

SZAKDOLGOZAT

Fogas Gábor

NYÍREGYHÁZA, 2013.



NYÍREGYHÁZI FŐISKOLA
TERMÉSZETTUDOMÁNYI ÉS INFORMATIKAI KAR
MATEMATIKA ÉS INFORMATIKA INTÉZET

Internetes v. Intranetes Társkereső Szerverprogram kivitelezése valamilyen közös érdeklődésre számot tartó témában

Konzulens:

Simon Béláné dr.
főiskolai tanár

Készítette:

Fogas Gábor
Programtervező Informatika, alapszakos
(BSc) hallgató

Nyíregyháza, 2013.

TARTALOMJEGYZÉK

1. BEVEZETÉS	1
2. TERVEZÉS.....	2
2.1 AZ OLDAL SZERKEZETE.....	3
2.2. ADATBÁZIS.....	5
2.3 FELHASZNÁLT JAVASCRIPTEK	10
2.4 BIZTONSÁGI INTÉZKEDÉSEK.....	10
3. FELHASZNÁLT TECHNOLÓGIÁK.....	12
3.1 SZERVEROLDALI PROGRAMOZÁS.....	12
3.1.1 A PHP TÖRTÉNETE.....	13
3.1.2 A PHP SZINTAKTIKÁJA	13
3.2 A HTTP PROTOKOLL RÖVID ISMERTETÉSE	14
3.3 A HTML RÖVID ISMERTETÉSE	14
3.4 A CSS RÖVID ISMERTETÉSE	15
3.5 A JAVASCRIPT RÖVID ISMERTETÉSE	16
3.6 SQL, MYSQL RÖVID ISMERTETÉSE.....	17
4. AZ OLDAL FUNKCIÓI.....	20
4.1 REGISZTRÁLÁS	20
4.2 BEJELENTKEZÉS.....	22
4.3 ÜZENETEK.....	23
4.3.1 ÚJ ÜZENET ÍRÁSA.....	23
4.3.2 ÜZENETEK OLVASÁSA ÉS TÖRLÉSE	25
4.4 KÉPEK.....	27

4.5 PROFIL.....	29
4.5.1 FELHASZNÁLÓ ADATAINAK MEGJELENÍTÉSE ÉS SZERKESZTÉSE...	29
4.5.2 PROFIL TÖRLÉSE.....	30
4.6 KEDVENCEK	31
4.7 TÁRSKERESŐK.....	31
4.8 KERESÉS	31
4.9 VENDÉGKÖNYV.....	32
4.10 ADMIN	32
4.11 KIJELENTKEZÉS.....	32
5. FELHASZNÁLÓI KÉZIKÖNYV	33
6. TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK	34
7. ÖSSZEGZÉS.....	35
8. KÖSZÖNETNYILVÁNÍTÁS.....	36
IRODALOMJEGYZÉK	37

1. Bevezetés

Az internet egyre nagyobb szerepet tölt be a mindennapi életben, egyre jobban az életünk részévé válik. A mai fiatalság nagy része rengeteg időt töltenek el számítógép előtt, ami nem feltétlenül jó dolog. Tanulmányaim során legjobban a webes programozás keltette fel az érdeklődésemet.

Témaválasztásom az internetes közösségek összehozására alkalmas weboldal elkészítése remek lehetőséget kínált arra, hogy tovább bővítsem tudásomat és tapasztalatokat gyűjtsek. A szakdolgozatom a társkeresésben és az ismerkedésben segít az embereknek. Egy olyan oldal elkészítését tűztem ki célul, ami letisztult, biztonságos és egyszerűvé teszi a kapcsolatfelvételt a másik féllel.

Nagyon sok társkereső oldal szolgált példaként. Az elkészítés során próbáltam ötvözni ezeket és pótolni a hiányosságait.

A szakdolgozatom során bemutatom, hogyan terveztem meg a weboldalt és a hozzá tartozó adatbázist, amiben az adatokat tárolom. Röviden ismertetem az elkészítés során felhasznált technológiákat, így azok is informálódhatnak - betekintést nyerhetnek a programozás világába -, akik nem szakmabeliek. Bemutatom az oldal fő funkcióit, képekkel és ábrákkal szemléltetve, valamint megosztom milyen tapasztalatokat gyűjtöttem az elkészítés során.

2. Tervezés

A tervezés során az elsődleges szempont az átláthatóság és letisztultság volt. Ezeket a szempontokat vettem alapul az oldal dizájnjának elkészítése során is. Ne legyenek fölösleges menüpontok, könnyen kezelhető legyen, ezzel megcélózva az idősebb korosztályt, akik kevésbé jártasak az informatika világában.

Az oldalt eredetileg úgy terveztem, hogy csak regisztrálás és bejelentkezés után váljanak elérhetővé a különböző funkciók, amíg ez nem történik meg, csak egy bejelentkező képernyőt látott volna a felhasználó. Ezt az ötletemet elég korán elvetettem, mert fontos tudatni a felhasználókkal, hogy milyen plusz dolgokkal rendelkezik a közösségi oldal a konkurens oldalakhoz képest, valamint hányan regisztráltak, mert a sok felhasználó még több felhasználót vonz. Felhasználók nélkül, az oldal nem életképes.

Egyre fontosabb szerepet tölt be a mobiltelefonok és a táblagépek használata, sokan már csak ezekről az eszközökről interneteznek, ezért is volt lényeges, hogy az oldal könnyen kezelhető legyen, kisebb érintő kijelzőkön is egyszerűen és gyorsan elérhetőek legyenek a funkciók.

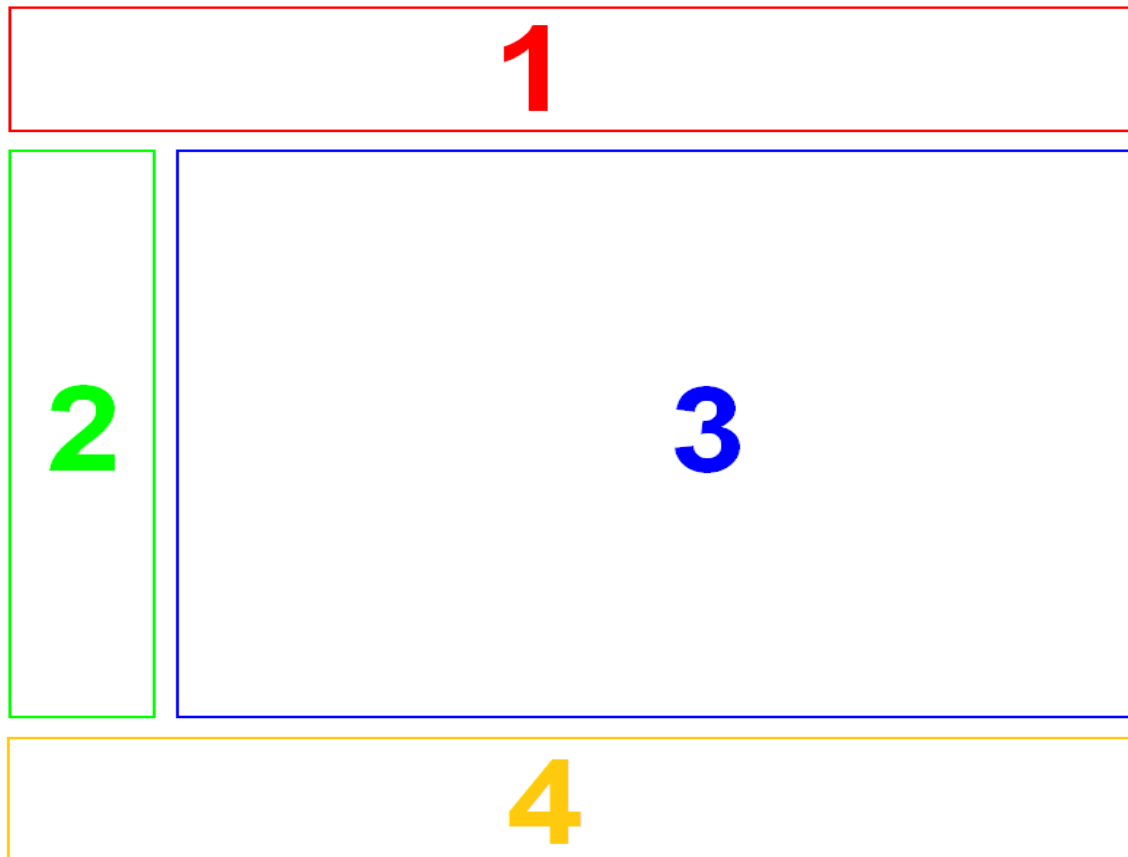
A társkeresés, ismerkedés az oldal fő funkciója, ezért sok munkát fektettem az üzenetküldés megvalósítására. Próbáltam egyszerűvé, kényelmessé, de jól használhatóvá tenni. Lényeges szempont volt, hogy ha valakivel már felvettük a kapcsolatot, akkor az a partner a későbbiek során is könnyen elérhető legyen. Ezért találtam ki a kedvencek menüpontot, ahol a szimpatikus felhasználókat lehet egy listába gyűjteni és egyszerűen elérni. Törekedtem az esetleges és a tipikus hibák kiszűrésére. A dolgozatom folyamán be is fogom mutatni hogyan valósítottam meg.

A tervezés folyamata sok időt vett igénybe, hogy mégis milyen legyen az oldal kinézete, milyen menüpontok legyenek, hogyan legyenek elhelyezve és hol húzom meg a határt a fejlesztés során. Nagyon sok verzió készült el az oldal megalkotása során, számos dolog megváltozott a kezdetekhez képest, amíg az oldal el nem nyerte végleges formáját. A jól bevált dolgok mellett próbáltam egyedi ötleteket is megvalósítani, amelyek szerintem fontosak egy ilyen típusú oldalnál. A regisztrálás során is sok kérdés vetődött fel bennem. Például, hogy milyen információkat kérjek be a felhasználóról, amelyek szükségesek és segíthetik az ismerkedést. Úgy gondoltam a vallásról és alkoholfogyasztásról nem kérek be

semmilyen információt, mert ez sértő lehet. A telefonszámok megadását azért mellőztem, hogy a felhasználók elsődlegesen az oldalon keresztül kommunikáljanak egymással.

2.1 Az oldal szerkezete

Az oldal szerkezete négy fő részből épül fel. Tartalmuk dinamikusan változik.



1. ábra - Oldal szerkezet

1. Fejléc: tartalmazza az oldal logóját, valamint egy felső menüsort melynek tartalma dinamikusan változik, bejelentkezett és kijelentkezett állapottól függően.

A fejlécben az alábbi menüpontok találhatóak.

- **Hírek:** Ez tartalmazhat cikkeket, híreket, amit csak az Admin tud posztolni a főoldalra. Ez a menüpont sohasem tűnik el.
- **Bejelentkezés:** A már regisztrált felhasználók itt tudnak bejelentkezni az oldalra, illetve a már regisztrált felhasználók, akik elfelejtették a jelszavukat, kérhetnek új jelszót. Ez utóbbi funkció csak kijelentkezett állapotban érhető el.

- **Regisztrálás:** Lehetőség van regisztrációra, amivel az oldalt teljes jogúan lehet használni. Ez csak kijelentkezett állapotban elérhető.
- **Vendégkönyv:** Fő szerepe a felhasználók által írt vélemények megjelenítése. Továbbá hasznos linkek, információk és statisztikák (regisztrált és online felhasználók számának) megjelenítése. Ez a funkció kijelentkezett állapotban is elérhető.
- **Társskeresők:** Ez az oldal fő funkciója. Itt lehet új barátokra, ismerősökre szert tenni. Megtekinteni a már regisztrált felhasználók adatait, lehetőség van üzenetküldésre a kiválasztott személynek és kedvencnek jelölni. Ha valaki szimpatikussá válik, később gyorsabban elérhető lesz a kedvencek menüpontból. Csak bejelentkezett állapotban érhető el.
- **Keresés:** Lehetőség nyílik már regisztrált felhasználók keresésére. Kereshetünk felhasználónév, vezetéknev, keresztnév alapján. Találat esetén megjelennek azok a személyek, akikre a keresési feltétel illik. A találati listából elérhető az üzenetküldés, kedvencnek jelölés, illetve az adatlap is megtekinthető. Csak bejelentkezett állapotban érhető el.
- **Admin:** Ez csak az oldal adminisztrátora számára elérhető. Itt tud híreket posztolni, regisztrált felhasználók egyes adatait megtekinteni valamint szabálytalanság esetén a felhasználót törölni.

2. Baloldali menüsáv: Tartalma dinamikusan változik, attól függően, hogy bejelentkezett-e a felhasználó.

- **Profilkép:** 128x128 pixel méretű kép, melyet a felhasználó maga állít be vagy tölt fel. Ha nem tölt fel képet, akkor egy alapértelmezett kép jelenik meg, nemtől függően.
- **Üdvözlés és a felhasználó nevének kiírása,** alatta pedig az utolsó bejelentkezésnek a dátuma látható.
- **A felhasználó saját menüje,** amely öt részből áll.
 - **Üzenetek:** Itt lehet megnézni és törölni a beérkezett és elküldött üzeneteket, valamint üzenetet küldeni más felhasználónak.
 - **Képek:** A felhasználó által feltöltött képek megtekintése, képek feltöltése, profilkép kiválasztása.
 - **Profil:** A saját adatok megtekintése mellett lehetséges azok módosítása, kibővítése is pár információval. Például: idézet, magamról mező, mellyel

pár sorral még érdekesebbé tehető az adatlap. Itt lehet a profilt törölni a jelenlegi jelszó megadása után. A törlés nem visszavonható és azzal minden adat törlődik (beleértve a képeket és az üzeneteket is). Ezen az oldalon lehet a jelszót módosítani is.

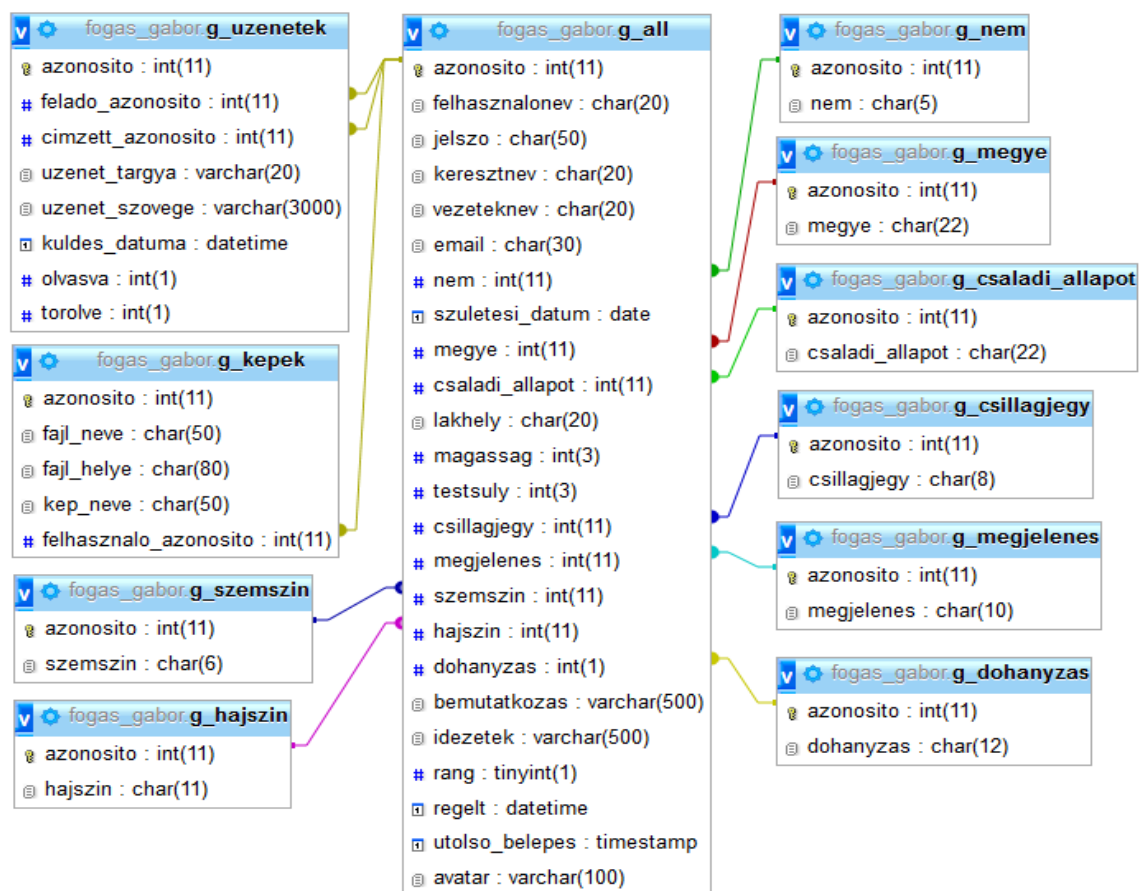
- **Kedvencek:** A felhasználó kedvencnek jelölt partnereit lehet itt megtekinteni, üzenetet írni és esetleg eltávolítani a listáról.
- **Kijelentkezés:** A saját profil elhagyása, kijelentkezés után visszakerül a kezdő oldalra.

3. Tartalom: Itt jelenik meg a tényleges tartalom.

4. Lábléc: Itt jelenik meg a készítés éve, a készítő neve valamint az oldal adatvédelmi nyilatkozata.

2.2. Adatbázis

Az oldal tökéletes működéséhez tizenöt táblára volt szükségem.



4. ábra - Tábla kapcsolatok

A `g_all` tábla a legfontosabb, ebben vannak eltárolva a felhasználó adatai. Huszonnégy mezőt tartalmaz, amelyek az alábbiak:

- **azonosito** - minden felhasználó saját azonosítóval rendelkezik, ez a mező egyben a kulcs is. Értéke nem lehet NULL és automatikusan növekszik. (A NULL egy olyan speciális érték, melynek a jelentése az, hogy nincs érték.)
- **felhasznalonev** – Ez a mező tartalmazza a felhasználó nevét, amit a regisztrálás során ad meg, maximum 20 karakter hosszúságú lehet.
- **jelszo** – Jelszót tartalmazó mező maximum 32 karakter lehet, ez a jelszó titkosítása (kódolása) miatt lett ennyi.
- **keresztnev** – Keresztnevet tartalmazó mező maximum 20 karakter lehet.
- **vezeteknev** – Vezetéknévet tartalmazó mező maximum 20 karakter lehet.
- **email** – E-mail címet tartalmazó mező maximum 30 karakter lehet.
- **nem** – Ez a mező a nemet tartalmazza, csak kettő értéket vehet fel 1 és 2, NULL nem lehet. A `g_nem` táblával van JOIN-olva, amelyek tartalmazzák az 1 és 2 értékeit, Férfi illetve Nő.
- **szulesi_datum** – Születési dátumot tartalmazó mező, `date` típusú (év, hónap, nap)
- **lakhely** – Lakhelyet tartalmazó mező maximum 20 karakter lehet.
- **magassag** – Magasságot tartalmazó mező csak számot tartalmazhat.
- **testsuly** – Testsúlyt tartalmazó mező csak számot tartalmazhat.
- **bemutakozas** – A gyorsabb társkeresés, figyelemfelkeltés érdekében a felhasználó írhat magáról egy kis bemutatkozó szöveget, ez a mező maximum 500 karakter lehet.
- **idezetek** – A felhasználó, ha szeretne idézetet is megadhat, amely az adatlapján jelenik meg, maximum 500 karakter lehet.
- **rang** – Ez a mező jelöli a felhasználó rangját, alapértelmezetten 0 azaz sima felhasználó, 1-es értékkel admin jogosultságot kaphat a felhasználó.
- **regelt** – A regisztrálás időpontja tartalmazza ez a mezőben `DATETIME` típusú (év, hónap, nap, óra, perc, másodpercet tartalmaz.)
- **utolso_belepes** – Ez a mező tartalmazza, hogy a felhasználó mikor volt utoljára bejelentkezve, ami kijelentkezéskor íródik felül. `TIMESTAMP` típusú (év, hónap, nap, óra, perc, másodpercet tartalmaz.)

- **avatar** – Ez a mező a kiválasztott profilképnek a helyét tárolja, alapértelmezetten üres, nemtől függően változik az alapértelmezett profilkép.
- **megye** – A `g_megyek` táblával van kapcsolatban ami a 19 megyénk nevét tartalmazza, ebben a mezőben csak 1 és 19 közötti érték lehet, NULL nem. Minden szám (azonosító) egy megyének a nevét tartalmazza. A megyék sorrendjét először a lakosság száma szerint csökkenő sorrendben akartam felsorolni, de végül az ABC sorrend mellett maradtam.

```
function getMegyek() {
    $resp="";
    $query="SELECT azonosito,megye FROM g_megye";
    $result = mysql_query($query) or die('Hibaüzenet!!!' .
mysql_error());
    while ($sor = mysql_fetch_assoc($result)) {
        $resp.="<option value=\"\" . $sor["azonosito"] .
"\">>".$sor["megye"]."</option>";
    }
    return $resp;
}
```

A kódrészlet segítségével történik a megyék nevének kiválasztása, hasonló jellegű tábláknál is ezt az elvet alkalmaztam.

A további adatok megadásánál (megye, szemszín, hajszín, családi állapot, csillagjegy, megjelenés, dohányzás) egy-egy legördülő listából lehet kiválasztani a különféle kategóriákat: szemszín, hajszín, stb. A legördülő lista előnye, hogy csak fix értékek közül lehet választani.

Az értékek a kapcsolótáblából jönnek, aminek az általános szerkezete: `id`, - tulajdonság azonosító (szám típus), leírás: szöveg (ez a rész tartalmazza a kategóriának megfelelő tulajdonságát, kék szemszín vagy például azt, hogy az ikrek csillagjegyében született).

Előnye, hogy ha változtatni, bővíteni kell ezeket a listákat, akkor elég egyetlen egy helyen átírni az adatbázisban, és nem kell a fájlokat módosítani, ezért is választottam ezt a megoldást, gondolva a majd esedékes fejlesztésekre.

Az oldal további funkcióihoz szükségem volt még mellék táblákra is ezek a táblák nincsenek kapcsolatban egyik táblával sem kivétel a `g_uzenetek` tábla.

fogas_gabor.g_kedvencek
azonosito : int(11)
saját_azonosito : int(11)
kedvenc_azonosito : int(11)

fogas_gabor.g_online_felhasznalok
azonosito : int(11)
ip : char(30)
timer : char(100)
datum : datetime

fogas_gabor.g_vendegkonyv
azonosito : int(11)
vendegkonyv_cim : char(30)
vendegkonyv_szoveg : char(255)
datum : datetime

fogas_gabor.g_hirek
azonosito : int(11)
hir_cime : varchar(50)
hir_szoveg : varchar(2000)
datum : datetime

5. ábra - Táblák

A `g_hirek` tábla: tartalmazza a főoldalon látható híreket. Négy mezőből áll. Az első mező a kulcs mező, ami az azonosító. A hír címe mező: egy szöveges és maximum 50 karakter hosszúságú szöveg adható meg. A hír szövege mező: maximum 2000 karakter lehet és itt kerülnek eltárolásra a hír szövege, végül pedig egy dátum mezővel rendelkezik a tábla ahol az aktuális posztolás dátuma található meg. Más táblával nem áll kapcsolatban.

A `g_kedvencek` tábla: tartalmazza a felhasználók kedvenceit, azaz kiket jelöltek kedvencnek. Rendelkezik egy kulcs mezővel, ami az azonosító. Saját azonosító mezővel ahol a bejelentkezett felhasználó azonosítója szerepel majd. Kedvenc azonosító, ebbe a mezőbe kerülnek a felhasználó által kedvencnek jelölt partnerek azonosítója.

A `g_kepek` tábla: tartalmazza a képgaléria megjelenítéséhez szükséges adatokat, legfőbbképpen azt, hogy a feltöltött képek merre találhatóak. A tábla rendelkezik egy kulcs mezővel, ami az azonosító lesz. Fájl neve mező: ide kerülnek a feltöltött képfájlok nevei, amik már átvannak nevezve, erről későbbiek során foglalkozom hogyan és miért. A fájl helye mező: itt kerülnek eltárolásra a képek elérési útja ilyen formában felhasználónév/képneve.kiterjesztés. Így könnyedén megtudom jeleníteni a képeket a galériában a script segítségével. A kép neve mező, ennek kitöltése nem kötelező,

értelmszerűen a képek neve kerül eltárolásra, ha a feltöltés során megadta a felhasználó a kép nevét, akkor a későbbiek során ez a név látható a képgalériában. Végül található egy felhasználó azonosító mező: ami arra szolgál, hogy meg lehessen különböztetni ki töltötte fel az adott képet.

A `g_vendegkonyv` tábla: tartalmazza a látogatók/regisztrált felhasználók által beírt véleményeket, észrevételeket. Négy mezőt tartalmaz a tábla. Az első egy kulcsmező, ami itt is az azonosító nevet kapta. Vendégekönyv címe: itt található a vélemény címe ami maximum 30 karakter lehet. Vendégekönyv szövege: maximum 255 karakter lehet a véleményt tartalmazó szöveg. Dátum mező: a vélemény beküldésének időpontja található itt meg.

A `g_online_felhasznalok` tábla: tartalmazza az online regisztrált felhasználók számát. A tábla rendelkezik egy azonosító mezővel, ami a kulcs is egyben. IP mező: ide kerülnek az IP címek. Timer mező: az idő eltárolása kerül ide a `time()` függvény segítségével. Dátum mező: az oldal megtekintésének idejét tárolom itt.

A `g_uzenetek` tábla: tartalmazza a felhasználók leveleit. Azonosító mező: ez egyben a kulcs mező is. Feladó azonosító: ide kerül az éppen bejelentkezett felhasználó azonosítója. Címzett azonosító: ide kerül a annak a felhasználónak az azonosítója akinek a levél címezve van, ez a két mező kapcsolatban van a `g_all` táblával így csak azokkal az értékkel rendelkezhetnek amik a `g_all` táblában megtalálható, azaz regisztrált felhasználó. Üzenet tárgya: a levél tárgyát tartalmazó mező, ami maximum 20 karakter lehet. Üzenet szövege mező: ez maximum 3000 karakter lehet, itt kerül eltárolásra az üzenet szövege. Küldés dátuma: az üzenet küldésének dátuma található meg itt. Év, hónap, nap, óra, perc, másodperc pontossággal. Olvasva mező: alapértelmezetten 0 az értéket ez azt jelenti, hogy az üzenet még nem volt olvasva, ha az adott felhasználó megnyitotta olvasásra, akkor az értéke 1-re változik. Törölve mező: tartalmazza, hogy ki az a felhasználó aki már törölte az üzenetet, alapértelmezetten 0. Ha egy adott felhasználó törölte akkor az ő azonosítója kerül a mezőbe.

2.3 Felhasznált JavaScriptek

Az oldal elkészítéséhez felhasználtam az ingyenes ¹prettyPhoto nevű képnézegető scriptet. Azért esett a választásom erre, mert támogatja a jquery-1.9.1-et és a népszerűbb böngészőket is (Firefox 3.0+ Google Chrome 10.0+ Internet Explorer 6.0+ Safari 3.1.1+ Opera 9+). Egyszerűen testre szabható, a nem kívánt funkciók kiszedhetőek. Az oldalba való beépítése is nagyon egyszerű, tartalmaz automatikus képméretezést, így a bélyeg képek mérete meghatározható.

Az alábbi kódrész segítségével ágyaztam be az oldalba.

```
<a href="<?php echo $data['fajl_helye'] ?>" target="_blank"
rel="prettyPhoto[galery]">"
```

Az üzenetek küldéséhez, a vendégkönyvhöz, az admin oldal hírek szerkesztéséhez a ²CKEditor nevezetű scriptet használtam, ami nagyon hasznos lehet a szöveges üzenetek szerkesztéséhez, nyílt forrású, ingyenesen letölthető és teljes mértékben személyre szabható, nagyon sok funkciót tartalmaz és sok extrát.

Az alábbi módon ágyaztam az oldalba:

```
<textarea name="uzenet_szovege" class="ckeditor" id="editor1"
rows="15" cols="80"></textarea>
```

2.4 Biztonsági intézkedések

A bejelentkezés nélküli nem elérhető oldalak védelmében egy egyszerű logika feltétel megadásával védekezem, ami két függvénybe van megírva. Az első függvény megnézi, hogy be van-e jelentkezve a felhasználó, két értékkel térhet vissza, ha üres a `$_SESSION['felhasznalonev']` akkor a felhasználó nem jelentkezett be, mert ebbe a változóba kerül a bejelentkezett felhasználó neve, ha pedig nem üres, akkor tartalmaz adatot, tehát be van jelentkezve.

¹ <http://www.no-margin-for-errors.com/projects/prettyphoto-jquery-lightbox-clone/>

² <http://ckeditor.com/>

```
function isLoggedIn() {
    return isset($_SESSION['felhasznalonev']) &&
!empty($_SESSION['felhasznalonev']);
}
```

A második függvény, amire szükségem volt szintén nagyon egyszerű.

```
function atiranyitas($atdob) {
    if($atdob) {
        header("Location: index.php");
        exit;
    }
}
```

Egyszerű a funkciója, átirányít a kezdő oldalra. Minden oldal tartalmazza ezt a kódrészletet.

```
atiranyitas(!isLoggedIn());
```

Ha nincs bejelentkezve, akkor átirányít a kezdőlapra.

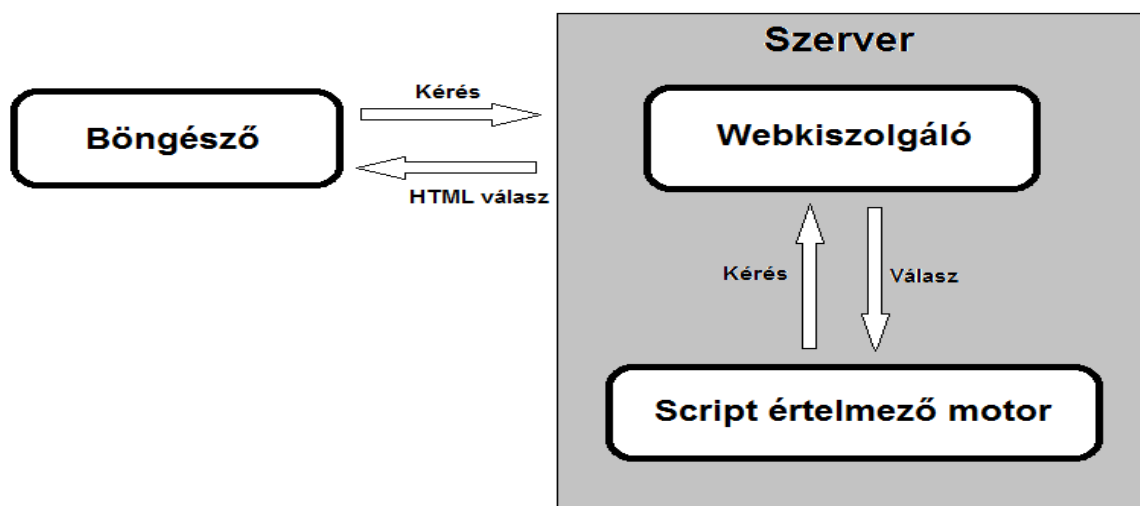
3. Felhasznált technológiák

Az elkészítéséhez sokféle technológiát kellett alkalmaznom, ami működőképessé és szebbé teszi a weboldalt. Szükségesnek éreztem a CSS használatát az oldal dizájnjának elkészítéséhez. PHP-t az oldal funkcionak elkészítéséhez. MySQL-t az adatok tárolásához, JavaScriptet, hogy az oldal könnyebben kezelhető legyen. A HTML4-et választottam a HTML5 helyett, mert az nem elég kiforrott és még nincs véglegesítve, jelenlegi formájában még csak egy ajánlás a w3c konzorciumtól. Ezzel szemben a 4.01-es 1999-ben jelent meg és a mai napig ez a verzió a legelterjedtebb és széles körűen támogatott.

3.1 Szerveroldali programozás

A PHP (PHP Hypertext Preprocessor) egy olyan szabad szerver oldali programozási nyelv, amivel dinamikus weblapok készíthetők. Nagy előnye hogy képes adatbázis kezelésre, kódolásra, adatfeldolgozásra, fájlkezelésre, kapcsolatok létrehozására, űrlapok kezelésére. Nagy mennyiségű képek listázására kitűnő segítség.

A program futásához szerver szükséges, fejlesztési fázisban célszerű a saját gépünkre telepíteni egy Apache webservert (én a XAMPP programcsomagot választottam) ami majd az általunk megírt PHP kódokat fogja értelmezni, lefordítani. Az alábbi ábrán látható hogyan kommunikál a böngészőnk a webszerverrel.



2. ábra - Dinamikus HTML oldal

A böngésző elküldi a kérést, kódot a szerver felé és a scriptet értelmező motor feldolgozza az adatokat és válaszul a már lefordított HTML kódot kapjuk. A saját kódunkat sosem fogjuk látni, mert a szerver már a lefordított kódot küldi, adja vissza eredményül. Az oldal

forrásában a PHP kód már nem látható. Ezeknek az oldalaknak a tartalma dinamikusan változik, mert vagy paraméterekhez, felhasználókhöz fog igazodni.

3.1.1 A PHP története

³„A PHP-t Rasmus Lerdorf agyalta ki valamikor 1994 őszén. Az első kiadatlan verziókat a saját honlapján használta, hogy figyelemmel kísérje, kik látogatják az oldalait. Az első mások által is használt változat 1995 elején látott napvilágot és "Personal Home Page Tools" néven volt ismert. Egy nagyon egyszerű feldolgozó programból állt, ami csak néhány speciális makrót értett meg, valamint tartalmazott számos eszközt, amiket akkoriban gyakran használtak a honlapokon (számláló, vendégkönyv és hasonlók). A feldolgozó program újraírása után 1995 közepén a "PHP/FI 2. verzió" nevet kapta. A FI Rasmus egy másik szoftveréből került bele, amit HTML űrlap információk feldolgozására készített. Ötvözte a "Personal Home Page Tools" programját a "Form Interpreter"-el és mSQL támogatást adott hozzá. Így született a PHP/FI. A program bámulatos ütemben fejlődött, és az emberek elkezdtek kódokkal segíteni a fejlesztést.

Nehéz lenne pontos adatokat megadni, de a PHP/FI-t körülbelül 15.000 webhelyen használták 1996 végére világszerte. 1997 közepére ez a szám 50.000 fölé nőtt. 1997 közepe nagy változást jelentett a PHP fejlesztésében. Rasmus saját kis projektjéből egy sokkal jobban szervezett csapatmunka lett. A feldolgozó programot Zeev Suraski és Andi Gutmans teljesen újraírták, és ez alkotta a lelkét az új PHP 3-nak. Sok kódot sikerült átvenni a PHP/FI-ből, másokat pedig teljesen újra kellett írni.

A legfrissebb változat (a PHP 4) a Zend Scripting Engine használatával nagyobb teljesítményt nyújt, még több külső könyvtárat és kiterjesztést támogat, és modulként képes futni minden népszerű webkiszolgálón.”

3.1.2 A PHP szintaktikája

```
<?php print("Helló Világ!"); ?>
```

Az alábbi kódrészletben a PHP szintaktikája látható. A mára már klasszikus „Helló Világ” szöveg kiírása. Megfigyelhető, hogy a kódrész egy záró és egy nyitó tag közé kerül, jelen

³ <http://www.kando-kkt.sulinet.hu/mirror/phphu/intro-history.html>

esetben a `<?php` ez lesz a nyitó tag, egyes esetekben elhagyható a `php` szó, de akkor XML-el nem fog megfelelően működni a kódrészlet, ezért ajánlott mindig kiírni. A záró tag pedig a `?>` végződik. A szintaktika nagyon hasonlít a C nyelvhez, minden utasítás pontosvesszővel végződik. Egy vagy több utasítást blokkokba lehet csoportosítani, ezek kapcsos zárójelek közé kerülnek. A PHP segítségével dinamikus HTML oldalakat tudunk generálni. A változó `$` jellel kezdődik és betűvel folytatódik, majd betűvel és számmal folytatódhat. Kis és nagybetűk között különbséget tesz, valamint a `_` jel is betűnek számít. A változó típusát a felvett érték határozza meg.

3.2 A http protokoll rövid ismertetése

⁴„A http protokoll a Hyper Text Transfer Protocol betűszavai, magyarul hiperszöveg átviteli protokoll. A protokoll meghatározza milyen szabályok szerint kommunikál a weblapot kiszolgáló gép a böngészőprogramunkkal. Kérés-válasz protokollnak is hívják. A böngésző program kér egy weboldalt a kiszolgáló számítógép pedig el küldi azt. A http-nek van egy biztonságos változata https. Ebben az esetben a weboldalak az ún. SSL vagy TLS protokollon keresztül titkosított formában kerülnek átvitelre. A https-t a Netscape böngészőgyártó fejlesztette ki olyan weboldalakhoz ahol fokozottan szükséges a biztonság, mint a fizetési átutalások, stb.”

3.3 A HTML rövid ismertetése

⁵„A HTML angol mozaikszó, jelentése HyperText Markup Language, vagyis hiperszöveges jelölőnyelv. A weboldalak leíró jelnyelv, vagy az a kódrendszer, amit a weboldalak elkészítéséhez használnak.

A HTML nem programozási nyelv, hanem kódnyelv, aminek segítségével akár a Jegyzettömb segítségével is alkothatóak weboldalak. A hipertext jelenti valójában az interneten található oldalakat (dokumentumokat), amelyek szöveg, kép, videó, hang, animáció, vagy ezek valamilyen kombinációjából áll. A HTML ezeknek a dokumentumoknak az elrendezését, formázását tartalmazza saját jelölőnyelvén.”

⁴ http://informatika.gtportal.eu/index.php?f0=web_elm_113

⁵ <http://www.matebalazs.hu/html.html>

A HTML szerkezetének felépítése, ami XML (Extensible Markup Language) alapú ezért is jelölő nyelv. Az elemnevek, amik a reláció jelben találhatóak, érzékenyek a kis és nagybetűk közötti különbségre. A dokumentum felépítése `<html>` tag-el kezdődik és `</html>` tag-el záródik. Nem minden elemnek van záró eleme, de ha kiírjuk az nem hiba, és úgy az oldalunk XHTML kompatibilissé tehető. (Az XHTML-ben mindig van egy záró és egy nyitó tag, a záró tagok mindig per jelel kezdődnek.) Az XHTML oldal olyan HTML kódot tartalmaz, amely egy érvényes XML dokumentumnak felel meg.

Az alábbi kódrészlet az általános HTML oldalt mutatja be leegyszerűsítve.

```
<html>
  <head>
    <title>A weboldal címe</title>
  </head>
  <body>
    A dokumentum tartalma
  </body>
</html>
```

3.4 A CSS rövid ismertetése

⁶„A CSS (angolul Cascading Style Sheets) a számítástechnikában egy stílusleíró nyelv, mely a HTML vagy XHTML típusú strukturált dokumentumok megjelenését írja le. Ezen kívül használható bármilyen XML alapú dokumentum stílusának leírására is, mint például az SVG, XUL stb. A HTML kevés lehetőséget az oldalak formázására, tulajdonságaik beállítására.

Ezért bevezették a CSS-t (cascading style sheets - egymásba ágyazott stíluslapok), amely a HTML-t hivatott kiegészíteni.

A CSS nagymértékben megkönnyíti, és gyorsabbá teszi munkánkat, mivel elég csak az oldalainkhoz tartozó stíluslapot módosítani.

Míg HTML-ben például egy száz soros szöveg minden mondatának kezdőbetűjét zöldre akarjuk színezn, akkor az összes ilyen helyre be kellene írunk a megfelelő kódot, ami elég sokáig tartana. Ugyanezt CSS-ben elérhetjük pár másodperc alatt.”

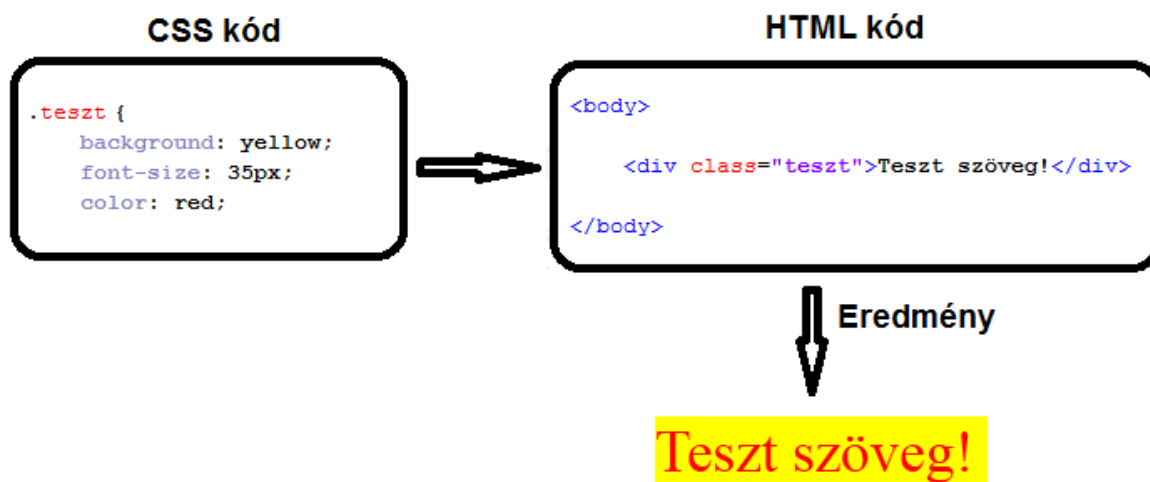
⁶ <http://www.hotdog.hu/css-kodok/mindenhez/mi-a-css>

Célszerű külön fájlba tenni a CSS-t és utólag importni a fájlt. A HTML fájl head részébe célszerű elhelyezni az alábbi kódot.

```
<link rel="stylesheet" href="css/main.css" type="text/css" media="all" />
```

Én az előbbi megoldást alkalmaztam, de természetesen másféle módokon is beágyazható, például így:

```
<style type="text/css"> tartalom </style>
```



3. ábra - CSS szintaktika

A fent látható példán egy egyszerű CSS osztály használata látható. Három tulajdonságot kapott:

- a háttér színe legyen sárga,
- a betűmérete legyen 35 pixel,
- és a szöveg színe pedig legyen piros.

3.5 A JavaScript rövid ismertetése

⁷„A JavaScript, mint neve is mutatja, egy script nyelv, amit legelőször a Netscape Navigator 2.0 támogatott. A JavaScript-eket HTML lapokba lehet beágyazni, a lappal együtt töltődnek le, majd a böngésző értelmezi és futtatja azokat.

⁷ <http://www.nejanet.hu/~ksanyi/java/JAVAS.HTM>

- A kliens értelmezi (interpretálja)
- Objektum-alapú. Beépített, kiterjeszthető objektumokat használ, de nincsenek osztályok és öröklődés.
- A program beágyazódik a HTML dokumentumba.
- A változók típusát nem kell deklarálni. (Nem típusos nyelv.)
- Dinamikus kötés. Az objektum-hivatkozások ellenőrzése futásidőben történik.
- Nem tud önállóan a merevlemezre írni.

A JavaScript-et azért fejlesztették ki, hogy az egyébként statikus lapokba dinamizmust vigyenek. Egyrészt látványosabbá tehetjük vele azokat (dinamikusan, futásidőben generálhatunk HTML lapokat), másrészt számos praktikus dologra használhatjuk fel (pl. jelszót kérhetünk a lapunk megtekintéséhez, vagy pl. ellenőrizhetjük az elküldendő adatok helyességét még a kliens oldalon - hálózati erőforrásokat spórolva meg, stb.).”

3.6 SQL, MySQL rövid ismertetése

⁸„Az SQL egy szabványosított strukturált lekérdező nyelv (Structured Query Language) rövidítése, melyet az IBM dolgozott ki és több relációs adatbáziskezelő ismer, különböző operációs rendszeri környezetben.

Az SQL egy szabványosított lekérdező nyelv, melyet Ennek óriási jelentősége van az adatbázis alkalmazások fejlesztőinek körében, mert így az alkalmazások a különböző operációs rendszerek és adatbáziskezelők között módosítás nélkül vagy csekély módosítással átvihetők.

Az SQL nem algoritmikus nyelv, nem tartalmaz algoritmus szerkezeteket (elágazás, ciklus stb.). Az SQL halmaz orientált nyelv, mely a relációkon dolgozik. A halmaz orientáltság azt jelenti, hogy nem kell definiálni a művelet végrehajtásának lépéseit, hanem a feladat nem eljárás szerű megfogalmazását kell megadni, melyek a reláció vagy relációk kiválasztott sorain hajtnak végre. A művelet végrehajtásához optimális megoldás megtalálása a nyelvi processzor feladata, nem a programozóé. Például annak eldöntése,

⁸ <http://www.agt.bme.hu/szakm/adatb/db5.htm>

hogy egy adott visszakeresésben alkalmazhatók-e indexek, vannak-e indexek vagy építsen-e fel új indexet, a nyelvi processzor feladata. Az SQL nem rekurzív nyelv.”

⁹„MySQL a világ egyik legnépszerűbb, nyílt forráskódú adatbázis szervere. Több platformon, párhuzamosan zajlik a fejlesztése. Olcsó, hatékony alternatívát jelent mind a programfejlesztők, mind a rendszergazdák számára.

A termék mindenki számára elérhető verziói 100%-ig a GPL (Gnu General Public License) licence alá esnek, ami annyit jelent, hogy ingyenesen használhatók és terjeszthetők. A termék letölthető az Internetről, számos tüköroldal áll rendelkezésre, köztük magyarországi is. Vannak olyan kereskedelembe szánt verziók, melyekért fizetni kell. A csatlakozó kliensek száma itt sincs korlátozva, de maga a szerver nem esik GPL alá (non-GPL).

Néhány jellemző címszavakban:

- gyorsaság - több díjat nyert az adatkezelési sebességével
- többszálú
- megbízható
- több felhasználós
- könnyű használat
- állíthatók a hozzáférési jogosultságok
- elterjedt (főleg Unix/Linux körökben)
- nagy teljesítményű
- nagy adatbázisméret mellett is megállja a helyét (eredetileg ilyen feltételek használatára fejlesztették)
- relációs adatbáziskezelő
- magyar nyelvű hibaüzenetek
- TCP/IP protokollt használ
- 100%-ig SQL kompatibilis
- Weblapokról és szinte az összes szkript nyelvből elérhető (VBScript, JavaScript, PHP, Perl, stb.)
- az ingyenes verziók is kiválóan használhatók

⁹ <http://www.softwareonline.hu/art3216/mysql+szerver+bemutatasa.html>

Negatívumként megemlíthető, hogy nem áll rendelkezésre olyan minden funkciót ellátni képes, könnyen kezelhető adminisztrációs felület, mint létezik az MS SQL-nél. Bár a fejlesztések igyekeznek bepótolni ezt a hiányosságot és már vannak biztató lépések (pl.: MyCC), még egy kicsit várnunk kell az igazi megoldásra.”

4. Az oldal funkciói

4.1 Regisztrálás

Ez a funkció csak kijelentkezett állapotban érhető el, a *reg.php* oldalról. A felhasználó itt adhatja meg az adatait, hogy létrehozzon egy accountot és azután betudjon jelentkezni az oldalra, hogy megkezdhesse az ismerkedést a többi felhasználóval. Az űrlap kitöltése során majdnem minden adat kitöltése kötelező, ha ez nem történik meg hibaüzenetet kap, mint például ami az ábrán is látható.

REGISZTRÁLÁS

Felhasználónév:
Felhasználónév megadása kötelező!

Jelszó:
Nincs megadva jelszó!

Jelszó megegyeszer:

Keresztnév:

Vezetéknév:

E-mail Cím:
Az E-mail megadása kötelező!

Nem:

Születési Dátum:

Megye:

Lakhely:
A Lakhely megadása kötelező!

Családi állapot:

Magasság:
A magasság megadása kötelező!

Testsúly:
A testsúly megadása kötelező!

Csillagjegyed:

Megjelenés:

Szemszín:

Hajszín:

☐ Elfogadom a feltételeket
A feltételeit nincs elfogadva!

6. ábra - Regisztrálás

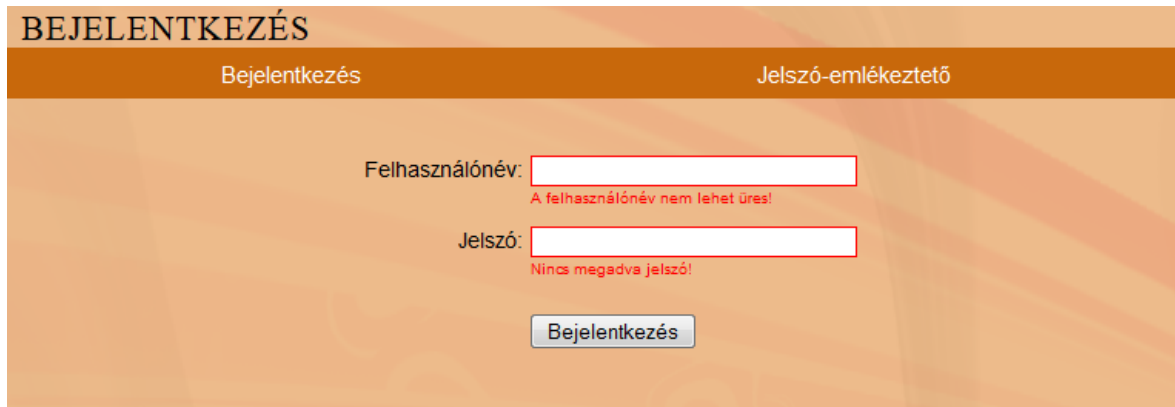
- Legelsőnek a felhasználónevet kell megadni. Vannak bizonyos megkötések, csak az ABC kis és nagybetűi, számok, alsó vonal és pont szerepelhet. Szűrve van, hogy azonos felhasználónév megadása nem lehetséges (ha már regisztráltak ilyen

névvel), a mezőt nem lehet üresen hagyni, kötelező kitölteni, minimum 5 karakterből kell állnia és maximum 12 karakter lehet.

- A jelszó megadása kötelező, a mező nem lehet üres, bármilyen karakter megadható, nincs megkötés, a jelszó tárolása MD5-el van titkosítva. Minimum 5 karakternek és maximum 12 karakternek kell lennie, a jelszó bekérése 2x történik meg a Jelszó megegyezés mező kitöltésével, erre azért van szükség, mert a mező típusa password a felhasználó nem látja milyen karaktereket ütött be, a jelszó elgépelését lehet elkerülni így.
- A keresztnév és a vezetéknév maximum 20 karakter lehet, egyéb megkötések nincsenek.
- E-mail cím megadás maximum 30 karakter lehet, szűrve van, hogy csak megfelelő formátumú címet lehet megadni, tehát kötelező megadni a @-ot. továbbá ellenőrizve van, hogy valaki regisztrált-e már ezzel az email címmel. Megadása kötelező!
- Nem kiválasztása legördülő menüből, férfi és nő választható ki.
- Születési dátum megadásánál csak érvényes dátum típus adható meg.
- Megye megadása, szintén legördülő menüből választható ki mind a tizenkilenc megyénk ABC sorrendben felsorolva.
- Lakhely megadása kötelező maximális hossza 20 karakter lehet és csak az ABC kis és nagybetűi használhatóak.
- Családi állapot megadása, legördülő menüből 6 választható állapot lehetséges.
- Magasság megadásánál csak szám adható meg minimum 80cm és maximum 250cm lehet!
- Testsúly megadásánál csak szám adható meg minimum 40kg és maximum 180kg lehet!
- Csillagjegy megadása legördülő menüből választható ki az év hónapjainak megfelelően növekvő sorrendben.
- Megjelenés 5 opció közül lehet kiválasztani.
- Szemszín 5 opció közül lehet választani.
- Hajsza 6 féle szín választható.
- Feltételek elfogadása, ha a felhasználó nem pipálja be nem lehetséges a regisztráció.

4.2 Bejelentkezés

A helyes adatok megadása után a regisztrációnk sikeres és átdob automatikusan a bejelentkező oldalra (*bejelentkezes.php*).



The screenshot shows a web form titled "BEJELENTKEZÉS". It has two tabs: "Bejelentkezés" (selected) and "Jelszó-emlékeztető". The form contains two input fields: "Felhasználónév:" and "Jelszó:". Below the "Felhasználónév:" field is a red error message: "A felhasználónév nem lehet üres!". Below the "Jelszó:" field is a red error message: "Nincs megadva jelszó!". At the bottom of the form is a button labeled "Bejelentkezés".

7. ábra - Bejelentkezés

Itt lehetőségünk van bejelentkezni, illetve jelszó emlékeztetőt kérni, amely a regisztrált email címünkre automatikusan elküldődik egy új generált jelszóval.

Bejelentkezéskor fontos ellenőrizni, hogy létezik-e a beírt felhasználó név, ez egyszerű lekérdezéssel kivitelezhető.

```
$belepes_ell = mysql_query("SELECT count(azonosito) FROM  
g_all WHERE felhasznalonev = '". $felhasznalonev. "' AND jelszo  
= '". $jelszo. "' ");  
$a=mysql_fetch_array($belepes_ell);  
if ($a[0] != 1) {  
    $hiba['felhasznalonev']="Érvénytelen felhasználónév vagy  
jelszó!";  
}
```

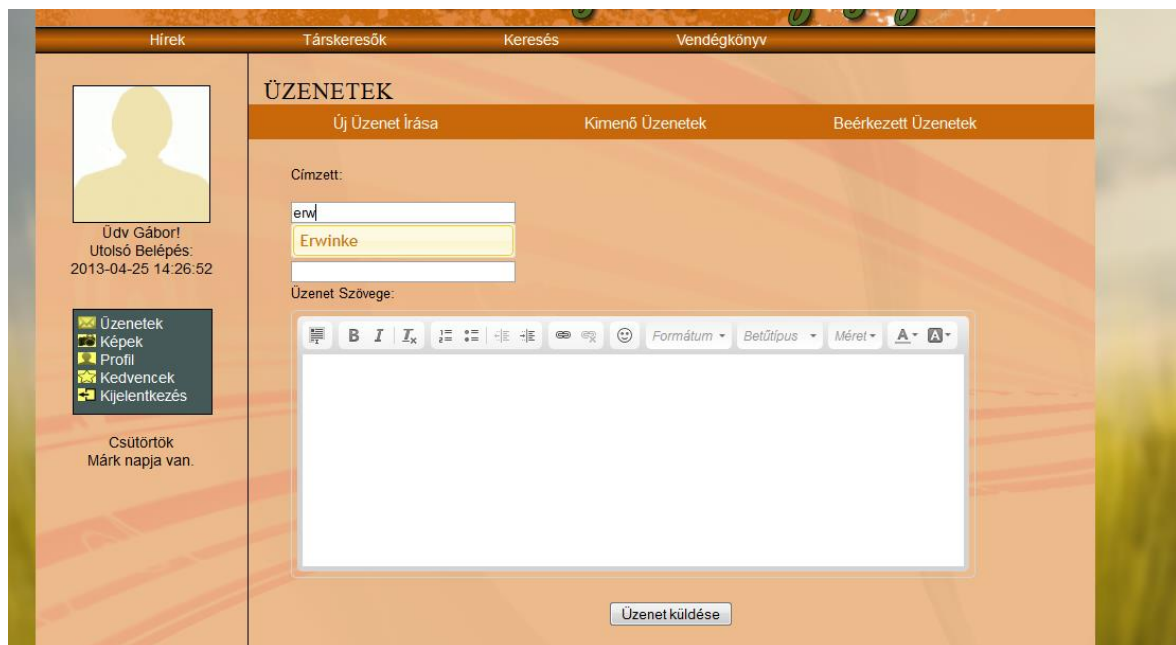
Kiválasztjuk az *azonosito* mezőt amit összesítünk (*count*) a *g_all* táblából, ahol a bekért jelszó és felhasználónév megegyezik akkor 1 lesz az eredmény, ezt lekérjük a *\$a* változóba és feltételbe megadjuk, ha *\$a[0]* azaz a nulladik eleme nem egyenlő 1-el, tehát nem volt egyezés, ekkor kapjuk a hibaüzenetet. Hiba elárulni, hogy a jelszó vagy a felhasználónév nem egyezik, ezzel is bonyolítva a rosszindulatú felhasználók dolgát. A helyes adatok megadása után a *session* (*munkamenet*) változóba (*\$_SESSION*) adatokat tárolok el. Azért hasznos alkalmazni, mert ezek az ideiglenesen eltárolt adatok nem a felhasználó gépén sütiben tárolódnak, hanem a szerveren. Eltárolásra kerül a felhasználó

neve, rangja, azonosítója, utolsó belépésének dátuma, kilépésének dátuma. Ezekre a későbbiek során szükség lesz.

Ha megtörtént a bejelentkezés átkerülünk a hírek oldalra (*index.php*). Itt olvashatóak az aktuális hírek, de bal oldalt megjelenik a felhasználó saját menüje és profilképe is. Az új tagok profilképe alapértelmezetten egy nemtől függő alakot ábrázoló kép.

A felhasználó profilképe alatt található egy üdvözlő szöveg a saját nevével és hogy mikor lépett be utoljára az oldalra.

4.3 Üzenetek



8. ábra - Üzenetküldés

4.3.1 Új üzenet írása

A címzetthez a felhasználó nevét kell megadni. Úgy gondoltam, hogy használhatóbb lesz az oldal az automatikus kiegészítéssel. Az első három betű beütése után automatikusan segít, felajánlja azokat a felhasználókat akiknél van egyezés, így gyorsan és egyszerűen ki lehet választani a neveket.

4.3.1.1 Autocomplete (automatikus kiegészítés)

Az automatikus kiegészítés jQuery-vel történik. Itt található a függvény, amelyik ezt elvégzi:

```

$("#cimzettazonosito").autocomplete({
    source: function(request, response) {
        $.getJSON("kiegeszites.php", {kulcs: request.term},
response);
    },
    select: function(event, ui) {
        $("#cimzettazonosito_id").val(ui.item.id);
    },
    minLength: 3
});

```

Az első sorban a `cimzettazonosito` id-val rendelkező html elemre hívódik meg az `autocomplete` jquery függvény. Az `autocomplete` függvény paramétere egy JavaScript objektum, amelynek három attribútuma van.

- `source`: (forrás) honnan szedje az adatokat. A kód a `kiegeszites.php` oldalt hívja meg egyetlen egy `kulcs` nevű paraméterrel, amely a felhasználó által a címzett beviteli mezőbe (melynek az id-ja `cimzettazonosito`) begépelt szövegtöredék.
- `select`: (kiválasztás) mi történjen, amikor a felhasználó kiválaszt valamit a listából. A `cimzettazonosito_id` idval (azonosítóval) rendelkező hidden (rejtett) input mező értéke a kiválasztott felhasználó IDja lesz.
- `minLength`: Mekkora hosszúságú szövegnél küldje el a kérést a servernek. A minimum hosszúságot tetszés szerint változtathatjuk jelen esetben az értéke három.

4.3.1.2 Kiegészítés.php

Ez egy JSON formátumú választ ad vissza. Először ellenőrzi, hogy a felhasználó bejelentkezett-e és megkapta-e a `kulcs` paramétert. A `$kulcs` változóban eltároljuk a `kulcs` nevű paramétert. Nagybetűssé alakítjuk, majd meghívjuk rá a `mysql_real_escape_string()` nevű függvényt, ami levédi a szövegben a speciális karaktereket a MySQL lekérdezés számára. A `$data` nevű változóba lekérjük a `g_all` táblából azokat az azonosítókat és felhasználóneveket ahol a felhasználónév nagybetűs változatának része a `$kulcs` változóban lévő szövegtöredék. Valamint saját magunkat kihagyjuk a találati listából. A `$names` egy tömb, aminek az elemei tömbök. A tömbben egy `label` és `id` található, az `id` a felhasználó azonosítója lesz, a `label` pedig a felhasználónév. Végül a JSON formátumra hozzuk a `json_encode()` függvény segítségével.

```

if(isLoggedIn() && isset($_GET["kulcs"])) {
    $kulcs =
mysql_real_escape_string(strtoupper($_GET["kulcs"]));
    $data = mysql_query("SELECT azonosito, felhasznalonev FROM
g_all WHERE upper(felhasznalonev) LIKE '%" . $kulcs . "%' AND
azonosito <> '".$_SESSION["azonosito"]");

    $names = array();
    while ($row = mysql_fetch_assoc($data)) {
        $names[] = array(
            'label' => $row['felhasznalonev'],
            'id' => $row['azonosito']
        );
    }
}
echo json_encode($names);

```

4.3.2 Üzenetek olvasása és törlése

A felhasználónak címzett levelek találhatóak itt táblázatos formában megjelenítve. A beérkezett és kimenő üzenetek külön oldalon találhatóak. Az első oszlopban egy checkbox (jelölő négyzet) található, aminek segítségével könnyen kijelölhetjük azokat a leveleket, amiket törölni szeretnénk. Ha minden levelet kisseretnénk jelölni, akkor egyszerűen a fejlécben található jelölőnégyzetre kell kattintanunk.

A levelek törlésénél a fizikai törlést választottam a logikai helyett, mert ha a felhasználó törli magát a rendszerből és megmaradnak a levelei az szerintem etikátlan. Sok problémát vetett fel, hogy akkor mégis hogyan töröljem a leveleket.

```

function levelTorlese($level_azon) {
    $torolve_query = mysql_query("SELECT torolve FROM
g_uzenetek WHERE azonosito = '".$_SESSION["azonosito"]."");
    $torolve = mysql_fetch_array($torolve_query);
    $torles_query='';
    if($torolve[0]==0) {
        $torles_query="UPDATE g_uzenetek SET
torolve='".$_SESSION["azonosito"]." WHERE
azonosito='".$_SESSION["azonosito"]."";
    } else {
        $torles_query="DELETE FROM g_uzenetek WHERE
azonosito='".$_SESSION["azonosito"]."";
    }
    return mysql_query($torles_query);
}

```

A fent látható kódrészletben a levéltörlés függvény látható.

Először lekérjük a `g_uzenetek` táblából a törölve oszlop adatait ahol az azonosító egyenlő lesz a kiválasztott levél azonosítójával, ha van ilyen akkor lekértem a sor adatait és a `$torolve` változóba tárolom el. A `$torles_query`-t deklarálom, kezdeti értéke egy üres sztring lesz. Ha a `torolve` nulladik eleme egyenlő nullával, azaz üres akkor a `torolve` mezőt frissítem. A bejelentkezett felhasználó azonosítója kerül beírása ott ahol a levél azonosítója megegyezik a kiválasztott levél azonosítójával, és így a levelek között már az adott felhasználónál nem fognak azok a levelek megjelenni ahol a saját azonosítója szerepel a `torolve` mezőben.

Ez azért fontos, mert ha az egyik fél törölte az üzenetet - legyen az elküldött vagy beérkezett -, attól a másik fél nem biztos, hogy törölni szeretné vagy már törölte azt. Csak akkor történik meg a végleges (fizikai) törlés, ha már mindkét felhasználó törölte a levelet. Ahogy a függvényben is látható, ha már volt bejegyzés a `törölve` mezőben, tehát nem üres, akkor törlődik a kiválasztott levél. A függvény meghívása az `uzenetek.lista.php` fájlban hívódik meg.

```
if(isset($_POST['jelolt_level'])) {  
    foreach($_POST['jelolt_level'] as $level_azon) {  
        if(!levelTorlese($level_azon)) {  
            die("Hiba - Az adatainak módosítása közben:  
".mysql_error());  
        }  
    }  
}
```

Ha a `jelolt_level` nem üres, akkor a `jelolt_level` legyen `$level_azon`-al és ha a `levelTorlese($level_azon)` nincs egyező visszatérési értéke akkor megállítjuk a kód futását és hibaüzenetet jelenítünk meg.

Egy oldalon jelennek meg a beérkezett és a kimenő üzenetek, de egyszerre csak az egyik látszik. Egy paraméter dönti el, hogy melyik látszik éppen. Alapértelmezetten (paraméter nélkül) a kimenő üzeneteket láthatjuk. Ha a paraméter értéke 1, akkor a beérkezett üzenetek jelennek meg.

A második oszlopban található a levelek státusza, hogy olvasta-e már a felhasználó vagy sem. A zárt boríték az olvasatlan üzeneteket jelöli, a nyitott boríték pedig a már olvasott üzeneteket. A következő oszlopban a címzett vagy a feladó neve található, mellette az üzenet tárgya, aztán az üzenet szövegének egy részlete, és végül egy link az üzenet elolvasásához.

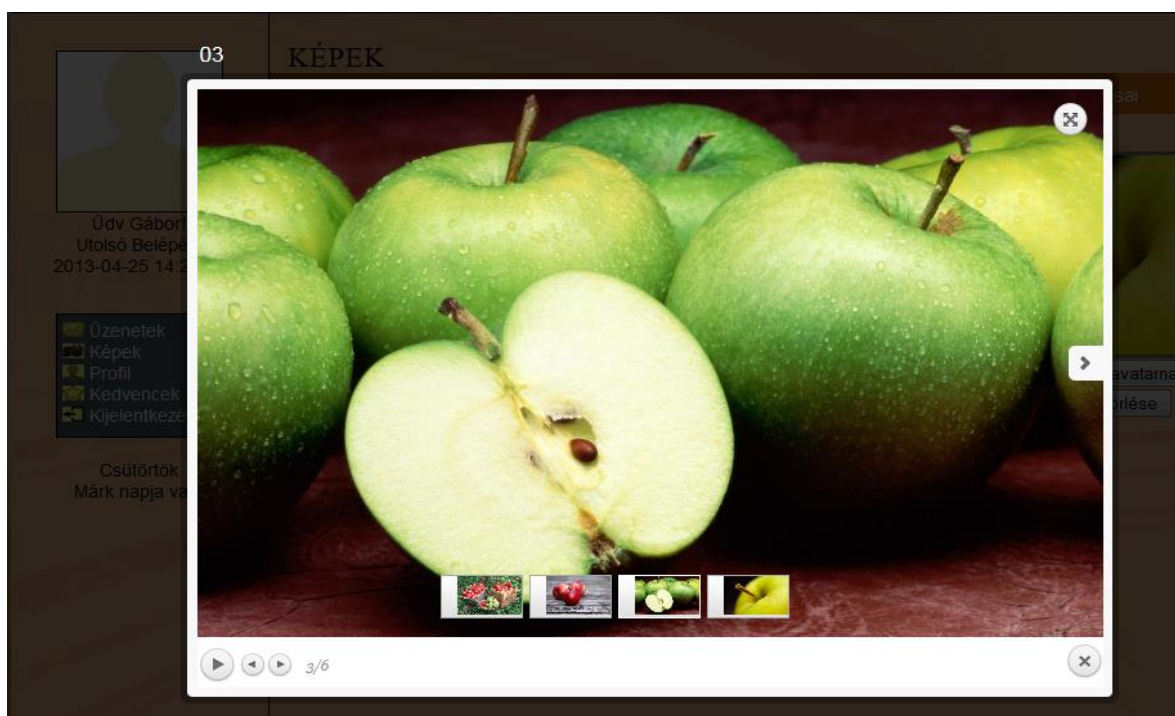
4.4 Képek

Itt található a felhasználó által feltölt képek, lehetőség van a profilkép kiválasztására, és a már nem szükséges képek törlésére.



9. ábra - Képek beállítási lehetőségei

A képek egy JavaScript segítségével jelennek meg. A script neve prettyPhoto ingyenesen letölthető és felhasználható.



10. ábra - Képnézegető (prettyPhoto)

Tartalmaz képvetítés funkciót is. Több feltöltött kép esetén az alsó sorban látható a többi kép is és így könnyen kiválasztható más kép, nem muszáj sorba haladni. A jobb felső sarokban található ikonra kattintva a kép az eredeti méretében jelenik meg.

Három menürésze van a képek menüpontnak.

1. menüpont a már feltöltött képek megjelenítése.
2. menüpont, képek feltöltése és a kép címének megadása (nem kötelező megadni, maximum 30 karakter lehet). A képek feltöltésekor, feltöltött fájl ellenőrzés alá kerül, először megnézzük a feltöltött fájl kiterjesztését, meg kell néznünk a mime típusát, azaz hogy valóban kép-e és nem csak átvan nevezve valamilyen képformátumra a fájl. Három képtípust engedélyeztem `jpg`, `gif`, `png`. Vizsgáljuk a kép méretét is, hogy az 5 megabájttól nagyobb ne lehessen. Ha nagyobb, akkor hibaüzenetet kapunk. A feltöltött képek egy `images` mappába kerülnek és azon belül a felhasználó nevével létrehozott mappába. A feltöltött képek átnevezése történik a felhasználó nevére és utána véletlenszerűen generálódik egy háromjegyű szám 1 és 1000 között. Ha megtörtént az átnevezés a szerver ideiglenes `temp` könyvtárból átkerül a képe az általam megadott `temp` könyvtárba még hozzá a `images/tmp`-be. Ha a felhasználó nem rendelkezett még a saját mappájával, tehát nem volt még feltöltött képe, akkor a mappa létrejön és a `temp` könyvtárból átkerül oda a kép és végül a `temp` könyvtárból törlődik a fájl. A kép neve, helye, elérési útja és a felhasználó azonosítója eltárolásra kerül az adatbázisban a `g_kepek` táblában.
3. menüpont a képek törlése és profilkép beállítása.

4.5 Profil

4.5.1 Felhasználó adatainak megjelenítése és szerkesztése

A felhasználó adatainak megjelenítése: minden információ megjelenik a felhasználóról, amit megadott.

A felhasználó adatainak módosítása: lehetőség van az adatok módosítására, illetve olyan plusz adatok megadására, ami a regisztráció során nem volt elérhető.



The image shows a web form for editing a user profile. The form is set against a light orange background with a subtle pattern. It contains the following fields and values:

- Kor:** 23 éves
- Születési Dátum:** 1989-06-04
- Nem:** Férfi
- Megye:** Szabolcs-Szatmár-Bereg (dropdown menu)
- Lakhely:** Kisvárdá (text input)
- E-mail:** fogasgabor@gmail.com
- Családi állapot:** Egyedülálló (dropdown menu)
- Magasság:** 185 (text input) cm
- Testsúly:** 99 (text input) kg
- Csillagjegy:** Ikrek (dropdown menu)
- Megjelenés:** Sportos (dropdown menu)
- Szemszín:** Barna (dropdown menu)
- Hajszín:** Barna (dropdown menu)
- Dohányzás:** (empty dropdown menu)
- Bemutakozás:** (empty text area)
- Idézetek:** (empty text area)

At the bottom right of the form is a button labeled "Adatok módosítása".

11. ábra - Adatlap módosítás

Itt adható meg, hogy dohányzik-e, rövid bemutatkozás és esetleg valami kedvenc idézet. Egyes adatokat legördülő menüből lehet kiválasztani, ami szintén egy függvényben, van megírva.

4.5.2 Profil törlése

Szükséges hozzá a jelenlegi felhasználói jelszó megadása.

```
$torles_level = mysql_query("SELECT azonosito FROM g_uzenetek  
WHERE (felado_azonosito = '". $_SESSION['azonosito']."' OR  
cimzett_azonosito = '". $_SESSION['azonosito']."' ) AND torolve  
<> '". $_SESSION['azonosito']."'");  
while($level_azon = mysql_fetch_array($torles_level)) {  
    if(!levelTorlese($level_azon[0])) {  
        $hiba['altalanos']="Hiba történt! Kérjük próbálja meg  
ismét!";  
        break;  
    }  
}
```

Ahogy a kódrészletben is látható, szükséges a leveleket törölni. A fent említett levéltörlés elve szerint, amíg nem törölte mind a két felhasználó, addig a levél még az adatbázisban marad. Első körben kiválasztjuk a g_uzenetek táblából a levél azonosito-kat, ahol a felado_azonosito megegyezik a bejelentkezett felhasználó azonosítójával vagy a cimzett_azonosito egyenlő a bejelentkezett felhasználó azonosítójával és a torolve oszlopban nem egyenlő a felhasználó azonosítójával. Következik egy ciklus, ami végig megy az így elkészült listán, ha üres akkor beállítunk egy hibaüzenetet.

```
$eredmeny=mysql_query("UPDATE g_uzenetek SET felado_azonosito  
= NULL WHERE felado_azonosito =  
'". $_SESSION['azonosito']."'");  
if(!$eredmeny) {  
    $hiba['altalanos']="Hiba történt! Kérjük próbálja meg  
ismét!";  
}
```

Ha ez sikeres volt, akkor következik a felhasználó azonosítójának törlése a g_uzenetek táblából. A felado_azonositot átállítjuk NULL-ra, ahol a felado_azonosito egyenlő a bejelentkezett felhasználó azonosítójával. Ha sikertelen a lekérdezésünk, akkor hibaüzenetet kapunk.

```
$eredmeny=mysql_query("UPDATE g_uzenetek SET  
cimzett_azonosito = NULL WHERE cimzett_azonosito =  
'". $_SESSION['azonosito']."'");  
if(!$eredmeny) {  
    $hiba['altalanos']="Hiba történt! Kérjük próbálja meg  
ismét!";  
}
```

A művelet megismétljük csak a `cimzett_azonosito` oszlopra.

```
$storles_felhasznalo = mysql_query("DELETE FROM g_all WHERE  
azonosito='".$$_SESSION['azonosito']."' ");  
if(!$storles_felhasznalo) {  
    $hiba['altalanos']="Hiba történt! Kérjük próbálja meg  
ismét!";  
} else {  
    header("Location: kijelentkezes.php");  
    exit;  
}
```

A levelek törlése után következik a felhasználó adatainak törlése a `g_all` táblából egyszerű törlés, törölje azt a sort ahol az azonosító megegyezik a bejelentkezett felhasználó azonosítójával.

Profil törléséhez és a felhasználó jelszavának módosításához a jelenlegi jelszó megadása szükséges. Teljes fiók törlésénél minden adat törlődik a felhasználóról. Beleértve a leveleit, képeit, valamint ha volt küldött levele egy másik felhasználónak, akkor egy függvény segítségével, „TÖRÖLT FELHASZNÁLÓ” lesz a neve.

4.6 Kedvencek

A felhasználó által kedvencnek jelölt partnerek jelennek meg itt. Innen könnyen elérhetőek a számára szimpatikus emberek, valamint lehetősége van levélküldésre és profil megtekintésére.

4.7 Társkeresők

Itt tekinthető meg a regisztrált felhasználók listája profil képpel, névvel, korral, nemmel, lakóhellyel és az utolsó bejelentkezés dátumával. Lehetőség van felvenni a kedvencek közé vagy üzenet írására az adott felhasználónak. A már kedvencnek jelölt partner neve mellett egy csillag jelenik meg. A felhasználó saját maga nem jelenik meg a listában, és akik már jelölve vannak őket többször nem lehet.

4.8 Keresés

Bejelentkezett állapotban érhető el, lehetőség van a regisztrált felhasználókra keresni az alábbi szempontok szerint:

- Felhasználó névre,
- Keresztnévre és
- Vezetéknévre

Ahol megegyezik a keresési feltétel, azok listázódnak ki.

4.9 Vendégekönyv

Ez a funkció ki és bejelentkezett állapotban is elérhető. Itt található egy rövid ismertetés az oldalról és a szakdolgozat témám is fel van tüntetve. Két fontos információ is megtalálható itt, hogy hány regisztrált felhasználóval jelentkezik az oldal, és hány regisztrált online felhasználó tartózkodik az oldalon. A lap alján megtalálhatóak a felhasznált programok bannerjei és az ismertebb böngészőké is, ahonnan gyorsan és egyszerűen letölthetőek. Két további opció található még, az idegenek vagy a felhasználók által írt vélemények megtekintése és új vélemény beküldése.

4.10 Admin

Ez az opció csak admin jogosultsággal rendelkező felhasználók számára jelenik meg. Az alábbi funkciók érhetőek el itt:

- Hírek posztolása,
- Regisztrált felhasználók néhány adatának megjelenítése,
- Felhasználók törlése az adatbázisból.

4.11 Kijelentkezés

Kilépéskor az eltárolt bejelentkezési dátum frissítve lesz az adatbázisban ott, ahol a bejelentkezett felhasználó azonosítója megegyezik az adatbázisban szereplő felhasználó azonosítójával. Ezután a bejelentkezéskor beállított munkamenet (\$_SESSION) kiürítésre kerül és a felhasználót átirányítjuk a kezdő oldalra.

5. Felhasználói kézikönyv

Ha saját gépünkön szeretnénk használni a weboldalt szükség lesz egy Apache-ra és egy phpMyAdminun-ra. A XAMPP programcsomag telepítésével minden ehhez szükséges programot megkapunk. Az első lépés, a táblák importálása az adatbázisba.

A phpMyAdminunkat megnyitva létrehozunk egy adatbázist tetszőleges névvel és utf8_hungarian_ci illesztéssel, ez az ékezetes betűk miatt fontos. Ha ez megtörtént már csak importálnunk kell a táblákat. Az importálás menüpontra kattintva betallózzuk az SQL fájlunkat (*adatbazis.sql*) és máris a rendelkezésünkre állnak az oldalhoz szükséges táblák.

Tábla	Művelet	Sorok	Típus	Illesztés
☐ g_all	Tartalom Szerkezet Keresés Beszúrás Kiürítés Eldobás	0	InnoDB	utf8_hungarian_ci
☐ g_csaladi_allapot	Tartalom Szerkezet Keresés Beszúrás Kiürítés Eldobás	6	InnoDB	utf8_hungarian_ci
☐ g_csillagjegy	Tartalom Szerkezet Keresés Beszúrás Kiürítés Eldobás	12	InnoDB	utf8_hungarian_ci
☐ g_dohanyzas	Tartalom Szerkezet Keresés Beszúrás Kiürítés Eldobás	3	InnoDB	utf8_hungarian_ci
☐ g_hajszin	Tartalom Szerkezet Keresés Beszúrás Kiürítés Eldobás	6	InnoDB	utf8_hungarian_ci
☐ g_hirek	Tartalom Szerkezet Keresés Beszúrás Kiürítés Eldobás	0	InnoDB	utf8_hungarian_ci
☐ g_kedvencek	Tartalom Szerkezet Keresés Beszúrás Kiürítés Eldobás	0	InnoDB	utf8_hungarian_ci
☐ g_kepek	Tartalom Szerkezet Keresés Beszúrás Kiürítés Eldobás	0	InnoDB	utf8_hungarian_ci
☐ g_megjelenes	Tartalom Szerkezet Keresés Beszúrás Kiürítés Eldobás	5	InnoDB	utf8_hungarian_ci
☐ g_megye	Tartalom Szerkezet Keresés Beszúrás Kiürítés Eldobás	19	InnoDB	utf8_hungarian_ci
☐ g_nem	Tartalom Szerkezet Keresés Beszúrás Kiürítés Eldobás	2	InnoDB	utf8_hungarian_ci
☐ g_online_felhasznalok	Tartalom Szerkezet Keresés Beszúrás Kiürítés Eldobás	0	InnoDB	utf8_hungarian_ci
☐ g_szemszin	Tartalom Szerkezet Keresés Beszúrás Kiürítés Eldobás	5	InnoDB	utf8_hungarian_ci
☐ g_uzenetek	Tartalom Szerkezet Keresés Beszúrás Kiürítés Eldobás	0	InnoDB	utf8_hungarian_ci
☐ g_vendegkonyv	Tartalom Szerkezet Keresés Beszúrás Kiürítés Eldobás	0	InnoDB	utf8_hungarian_ci
15 táblák	Összeg	58	InnoDB	utf8_hungarian_ci

12. ábra - Importált táblák

Ezután szükségünk lesz egy felhasználóra az adatbázishoz. A *config.ini* fájlunkat kell szerkeszteni az adatok megadásához:

- MYSQL_SERVER=Szerver címe
- MYSQL_USER=A felhasználó neve.
- MYSQL_PASS=A felhasználó jelszava.
- MYSQL_DB=Adatbázis neve.

Az adatok kitöltésével már létre is hoztuk a kapcsolatot a weblap és az adatbázis között, az oldalhoz szükséges fájlokat feltöltjük a tárhelyre és az már üzemkés.

6. Továbbfejlesztési lehetőségek

Úgy gondolom nincs tökéletes program, és mindegyiket lehetne tovább fejleszteni. A szakdolgozat megírása közben sok ötlet jutott eszembe, de ezek közül nem mind került megvalósításra. Néhány ötlet, amivel a jövőben bővíteni lehetne az oldalt:

- Regisztrálásnál nem dobna át az oldal a bejelentkezés részhez, hanem automatikusan beléptetné az oldalra a felhasználót.
- Megerősítő e-mail küldése a regisztráláskor megadott címre, amely tartalmazna egy aktiváló linket, amire kattintva a felhasználó véglegesítheti a regisztrációját.
- Egyszerre több kép feltöltése a galériába, albumok létrehozása.
- Az oldal többnyelvűségének megvalósítása, angol és magyar nyelven.
- Valós idejű chat beépítése.

7. Összegzés

A szakdolgozatom megírása során törekedtem minden olyan trükkös dolgot bemutatni, ami sok fejtörést okozott a megvalósítás során. Próbáltam kódrészletekkel szemléltetni az oldal főbb funkcióit. Nem gondoltam volna, hogy ilyen sokrétű tudásra van szükség egy jól működő és szép weboldal elkészítéséhez.

Úgy érzem sikerült elérnem a kezdeti célkitűzéseimet, de persze sok apró dologgal lehetne bővíteni még az oldalt. Remélem sikerült érthetően elmagyaráznom, milyen megoldásokat alkalmaztam az egyes helyeken.

Az elkészült társskereső weboldallal elégedett vagyok, rengeteg örömet leltem az oldal elkészítésében. A legfelemelőbb érzés, ha kitűz az ember egy célt és sikerült megvalósítani.

Egyetlen negatívummal találkoztam: sok odafigyelést igényel, hogy olyan weboldat hozzak létre, amely böngésző független azaz mindegyikben ugyanúgy jelenik meg.

Sikerült bővítenem PHP, CSS, MySQL tudásomat és úgy érzem olyan használható tudásra tettem szert, amelyet a későbbiek során is kamatoztathatok. Bízom benne, hogy a közeljövőben az oldalt továbbfejlesztve a nagyközönség elé is tárhatom valamilyen formában.

8. Köszönetnyilvánítás

Legelőször köszönetet szeretnék mondani testvéremnek, Fogas Jánosnak, aki szakmai tudása töredékét átadva támogatott a dolgozatom elkészítésében és mindig bizalommal fordulhattam hozzá szakmai támogatásért.

Továbbá sok segítséget kaptam Simon Béláné dr.-től is a témavezető tanárnőmtől.

Hálás vagyok barátaimnak, csoporttársaimnak, akik bízattak és segítettek a munkámat észrevételeikkel, szakmai tanácsaikkal.

Köszönettel tartozom családomnak, akik mindvégig mellettem álltak és segítséget nyújtottak főiskolai éveim alatt felmerülő problémáim megoldásában.

Irodalomjegyzék

1. A PHP története - <http://www.kando-kkt.sulinet.hu/mirror/phphu/intro-history.html>
2. Http protokoll ismertetése - http://informatika.gtportal.eu/index.php?f0=web_elm_113
3. HTML ismertetése - <http://www.matebalazs.hu/html.html>
4. CSS - <http://www.hotdog.hu/css-kodok/mindenhez/mi-a-css>
5. JavaScript - <http://www.nejanet.hu/~ksanyi/java/JAVAS.HTM>
6. SQL - <http://www.agt.bme.hu/szakm/adatb/db5.htm>
7. MySql - <http://www.softwareonline.hu/art3216/mysql+szerver+bemutatasa.html>
8. PrettyPhoto - <http://www.no-margin-for-errors.com/projects/prettyphoto-jquery-lightbox-clone/>
9. CKEditor - <http://ckeditor.com/>
10. Peter Moulding: PHP haladóknak - Fekete könyv 2002, Perfact-Pro Kiadó
12. Kevin Tatroe, Peter MacIntyre & Rasmus Lerdorf: Programming PHP 2013, O'REILLY
13. Eric A. Meyer: CSS zsebkönyv 2006, Kiskapu Kiadó
14. Julie C. Meloni, Michael Morrison : Tanuljuk meg a HTML és a CSS használatát 24 óra alatt 2011, Kiskapu kiadó
15. Lynn Beighley, Michael Morrison: Agyhullám - PHP és MYSQL 2009, Kiskapu Kiadó