjuk a felhasználót, megadjuk az ftp kapcsolathoz szükséges adatokat. Megadjuk a cél mappánt elérési útvonalát a már azonosított felhasználói név segítségével. Létrehozzuk az FTP kapcsolatot, ami a törléshez szükséges. Html oldalon kilistázzuk a cél mappánk tartalmát. Mellé a checkboxokat generálunk.

```
while ($fajl = readdir($handle))
 {
    echo "<tr onMouseOver=\"this.bgColor = '#E6E8FA'\" onMouseOut =\"this.bgColor = '#FFFFFF'\">";
                echo "<td>";
                echo  iconv('ISO-8859-2', 'UTF-8', $fajl);
                echo "</td><td align=center><input type='checkbox' name='$k'></td></tr>\n";
                $k++;
 }
```

A checkboxok neve fogja majd megadni, hogy melyik zenéről van éppen szó. A neve egy szám, ami folyamatosan növekszik nullától. Amikor a FORM elküldi az adatokat a fogadó oldalnak, ami ebben az esetben ugyanaz az oldal, a php kód ismét végig megy a zene mappa tartalmán, közbe vizsgálja, hogy nem-e ahhoz a mp3 fájlhoz ért oda, amit ép, törölni kell.

```
if(isset($_POST['generate'])){
                        $szamlalo=0;
                        while ($fajl = readdir($handle)){
                        if ($_POST[$szamlalo] == 'on') {
                         if (ftp_delete($conn, $Leleres.$fajl)) {
                                echo "$fajl törlése sikeres\n";
                                } else {
                                echo "Nem tudtam törölni $fajl\n";
                                } }
                        $szamlalo++;}
```

echo '<meta http-equiv="refresh" content="0;URL='.$settings['hostname'].'/szakdolgozat/?module=zenetorles">';

}

Ha a fájl sorszáma egyezik azzal a checkbox nevével, ami be van pipálva akkor a zene fájl törölve lesz. Miután a ciklus végig ment a mappa tartalmán és törölte azokat a zene fájlokat, melyeket kellet, az oldal frissül és visszatérünk a kiinduló helyzetbe.

IV.4. Jegyzettömb

Az oldal szolgáltatásai között szerepel a jegyzettömb elnevezésű modul. Segítségével az oldalon feljegyzéseket készíthetünk txt formátumba. Több feljegyzésünk is lehet, amiket később törölhetünk is. Három php modul kapcsolódik a szolgáltatáshoz, jegyzettomb.php, ujjegyzet.php és a jegyzettorles.php

4.a jegyzettomb.php

Ez a modul fellelős a jegyzet tartalmának megjelenítéséért, a jegyzet tartalmának módosításáért, illetve a módosítandó jegyzet kiválasztásáért. Az általa megjelenített oldalon szerepel egy textarea alatta a felhasználó jegyzetfájlainak listája majd egy mentés gomb. A textarea ba jelenik meg az aktuálisan kiválasztott jegyzet fájl tartalma.(4.ábra)

Kódmagyarázat: Elsőként lekérjük a jegyzet fájl nevét, kiszűrűnk néhány speciális karaktert az oldal biztonsága kedvéért,

$fajl= mysql_real_escape_string($_GET['fajl']);

If (0!=substr_count($fajl,".."))){

$fajl="jegyzet.txt";

Echo "Érvénytelen Fáljnevet adtál meg";

}

ha érvénytelen vagy nem létező jegyzet fájlnevet adtak meg az urlbe, akkor a kód automatikusan egy alapértelmezett „jegyzet.txt” értéket ad meg a fájlnévnek.

If ($fajl=="") {

$sourcesmall = "/szakdolgozat/".$felhasznalo."/jegyzetek/";

$ftp_root ="ftp://".$ftp_user_name.":".$ftp_user_pass."@".$ftp_server.$sourcesmall;

```
$handle = opendir($ftp_root);
$fajl = readdir($handle);
If ($fajl==""){
$fajl="jegyzet.txt";
echo '<meta http-equiv="refresh" content="0;URL='.$settings['hostname'].'/szakdolgozat/?module=jegyzettomb&fajl='.$fajl.'">';
}
```

Utána azonosítjuk a felhasználót, és megadjuk az ftp kapcsolathoz szükséges adatokat, illetve a felhasználó jegyzetek könyvtárának elérési útvonalát. Létrehozunk egy ideiglenes fájlt, ami a biztonságos mentéshez szükséges. Csatlakozunk a szerverhez ftpén keresztül Ezután ellenőrizzük azt, hogy az urlbe megkapott fájlnév értékét. Ha üres, akkor megnyitjuk a felhasználó jegyzetek mappáját és az ott található első jegyzett neve lesz az értéke. Ha nem üres, akkor azzal az adattal dolgozunk tovább. Megnyitjuk a megadott fájlt és tartalmát egy változóba olvassuk, majd ez a változót beleírjuk az ideiglenes fájlba. A létező jegyzetek ki listázása úgy történik, hogy a felhasználó jegyzetek mappáján végig megyünk közbe linkeket generálva.

```
$sourcesmall = "/szakdolgozat/".$felhasznalo."/jegyzetek/";
$ftp_root ="ftp://".$ftp_user_name.":".$ftp_user_pass."@".$ftp_server.$sourcesmall;
$url="http://www.kerecsengergo.clans.hu/szakdolgozat/?module=jegyzettomb&fajl=";
$handle = opendir($ftp_root);

 while ($fajl = readdir($handle))
  {

     echo "<tr onMouseOver=\"this.bgColor = '#E6E8FA'\" onMouseOut =\"this.bgColor = '#FFFFFF'\">";
                echo "<td>";
                echo "<a href=".$url.$fajl.">".$fajl."</a>";
                echo "</td></tr>\n";
                $k++;
```

```
}
```

A linkek urlje a jegyzettömb modul url lesz kiegészítve értéknek adva a fájl nevével, amit a könyvtáron végigfutó ciklus ad meg nekünk. Így, a linkek kattinthatóak lesznek és rákattintva a linkre újból bejön az oldal de annak a jegyzettömbnek a tartalmát fogja megjeleníteni amire kattintottunk. Ha menteni szeretnénk az adott jegyzetett végül a mentés gombra kell kattintani, aminek a kódbeli azonosítója a „generate”. A kódba a if(isset($_POST['generate'])) feltétel igazzá válik és a kód fut tovább a benne lévő utasításokkal. Leellenőrizzük a szükséges fájlok létezését és hogy helyes ideiglenes fájl van-e megnyitva és attól függően, hogy a feltételek milyen állítást dobnak vagy hiba üzenetet kapunk vagy újraírjuk a ideiglenes fájlt.

```
if  (!file_exists ("ftp://".$ftp_user_name.":".$ftp_user_pass."@".$ftp_server.$source)) {
  if (!$handle = "jegyzet.txt". $random) {
echo "<h2><span>Hiba történt</span></h2>
<p><span class='ipcim'>Hiba leírása: </span>A üdvözlő üzenet szerkesztő nem tudta megnyitni a note.txt fájlod, így az adatok betöltése nem lehetséges.</p>";
        exit;}
                $handle = fopen("tmp/jegyzet.txt". $random, "w");
            fwrite($handle, $_POST['note']);
                fwrite($handle, "\r\n");
} else {
  if (!$handle = "jegyzet.txt". $random) {
echo "<h2><span>Hiba történt</span></h2>
<p><span class='ipcim'>Hiba leírása: </span>A generáló nem tudta megnyitni a fájlod, így az adatok betöltése nem lehetséges.</p>";
        exit;}
                $handle = fopen("tmp/jegyzet.txt". $random, "w");
            fwrite($handle, $_POST['note']);
                fwrite($handle, "\r\n");}
```

Ezek után beleírjuk a végleges fájlba az adatokat.

ftp_fput($conn, $source, $handle, FTP_BINARY)

4.b ujjegyzet.php

Az új jegyzet létrehozásáért felelős. Maga a kód végtelenül egyszerű. Tartalmaz egy input mezőt amibe majd a új jegyzetünk nevét fogjuk megadni, illetve egy mentés gombot. A kód mentés gomb lenyomására aktiválódik az inputmező tartalmát változóba rakja hozzá fűz egy „.txt” karaktersorozatot majd az oldal befrissül a jegyzettömb oldalra irányulva, ahol a fájlnév helyén az előzőleg inputmezőbe megadott érték fog szerepelni. A jegyzettomb.php kódja pedig az elérési útvonalat úgy fogja megnyitni, hogy ha a fájl nem létezik akkor létrehozza. Így a fájl létrehozataláról végülis a jegyzettomb.php gondoskodik. A ujjegyzet.php nak csak az a feladata, hogy a jegyzettomb.php-ét paraméterrel lássa el az urlben.

4.c jegyzettorles.php

A modul segítségével törölni tudjuk a jegyzeteket a szerverről. A felhasználói felületen megtekinthetjük a jegyzetek listáját és a mellette lévő checkbox-ot kipipálva, s majd a „jegyzetek törlése” gombra kattintva törölhetjük.

Kódmagyarázat: Az oldalon jegyzetek listájának előállításához szükségünk van a FTP kapcsolathoz. Ehhez kérjük be elsőként az adatokat. A felhasználót azonosítjuk. Megadjuk a cél mappánt elérési útvonalát a már azonosított felhasználói név segítségével. Létrehozzuk az FTP kapcsolatot, ami a törléshez szükséges. Html oldalon kilistázzuk a cél mappánk tartalmát. Mellé a checkboxokat generálunk.

```
while ($fajl = readdir($handle))
 {
    echo "<tr onMouseOver=\"this.bgColor = '#E6E8FA'\" onMouseOut
=\"this.bgColor = '#FFFFFF'\">";
                echo "<td>";
                echo  $fajl;
```

```
            echo "</td><td align=center><input type='checkbox' name='$k'></td></tr>\n";
            $k++;
 }
```

A checkboxok neve fogja majd megadni, hogy melyik jegyzetről van éppen szó. A neve egy szám, ami folyamatosan növekszik nullától. Amikor a FORM elküldi az adatokat a fogadó oldalnak, ami ebben az esetben ugyanaz az oldal, a php kód ismét végig megy a jegyzetek mappa tartalmát, közbe vizsgálja, hogy nem-e ahhoz a jegyzethez ért oda, amit ép, törölni kell. Ha a jegyzet sorszáma egyezik azzal a checkbox nevével, ami be van pipálva akkor a jegyzet törölve lesz. Miután a ciklus végig ment a mappa tartalmán és törölte azokat a jegyzeteket, melyeket kellet, az oldal frissül és visszatérünk a kiinduló helyzetbe.

```
Leleres="/szakdolgozat/".$felhasznalo."/jegyzetek/";
if(isset($_POST['generate'])){

                    $szamlalo=0;
                    while ($fajl = readdir($handle))
        {
                    if ($_POST[$szamlalo] == 'on') {
                     if (ftp_delete($conn, $Leleres.$fajl)) {
                            echo "$fajl törlése sikeres\n";
                            } else {
                            echo "Nem tudtam törölni $fajl\n";
                            }
                    }
                    $szamlalo++;
                   }
                    echo '<meta http-equiv="refresh" content="0;URL='.$settings['hostname'].'/szakdolgozat/?module=jegyzettorles">';
}
```

V. Fejlesztésnél használt nyelvek jellemzése

V.1. PHP

„A PHP (PHP: Hypertext Preprocessor) nyílt forráskódú, számítógépes szkriptnyelv, legfőbb felhasználási területe a dinamikus weboldalak készítése. Emiatt a PHP-t jórészt szerver-oldalon használják, bár létezik parancssori interfésze is, illetve önálló, grafikus felületű alkalmazások is létrehozhatóak vele.”[w]

„Szintaxis:
A PHP utasításokat mindig egy meghatározott karaktersorral kell kezdeni és bezárni.
Ez lehet a:
<? ?> //(ha a php.ini-ben be van kapcsolva az "short_open_tag"), a
<?php ?> //, a
<script language="PHP"> </script> //, és a
<% %> //csak ha a php.ini-ben az "asp_tags" On-ra van állítva
//Végül egy speciális eset:
<?=$valtozo;?> //Ez sima változók kiírásának egyszerű, speciális módja.

Sok hibát kizárhatunk, ha programjainkban a "<?php" nyitó és a "?>" záró karaktersorokat használjuk. Ugyanis az érvényes nyitó és záró elemek a php.ini konfigurációs fájlban vannak deklarálva, amik szerverenként eltérhetnek egymástól, így előfordulhat, hogy egyes szerveren nem értelmezi az interpreter a php programunkat, hanem közvetlenül a kimenetre küldi a forráskódot
Minden változót és összetett adatszerkezetet(objektumot,tömböt) a $ jellel kezdünk (például $valtozo="béka"; echo $valtozo;). Kivételt képeznek ez alól a konstansok (például define("KONSTANS","123456"); echo KONSTANS;).
A tömb indexelése – hasonlóan más C alapú nyelvekhez – 0-tól kezdődik, tehát a $tomb[1] eleme valójában a tömb 2. eleme.

A legegyszerűbb utasítás a már említett echo, vagy az ezzel egyenértékű print. Ennek segítségével lehet kihasználni igazán a php-t: közvetlenül lehet írni a készülő HTML dokumentumba, akár formázott szöveget is."[w]

V.1. SQL

„Az SQL alapjait az IBM-nél fektették le, még az 1970-es években. Elvi alapot a relációs adatmodell szolgáltatott, amit Edgar F. Codd híres 12 szabályával írt le először, 1970-ben.
Az IBM, az Oracle és más gyártók is érdekeltek voltak egy szabványos lekérdező nyelv kifejlesztésében, amivel a relációs adatbázisok programozhatók. Az iparági összefogással létrejött ANSI NCITS (National Committee on Information Technology Standards) H2 csoport lerakta az SQL alapjait.
A szabványt az ANSI (Amerikai Nemzeti Szabványügyi Intézet – American National Standards Institute) 1986-ban, az ISO (Nemzetközi Szabványügyi Szervezet – International Organization for Standardization) 1987-ben jegyezte be. Az első változatot SQL86 néven is szokták emlegetni." [sql]

VI. Munkamenet (session)

Weboldalainkat egyszerre többen is olvasgatják. Ezért szükség van rá, hogy látogatóinkat egyedileg azonosíthassuk, követhessük mozgásukat az oldalaink között. Az adott oldalunkon a munkamenet tárolja a bejelentkezettség állapotát. Mivel a webes kommunikáció alapjául szolgáló HTTP állapot nélküli protokoll alapértelmezetten nem azonosítja a látogatót. Ezért sütiket használunk, bár ez nem biztonságos megoldás mivel a sütiket a kliens oldalon tároljuk le így azok könnyen, hamisíthatóak hozzáférhetőek a felhasználók számára. Tehát egy látogatás során sokall biztonságosabb ,ha a „sütiket" szerver oldalon tároljuk és ebben nyújt nekünk segítséget a sessionok.

A sessionok hasonló rövid, skaláris adatsorok, melyeket a szerveren tárolunk. Formája lehet ideiglenes adatfájl, de a PHP beállításai szerint tárolhatók azok külső adatbázisban is. Így biztosított, hogy azokhoz a kliens közvetlenül nem férhet hozzá, csak a mi PHP scriptünk.
A PHP sessionok használatával lehetővé válik adatok rögzítése a klienstőr amit később azonosíthatunk magával a kliensel.
Sessionok használatakor a PHP minden látogatóhoz egy egyedi azonosítót rendel. Az oldal elküldése után a klienshez kapcsolunk a megőrzendő adatokat majd ehhez az azonosítóhoz kötve tároljuk a szerveren. A PHP minden oldal lekéréskor ellenőrzi, hogy létezik-e már ilyen azonosító, ha nem talál (új látogató az első oldalt kéri), akkor létrehoz egyet.
Az adatokat a szerver tárolhatja egy külön mappában, szöveges adatfájlokban a szerveren, vagy akár adatbázisban is. A böngészőnek csak a kliens saját azonosítóját küldi el, amit a következő oldal kérésekor vissza fog kapni a böngészőtől. Az egyedi azonosító egy véletlen-generált 32 jegyű karaktersor. Gyakorlatilag nem lehetséges hogy két felhasználó ugyanazt a azonosítót kapja.

Sessionok létrehozása:
Ha a PHP beállításainál a session.auto_start be van kapcsolva, akkor a PHP automatikusan létrehozza a sessiont. Ha ez a beállítási opció kikapcsolt, akkor a session_start() függvény beírásával tudjuk elindítani a munkamenetet.
Ekkor a munkamenet fut és új adatot a $_SESSION tömbbe felvéve rögzíthetünk:
<?php

```
session_start();
$_SESSION["felhasznalo_nev"] = "valaki";
?>
```

A PHP a $_SESSION tömb tartalmát el fogja tárolni az oldal feldolgozása után. A következő oldal kérésekor így is elérhetjük a tárolt adatot:

```
<?php
session_start();
print $_SESSION["felhasznalo_nev"]; // kiírja, hogy 'valaki'
?>
```

Session változók kezelése, törlése

Munkamenet változókat egyszerűen a változó értékének módosításával módosíthatunk. Ügyeljünk arra, hogy a módosított érték munkamenet változóként csak a következő oldal feldolgozásakor lesz elérhető.

```
<?php
session_start();
print $_SESSION["felhasznalo _nev"]; // kiírja, hogy 'valaki'
$_SESSION["felhasznalo _nev"] = "másik valaki"; // megváltoztattuk az értéket
// a következő oldal feldolgozásakor a ' felhasznalo _nev' értéke már 'másik valaki' lesz.
?>
```

Egy adat törléséhez a sessionból elég a $_SESSION tömb megfelelő tömbelemét törölni.

A szerver kétféleképpen küldheti el a böngészőnek a session azonosítót:

1. automatikusan elhelyezhet a kliens gépen egy sütit, benne az azonosító nevével (ez alapértelmezetten: PHPSESSID) és az egyedi azonosító értékével.
2. Ha a PHP nem kap vissza a böngészőtől sütit (pl. mert ez az első oldala, vagy mert a böngészőben tiltva van a süti használata), akkor az URL-ben GET kérésként küldi el a session azonosítót. Ehhez a kód feldolgozása során minden belső - http://masdomain.com nélküli - linket és űrlapot kiegészít egy PHPSESSID GET kéréssel.

   Használható még a SID PHP állandó is, mely a session azonosító értékét tartalmazza, ha nem küldte vissza a böngésző süti értékként a session azonosítót.

VI. Sütik (*cookie)*

Számos weboldalon találkozunk olyan háttértárolókkal, mint például a süti (cookie). A sütik olyan, a felhasználó gépén tárolt, állományok, amelyek adatokat tárolnak a weboldal információival. Például a süti tárolhatja azt hogy a felhasználó be van-e jelentkezve az oldalon.

A süti hátránya a munkamenetekkel szemben az, hogy mivel a kliens oldalon van, nagyon nagy veszélyforrás a személyi adatainkra nézve, hogyha egy oldal sütiben tárolja a belépési információnkat. A sütiket több féle képen is lehet kezelni, mint például JavaScript segítségével vagy PHPval is.

A setcookie függvény létrehoz egy sütit, ami a többi header információval együtt kerül az olvasó böngészőjéhez. A paraméterek száma egytől hatig terjedő. Kizárólag az első paraméter kötelező, ami a változó nevét tárolja.
név: a süti neve (a tárolt változó neve) $_COOKIE['név']
érték: értéket rendelhetünk mellé, de titkosított fontos adatot ne tároljunk a sütibe mivel az kliens oldalon tárolódik és ehhez könnyen hozzá lehet férni
lejárás: az az idő intervallum, amíg a süti él. Az értéket másodpercekben kell megadni. time()+60*60*24*30 például 30 napos intervallumot rendelt a süti lejártához. Ha nem adjuk meg, akkor a böngésző bezárásával megszűnik a süti.
elérési út: az az elérési út a szerveren, ahol a süti használható pl.: /admin/ ekkor az admin és az admin alkönyvtáraiban használható a süti.
domain: az a domain ahol érvényes a süti, pl.: www.index.hu akkor a süti csak ezen a domainon fog müködni.
biztonság: kiválaszthatjuk, hogy a süti biztonságos https protokolon küldjük el vagy nem

A sütiben tárolt értékeket elérhetjük több változón keresztül is, a leggyakoribbak: $HTTP_COOKIE_VARS (a PHP 3 óta) és a $_COOKIE (a PHP 4.1.0 óta). Az alábbi kód részlet egy példa süti készítésére:

```
setcookie('cookie','suti');
```

Elküldi a ’cookie’ nevű változóban a megadott értéket.

VIII. Php függvények

mysql_connect()

Kapcsolatot nyit meg egy MySQL szerverhez. Pozitív MySQL azonosítóval tér vissza, ha a csatlakozás sikerült, FALSE-sal ha nem. A mysql_connect() függvény kapcsolatot nyit meg egy MySQL szerverhez. A paramétereket elhagyhatod. Az alapértelmezett értékek: server = 'localhost:3306', username = a folyamat tulajdonosának belépési neve password = üres karakterlánc. A mysql_connect() függvény kapcsolatot nyit meg egy MySQL szerverhez. A paramétereket elhagyhatod. Az alapértelmezett értékek: server = 'localhost:3306', username = a folyamat tulajdonosának belépési neve password = üres karakterlánc.

A server paraméter tartalmazhat egy portszámot is, például: "hostname:port" vagy tartalmazhatja a MySQL socket elérési útvonalát, például: ":/path/to/socket". Az utóbbi hostname paramétert használva is a helyi MySQL szerverhez próbál majd kapcsolódni a függvény.A sikertelen kapcsolatkor kiírt hibaüzenetet elnyomhatod, ha '@' jelet írsz a függvény elé. Ha a mysql_connect() függvényt kétszer ugyanazokkal a paraméterekkel hívod meg, akkor nem jön létre újabb kapcsolat; a függvény a már meglévő kapcsolat azonosítóját fogja vissza adni. A kapcsolat a PHP program végén bezárul, ha előbb nem zártuk volna le a mysql_close() függvénnyel.

mysql_select_db()

Kiválaszt egy MySQL adatbázis. Siker esetén TRUE értékkel tér vissza, ellenkező esetben FALSE értéket ad. A mysql_select_db() függvény az adott kapcsolat-azonosítójú szerverkapcsolat adatbázisát módosítja. Ha nincs kapcsolat-azonosító megadva, akkor az utoljára megnyitott kapcsolatban választ adatbázist. Ha ilyen sincs, akkor megpróbál a MySQL szerverhez kapcsolódni úgy, mintha a mysql_connect() függvény lett volna meghívva paraméterek nélkül.

mysql_query()

mysql_query -- MySQL kérést küld a szervernek

Leírás

resource mysql_query (string query [, resource link_identifier])

A mysql_query() függvény, kérést küld a megadott kapcsolat-azonosítójú szerver aktív adatbázisához. Ha nem adsz meg link_identifier-t, akkor a legutóbb megnyitott kapcsolatot használja a függvény. Ha nincs nyitva ilyen kapcsolat, akkor a függvény megpróbál nyitni egyet, mintha a mysql_connect() függvényt hívtuk volna paraméterek nélkül.A SELECT utasításra alkalmazott mysql_query() függvény eredményazonosítóval vagy FALSE-sal tér vissza a kérés végrehajtásától függően. Egyéb esetekben mysql_query() függvény TRUE-val (nemnulla) vagy FALSE-szal tér vissza, attól függően, hogy a kérés teljesítése sikeres volt-e. A TRUE visszatérési érték azt jelenti, hogy a kérés szintaktikailag helyes volt, és lefuttatta a szerver. Az érintett sorok számáról azonban nem mond semmit. Előfordulhat ugyanis, hogy a kérés sikeresen lefutott, de nem érintett egyetlen sort sem, vagy az eredményben egyetlen sor sincs.

mysql_fetch_array()

mysql_fetch_array -- Kérés egy sorát adja vissza (tetszőleges) tömb formájában.

Leírás

Az eredmény következő sorával tér vissza tömb formájában, vagy FALSE-sal, ha már nincs több sor.

A mysql_fetch_array() függvény a mysql_fetch_row() függvény kiterjesztett változata. Az eredményeket akár asszociatív tömbbe is írhatja aminek a tömb indexe egy név pl.: tomb[’password’]

Ha az eredmény több sorának ugyanaz a neve, akkor a később szereplő oszlop marad meg. Ha szeretnéd az összes mezőt elérni ilyenkor is, akkor számmal indexeld a tömböt kell használnod.

Jó tudni, hogy a mysql_fetch_array() függvény használata nem jelentősen lassabb a mysql_fetch_row() használatánál, de a kapott eredmény feldolgozása jóval kényelmesebb.

A mysql_fetch_array() függvény elhagyható result_type paramétere a következő lehet: MYSQL_ASSOC, MYSQL_NUM, vagy MYSQL_BOTH. Ez a lehetőség a PHP 3.0.7-es változatában került a nyelvbe. A paraméter alapértelmezett értéke a MYSQL_BOTH.

A MYSQL_BOTH használatával egy olyan tömböt kapsz, amelyben az elemek számmal és karakterlánccal is indexelve vannak. MYSQL_ASSOC értékkel használva csak asszociatív tömböt kapsz (mint a mysql_fetch_assoc() függvénnyel), MYSQL_NUM értékkel meghívva a fügvényt számozott indexű tömböt kapsz (mint a mysql_fetch_row() függvénnyel).

Fopen():

Az fopen fájl vagy url megnyitására használható.

Szintaktika:

erőforrás fopen (string $fájlnév , string $megnyitásmódja);

A fájlnév lehet URL vagy lokális fájl elérési útvonala.

PL.:

```
<?php
$eroforras = fopen("c:\\adat\\informacio.txt", "r");
?>
```

vagy

```
<?php
$eroforras = fopen("http://www.clans.hu/data.txt", "r");
?>
```

A megnyitás módja többféle lehet:

'r' Megnyitás csak olvasára; a fájl mutatót a fájl elejére viszi.

'r+' Megnyitás olvasára és írásra; a fájl mutatót a fájl elejére viszi.

'w' Megnyitás csak írásra; a fájl mutatót a fájl elejére viszi és a fájl hosszá lenullázza. Ha a fájl nem létezik, akkor létrehozza.

'w+' Megnyitás olvasára és írásra; a fájl mutatót a fájl elejére viszi és a fájl hosszá lenullázza. Ha a fájl nem létezik, akkor létrehozza.

'a' Megnyitás csak írásra; a fájl mutatót a fájl elejére viszi. Ha a fájl nem létezik, akkor létrehozza.

'a+' Megnyitás olvasára és írásra a fájl mutatót a fájl elejére viszi. Ha a fájl nem létezik, akkor létrehozza.

'x' Létrehozás és megnyitás csak olvasára; a fájl mutatót a fájl elejére viszi.ha a fájl már létezik akkor hibát fogunk kapni. Ha nem létezik, megpróbálja létrehozni.

'x+' Létrehozás és megnyitás, olvasásra és írásra, a fájl mutatót a fájl elejére viszi. Ha a fájl létezett, akkor az fopen hibát fog dobni ha nem akkor megpróbálja létrehozni.

A fopent gyakran használjuk az oldal kódjában fájlok megnyitására. A mi esetünkben ftp url adunk meg a kapcsolódásnál. Így a fájlt ftp keresztül fogja megnyitni

mysql_real_escape_string();

Ez a függvény levédi a speciális karaktereket az *unescaped_string* -ben, figyelembe véve az aktuális kapcsolat karakterkészletét. Így elkerülhetjük az oldal feltörését, sql ijection technikával való támadást. Biztonságos, ha mysql_query()-n belül használjuk. Ha bináris adatot akarunk beszúrni, akkor ezt a függvényt kell használnunk. Paraméterként megadhatjuk a levédeni kívánt stringet meg a mysql kapcsolat azonosítóját. Ha a kapcsolatazonosító nincs megadva, akkor az utólsó mysql_connect()-el megnyitott kapcsolatot használja. Ha nem talál semmilyen kapcsolatot, megpróbál létrehozni egyet úgy, mintha a mysql_connect() paraméterek nélkül lett volna meghívva. Ha esetleg semmilyen kapcsolatot nem talál és nem is sikerül létrehoznia, akkor egy E_WARNING szintű figyelmeztetés generálódik.

Visszatérési értékeke egy levédett string, hiba esetén pedig FALSE.

ftp_connect()

ftp kapcsolatot hoz létre.

pl.: $conn = ftp_connect($ftp_server)

ftp_login()

Bejelentkezik a nyitott ftp kapcsolatba.

pl.: ftp_login($conn,$ftp_user_name,$ftp_user_pass);

Az első változó a ftp kapcsolat erőforrása, második a felhasználói név, a harmadik a jelszó.

Siker esetén TRUE értékkel tér vissza, ellenkező esetben FALSE értéket ad. Ha a bejelentkezés sikertelen, akkor a PHP figyelmeztetést ad.

ftp_pasv()

Be és kikapcsolhatjuk a passív módot az adott ftp kapcsolatban.

Passzív módban adatkapcsolatokat az ügyfél kezdeményezi és ritkán a szerver. Erre szükség lehet, ha az ügyfél tűzfal mögött van.

opendir()

Megnyit egy könyvtárat

Egy könyvtárazonosítóval tér vissza, amit későbbi closedir(), readdir(), és rewinddir() hívásokban használhatsz.

Ha az útvonal nem egy érvényes könyvtárat ad meg, vagy a könyvtár nem megnyitható jogosultsági korlátozások, vagy filerendszer hibák miatt, az opendir() FALSE értéket ad vissza, és PHP hibát ad. Letilthatod az opendir() során fellépő hiba kiírását, ha egy `@' jelet teszel a függvény neve elé.

readdir()
Adott könyvtárból beolvas egy bejegyzést
A könyvtárban levő következő file nevével tér vissza. A fileneveket nem rendezetten adja vissza.
Pl.:
while ($fajl = readdir($handle)) {echo $fajl;}

closedir()
Bezár egy opendir() függvénnyel megnyitott mappát.

file_exists()
Megvizsgálj, hogy az adott fájl vagy könyvtár létezik-e.
Ha létezik a fájl vagy könyvtár, akkor TRUE értékkel tér vissza, ellenkező esetben FALSE értéket ad.

feof()
Ellenörzi a fájl mutatót hogy nem e „End of the file”.Tehát jelez, ha a fájlmutató a fájl végére ért. Ha a fájl végére ért a mutató, akkor TRUE értékkel tér vissza, ellenkező esetben FALSE értéket ad. Paraméternek egy létező kapcsolatot kell megadni, mint például egy fopen-el megnyitott fájlt függvény kimenetét.

fgets()
Beolvas egy sor az adot fájlból a fájlmutató helyzetétől kezdve. Paraméternek meg kel adni egy élő kapcsolatot egy fájlal, illetve megadhatjuk, hogy hány bitet olvasson ki a fájlból
Visszatérési értéke egy string, A fájl hosszuság – 1 bit méretet képes kiolvasni, egyébként FALSE értékel tér vissza.

ftp_delete()
Kitöröl egy megadott fájlt az Ftp szerverről.
Siker esetén TRUE értékkel tér vissza, ellenkezõ esetben FALSE értéket ad.

htmlspecialchars()
Ha valaki html kódokat fog írni a szövegmezőbe és azt elküldi, akkor az meg fog jelenni a böngészőben, így akár javascripteket is futtathatnak a lapon, enne kivédésére a htmlspecialchars() függvényt alkalmazzuk, mely átalakítja speciális karaktereket úgy, hogy azokat a böngésző nem tudja lefordítani, de számunkra úgy jelennek meg ahogy az beírtuk.

fwrite()
Ez a függvény egy fájlba írja a bemenetként megkapott string tartalmat adatfolyamként bytenként írja.
Például.: fwrite($target,$t);
$target a mi esetünkben az ideiglenes fájl fopenel megnyitott kapcsolata, A $t pedig egy string. Megadhatjuk még ezek mellet, hogy milyen hosszúságig írjon a fájlba. Ha sikeres az írás, akkor a beírt bytek számát adja vissza, ha sikertelen, akkor FALSE.

fclose()
Bezárja a paraméterként megkapott kapcsolatot. Visszatérési értéke, ha sikeres, akkor TRUE, ha sikertelen akkor FALSE.

isset()
Változó beállítottságának ellenőrzésére használjuk.
TRUE értéket ad vissza, ha var létezik, FALSE értéket ad egyébként.
Ha egy változót megszüntettél unset()-el, az isset() hamisat fog adni. Az isset() továbbá FALSE érétket fog adni, ha az ellenőrzött változó értéke NULL. Szintén fontos, hogy a NULL bájt ("\0") nem egyezik meg a PHP NULL konstans értékével.

ftp_put()
Feltölt egy megnyitott fájl az ftp szerverre
Paraméterek
ftp_stream A link, ami azonosítja a ftp kapcsolatott.
remote_file Távoli fájl elérési útvonala.
handle A megnyitott fájl, amit fel akarunk tölteni a fájl mutatótól. Fájl végénél abba haggya az olvasást
mode Az átviteli mód. FTP_ASCII vagy FTP_BINARY.
startpos Kezdő pozíció
Visszatérési értékek: Siker esetén TRUE értékkel tér vissza, ellenkező esetben FALSE értéket ad.

include()
Az include() beilleszti és feldolgozza a megadott fájlt.
Az alábbiak igazak a require()-ra is. A require() és az include() megegyezik egymással a hibakezelését leszámítva. Az include() figyelmeztetést generál, a require() viszont fatális hibát jelez. Magyarán, ahol az igényelt fájl nemlétekor a futást meg kell szakítani, ajánlott a require(). Az include() nem így viselkedik, a hibától függetlenül a szkript futtatása folytatódik.

IX. Összefoglalás

Az oldal megtekinthető az alábbi címen **http://www.kerecsengergo.clans.hu/szakdolgozat/** Ahol próba felhasználóként bárki beléphet a „user” felhasználói névvel és mellé a „user” jelszóval. Az oldal a clans.hu tárhelyén van elhelyezve, ahol a PHP 5.2.9 verziója, MySQL 5.0.67 verziója, és egy Apache/2.2.12 (Linux/SUSE) webkiszolgáló fut. Az oldal össze funkciója elérhető a próbafelhasználó számára, így láthatja működés közben is a rendszert. Megbizonyosodhat a funkciók gyakorlati hasznáról. Szakdolgozatom célja, hogy az oldal forráskódját bemutassam és érthető képet nyújtsak az oldal működéséről. Az oldal több funkciója is hasznosnak bizonyult a tesztelési időszak alatt. Bár ezek a funkciók külön-külön nem számítanak újdonságnak az interneten, de mégis újdonság lehet az, hogy ezek a funkciók együtt egy oldalba foglalva jelenek meg. A forráskód ismertetése mellet, a szakdolgozatom tartalmazza mind azon php függvények definícióját, amiket használtam a kódban és néhány rövid ismertetőt a php nyelvezetről, a munkamenetekről és a sütikről általában. Az oldalon fejlesztés közbe több biztonsági rész lett kijavítva, így növelve az oldal biztonságát. Remélem mások számára is hasznos és egyszerű rendszert készítettem, ahhoz, hogy ha ép nem a saját gépünknél ülünk, tudjuk azért biztonságosan kezelni saját adatainkat. Mindenkinek, aki érdeklődik a rendszer iránt, kívánom, hogy használja becsülettel, s ne alkalmazza illegális tartalom megosztására.

Irodalomjegyzék

[w] Wikipedia php definíciója
http://hu.wikipedia.org/wiki/PHP

[sql]Wikipedia sql definíciója
http://hu.wikipedia.org/wiki/SQL

[p]PHP hivatalos honlapja
http://www.php.net/

[pb]Php függvény gyűjtemény magyar nyelven
http://www.php-blog.hu/php-magyar-kezikonyv/

http://www.w3.org/
http://w3schools.com/php/default.asp
http://weblabor.hu/
http://sql.lap.hu/
http://flash-mp3-player.net/
http://www.thomas98.hu/webmuhely.php?sw=1024&lg=hu
http://www.php-blog.hu/php-magyar-kezikonyv/
http://flash-mp3-player.net/

Függelék:

1.ábra

2.ábra

3.ábra


Szakdolgozat
2009
Könyvjelzők
Képek
Zene
Jegyzettömb
Beállítások
Kilépés
Mobil Portál

4.ábra


Szakdolgozat
2009
Könyvjelzők
Képek
Zene
Jegyzettömb
Beállítások
Kilépés
Mobil Portál
Jegyzetfájlaid listája
jegyzet.txt
jegyzet2.txt
Beállítások mentése
Gombra kattintva elmentheted a jegyzetedet
Mentés

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani témavezetőmnek, Dr. Kuki Atillának, hogy megosztotta velem tapasztalatait, valamint útmutatást nyújtott dolgozatom megírásában. Ezen kívül köszönetet mondanék Szelkó Péternek a www.clans.hu üzemeltetőjének az általa biztosított tárhelyért.