

Miskolci Egyetem  
Gépészmérnöki és Informatikai Kar  
Általános Informatikai Tanszék



# **Webalkalmazás készítése PHP nyelven**

## **Szakdolgozat**

**Konzulens:**  
Szűcs Miklós

**Készítette:**  
Kovács Dániel  
TA4E8D  
Mérnök Informatikus Bsc  
Logisztika szakirány

## Tartalom

|                                                 |    |
|-------------------------------------------------|----|
| 1. Bevezetés.....                               | 1  |
| 1.1 Szakdolgozatom célja.....                   | 1  |
| 1.2 Versenytársak elemzése .....                | 2  |
| 2. Felhasznált technológiák.....                | 3  |
| 2.1 HTML .....                                  | 3  |
| 2.2 MYSQL.....                                  | 3  |
| 2.3 PHP .....                                   | 4  |
| 2.4 CSS.....                                    | 5  |
| 2.5 Felhasznált szoftverek.....                 | 6  |
| 3. Böngészők.....                               | 7  |
| 4. Az alkalmazás .....                          | 8  |
| 4.1 Szoftver-folyamat tevékenységei .....       | 8  |
| 5.Specifikáció.....                             | 10 |
| 5.1 Regisztráció.....                           | 10 |
| 5.2 Bejelentkezés.....                          | 10 |
| 5.3 Bejelentkezés utáni funkciók .....          | 10 |
| 6.Grafikus terv.....                            | 13 |
| 7. Adatbázis felépítés .....                    | 15 |
| 7.1 tars_user tábla.....                        | 16 |
| 7.2 tars_topic tábla .....                      | 16 |
| 7.3 tars_cat tábla.....                         | 16 |
| 7.4 tars_kviz tábla .....                       | 17 |
| 7.5 messages tábla.....                         | 17 |
| 7.6 messages_goup tábla.....                    | 17 |
| 7.7 tars_level tábla .....                      | 18 |
| 8. Séma megtervezése .....                      | 19 |
| 9. Kódolás .....                                | 21 |
| 10. Szoftver validáció.....                     | 33 |
| 10.1 Megjelenés .....                           | 33 |
| 11. Továbbfejlesztési lehetőségek, ötletek..... | 35 |
| 12. Felhasználói dokumentáció.....              | 37 |
| 12.1 Hardver és szoftver igény.....             | 41 |
| 12.2 Telepítés .....                            | 41 |
| 13. Tesztelés.....                              | 42 |

## Webalkalmazás készítése PHP nyelven

|                          |    |
|--------------------------|----|
| 14. Összefoglalás.....   | 44 |
| 15. Summary .....        | 45 |
| 16. Melléklet .....      | 46 |
| 17. Irodalomjegyzék..... | 48 |

## 1. Bevezetés

Napjainkban a társadalmi változások, a technika fejlődése, valamint a globalizáció okozta hatásoknak köszönhetően az internet egyre inkább középpontba került. Mind a civil, mind az üzleti szférában (termelés, gazdaság) meghatározó szerepet kapott az információs-technológia, mely ugyan nem csupán a világhálót jelenti, de nyilvánvalóan ezt nevezhetjük az információs társadalom egyik legdominánsabb erőforrásának. Ezért az idő előrehaladtával mint minden más ágazat az internetes világ is megkövetelte a fejlődést.

Régebben egy weboldal csak statikus tartalmakat lehetővé tevő HTML elemekből épült fel. A kezdetben statikus leírásokat, dokumentációkat tartalmazó web mára a feledés homályába merült, felváltotta a dinamikus tartalmú, felhasználó központú weboldalak tömkelege. Az új „trend” lehetővé tette az olyan weboldalak megjelenését melyek komplexitásuknak köszönhetően webes alkalmazásoknak nevezünk.

Manapság szinte mindenkinek van internet hozzáférése, vagy internet vételére alkalmas eszköze. Sok olyan terület van, ahol úgy végezhetünk munkát, hogy egy böngészőn kívül más alkalmazás nem is fut a gépen. Például egy online marketinges elemzi egy oldal látogatottságát a Google Analytics programmal, aminek az eléréséhez internetre van szükség vagy egy újságíró utánanéz a híreknek (esetleg egy webes programmal vagy egy bővítménnyel folyamatosan értesül róluk).

Az emberek egy jelentős része interneten vásárol, banki tranzakciókat végez, és még sorolhatnám. A kapcsolattartást is jelentősen felgyorsította a web ilyen irányú fejlődése. Napjainkban egyre nagyobb teret hódít az online társkeresés is, de ezeken legtöbbször idő kell a sikerhez, melyben az emberek manapság igencsak szűkölködnek.

### 1.1 Szakdolgozatom célja

A szakdolgozatomban egy saját készítésű társkereső weboldalt fogok elkészíteni. Az online társkereséshez idő és nem ritkán pénz is kell, ezen szeretnék némileg változtatni egy sajátos ötlettel, melyben a felhasználók egy online kérdőív segítségével megadják az érdeklődési köreiket és a dinamikus tartalomkezelésnek köszönhetően a program megadja a hozzájuk leginkább illő személyt. Szeretném demonstrálni, hogy hogyan lehet egy webalkalmazást elkészíteni, mely dinamikusan képes kezelni a felhasználókat és a köztük lévő kapcsolatokat. Az általam készített webalkalmazás során a legtöbb készítési probléma előkerül, így szerintem meglehetősen jól körbe lehet járni ezt a területet.

A megvalósításhoz a PHP, MySQL, Apache<sup>[1]</sup> triót fogom használni. Az adatok tárolása adatbázisban fog történni, melyből a PHP segítségével nyerem ki az adatokat, így egy dinamikus könnyen karbantartható oldalt készítek.

Indíttatásom a téma kiválasztása során az volt, hogy a jövőben szeretnék ilyen területen dolgozni, így az elkészítés során szerzett tudás a hasznomra válhat a későbbiekben

### 1.2 Versenytársak elemzése

Szakdolgozatom készítése előtt körülnéztem az interneten, hogy mit nyújtanak az ilyen jellegű honlapok. Regisztráltam is néhányra, de ezek körülbelül 50 százalékan csak akkor vehettem volna igénybe bizonyos szolgáltatásokat, ha előfizetek rájuk. Ezenfelül a regisztrációs folyamat olyan hosszú volt, hogy, ha nincs indíttatásom azok kitöltésére nagy valószínűséggel nem is csinálom végig. Másik oldalakra belépve észrevettem, hogy szinte pontról pontra ugyanolyan szolgáltatásokat nyújtanak, mint például az üzenetküldés, keresés, album, böngészés. Persze láttam olyan funkciókat is, amit a jövőben szeretnék megvalósítani, például a profil oldalon a látogatók száma, barát rendszer, több kép feltöltése, lakhely szerinti keresés. Mindegyik általam tesztelt oldalon túl sok funkció volt, ami miatt túl bonyolult volt azokat a funkciókat használni amiket valójában akar a felhasználó.

Ezért szeretnék elkészíteni egy átlátható letisztult a fő funkciókkal rendelkező társskeresőt létrehozni és belevinni egy olyan funkciót, amit én hiányoltam. Pontosabban volt ilyen opció csak azok szerintem túl voltak bonyolítva. Többek között ennek a funkciónak a kidolgozását fogom részletezni a dolgozatom során.

1: Az Apache HTTP Server (röviden Apache) nyílt forráskódú webkiszolgáló alkalmazás.

### 2. Felhasznált technológiák

A webalkalmazásom elkészítése során igyekeztem minél jobban szem előtt tartani a böngésző függő sajátosságokat, így választottam meg a felhasznált technológiákat is. Az ilyen eltérések legfőképpen a megjelenésben voltak ennél fogva a CSS böngésző specifikus beállításaira kellett figyelnem.

#### 2.1 HTML

A weboldalak strukturált tartalma a HTML jelölőnyelvben adható meg (angolul: HyperText Markup Language = hiperszöveges jelölőnyelv). A HTML egy leíró nyelv, melyet weboldalak készítéséhez fejlesztettek ki, és mára már internetes szabvánnyá vált a W3C (World Wide Web Consortium) támogatásával. A HTML nem programozási, hanem egy kód nyelv, ami azt jelenti, hogy akár egy Jegyzetömbben is alkothatnánk weboldalakat. Igazából a hipertext jelenti az interneten található oldalakat, amelyek képek szövegek animációk kombinációjából tevődik össze. A HTML ezeknek a dokumentumoknak az elrendezését, formázását tartalmazza. A nyelvet 1990 óta használják a hálózatok hálózatán vagyis az interneten tartalmak megjelenítésére. Jelenleg a HTML 5-ös verziója van színpadon, melyet már 2004-óta fejlesztenek.

#### 2.2 MYSQL

A MySQL egy stabil, gyors relációs adatbázis kezelő rendszer. Lehetővé teszi az adatok rendszerezését, hatékony tárolását és a kinyerésüket. Biztosítja az adatok párhuzamos hozzáférését és megakadályozza, hogy a jogosulatlan felhasználók információkat kaphassanak. Ezért a MySQL egy többszálú, több felhasználós lekérdező nyelv. Az adatbázisok kezeléséhez parancssori eszközöket használhatunk vagy választhatunk különböző grafikus felületű adminisztrációs eszközöket is. Például: a phpMyAdmin, melyet PHP nyelven írtak, nyílt forráskódú és rengeteg lehetőséget biztosít az adatok kezelésére. A dolgozatom során én is ezt használtam az SQL műveletek végrehajtásához. Egy külön fájlba helyeztem el a konstansokat, így nem kellett minden oldalon leírni a csatlakozási metódus során a különböző paramétereket, azokon, amelyeken kapcsolatot létesítenek a szerverrel.

Így mindössze ezt a két sort kellett minden php fájl elejére beírnom a csatlakozáshoz.

```
<?php
    require_once('connectvars.php');
    $dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD,
        DB_NAME);
?>
```

### 2.3 PHP

A PHP életre kelti a weboldalakat. A PHP születésekor csupán egy makro készlet<sup>[2]</sup> volt, azonban mára egy önálló programozási nyelvnek nőtt ki magát, mely képes nagyméretű adatbázis alapú alkalmazások működtetésére is.

Ahhoz, hogy egy weboldal statikus HTML megjelenítésből, dinamikus webes alkalmazássá váljon a webkiszolgálónak is dinamikussá kell válnia. Ezt teszi lehetővé a PHP. A kiszolgálón tárolt hagyományos HTML oldalaktól abban tér el, hogy a kiszolgáló oldalán a PHP - értelmező feldolgozza ezeket. Vagyis a HTML elemek érintetlenek maradnak, de a PHP kódok feldolgozódnak. A nyelvvel létrehozhatunk dinamikus képeket, fájlokat olvashatunk, adatbázis lekérdezéseket eszközölhetünk, tehát a lehetőségek szinte végtelenek.

#### A PHP alkalmazásának három fő területe van

1. Grafikus alkalmazás

Nem ez a legmegfelelőbb nyelv ilyen alkalmazások készítésére, de ha már elég jól ismerjük a nyelvet, akkor használhatjuk e képességeit a kliens-oldali alkalmazásainkban.

2. Parancssori alkalmazás

Alkalmas lehet parancsfájlok készítésére is, Ilyenkor nincs szükségünk se szerverre se böngészőre csupán egy PHP értelmezőre. Alkalmazták automatizálási célokra vagy egyszerű szöveg feldolgozásra.

3. Szerver oldali alkalmazás

Ez a leghatékonyabb területe a PHP alkalmazásának. Ekkor három dologra lesz szükségünk: web szerverre, böngészőre és egy PHP értelmezőre.

2: Tulajdonképpen számítógépes utasítások sorozata, amelyeket rögzíthet, majd hozzárendelhet egy billentyűkombinációhoz vagy makró névhez.

### Amiért a PHP nyelvre esett a választásom

A PHP-t úgy tervezték, hogy minden operációs rendszeren használható legyen és együttműködjön az adatbázis-kezelőkkel, kiszolgálókkal.

Elsődleges versenytársa a következő három programozási technológia ASP.NET (.NET keretrendszeren), a Perl (interpretált programozási nyelv) és a Ruby (Rails keretrendszeren).

Ezekkel összevetve a PHP néhány erőssége

- Nagyméretű, hozzáférhető dokumentáció és támogatás
- Hordozhatóság
- Forráskódokhoz való hozzáférhetőség
- Rengeteg beépített könyvtár, függvény
- Adatbázis - integráció

### 2.4 CSS

A lépcsőzetes stíluslapok (Cascading Style Sheets, CSS) adják a leírókódokon belüli elemeknek a megjelenési tulajdonságokat. A CSS létrejöttének oka az volt, hogy a weboldalakon megnőtt az igény az egyre bonyolultabb, kiforrottabb elemekre és formázásokra, amelyeket már az eredetileg egyszerű szöveges információk ábrázolására kitalált HTML-lel nem nagyon volt megvalósítható. (Sok formázást meglehetősen valósítani a HTML-el is, csak az egyes beállításokat külön-külön mindegyik oldalon végre kellett hajtani, így a nyelv elveszette egyszerűségét és áttekinthetőségét.) Ekkor támadt igény a CSS kifejlesztésére, mely sokkal rövidebbé, rugalmasabbá és szabadon kezelhetőbbé tette a HTML dokumentumokat. A CSS segítségével beállíthatjuk a betűtípusokat, színeket, margókat, távolságokat, egyes elemek helyeit a weblapon belül és még sok más. Egy stílus lapot több oldalhoz is hozzá rendelhetünk, így rugalmasságot és időt spórolhatunk velük. Ha egy stílus lapon belül módosítunk például egy háttér színt, akkor az összes olyan HTML oldalon módosulni fog a background, amihez hozzá van csatolva a CSS fájl. Ha a böngészőben megnyitunk egy oldalt akkor az egy úgynevezett cache (gyorsítótár) mappába lementi a stíluslapot, és, ha legközelebb az oldalra navigálunk, akkor nem kell megvárni, hogy letöltődjön a CSS fájl, mert a böngésző már a gyorsítótárban tárolja. Így időt spórol a felhasználóknak.

Maga a nyelv már viszonylag régóta létezik; 1996. december 17-én készült el. Jelenleg CSS 3-as verziója a legújabb, melyet már 15 éve pontosabban 1999-óta fejlesztenek, így ez a leghosszabb ideig fejlesztett webes szabvány.

A legelterjedtebb böngészők nagy része ma már támogatja a CSS nyelv elemeit, de még mindig vannak olyan böngészők is, amelyek egyes nyelvi elemeket rosszul jelenítenek meg vagy kihagynak, mert nem tudják értelmezni.

### 2.5 Felhasznált szoftverek

A megvalósítás kezdetén localhoston<sup>[3]</sup> dolgoztam így elegendő volt egy editor, melyből számtalan ingyenes programot is találunk, de alapvetően a Windows operációs rendszer is tartalmaz ilyet. Mivel helyi gépen dolgoztam így szükségem volt egy webkiszolgálóra, (előzetes ismereteim alapján) a Wampserverre esett a választásom, mivel ez egyben tartalmazza az Apache2-t a PHP és a MySQL adatbázist, így nem kellett külön-külön beszerezni őket. Másik nagy előnye, hogy lehetőséget ad az adatbázisok és a táblák könnyű kezelésére a PhpMyAdmin segítségével.

Későbbi munkám során szükségem volt egy tárhelyre, amire feltöltve a php fájlaimat elkezdhettem „élni” a weboldalam. Letöltöttem a Notepad++ (választott editor) FTP pluginját, hogy közvetlenül a szerveren lévő fájlaimat tudjam szerkeszteni.

3: Közvetlenül a saját számítógépen felállított webkiszolgáló.

### 3. Böngészők

A jelenleg a piacon lévő, szolgáltatásaikban egymással versengő böngészők a HTML5 és a CSS3 kompatibilitásra 2011-től kezdtek nagyobb hangsúlyt fektetni. A HTML5-öt minden jelentősebb mobil eszköz támogatja. Az egyik fejlesztési tervem a jövőre nézve, hogy ezt a kompatibilitást kifejlesszem, tehát responsívá tegyem a weboldalt (megjelenítő eszköz méretéhez igazodva az optimális megjelenést biztosító megjelenési forma). A CSS-el lehetőségünk volt külön megjelenést definiálni a nyomtatási illetve a képernyőre szánt felületre külön-külön. A CSS3-al azonban nőttek a lehetőségeink. Lehetőségünk van külön CSS kódot beállítanunk a fekvő és az álló tájolásra is az orientáció lekérdezésével. Példa az álló tájolásra.

```
@media screen and (orientation:portrait)
```

Példának okáért az Iphone készülékek nem támogatják a media query lekérdezéseket. Így ha Iphone-ra akarunk optimalizálni, akkor a következő beállításokat kell elhelyezni a CSS fájlban álló tájoláshoz.

```
@media screen and (max-width: 320px)
```

Szakdolgozatom készítése során vált fejlesztésiből ajánlott státuszba <sup>[4]</sup> a HTML5 szabvány. A leginkább használt böngészők: a Google Chrome, az Internet Explorer, a Mozilla Firefox és a Safari. Ezt az első táblázat támasztja alá, mely adatai a W3School honlapjáról származnak és a böngészők használatának részesedését mutatja a világon.

| 2014              | Chrome | Internet Explorer | Firefox | Safari | Opera |
|-------------------|--------|-------------------|---------|--------|-------|
| <b>Július</b>     | 59.8 % | 8.5 %             | 24.9 %  | 3.5 %  | 1.7 % |
| <b>Augusztus</b>  | 60.1 % | 8.3 %             | 24.7 %  | 3.7 %  | 1.8 % |
| <b>Szeptember</b> | 59.6 % | 9.9 %             | 24.0 %  | 3.6 %  | 1.6 % |
| <b>Október</b>    | 60.4 % | 9.5 %             | 23.4 %  | 3.9 %  | 1.6 % |

1. táblázat Böngészők használati arány

Forrás: [http://www.w3schools.com/browsers/browsers\\_stats.asp](http://www.w3schools.com/browsers/browsers_stats.asp) [1]

4: <http://www.w3.org/TR/2014/REC-html5-20141028/> [2]

## 4. Az alkalmazás

A dolgozatomban szeretnék megvalósítani egy tárkereső honlapot, amely jelentősen felgyorsítja a keresést azáltal, hogy a megadott információk alapján a program adja meg a leginkább hozzánk illő partnert, illetve a legtöbb ellentétes véleményt alkotó személyt. Az oldalra navigálva rögtön a bejelentkezési képernyőt látják a látogatók, mivel az oldal a felhasználó bejelentkezése nélkül nem használható. Ezt követően a profilszerkesztés menü pont alatt megadhatják a személyes adataikat. Kitölthetik a személyes kérdőívüket. Megtekinthetik a legtöbb közös érdeklődési körrel rendelkező partnereiket továbbá üzeneteket küldhetnek a felhasználóknak, majd személyesen is találkozhatnak.

### 4.1 Szoftver-folyamat tevékenységei

A szoftver folyamat olyan tevékenységek és a hozzájuk kapcsolódó eredmények egymás utáni sora, amelyek lehetővé teszik a követelmények meghatározását és egy a követelményeket kielégítő szoftver létrehozását.

#### Négy alapvető tevékenységre bontható fel a szoftver folyamat

1. Specifikáció:  
Itt történik a követelmények teljes és következetes leírása. A szoftver által elvárt feladatok rendszerbe foglalása és meghatározása.
2. Tervezés és implementáció:  
Tulajdonképpen ez a specifikáció futtatható rendszerré való átalakítása. Magában foglalja a szoftver tervezését, megvalósítását, vagyis a programozást és az esetleges specifikációbeli finomításokat is. Ide tartozik többek között az architekturális tervezés, az interfész tervezése, adatszerkezet tervezés és a tesztelés.
3. Validáció  
A verifikációt és a validációt (V&V) általánosan szoftvervalidációnak nevezzük. Itt bizonyosodunk meg róla, hogy a termék megfelel-e a specifikációban leírtaknak és, hogy „megfelel-e a célnak”.
4. Evolúció  
A használatba vett szoftvert az elvárásoknak megfelelően bővíteni, javítani, változtatni kell, ez a leghosszadalmasabb tevékenységi egység a kivitelezés során.

A fejlesztésekhez szükséges erőforrások volumenének és a fejlesztések számának növekedése igényelte a fejlesztési folyamatok ésszerűsítését. A fejlesztés gyorsabb és egyszerűbb végrehajtását ennek érdekében különböző modellek jöttek létre. Vannak statikus és dinamikus modellek is, utóbbira példa lehet a Scrum. Statikusok közül talán a két legismertebb a vízesés (Waterfall) vagy az evolúciós modell (Evolutionary Prototyping). A leírt négy fő és a hozzájuk tartozó altevékenységek egy általános szoftver létrehozását adják meg. A webalkalmazásom adottságaiból, összetettségéből kifolyólag a fentebb leírt lépéseket nem lehet jól elkülöníthető egységekre bontani, mert nem igényelt olyan mélységű tervezést ahol a lépések élesen szétválának, így a leírt egységeket rugalmasan kezelve kezdtem el a munkát. Ez egy ilyen kisebb projekt során megengedett, de egy nagyobb komplexitású program során ezek a lépések elvárhatóak.

### 5. Specifikáció

Ebben a fejezetben határozom meg, hogyan is kellene működnie a rendszernek, milyen szolgáltatásokat kell nyújtania és a bemeneti adatokra, hogyan kell reagálnia, valamint, hogyan kell viselkednie az egyes helyzetekben.

Meghatározom milyen használati (use-case) esetek fordulhatnak elő az egyes felhasználókkal, milyen funkciókat érhetnek el és milyen módon használhatják a rendszert. Ezen esetek leírása megkönnyíti a specifikáció második fázisát az által, hogy minden elvárt funkciót ismerek, amit a megjelenési felületen keresztül a felhasználók elérhetnek. Az itt leírtaknak megfelelő megvalósításokat a hatodik fejezetben fogom részletesen bemutatni.

#### 5.1 Regisztráció

Az alkalmazásnak biztonságosan kell tudnia tárolni a felhasználók adatait és biztosítani akarom, hogy csak a regisztrált tagok láthassák egymást és léphessenek egymással kapcsolatba. A felhasználó neveknek egyedieknek kell lenniük, a sikeres regisztrációhoz pedig e-mail aktiváció szükséges. Azért tartottam ezt a lépést szükségesnek, hogy valóban csak azok az emberek regisztráljanak, akik használni is akarják az alkalmazást. A jelszavak tárolásának felfedése még az adatbázisban is elfogadhatatlan, így szükséges a jelszavak eltárolásának a titkosítása.

#### 5.2 Bejelentkezés

A sikeres regisztrációt követően lehet bejelentkezni az oldalra a megadott felhasználónév és jelszó párossal, ha valamelyik mező üresen marad vagy nem illik össze a két adat, akkor ezt jelezni kell a felhasználó felé.

#### 5.3 Bejelentkezés utáni funkciók

##### **Kezdőlap**

Csupán a regisztrált felhasználók kis képei jelennek meg, a regisztráció sorrendjében, és a menü sáv, amelyen választhatnak az egyes funkciók közül.

##### **Profil szerkesztése**

Itt lehetősége van a társskeresőnek megadni a személyes adatait, valamint profilképet is feltölthet, de ez utóbbi opcionális. Ebben a funkcióban is jelezni kell a felhasználó felé az esetleges hiányosságot és/vagy az adatok szerkesztésének sikerességét.

### **Kérdőív**

Ebben a menüpontban lehet kitölteni a kérdőívet. Véleményt formálhatunk az egyes kategóriákba tartozó jellemzőkről, melyeket a későbbi feldolgozás érdekében letárolunk az adatbázisban.

### **Párom**

Ennek a lehetőségnek a leírását két pontba szedem.

- E menüpont alatt láthatjuk azt a felhasználót, aki a kérdőív alatt megadott válaszok alapján a leginkább hasonlít hozzánk. Meg kell jeleníteni a közös véleményekhez vonatkozó tulajdonságokat. A vizuálisabb típusú felhasználóknak kedvezve kirajzoltattam a Google Chart segítségével egy oszlopdiagramot, mely a kategóriáknak megfelelően alakul.
- A „Kíváncsi vagyok, ki lenne az ellentétem” választógomb választása esetén, ugyanazok a beállítások érvényesek, mint a leginkább hozzám illőnél, azonban itt a program azt a felhasználót adja meg, akivel a legtöbb ellentétes vélemény volt.

### **Üzenet**

Erre a pontra kattintva két lehetősége van a felhasználónak

- Új üzenetet kezdhet, ekkor lát minden regisztrált társkeresőt.
- Másik opció a Beszélgetések, itt egy-egy userrel kezdeményezett beszélgetéseket láthatja, rákattintva egyre megjelenik maga a beszélgetés.

### **Tagok**

Itt az összes regisztrált felhasználót láthatjuk, ha be vagyunk jelentkezve.

### **Admin**

Ez a menüpont csak akkor jelenik meg, ha a bejelentkezett felhasználó 'level' értéke '1' azaz Admin.

A következő módosításokat hajthatja végre a felhasználók között.

- Látja az összes felhasználót.
- Egy adott felhasználót aktiválhat, deaktiválhat és törölhet.

- A jövőben szeretném bővíteni, ezt a funkciót úgy, hogy az admin láthatja az üzeneteket is. Valamint közvetlenül erről a felületről küldhet személyeknek üzeneteket.

### **Kijelentkezés**

Ez a menüpont az utolsó, amit megvalósítok a szakdolgozatom elkészítése során. Értelemszerűen itt tudja magát kijelentkeztetni a felhasználó. Ha az ember bejelentkezik egy idő után ki is akar lépni, ha nem kíván tovább az oldalon maradni és nem akarja, hogy más emberek ugyanazon a számítógépen bejelentkezve legyenek a személyes profiljával. Erre minden esetben lehetőséget kell biztosítani egy ilyen jellegű alkalmazásnál.

## 6. Grafikus terv

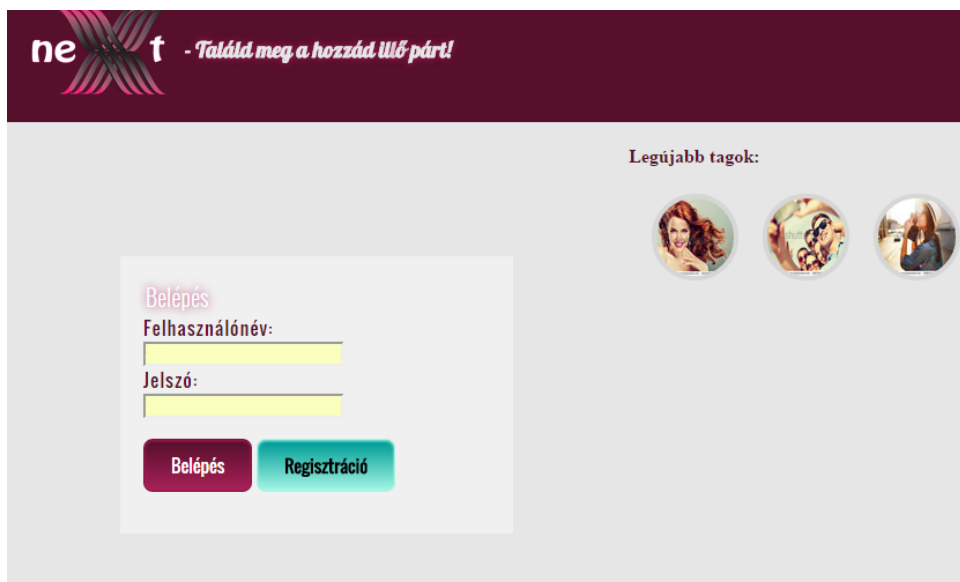
Társkereső honlaphoz híven igyekeztem barátságos, fiatalos, összeillő színeket találni, így egy olyan színekombinációra esett a választásom, melyet az alábbi linken találtam <https://color.adobe.com/Viximo-color-theme-4056754/> [3]



1. ábra Szín kombináció

Úgy alakítottam a weboldal kinézetét és összeállítását, hogy a későbbi szerkesztések a lehetőségekhez mérten a legegyszerűbbek legyenek. Így külön fájlba tettem a fő részeket például header.php, footer.php, navmenu.php, valamint a különböző menük fájljai is külön vannak. Továbbá külön fájlba van a kinézetért felelős css kódsor.

Az oldalra navigálva a bejelentkező képernyő fogad, melyen a legújabb tagok láthatóak valamint a login form. Az oldal logóját Adobe Illustrator programmal készítettem. (2. ábra).



2. ábra Bejelentkezés

A következő ábra az admin felület mutatja, ahol lehetőségünk van a felhasználók kezelésére, törlésére, deaktiválására. (3. ábra). A lehetőségeket a jövőben szeretném bővíteni.

## Webalkalmazás készítése PHP nyelven

Felhasználó beállítások

| Felhasználók | Beállítások                 | Törlés                 | Email                  | Kép                                                                                 |
|--------------|-----------------------------|------------------------|------------------------|-------------------------------------------------------------------------------------|
| ellentet     | <a href="#">Deaktiválás</a> | <a href="#">Törlés</a> | demo@gmail.com         |  |
| ellenno      | <a href="#">Deaktiválás</a> | <a href="#">Törlés</a> | kovacsdani04@gmail.com |  |
| demo         | <a href="#">Deaktiválás</a> | <a href="#">Törlés</a> | kovacsdani04@gmail.com |  |

3. ábra Admin funkciók

## 7. Adatbázis felépítés

Az adatmodell kialakítása egy nagyon fontos lépés egy ilyen alkalmazás elkészítése során, tehát ebben a fejezetben meghatározom az adatbázisban tárolt adatok típusait, méreteit és nem utolsósorban az adatmodell felépítését.

### Szükséges lépések

1. Az adatbázis céljának a meghatározása

El kell dönteni és meg kell határozni, magának az adatbázisnak a használati módját, a tárolandó adatok méreteit, típusait és azt, hogy milyen műveleteket fog elvégezni az adatbázis.

2. Táblák meghatározása

Az összegyűjtött adatokat témakörökre kell bontani, és ajánlott több táblába elhelyezni az összetartozó adatokat, ez a normalizálás. Ez a későbbi lekérdezéseket összekapcsolásokkal bonyolítja, de így sokkal rendszerezettebb és átláthatóbb a rendszer. Valamint el kell kerülni a többszörös adatbevitt, a redundanciát.

3. Mezők meghatározása

Ez az a lépés, amikor el kell döntenünk a táblák egyedeiről, hogy mit kell tudnunk róluk és milyen jellemző adataikra lesz szükségünk. Ezek ismeretében kell meghatározni a mezőneveket, amik lehetőség szerint beszédesek legyenek. Minden táblának tartalmaznia kell egy mezőt, amely egyértelműen azonosítja a tábla rekordjait, ennél fogva betöltheti az elsődleges kulcs szerepét. Tehát ajánlott felvenni egy ilyen mezőt, amit Auto Increment-re (AI) állítunk (egyszerű sorszámozás) ezzel biztosítva van a rekordok egyértelmű azonosítása.

4. Kapcsolatok meghatározása

El kell dönteni, hogy milyen táblákból akarunk adatokat kinyerni továbbá, hogy ezek között a kapcsolatok biztosítva vannak-e. Ezt kapcsoló mezőkkel tudjuk biztosítani. Ezek a mezők a szülő táblában az elsődleges (Primary Key - PK) kulcsok, a hozzá kapcsolt mezőben (gyerekekben) pedig idegen kulcsok (Foreign Key - FK). Ez utal a szülő tábla elsődleges kulcsára. Az adatbázis felépíthető úgy is, hogy a kulcsok egyezősége kötelező legyen. Ez a hivatkozási épség, ami annyit jelent, hogy minden kulcshivatkozásnak érvényesnek kell lennie.

### 7.1 tars\_user tábla

Az alkalmazás személyes adatokra támaszkodik, ezeken az adatokon keresztül épít fel kapcsolatokat a felhasználók között. Adataik nyomon követését és tárolását a tars\_user tábla hajtja végre. A személyes adatokat tároló rekordokat nem sorolom fel, mert a tartalmukra nevükből lehet következtetni és mindegyiknek a típusa VARCHAR.

| Mezők                  |             |       |                                  |
|------------------------|-------------|-------|----------------------------------|
| Oszlopnév              | Oszloptípus | Hossz | Tartalma                         |
| <b>join_date:</b>      | DATETIME    |       | Regisztráció ideje               |
| <b>level:</b>          | INTEGER     | 11    | User határköre                   |
| <b>type:</b>           | VARCHAR     | 1     | Aktív vagy deaktív               |
| <b>confirmed:</b>      | INTEGER     | 11    | Aktiválta-e a regisztrációját    |
| <b>confirmed_code:</b> | INTEGER     | 11    | Az aktiváló linkhez generált kód |

### 7.2 tars\_topic tábla

Az alkalmazás adatainak az alapját ez a tábla képezi. Ebben a táblában lévő paraméterek azok a témák, amelyekre a tetszésüket vagy nem tetszésüket fejezhetik ki a felhasználók.

| Mezők            |                     |       |                            |
|------------------|---------------------|-------|----------------------------|
| Oszlopnév        | Típus               | Hossz | Tartalma                   |
| <b>topic_id:</b> | INTEGER, PK, AI, NN | 11    | Elsődleges kulcs           |
| <b>name:</b>     | VARCHAR             | 255   | Témák nevei                |
| <b>kat:</b>      | VARCHAR             | 255   | Kategóriák nevei           |
| <b>kat_id:</b>   | INETEGER, FK        | 11    | Hivatkozás a másik táblára |

### 7.3 tars\_cat tábla

Ez a tábla tartalmazza a különböző kategóriákat, amiken belül - de másik táblában - vannak az eldöntendő választ igénylő tulajdonságok. Az itteni elsődleges kulcsok, az előző tars\_topic tábla idegen kulcsai állnak egymással kapcsolatban.

| Mezők              |                 |       |                  |
|--------------------|-----------------|-------|------------------|
| Oszlopnév          | Típus           | Hossz | Tartalma         |
| <b>category_id</b> | INTEGER, AI, NN | 11    | Elsődleges kulcs |
| <b>name</b>        | VARCHAR         | 255   | Kategória nevek  |

#### 7.4 tars\_kviz tábla

Ez a tábla kapcsolja össze a ``user_id`` és a ``topic_id`` oszlopok alapján a felhasználókat az egyes kérdésekre adott személyenkénti válaszokkal. Amik alapján végül kiolvassuk az egyezőségeket.

| Mezők              |                     |       |                  |
|--------------------|---------------------|-------|------------------|
| Oszlopnév          | Típus               | Hossz |                  |
| <b>response_id</b> | INTEGER, PK, AI, NN | 11    | Elsődleges kulcs |
| <b>user_id</b>     | INTEGER             | 11    | User azonosítók  |
| <b>topic_id</b>    | INTEGER             | 11    | Téma azonosítók  |
| <b>response</b>    | TINYINT             | 11    | Válaszok         |

#### 7.5 messages tábla

Ez a tábla a személyes üzenetek tárolására lett létrehozva.

| Mezők             |                     |       |                          |
|-------------------|---------------------|-------|--------------------------|
| Oszlopnév         | Típus               | Hossz | Tartalma                 |
| <b>id</b>         | INTEGER, PK, AI, NN | 11    | Elsődleges kulcs         |
| <b>group_hash</b> | INTEGER             | 11    | Beszélgetés egyedi kódja |
| <b>from_id</b>    | INTEGER             | 11    | Küldő user_id értéke     |
| <b>message</b>    | VARCHAR             | 300   | Üzenet szövege           |

#### 7.6 messages\_goup tábla

Két személy között felépülő beszélgetés „útvonalának” tárolására szolgál. Szerepe, hogy a már kiépült csatornát két felhasználó között ne a messages táblában legyen tárolva.

| Mezők           |         |       |                        |
|-----------------|---------|-------|------------------------|
| Oszlopnév       | Típus   | Hossz | Tartalma               |
| <b>user_one</b> | INTEGER | 11    | Küldő user_id-ja       |
| <b>user_two</b> | INTEGER | 11    | Címzett user_id-ja     |
| <b>hash</b>     | INTEGER | 11    | Üzenethez generált kód |

### 7.7 tars\_level tábla

A felhasználók két kategóriába tartoznak. Az egyik az admin a másik a felhasználó, ez a tábla csupán a megkülönböztethetőség miatt lett létrehozva

| Mezők       |         |       |                        |
|-------------|---------|-------|------------------------|
| Oszlopnév   | Típus   | Hossz |                        |
| <b>id</b>   | INTEGER | 11    | Egyedi azonosító       |
| <b>name</b> | VARCHAR | 11    | Felhasználó szint neve |

## 8. Séma megtervezése

Ebben a fejezetben ismertetem az adatbázis felépítéséhez szükséges ismereteket. Majd részletezem a táblákat és végül modellezem az adatbázis felépítését.

### Háromféle kapcsolat lehetséges

1. Egy az egyhez (one-to-one)

Egy 'A' egyedhalmaz minden eleméhez legfeljebb egy elem tartozhat a 'B' (gyermek) egyedhalmazban, és a 'B' egyedhalmaz egyes eleméhez is csak egy egyed tartozhat az 'A' egyedhalmazban.

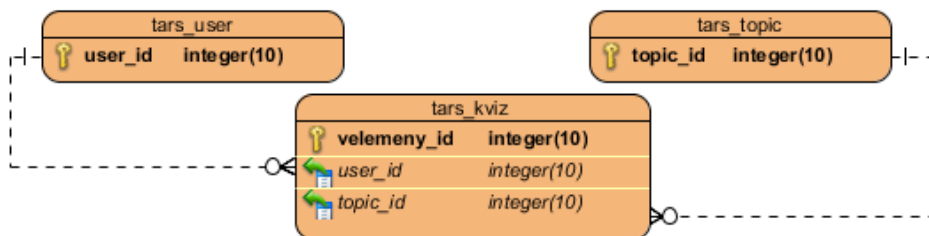
2. Egy a sokhoz (one-to-many)

Az 'A' tábla egy rekordjához több rekord tartozhat a 'B' táblában, de ez fordítva nem igaz. Ezt ki is használja az alkalmazás ugyanis egy felhasználó több témáról is véleményt formálhat.

3. Sok a sokhoz (many-to-many)

Az 'A' tábla több adatsora áll kapcsolatban a 'B' tábla több adatsorával, és a 'B' tábla valamely rekordjához több rekord tartozhat az 'A' táblában. Ennek a kapcsolat fajtának a megvalósítása csak úgy lehetséges, hogy definiálunk egy kapcsoló táblát. Mely tartalmazza az 'A' és a 'B' tábla elsődleges kulcsaira mutató idegen kulcsokat.

Tulajdonképpen a több a többhöz kapcsolat két egy a többhöz kapcsolat egy harmadik tábla részvételével. Az alábbi képen ezt a fajta kapcsolatot demonstráló kapcsolati diagram látható az alkalmazásomból. (4. ábra)



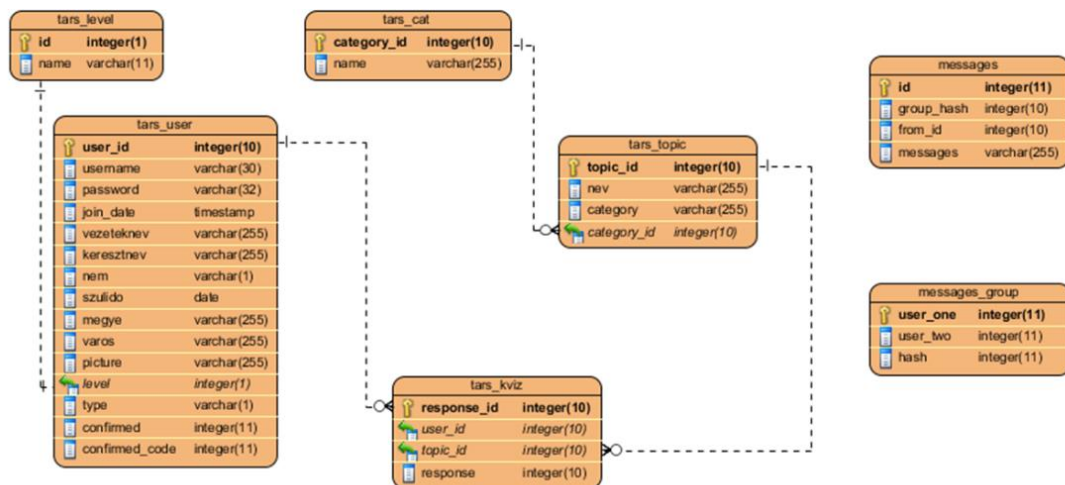
4. ábra Több a többhöz kapcsolat

A tars\_user táblában lévő `user\_id` Primary Key szülőjének tekinthető a tars\_kviz táblának, mely ugyancsak `user\_id` nevű rekordot tartalmaz, de itt Foreign Key.

Így ez a két tábla egy a többhöz kapcsolatban áll.

A tars\_topic tábla ugyancsak a szülője a tars\_kviz táblának. A `topic\_id` rekord Foreign Key és Primary Key alapján így ezek is egy a többhöz kapcsolatban állnak. Mindezek által a tars\_topic tábla és a tars\_user tábla több a többhöz kapcsolatban állnak egymással a tars\_kviz tábla révén.

Mellékletként csatolom a kapcsolati modellt, nagyobb méretben is. (5. ábra)



5. ábra Kapcsolati modell

## 9. Kódolás

Ismertetem a program írás során alkalmazott technikát és kiemelek pár kódsort, amelyet időigényesebb volt megalkotni és olyat, amelyet érdemesnek gondolok részletezni.

### Regisztráció

Magához a regisztrációhoz csak egy felhasználónevet, email címet, valamint egy jelszót kell megadni.

1. A signup.php fájlban ellenőrzést végzek a nick név egyediségére és arra, hogy nem maradt-e üresen a mező.
2. Elhelyeztem egy tooltip ikont, ami segítséget adhat a felhasználóknak a helyes jelszó megadásában.
3. Valamint egy kis visszajelzést, hogy milyen erősségű jelszót adott meg, ehhez a jquery.pwdMeter.js fájlt használtam mely ingyenesen letölthető és használható a <https://github.com/shouvikchatterjee/pwdMeter> oldalról.[4]
4. Az email cím szintaktikai ellenőrzésére a következő függvényt használtam

```
function isemail($email) {  
    $regex="^[a-z0-9_\\-]+(\\.[_a-z0-9\\-]+)*@[[_a-z0-9\\-  
]+\\.)+([a-  
z]{2}|aero|arpa|biz|com|coop|edu|gov|info|int|jobs|mil|museum  
|name|nato|net|org|pro|travel)$";  
    return preg_match("/$regex/i", $email);  
}
```

A *regex* a Regular Expression rövid formája. Magyarul a leghelyesebb fordítás a „szabályos kifejezés” lenne. A RegEX lehetőséget ad szabályok, minták egyszerű leírására. Sokféle felhasználási módja van, én itt adatellenőrzésre használtam.

#### Magyarázat

- `^[a-z0-9_\\-]`  
Alfanumerikus, kis-nagy betűk és a kötőjel is megengedett
- `+(\\.[_a-z0-9\\-]+)`  
Ezeket követhetik ugyanilyen karakterek plusz a pont is megengedett

- @  
Kukac karakternek kell következnie
  - Ez utáni zárójelpáros az ismétlődést teszi lehetővé például  
nem csak a user@domain.com megengedett hanem a  
user@domain.host.com.
5. Ha az e-mailben kapott linkre kattintva aktiválja a fiókot a felhasználó, akkor egy visszajelzést kap a képernyőre, hogy sikeres volt az aktiváció. Itt van egy link is, amelyre ha kattint, akkor automatikusan be is lesz léptetve.
6. A jelszó megadására három elvárást határoztam meg.  
A hossza négy és tizennégy karakter közé essen ezt az strlen() függvénnyel határoztam meg. Valamint kétszer kell megadni a jelszót és egyeznie kell a kettőnek. Nem megengedett még adatbázisban sem, a felhasználók jelszavának a felfedése. Ennek megfelelően titkosítást alkalmazok az SHA() függvény segítségével. Mely egy titkosítási algoritmus, ami negyven hexadecimális karakterből áll. Az SHA egy MySQL és nem PHP függvény ezért annak a részeként kell meghívni. A függvény nem teszi lehetővé a visszafejtést ennek célja az adatok biztonságának a megőrzése. Ezáltal az INSERT utasítás VALUES zárójel részében a jelszó beszúrásánál SHA(password1) értéket kell megadni.
7. A regisztrációs form sikeres kitöltése után bekerül a személy az adatbázisba, de bejelentkezni csak akkor tud, ha az aktivációs link segítségével aktiválja a regisztrációját. Ehhez a PHP mail() függvényét használtam fel.

### Paraméterei

1. \$to  
Regisztrációkor megadott email cím, ahová az aktivációs link küldendő
2. \$subject  
Az üzenet tárgya
3. \$message  
Az üzenet szövege
4. \$headers  
A küldő email címe

### Az aktivációs link felépítése

`domain/filenev.php?username=$username&code=$confirmcode`

A username-nek meg kell egyeznie a regisztrációkor megadott felhasználónévvel. Az & jel utána code-nak pedig egy RAND() függvénnyel előállított számsorral, amely egyedileg azonosítja a felhasználót, ami bekerül az adatbázisba addig, amíg nem kattint a linkre. Miután rákattintott az adott felhasználó *'confirmed'* rekordja 1-re változik és kikerül a *'confirmed\_code'*.

### Bejelentkezés

Meg kell adni a user nevet és a hozzá tartozó jelszót. A bejelentkezési adatokat kiolvasom, majd megtisztítom a trim() függvény segítségével.

Kikeresem a nick nevet, a jelszót és a felhasználó típusát.

```
$query = "SELECT user_id, username,type FROM tars_user WHERE  
username='$user_username'AND password=SHA('$user_password')";
```

Mivel a jelszót tikosítva tároltam és ez nem visszafejthető ezért kell a lekérdezésben is elérni az SHA címkét.

Ha a bejelentkezési adatok helyesek, akkor beállítjuk a bejelentkezési értékeket a \$\_SESSION szuper globális segítségével. Ezt a változót akkor kaphatjuk meg, ha a fájl elején elindítottuk a munkamenetet a session\_start() PHP függvény segítségével. A munkamenetek az adataikat munkameneti változóknak tárolják, ha egy ilyen változóba adatot írunk, akkor az a kiszolgálón tárolódik ellentétben a cookie-kal<sup>[5]</sup>, így ez is egyfajta biztonságot nyújt, de ezeket is tudjuk törölni, mint a sütitket, így biztosíthatjuk a kijelentkezést. A munkamenetnek, mint arra a neve is utal pontos kezdete és vége van, erre csak akkor nincs befolyásunk, ha a felhasználó bezárja a böngészőt, mert ekkor automatikusan megsemmisül. Tehát vagy a browser bezárásával vagy a session\_destroy() függvény használatával semmisíthető meg a munkamenet. Röviden a munkamenetek a sütik kiszolgálóbeli megfelelői.

5: Egy információcsomag, amelyet a szerver küld a böngészőnek, magyarul süti.

Következő lépés, hogy beállítjuk a bejelentkezési adatokat a munkameneti változóba. Egy ellenőrzés, ha a felhasználó *'type'* mezője egyenlő *'d'*-vel akkor nem tud bejelentkezni, mert az admin deaktiválta.

Azért volt szükséges munkamenetek alkalmazására, mert vannak felhasználók, akiknél a süti le vannak tiltva a böngészőben így ez nem lenne járható út. Mivel a munkamenetek csak addig élnek, amíg a böngésző aktuális példánya, ezért ha a belépett user bezárja a böngészőt, akkor ki is jelentkeztetődik. Emiatt kellett azt a megoldást alkalmazni, hogy kihasználjam a munkamenetek biztonságát és a süti maradárságát. Ugyanis a süti van lejárat idejük, így az azokban tárolt adatok mindaddig megmaradnak, amíg ez az idő le nem telik, ezt megadhatjuk napokban vagy akár években is, én itt húsz napra állítottam az élettartamot.

```
setcookie('user_id', $row['user_id'], time() + (60 * 60 * 24 * 20));
```

Végző soron a böngésző bezárása a munkamenetek végét jelenti, de a süti-be tárolt adatok megmaradnak. Így a felhasználó nem lesz kijelentkeztetve a böngésző bezárásával, ha a süti engedélyezettek.

### Profil szerkesztése

Mivel az alkalmazás munkameneteken keresztül azonosítja a felhasználókat az oldalak közti navigáció során, ezért a php fájlok elején includálom a startsession.php fájlt, amiben külön kiemeltem a munkamenet elindítását és a felhasználó azonosítást. Ezt én az include('startsession.php') helyett require\_once('startsession.php') utasítással oldottam meg. A különbség az include és a require között csak az, hogy az include() csak egy figyelmeztetést dob hiba esetén, de az azt követő script lefut a require() pedig végzetes hibát (fatal error) eredményez hiba esetén és a script futása meg áll.

Tehát ha be van jelentkezve a felhasználó, akkor láthatja a form-ot, üresen, mert még nem töltötte ki. Ha már kitöltötte, akkor lekérem az adatait a \$\_POST tömbből és ezeket jelenítem meg neki. Ezeket változás esetén szerkesztheti is. Valamint megadhatja a profilképét. Az ide vonatkozó konstansokat (képek helye, mérete) kiemeltem az appvars.php fájlba. Ellenőrzést végeztem a kép formátumára csak JPEG, PNG, PJPEG és GIF képeket engedélyeztem valamint méretkorlátot is szabtam. Így, ha ezek nem megfelelőek, akkor hibaüzenetet kap a felhasználó.

### Profilom

Itt a felvezetett adatokat tekintheti meg. Itt egy query-vel lekérem a szükséges adatokat és formázottan kiíratom.

### Kérdőív

Minden regisztrált felhasználónak, aki ide navigál meg kell jeleníteni a kérdőívet, ezért, akik még nem töltötték ki azoknak üres válaszokat szúrok be a tars\_kviz táblába. Az array\_push() függvényt használom, mely dinamikusan, elemenként épít fel egy tömböt, lényegében egy új elemet szúr a tömb végére (növeli eggyel) .

Arra használom a programomban ezt a függvényt, hogy a tars\_topic táblában található 'category\_id' azonosítókból egy üres véleményhalmazt generáljak, majd ennek a segítségével szúrom be az üres véleményeket a tars\_kviz táblába témánként.

Ha végül a véleményeket megadták és elmentették, akkor beszúrom, de csak a 'response' mező értéke változik.

### A kérdőív összeállítása

Az ehhez szükséges adatok három különböző táblában vannak eltárolva, amely három lekérdezést igényelne. Szerencsére a MySQL biztosít egy nagyon hasznos lehetőséget ennek elkerülésére, mégpedig az összekapcsolást (join). Talán a joinok közül az inner join a legismertebb (belső összekapcsolás), amely egy feltétel alapján választ ki adatokat két táblából.

```
$query = "SELECT response_id, topic_id, response, tt.name AS  
topic_name, tc.name AS category_name FROM tars_kviz AS tk  
INNER JOIN tars_topic AS tt USING (topic_id) INNER JOIN  
tars_cat AS tc USING (category_id) WHERE tk.user_id = '" .  
$_SESSION['user_id'] . "'";
```

### Az alábbiakat kérem le

1. 'response\_id', 'topic\_id', 'response' rekordokat a tars\_kviz táblából
2. 'tt.name' ez a rekord már egy másik táblában van, ezért INNER JOIN használatával hozzákapcsolom az előző táblához. Hozzárendelek egy alias AS kulcsszóval így

hivatkozhatok rá *'tt'* néven. A *'tt'* tulajdonképpen a *'tars\_topic'* aliasa (álneve) lesz. Ez a későbbi olvashatóságot teszi kényelmesebbé.

3. A USING utasítás segítségével egyszerűsíthetjük a kódunkat úgy, hogy megadjuk annak az oszlopnak a nevét ami alapján az összekapcsolást végre szeretnénk hajtani. A JOIN feltétel definiálja a kapcsolatot két tábla között egy lekérdezésben, mégpedig úgy, hogy adatmezőket kapcsol egymáshoz, Figyelni kell, hogy a JOIN bal oldalán lévő (szülő) táblában a megadott oszlop név elsődleges, míg, amihez hozzá kapcsoljuk, abban idegen kulcsként szerepeljen.
4. Hasonló lépéseket végeztem el a második összekapcsolás során is, csak itt a *'tars\_cat'* táblát kapcsoltam be, amivel elérhetővé tettem a kategória nevek elérését, amik szerint lesz egy-egy kérdés csoport.

A lekérdezés során kapott adatokat a `$responses = array();` tömbben tárolom el. Egy WHILE ciklussal végigmegyek a lekérdezésen. Majd előállítom a megjelenített kérdőívet. Ezért egy foreach ciklussal végigmegyek a responses tömbön. A mezőcsoport (fieldset) addig terjed, amíg az adott kategóriában vannak kérdések, ha elfogytak, akkor új mezőcsoport kezdődik.

Végül kiíratom, az egyes kategóriákon belüli témákat, melyekre radio gombok segítségével adhatják meg a nekik szimpatikus válaszlehetőséget. Az egy mezőcsoportot előállító kódrész kategóriánként lefut, ez teszi lehetővé, hogy ne csak egy kategória kérdéseit írassuk ki.

## Párom

### 1. lépés

Kiolvassuk a párosításhoz szükséges adatokat, de ezek három különböző táblában vannak elhelyezve, így újra az összekapcsoláshoz kell nyúlnunk.

Lekérdezést hajtunk végre a tars\_kviz táblából, melyet rövidebb álnévvé alakítottam és a pont operátorral hivatkoztam egy oszlopára, így lett *'tk.response\_id'* alakú a hivatkozás. A *'tt.name'* egy másik táblában foglal helyet, tehát szükséges egy INNER JOIN a táblára, ahonnan a neveket szeretném lekérni ez a *'tars\_topic'* AS (alias) *'tt'*. A *'tc.name'* a kategóriák nevei, melynek adtam egy alias, hogy értelmezhetőbb legyen a neve a későbbiekben így lett *'category\_name'*, ezt a tars\_kviz táblából kértem le, mely ugyancsak

egy külön tábla ezért összekapcsolásra volt szükség. Végül feltételben megadtam, hogy ez a lekérdezés csak az aktuálisan bejelentkezett felhasználó részéről hajtódjon végre.

```
$user_responses = array();  
while ($row = mysqli_fetch_array($data)) {  
    array_push($user_responses, $row);  
}
```

Az előbbi query egy `'$user_responses'` tömböt állít elő, ami a felhasználó véleményeit tartalmazza, de ezen végig kell menni egy while ciklussal, hogy fel is épüljön a tömb.

### 2. lépés

Itt történik a keresés eredményét elraktározó változók inicializálása, melyeket a keresés egész folyamata során használni fogok a legalkalmasabb pár megtalálása során.

```
$match_score = 0;
```

Ez tárolja a két személy „egyezőségi” pontszámát.

```
$stars_user_id = -1;
```

A `'user_id'`-t -1-re állítjuk, ha ennek a keresés végén is ennyi az értéke, akkor nem találtunk párt (de ez csak akkor lehetséges, ha rajtunk kívül senki más nem formált véleményt egy kategóriáról sem). Vagyis ez a változó tárolja a legtöbb pontszámot elérő személy azonosítóját.

```
$match_topics = array();
```

Ebben a tömbben vannak azok a témák, amikről a két felhasználónak egyező a véleménye.

```
$match_categories = array();
```

Ennek a tömbnek a jelentősége az ötödik lépésben lesz látható

### 3. lépés

Egy lekérdezéssel végighaladunk a `tars_user` táblán, melyben megadjuk feltételnek, hogy az aktuális felhasználó ne számítson. Majd elindítok egy `while` ciklust, és amíg vannak regisztrált felhasználók addig a ciklus feltétele igaz, tehát a következő részek mindig lefutnak. Végül egy lekérdezéssel lekérem a vélemény adatokat, melyeket egy tömbben tárolok el, majd egy `while` ciklussal végig megyek ezen is.

```
$stars_responses = array();
```

### 4. lépés

Még mindig a harmadik lépés első `while` ciklusán belül vagyok, itt számolom az egyezőségek számát. Egy lekérdezéssel lekérem az összes felhasználó véleményét melyet a `'tars_responses'` tömbben tárolok el. Egy `for` ciklussal végig megyek az első lépésben létrehozott `'$user_responses'` tömbön. Addig megy a ciklus, amíg a `'$i'` változó értéke kisebb nem lesz, mint a `'$user_responses'` tömbé. Ezt egy `count()` függvény segítségével nézem, ami a paraméterben megadott több elemeit tartja számon. Végül növelem a változó értékét eggyel.

Mivel a kérdőívben adható két válasz a `'Szeretem'` és `'Inkább nem'` melyek értéke rendre `'1'` és `'2'`. Így az összepárosítás eredményessége a matematikán alapszik, Azáltal, hogy két párkereső véleménye akkor egyezik egy adott témában, ha ugyan azon a kérdéshez `'Szeretem'` vagy `'Inkább nem'` válaszokat adtak tehát a válaszok ugyanahhoz a mezőhöz tartozó két user összeadott eredménye 2 vagy 4.

```
if($user_responses[$i]['response']+$stars_responses[$i]['response']==2||$user_responses[$i]['response']+$stars_responses[$i]['response'] == 4)
```

Ha az első pontban létrehozott `'user_responses'` tömb (mely a felhasználó véleményeit tartalmazza) `'$i'`-edik elemének `'response'` értéke egy vagy kettő, és a `'tars_responses'` (véleményeket tartalmazó) tömb `'$i'`-edik elemének `'response'` értéke, megegyezik. Vagyis (szeretem - szeretem) azaz egyenlő kettővel vagy egyenlő négygel (inkább nem – inkább nem) a számolás elején kinullázott `'$score=0'` értéket növeljük, eggyel. Majd az egyezőségeket egy tömbben tárolom, hogy megmutathassam a felhasználónak.

```
array_push($topics, $user_responses[$i]['topic_name']);
```

Végül megnézem, hogy a mostani pár '*\$score*' értéke jobb-e az eddigi párnál (1. lépés '*\$match\_score*'). Ha igen akkor frissítjük a keresés eredményét, vagyis őt állítom be a legjobb találatnak. Majd az első lépésben inicializált változóknak az értékeit aktualizáljuk.

### 5. lépés

Megjelenítés előtt ellenőrzöm, hogy valóban találtunk-e párt. Ha igen, lekérdezzük az adatait, amiket meg szeretnénk jeleníteni, majd végigmegyünk soronként a lekérdezésen és formázott alakban kiíratjuk. Végül megjelenítjük neki az egyezőségek számát. Egy ciklussal pedig kiíratom a témákat, melyekben egyezik a véleményük.

### 6. lépés

Elkészítem a vizuális megjelenítését az egyezőségeknek, eddig a téma neveket íratam ki, most viszont ehelyett a kategória nevekre van szükségem. Az 1. lépésben lévő query –ben szerepelt ez a rész, ahol a tars\_kviz táblához hozzákapcsoltam a tars\_cat táblát, hogy kinyerhessem a kategóriák neveit. Ennél fogva a '*category\_name*' értékek is bekerültek a '*user\_responses*' tömbbe.

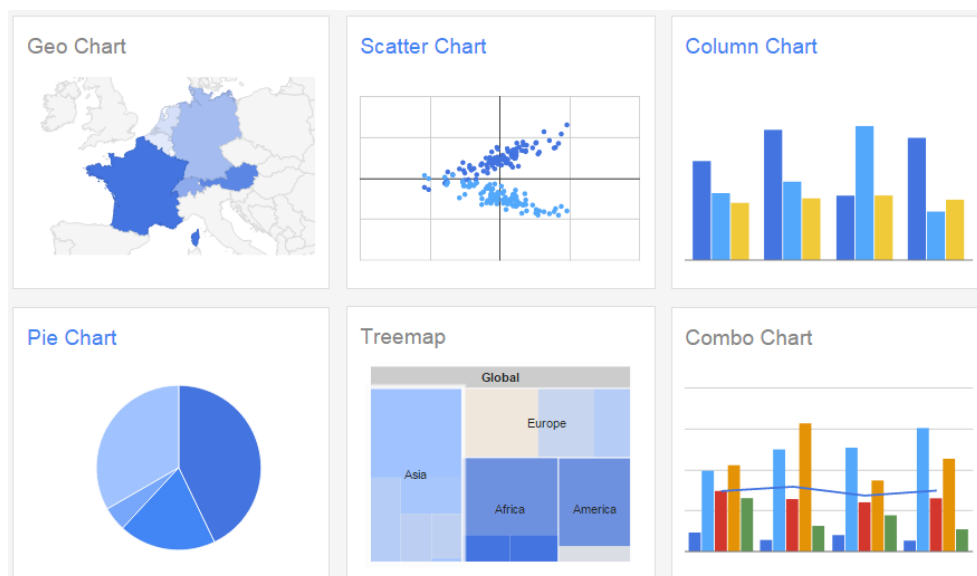
```
(lekérdezés részlet)
JOIN tars_cat AS tc USING (category_id)...
...tc.name AS category_name
```

Még a 3. lépés kódrészénél elhelyeztem az alábbi kódsort, ami tárolja a két felhasználó ellentétes válaszaihoz tartozó kategóriákat és nem a témákat.

```
array_push($categories, $user_responses[$i]['category_name']);
```

### Következzet a diagram felépítése

Ehhez a Google Chart API<sup>[6]</sup> <https://developers.google.com/chart/> [4] eszközét használtam. Ez egy olyan eszköz, ami lehetővé teszi, hogy könnyen teremtsünk diagramokat adatokból és ezeket megjelenítsük a weboldalon. A leggyakoribb táblázat (chart) típusokat tartalmazza. (6. ábra)



6. ábra Google Chart Diagram típusokra példa

A Google Chart egy „egyszerű” JavaScript, amit be kell illeszteni az oldalra. Az adatok megadása és a megjelenés testreszabása után csak egy `<div>` elemet kell elhelyeznünk, amibe a függvény meghívása után kirajzolódik a chart. A meghíváskor a függvény aktuális paraméter listájába a `<div>` id értékét kell megadni. Bővebben a Google Chart dokumentációs honlapján olvasatunk ennek az API-nak a lehetőségeiről és a funkcióiról, ami az alábbi címen érhető el.

<https://google-developers.appspot.com/chart/interactive/docs/index>[5].

Az alkalmazásomban az oszlopdiagram lehetőséget választottam.

A Script a felhasználó böngészőjétől függően SVG<sup>[7]</sup> vagy VML<sup>[8]</sup> vektorgrafikát állít elő. Ha pedig az adott oszlop felé visszük a kurzort, akkor megjelenik az oszlop értéke. A használatához csupán a forráskódban a megfelelő helyre kell illeszteni a scriptet oda, ahol szeretnénk megjeleníteni. Ehhez csupán egy script fájlt kellett beillesztenem az oldalamra. Ezt követően a megfelelő paraméterekkel ellátva máris kirajzolja a diagramot. Lehetőségünk van színeket, átlátszóságot, keretet definiálni.

7: XML alapú leíró nyelv. Macromedia és a Microsoft vezette be.

8: XML alapú leíró nyelv. SVG Working Group fejlesztette ki.

### Üzenet

Ha ide navigálunk először is megjelenik két opció

- Új üzenet

Itt megjelenik minden regisztrált user, nick neve a teljes neve és a profil képe. Egyet kiválasztva pedig bejön egy üzenet ablak ahol a kiválasztott személynek írhatunk privát üzenetet. A küldés gombra kattintva megjelenik egy üzenet „A beszélgetés elkezdődött” ez két dolgot jelent

- A beszélgetések menüpont alatt kezelheti az adott user a beszélgetést, itt vissza is tudja olvasni az előzményeket. Nem kell külön mindig kikeresni az üzenet címzettjét, mert az már a másik menüpont alatt van számon tartva.
- Bekerült a messages táblába az üzenet, küldő *'user\_id'*-ja mint *'from\_id'* és a *'group\_hash'* kód amit rand() függvénnnyel generáltam, hogy jobban nyilván lehessen tartani egy beszélgetést.

- Beszélgetések

Mikor egy személy egy másiknak küldött már üzenetet és a küldő bejelentkezik akkor a már kiépített csatornát a Beszélgetések menüpont alatt láthatja. Az adott felhasználónévre kattintva pedig magát a párbeszédet kapja meg. A csatornát a messages\_group tábla tárolja, valamint a küldő és a fogadó felek *'user\_id'* értékeit és a generált *'group\_hash'* kódot.

### Tagok

E menü pont szolgál a tagok közötti keresésre. Az egyszerűség kedvéért a férfi felhasználók profil képeinek a kerete zöld a nőké bordó, ha nem adtak meg profil képet, akkor is van keret, így egyszerűbb a felhasználók megtalálása.

### Admin

Az admin menü alatt kezelhetjük a felhasználókat. Itt lekérem a felhasználók adatait majd formázottam kiíratom a felhasználó és a személyes nevüket. Egy egyszerű feltétel vizsgálattal íratom ki, hogy az adott usert aktiválni vagy deaktiválni lehet. Ehhez segít az option.php fájl melyben egy-egy query van, ami az adott user *'type'* mezőjét *'a'* vagy *'d'* értékre állítja az interakciónak megfelelően.

Ezenfelül tudom törölni a felhasználót. Itt szintén segít egy másik fájl a delete.php. Ki vannak listázva még az e-mail címek és a profil képek.

### Kilépés

A felhasználóknak itt van biztosítva a kilépési lehetőség. A logout.php fájlba több dolgot is el kell végezni.

- Természetesen itt is el kell indítani a munkamenetet a session\_start() függvénnyel.
- A munkamenetet a session\_destroy() függvénnyel lehet „megölni”, de ilyenkor a munkameneti változók még megmaradnak, ezért egy üres tömböt kell a \$\_SESSION szuper globálishoz rendelni, a változók végleges törléséhez.
- Mivel bejelentkezésnél, amely böngészőkben engedélyezettek a sütik ott létre is lettek hozva ezért azokat is törölni kell, még pedig, úgy, hogy a lejáratí idejüket egy múltbeli időpontra állítjuk ekkor rögtön törlődnek, ha akarjuk.
- Nyomkövető sütik törlése.

```
setcookie('user_id', '', time() - 3600);
```

## 10. Szoftver validáció

Ebben a fázisban a kész programot kell vizsgálni, hogy megfelel-e a specifikációban leírtaknak. A specifikációban leírt összes egységet sikeresen kódoltam, és az egyes komponensek képesek együttműködni így egy rendszert alkotnak.

### 10.1 Megjelenés

Amikor elkezdtem más-más böngészőkben nézni a weboldalt előkerültek a böngésző függő megjelenési különbségek.

A fellépő problémák a következők voltak:

- Kódolás
  - A mezők a kapcsolat és a fájlok karakterkódolását is be kellett állítani, végül ezek eredményezték a megfelelő kódolást
  - utf8\_hungarian\_ci
  - `mysqli_query($dbc,"SET NAMES 'utf8'");`
  - `mysqli_query($dbc,"SET CHARACTER SET 'utf8'")`
  - UTF-8 kódolás BOM nélkül
- Google fontokat használtam a szöveges részeknél
  - linkelnem kellett a HTML <head> részébe
  - olyan fontot kellett keresnem, ami megjeleníti az ékezetes betűket
  - helyettesítő font készleteket kellett megadnom
- A stílusok megadásánál figyelniem kellett a böngésző specifikus beállításokra például színátmeneteket, lekerekítéseket, áttünéseket máshogyan kezelik (2. táblázat).

| Előkód (színátmenet példa)     | Böngésző név                             |
|--------------------------------|------------------------------------------|
| <b>-moz-linear-gradient</b>    | Firefox 1- 3.6                           |
| <b>-webkit-linear-gradient</b> | Safari 3-4, iOS 1- 3.2, Android 1.6- 2.1 |
| <b>-o-linear-gradient</b>      | Opera 9                                  |

2. táblázat Böngésző CSS előkódok

A következő böngészőkben teszteltem a megjelenést és a funkciókat Opera, Firefox, Safari, Chrome és Internet Explorer. Valamint Androidos okostelefonon amelyen a megjelenés hibátlan volt, de mivel nem responsive az oldal így nem vártam csodát, igaz nem lógott ki semmi a képből. Ez annak köszönhető, hogy a szélességeket nem

## Webalkalmazás készítése PHP nyelven

pixeleken, hanem százalékokban adtam meg a stílusokban, azonban a szöveg méret miatt nem volt felhasználóbarát a kezelése.

## 11. Továbbfejlesztési lehetőségek, ötletek

Már a program fejlesztése során felmerültek bennem továbbfejlesztési lehetőségek. A már bejáratott nagyobb költségvetésű, hasonló kategóriájú honlapokat nézegetve is sok új ötletem támadt, azonban, ezeket még nem tudtam implementálni.

A következő felsorolásban megemlítek néhány ilyen potenciált.

- Barát-rendszer kialakítása, ahol lehetőségük van egymást barátnak jelölni ezzel is egyfajta „közeledési” szándékot jelezve.
- Több kép feltöltésének lehetősége.
- Az elsőhöz hasonló funkciót betöltő LOVE gomb, amely minden képnél lenne és annak megfelelően változva a gomb kitöltöttségének a mértéke, hogy hány ember „szavazott” a képre.
- A gyakran ismételt kérdések (GYIK) oldal léthozása. Ahol az oldal üzemeltetése során felmerülő kérdések és válaszok gyűjteménye van olvasható formában.
- Fórum megvalósítása, amelyben a felhasználók nyithatnak chat szobákat.
- Mobilra optimalizálás.
- Java Script alkalmazása. Mindig is el akartam sajátítani ezt a nyelvet. Így a jövőben biztosan használni fogom ebben a projektben is. Adatok ellenőrzésére, hibaüzenetek megjelenítésére, a használhatóság és a vizuális élmény növelésére.
- Szeretném Ajax technikával kibővíteni az oldalt. Ez egy JavaScript és XML nyelveket kombináló interaktív webalkalmazások létrehozására szolgáló webfejlesztési technika.
- Külön tesztet szeretnék létrehozni a két nem képviselőinek. Mikor ezt kitaláltam rájöttem, mennyire összetett ez a kis funkció, és arra is, hogy mennyire javítaná ez a felhasználói élményt.
- Nem csak a végén lehetne eldönteni, hogy a páromra vagy az ellentétpáromra vagyok kíváncsi, hanem külön- külön kategóriánként meghatározom, hogy csak ebben a témában akarom az ellentétemet keresni.

## Webalkalmazás készítése PHP nyelven

- Kategóriák finomítása. Például a szórakozás címke alatt lévő témákat rangsorolja.
- Valamint a jövőben szeretnék egy olyan funkciót létrehozni, ami a teszt kérdések alapján személyre szabottan ad egy rövid leírást a felhasználónak. Hogy milyen emberek illenek a tulajdonságai alapján hozzá, csak ez mélyebb kutatást és elmélyülést igényel. Tesztek, előzetes kérdőívek, személyes kompetenciák megértését, stb. Ez egy jóval hosszabb folyamatot igényel.

## 12. Felhasználói dokumentáció

A felhasználói dokumentáció célja, hogy tartalmazza mindazon szükséges információt, amelyre a felhasználóknak szüksége lehet a rendszer használatba vételéhez. Ezért a szoftver használata előtt a felhasználónak ajánlott áttekinteni ezt a dokumentációt és részletesen megismerkedni vele.

Kérem adja meg az adatokat a regisztrációhoz

**Regisztráció**

Név:

Email:

Jelszó:  ? Gyenge  
Minimum: 4 Maximum: 12 karakter hosszú legyen

**Regisztráció**

7. ábra Regisztráció

A 7. ábrán a regisztrációs folyamatot mutatom be, ahol a felhasználónak csak négy adatot kell megadni, ami egy percet vesz igénybe. A jelszóerősség mérő mutatja, hogy milyen nehézségű jelszót választunk. A kis tooltip pedig segít abban, hogy milyen elvárás van a jelszóval szemben. Helyes email cím megadása után kap a regisztráló egy megerősítő e-mailt. Az ebben lévő linkre kattintva tud belépni, ami egyébként kattintás után automatikusan megtörténik.

8. ábra Profil szerkesztés

A 8. ábra a bejelentkezés utáni profilszerkesztési lehetőséget mutatja. Megadhatjuk a személyes adatainkat, amiket az első kitöltés után meg is változtathatunk. Egy profilképet is tölthetünk fel, de ez opcionális.

A kérdőív (9. ábra) kitöltése nem kötelező, de ajánlott, ha igénybe akarjuk venni az oldal szolgáltatását. Itt kategóriánként látunk eldöntendő válaszokat igénylő tulajdonságokat, témákat. A válaszadáshoz egy kis kérdőjel ad segítséget, ha fölé visszük az egeret.

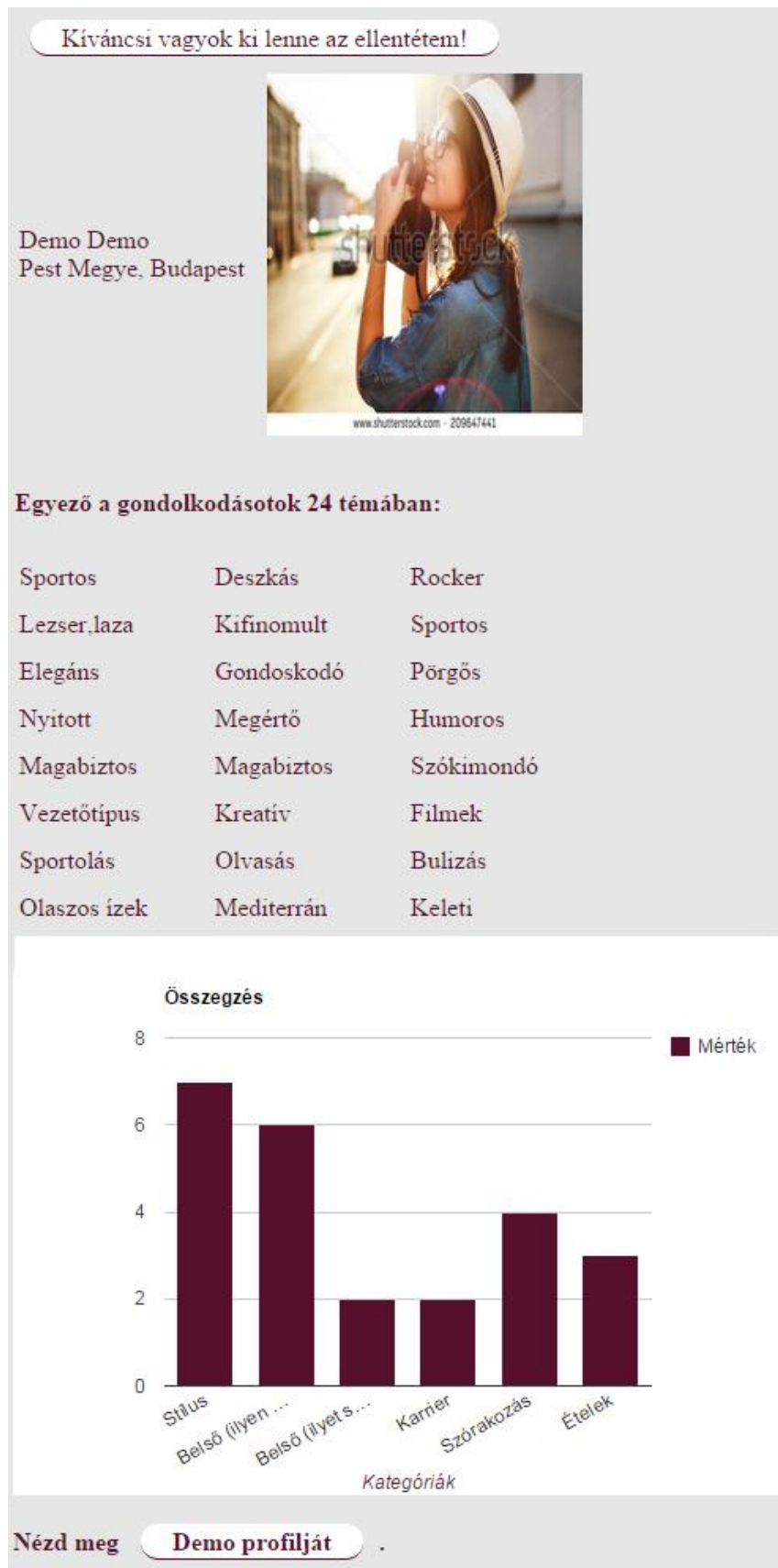
8. ábra Kérdőív

A tagok alatt láthatjuk a regisztrált felhasználókat. A női és a férfitagok képeinek a keretei bordók illetve zöldek. (10. ábra)

| Tagok:                               |                                                                                    |
|--------------------------------------|------------------------------------------------------------------------------------|
| Név                                  | Profil kép                                                                         |
| <a href="#"><u>Admin Admin</u></a>   |   |
| <a href="#"><u>Ellentét Pár</u></a>  |   |
| <a href="#"><u>Ellentépár Nő</u></a> |   |
| <a href="#"><u>Demo Demo</u></a>     |  |

9. ábra Tagok listája

Végül a 11. ábrán láthatunk egy példát a leginkább hozzánk illő személyről. Megjelenik a felhasználóhoz illő személy profilképe. A teljes neve valamint a lakhelye. Megjelenik azon témák nevei felsorolásként amelyekben, egyező a gondolkodásuk a társkeresőknek. A témák nevei alatt láthatjuk az egyezés mértékét. Kategóriánként egy-egy oszlopként ábrázolva, amelynek a mérete megegyezik az adott kategóriában lévő témákra adott egyező válaszok mértékével. Az oszlopok alatt a kategóriák nevei vannak, ha pedig fölé visszük a kurzort, akkor megjelenik a pontos mértéke az egyezésnek. Végül egy hivatkozás az adott felhasználó profiljára. Az oldal tetején van egy lehetőségünk az ellenté párunkra navigálni, ahol a profil és egyéb adatok megjelenése ugyanilyen, de értelemszerűen a témák nevei és a grafikon változik.



10. ábra Párom mutatása példa

### 12.1 Hardver és szoftver igény

Mivel böngészőben futó alkalmazásról szól a szakdolgozat, így minimális hardverigénye van csupán egy internet böngészőre van szükségünk, és egy internet kapcsolatra.

- Javasolt böngésző: Google Chrome
- Ajánlott felbontás: 1024 \* 786

Az alkalmazás telepítéséhez szükséges egy webszerver, egy adatbázis kiszolgáló, és egy php értelmező. Ezen eszközök legegyszerűbb telepítési módja a XAMPP telepítő csomag használata. Ami ingyenesen letölthető a következő címről (a mellékletben is megtalálható).

<https://www.apachefriends.org/hu/download.html> [7]

### 12.2 Telepítés

1. A portálhoz tartozó fájlok felmásolása a szerverre.
2. Az images mappa felmásolása a szerverre
3. A connvars.php fájlban be kell állítani a kapcsolódáshoz használt mysql paramétereket.
4. Az tables.sql állomány futatása az előző lépésben megadott adatbázison belül. A szkript létrehozza a szükséges táblákat és mezőket, valamint feltölti a táblákat példa adatokkal és hozzá ad egy admin felhasználót, amivel tesztelni lehet az admin funkciókat. Ha viszont léteznek már ilyen nevű táblák, akkor először törli azokat.

### 13. Tesztelés

A tesztelés során kell megbizonyosodnom arról, hogy a meglévő hibákat még az üzembe helyezés előtt kiküszöböljük és megbizonyosodjunk arról, hogy az alkalmazás megfelelően működik-e, Teljesíti-e a specifikációban leírt elvárásokat, funkciókat. Mivel a fejlesztés során folyamatosan teszteltem az egyre bővülő funkció listát, így ebbe a fázisba érkezve, már konkrét tapasztalataim voltak a teszteléshez szükséges visszaigazolásokhoz, a korábbi eredmények által.

Ebben a fázisban kell górcső alá venni az élesbe állított rendszert funkcionális szempontból. Ez az alkalmazás használatában, a kívánt lépések megfelelő eredményekkel történő végrehajtásaiban merül ki. A tesztelés előkészítésében egy Fehérdozós (white-box) tesztetet készítettem, ami azt jelenti, hogy a forráskód alapján kerültek kidolgozásra maga a teszt folyamán végrehajtandó lépések.

A tesztelés szintjei a következők:

1. komponens teszt: csak egy komponenst tesztelünk
2. integrációs teszt: kettő vagy több komponens együttes működését teszteljük
3. rendszerteszt: a teljes rendszert teszteljük
4. átvételi teszt: a felhasználók a kész rendszert tesztelik

Az első három szintet együttesen fejlesztői tesztnek nevezzük. Modultesztben a nem-funkcionális elemekre figyeltem, mint például a sebességre. A komponens teszt során a különböző komponensek együttműködését vizsgáltam.

Az integrációs tesztnek akkor van jelentősége, mikor egy programozó csapat fejleszt egy programot és összeillesztik az egységeket, így nálam erre nem volt szükség.

A rendszer teszt során vizsgáltam, hogy a kész alkalmazás megfelel-e a követelmény, a funkcionális specifikációnak és a rendszertervnek.

Szemantikai és szintaktikai ellenőrzést végeztem a forráskódon, hogy a mondatok sorozata megfelel-e a nyelvtannak és érvényesek-e.

A funkcionális teszt során végig próbáltam külön-külön a leírt funkciók működését, az egyes oldalak elérhetőségeit. Olyan eseményeket generáltam, amelyek éles helyzetben is előfordulhatnak.

Lépések

1. Regisztráltam egy felhasználót, itt szándékosan kihagytam a szükséges mezők közül egyet-egyet, hogy láthassam a hibaüzeneteket.

2. Belépve minden funkción és lehetőségen végigmentem, amin lehet felhasználóként.
3. Vizsgáltam a tevékenységek során letárolt adatok helyességét az adatbázisban.
4. Létrehoztam még néhány profilt a további lépések végrehajtásának érdekében.
5. A tesztelés talán leghosszabb fázisa volt, mikor teszteltem a program azon funkcióját mikor adott válaszok alapján meghatározza a hozzánk illő és a leginkább távol álló személyt. Ezt úgy csináltam, hogy bizonyos kérdésekre adtam ugyanolyan illetve ellentétes válaszokat is több user névvel és közben figyeltem mit mutat a program. Külön- külön beléptem az egyes felhasználói fiókokba és ezalatt figyelemmel kísértem (vagyis számoltam), hogy a program jól számol-e.
6. Mindemellett megkértem két barátomat is, hogy regisztráljanak és használják a funkciókat, majd az észrevételeiket juttassák el nekem.
7. Létrehoztam egy adminisztrátor felhasználót és vele teszteltem azon funkciókat amelyek admin jogosultsághoz kötöttek. Nevezetesen töröltem, aktiváltam, deaktiváltam az egyes felhasználókat. Természetesen ellenőriztem ezen akciók végrehajtásának hatásait a táblákban is.

### 14. Összefoglalás

Az alkalmazás megismerésének végére értünk. Bemutatásra kerültek az alkalmazás funkciói, és az ezekben rejlő lehetőség. Az alkalmazás ismertetése során tárgyalásra kerültek a különböző funkciók megvalósulásához szükséges algoritmusok létrehozatalának mikéntjei, és a szükséges adatok tárolásainak specifikációi és magának a tárolásnak a módja. A dolgozat elején bemutatásra kerültek az alkalmazott technológiák, azok sajátosságai, alkalmazhatóságuk.

Az alkalmazásom témájának, azért választottam az online társkeresést, kapcsolatteremtést, mert a mai emberek nem rendelkeznek túl sok szabadidővel, ezt betudhatjuk részben a mai rohanó világnak és egyéb demográfiai okoknak, de ez egy külön szakdolgozatban lehetne kifejtteni. Észrevételeim és kutatásaim szerint az online társkeresés egyre nagyobb teret hódít, a közösségi honlapok szerepéről nem is beszélve. Ezen okok miatt alakítottam ki ezt a weboldalt és emiatt szeretném a jövőben, még „közösségibbé” tenni.

Napjainkban drasztikusan megnőtt az igény a dinamikus weboldalak iránt és várhatóan a jövőben ez az igény növekedési tendenciát fog mutatni. Emiatt is választottam, hogy készítek egy dinamikus weboldalt, melynek a megvalósításához az egyik legnépszerűbb (szerver oldali) nyelvet a PHP-t választottam. A dinamikus weboldalak mindegyikéről elmondható, hogy adatokra támaszkodnak, vagyis adatbázissal állnak kapcsolatban erre a célra a MySQL adatbázis-kezelőt használtam az elkészítés során.

A dolgozatom kezdete óta átfogóbb képet kaptam az alkalmazott technikákról és szélesítettem tapasztalataimat, ismereteimet. A jövőben kívánom az ilyen irányú tudásomat a programmal együtt bővíteni. Reményeim szerint a szakdolgozatom jó ajánló levél lehet a későbbi munkahelyemen, mert e területen szándékozom elhelyezkedni az elkövetkezendő időkben.

## 15. Summary

We reached the end of the presentation of the application. The functions of the application and its possibilities were presented. During the presentation of the application, we discussed how we can create the algorithms which are necessary for realising the different functions, as well as the specifications of storage of the necessary data and the method of storage.

I chose online dating, relationship building as the topic of my application because nowadays people do not have too much free time due to our fast-paced world and other demographic reasons – this could be the topic another thesis. My observations and researches prove that online dating is taking over more and more, not to mention the role of the community websites. These were my reasons for creating this website, and that is why I would like to make it even more „social” in the future.

Nowdays, the demand for dynamic websites has dramatically increased and it is expected to show a continuously growing trend into the future. That is why I chose to create a dynamic website and used the help of one of the most popular (server side) language: PHP. It is commonly true for all dynamic websites that they rely on data; they are linked to a data base, I used MySQL data base management system during my work.

My thesis gave me broader picture of the applied techniques and I could also broaden my experiences and knowledge. In the future, I would like to expand my knowledge into the subject, together with the program.

I hope that my thesis will be a good reference to my future job, as my intention is to work in this field.

## 16. Melléklet

Dolgozatomhoz egy darab CD melléklet tartozik, melynek tartalma a következő:

- **Program** mappa

- Tartalma:
- weboldal forrása
  - forráskód, táblákat, teszt adatok létrehozó sql fájl
  - css, js mappák további szükséges állományok
  - images mappa a szükséges képekkel
  - Xampp telepítő csomag és kapcsolatok.png modell

- **Dolgozat** mappa:

- Tartalma:
- dolgozatom pdf formátumban
  - magyar, angol összefoglaló pdf formátumban

A programom éles környezetben megtekinthető a következő címen:  
<http://ddeal.emages.hu/book/>.

A következő adatokkal lehet bejelentkezni:

### **Teszt adatok**

#### Felhasználó

Felhasználó név: **demo**

Jelszó: **demo**

#### Admin

Felhasználó név: **admin**

Jelszó: **admin**

## **Köszönetnyilvánítás**

Ezúton szeretném megköszönni Szűcs Miklós mérnök tanár témavezetőmnek az egész félév során nyújtott értékes segítségét, észrevételeit, javaslatait.

Miskolc, 2014. november 21.

## 17. Irodalomjegyzék

### Internetes források:

- [1] Böngésző statisztika: [http://www.w3schools.com/browsers/browsers\\_stats.asp](http://www.w3schools.com/browsers/browsers_stats.asp) (2014.10.24.)
- [2] Aktuális böngésző szabvány: <http://www.w3.org/TR/2014/REC-html5-20141028/> (2014.11.01.)
- [3] Színkombináció: <https://color.adobe.com/Viximo-color-theme-4056754/> (2014.09.14)
- [4] PwdMeter.js: <https://github.com/shouvikchatterjee/pwdMeter> (2014.10.14)
- [5] Google Chart: <https://developers.google.com/chart/> (2014.11.10.)
- [6] Google Chart dokumentáció: <https://google-developers.appspot.com/chart/interactive/docs/index> (2014.11.10.)
- [7] Xamp letöltés: <https://www.apachefriends.org/hu/download.html> (2013.02.20.)
- [8] PHP dokumentáció: <http://php.net/docs.php> (2014.09.20)
- [9] Weblabor: <http://weblabor.hu/forumok/php> (2014.10.10.)
- [10] Tutorial.hu: <http://www.tutorial.hu/php-tombok/> (2014.09.30.)
- [11] Jason Lengstorf, Thomas Blom Hansen: PHP for absolute beginners ebook: <http://www.ebooooooooook.com/Upload/PHP%20for%20Absolute%20Beginners%20%282009%29.pdf> (2014.11.06.)
- [12] Ficsor Lajos, Krizsán Zoltán, Dr. Mileff Péter: Szoftverfejlesztés, elektronikus jegyzet, 5. 6. 7. 13. fejezet: <http://www.inf.unideb.hu/kmitt/konvkmitt/szoftverfejlesztes/book.xml.html> (2014.06.10)
- [13] Ficsor Lajos, Dr. Kovács László, Krizsán Zoltán, Dr. Kusper Gábor: Szoftvertesztelés, elektronikus jegyzet, 1. 2. fejezet: <http://www.inf.unideb.hu/kmitt/konvkmitt/szoftvertesztes/book.xml.html#id490923> (2014.11.12.)

### Nyomtatott források:

- [14] Virginia Debolt: HTML és CSS szerkesztés Webszerkesztés stílusosan  
2005 Kiskapu Kft.
- [15] Sági Gábor: Webes adatbázis kezelés MySQL és PHP használatával  
2005 BBS-INFO Kft.
- [16] Lynn Beighley - Michael Morrison Agyhullám: PHP & MySQL

2009 Kiskapu Kft.

[17] Julie C. Meloni: Tanuljuk meg a MySQL használatát 24 óra alatt

2003 Kiskapu Kft.

[18] Matt Zandstra: Tanuljuk meg a PHP4 használatát 24 óra alatt

2001 Kiskapu Kft.

## EREDETISÉGI NYILATKOZAT

Alulírott **Kovács Dániel**; Neptun-kód: **TA4E8D** a Miskolci Egyetem Gépészmérnöki és Informatikai Karának végzős **mérnök informatikus** szakos hallgatója ezennel büntetőjogi és fegyelmi felelősségem tudatában nyilatkozom és aláírással igazolom, hogy

### Webalkalmazás készítése PHP nyelven

című szakdolgozatom saját, önálló munkám; az abban hivatkozott szakirodalom felhasználása a forráskezelés szabályai szerint történt.

Tudomásul veszem, hogy szakdolgozat esetén plágiumnak számít:

- szószerinti idézet közlése idézőjel és hivatkozás megjelölése nélkül;
- tartalmi idézet hivatkozás megjelölése nélkül;
- más publikált gondolatainak saját gondolatként való feltüntetése.

Alulírott kijelentem, hogy a plágium fogalmát megismertem, és tudomásul veszem, hogy plágium esetén szakdolgozatom visszautasításra kerül.

Miskolc, 2014. november 21.

.....

Hallgató