

# **SZAKDOLGOZAT**

**Szamay Szabolcs**

**Debrecen**

**2010**

**Debreceni Egyetem**

**Informatikai Kar**

**Dokumentumok dinamikus megjelenítése a weben**

**Témavezető:**

**Dr. Bujdosó Gyöngyi**

**egyetemi adjunktus**

**Készítette:**

**Szamay Szabolcs**

**programtervező informatikus**

**Debrecen**

**2010**

# Tartalomjegyzék

<b>Bevezetés .....</b>	<b>4</b>
<b>1. Dinamikus weboldalak.....</b>	<b>5</b>
<b>2. Alkalmazott eszközök .....</b>	<b>7</b>
2.1. XAMPP .....	7
2.2. Apache.....	8
2.3. MySQL .....	9
2.4. PHP .....	9
2.5. phpMyAdmin.....	11
<b>3. Basketball Manager .....</b>	<b>12</b>
3.1. A Basketball Manager célja .....	12
3.2. Követelmények .....	12
3.3. Felhasználók, jogosultságok.....	13
<b>4. Az alkalmazás részletes bemutatása .....</b>	<b>14</b>
4.1. Telepítési útmutató .....	14
4.2. Adatbázisok .....	15
4.3. Az alkalmazás működése.....	19
4.3.1. Főoldal, bejelentkezés .....	19
4.3.2. Sikeres bejelentkezés .....	20
4.3.3. Adminisztrátori felület .....	21
4.3.4. Felhasználói felület.....	29
4.3.5. Vendég .....	35
<b>5. Az alkalmazás továbbfejlesztésének lehetősége.....</b>	<b>36</b>
<b>Összefoglalás.....</b>	<b>38</b>
<b>Irodalomjegyzék.....</b>	<b>39</b>
<b>Köszönetnyilvánítás .....</b>	<b>40</b>

## **Bevezetés**

A szakdolgozatom célkitűzése az, hogy létrehozzak és bemutassak egy webes alkalmazást, amelyet Basketball Manager névre kereszteltem el, és egy adott kosárlabda bajnokság eredményeit, statisztikáit dolgozza fel. Kosárlabdázóként sokszor szembesültem azzal a dologgal, hogy a lejátszott mérkőzésekkel kapcsolatos információk a lejátszástól számítva csak pár nappal később kerülnek fel az internetre, amelynek az egyik fő oka, hogy nehézkes, időigényes ezeknek feldolgozása, mivel ennek a munkafolyamatnak több komponense is van. Az általam létrehozott alkalmazással szeretném minimálisra csökkenteni az időt, ami az esemény, és annak publikálása között telik el. Emellett – ami jelentős szempont – megkönnyíti az adminisztrátorok feladatát is, akiknek bonyolult, időigényes munkába telt eddig, hogy az adatokat feldolgozzák, és megjelenítsék az érdeklődők számára. Ennek az alkalmazásnak a segítségével szeretném szemléltetni a dinamikus megjelenítés előnyeit.

Az első fejezetben a dinamikus weboldalak kifejlődéséről, tulajdonságairól lesz szó. A második fejezetben bemutatom a felhasznált eszközöket, technológiákat, és ezek kialakulását. A harmadik részben az alkalmazás céljáról, felhasználói követelményeiről esik szó bővebben. A negyedik fejezetben szemléltetem a Basketball Manager működését, funkcióit néhány ábra segítségével, egyrészt az adminisztrátor, másrészt az egyszerű felhasználó szemszögéből. Majd végül szó esik a továbbfejlesztés lehetőségéről.

# 1. Dinamikus weboldalak

Az Internetet böngészve alapvetően két fajta weboldallal találkozhatunk szembe: a statikus oldalakkal, melyeket akárhányszor is töltünk újra, mindig pontosan ugyanazt az adattartalmat adják vissza, és semmilyen interakciót nem biztosítanak számunkra. Ugyanakkor számos olyan oldal is fellelhető, amelyek tartalma dinamikusan változik, azaz műveleteinktől, a működési környezettől valamint egyéb tényezőktől függően több vagy kevesebb, de változó információt is tartalmaznak.

Ennek a "dinamizmusnak" több fajtája, szintje is létezik. Talán a legegyszerűbb ilyen dinamikus elemek az oldalszámlálót (page countereket) tartalmazó oldalak, melyeknél az oldal tartalmának legnagyobb része általában statikus, csak egy picit, számsort (a letöltések számát) tartalmazó képecske változik rajtuk. Ide sorolhatók a változó reklám-fejléceket tartalmazó oldalak is, hiszen ezek is pontosan ugyanazon az elven működnek, csak éppen számok helyett szöveges, képes, általában önmagában is animált információt jelenítenek meg.

A dinamizmus egy magasabb fokát valósítják meg azok az oldalak, amelyek tartalma ugyan többé-kevésbé folyamatosan változik. Ezek esetében már maga az oldal tartalma is dinamikusan kerül legenerálásra, szemben pl. az oldal-számlálósokkal, ahol maga a HTML kód végig változatlan marad, csak az egyik megjelenített képecske "rajzolata" változik.

A dinamizmus felső fokát azok az oldalak képviselik, melyekkel interaktív beviteli mezőkön (sorszerkesztők, választógombok, nyomógombok, stb.) keresztül mi magunk is közölhetünk adatokat és a visszakapott eredmény (oldal) tartalma a megadott információktól függ. Ezeknek is alapvetően két típusa létezik, amelyek megvalósításukban minimálisan, ám funkcióikban és komplexitásukban nagyságrendekkel különbözhetnek. Az "egyszerűbb" kategóriába a több-kevésbé statikus adatbázison a felhasználó által megadott paraméterek alapján legyűjtéseket/kereséseket végző oldalak tartoznak - tipikusan ilyen alkalmazások az on-line szótárak, a különböző ftp- és web-keresők. Ezeknél sokszor nagyságrenddel nagyobb mennyiségű mögöttes kódot tartalmaznak az interaktív web-alapú játékok, web-alapú home-banking rendszerek, az elektronikus áruházak, valamint az egyéb, sokszor rendkívül specifikus, kliens rétegeként a böngészőt használó ún. 3-rétegű alkalmazások melyek segítségével közvetlenül vagy közvetve egy kisebb/nagyobb adatbázisban módosíthatunk

információkat, programokat futtathatunk, vagy akár bármilyen más, tetszőleges műveleti sort is elvégeztethetünk a távoli számítógépen. Ide sorolhatók a rendkívül sok magáncélú site-on is megtalálható "vendégekönyv" oldalak is, amelyek azonban a legkevesebb esetben üzemelnek dedikált módon a magát az oldalt is tartalmazó szerveren és gyakorlatilag - generalitásukból adódóan - semmilyen szinten nem kötődnek a "beágyazó" oldalhoz, hanem publikus szolgáltatások keretében elérhetőek - hasonlóan az oldal-számálokhoz.

Meg kell említenünk az interaktivitás / dinamizmus megvalósításának egy, a fentiektől alapvetően eltérő formáját is, az ún. szkripteket, amelyek azonban az előbbi megoldásokkal ellentétben a működési logikát a kliens oldalra helyezik át, és bár bizonyos - elsősorban egyszerűbb számítási logikát használó, alapvetően kis adatmennyiségen, de mindenképpen kizárólagosan dolgozó - feladatok esetében rendkívül gyors, effektív és praktikus megoldást nyújtanak.

A dinamikus weboldalak térnyerésével megjelentek a tartalomkezelő rendszerek. A tartalomkezelő rendszerek (angolul Content Management System, CMS ) olyan szoftverrendszerek, amely nem strukturált információk, mint például az internetes portálok akár több felhasználó általi elkészítését, kezelését, és tárolását segíti. Továbbá gondoskodik a tartalmak strukturált megjelenítéséről, statisztikák készítéséről, kiegészítő funkciók integrálásáról. Strukturált információk, dokumentumok közös elkészítését, kezelését, és azok strukturált megjelenítését a DMS (Document Management System, dokumentum kezelő rendszerek) segítik. A CMS általában kereshetőséget, verziókövetést, és sok más szolgáltatást biztosít. A CMS-en igen gyakran, sajnos tévesen, azokat a webes alkalmazásokat értik, amelyek weboldalak, internetes portálok készítésére, folyamatos karbantartására – frissítésére – használatosak. A CMS olyan, általában webalapú program, amely lehetővé teszi változatos adatok általában hypertext formátumban való dinamikus publikálását programozói tudás nélkül. A szó egyébként mindenféle rendszerezett, decentralizált publikálási feladatot ellátó programrendszerre vonatkozik. Ilyen értelemben véve tehát egy vállalati / egyetemi / bármilyen közösségi használatú számítógépes felület. A CMS-ek legfontosabb tulajdonsága a dinamizmus. A tartalmat, annak megjelenését megfelelő jogosultság esetén bármikor, bárholnan meg lehet változtatni. Azokat a tartalomkezelő rendszereket, amelyekre ez nem érvényes, Document Management Systemnek (DMS) nevezzük. Meglehetősen sokféle tartalomkezelő rendszer található a piacon, a sok szabad licencű (és nyílt forrású) CMS

mellett sok a kereskedelmi termék és az egyedi fejlesztés is. A legtöbb CMS a PHP szerveroldali programozási nyelven íródott, és MySQL adatbázist használ. Léteznek ingyenes CMS-ek (például Drupal, e107, Joomla!, PHP-Nuke, PostNuke, magyar vonatkozásban TextPortál), de sok a fizetős is. A szinte univerzális, elterjedt CMS-eken kívül rengeteg kisebb, korlátozottabb tudású CMS létezik. [3.]

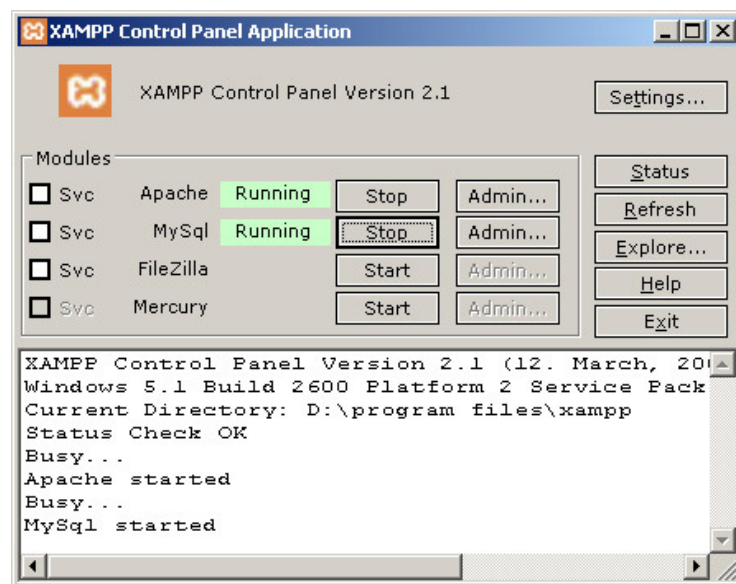
## 2. Alkalmazott eszközök

### 2.1. XAMPP

A XAMPP egy alkalmazás-kiszolgáló programcsomag, amelynek fő elemeit az

- Apache webservert
- MySQL adatbázis-kezelő rendszert
- PHP fordítót
- phpMyAdmin modult

alkotja.



XAMPP Control Panel

A XAMPP előnye, hogy minden lényeges – és emellett ingyenes – kiszolgálót egyesít. Egy csomagban megtalálható a webkiszolgáló, adatbázis-kiszolgáló, FTP-kiszolgáló és a levelező-kiszolgáló, melyeket az ábrán XAMPP Control Panel segítségével könnyedén kezelhetünk. Ezekből mindig a legfrissebb verzió áll rendelkezésünkre, és a frissítéseket is rövid idő alatt telepíteni lehet. Emellett praktikus is, mert könnyű a szükséges beállításokat elvégezni rajta.

## **2.2. Apache**

Az Apache egy nagyon elterjedt, nyílt forráskódú webkiszolgáló, amely számos operációs rendszeren érhető el. A fejlesztők célja az volt, hogy biztonságos, hatékony és bővíthető szerveret hozzanak létre, ami támogatja a HTTP szolgáltatásokat, szinkronban a jelenlegi http szabványokkal, melyeknek nagy része lefordított modulok formájában áll rendelkezésre a mag kiegészítéseként. A virtual hosting segítségével egyetlen Apache installáció ki tud szolgálni több különböző webhelyet is. Az Apache hibaiüzenetei konfigurálhatóak. Számos grafikus felhasználói felület (röviden GUI) támogatja a szerver könnyebb és intuitívabb konfigurálását. Statikus és dinamikus weboldalak közzétételére egyaránt használják. Sok webalkalmazást az Apache által nyújtott környezethez és szolgáltatásokhoz terveznek. Az Apache alkotja a webszerver komponenst az előbbieken említett XAMPP program-csomagban is. A webalkalmazásokat készítő szoftverfejlesztők is gyakran használnak a személyi számítógépükön telepített Apache webszervert a fejlesztett kód tesztelésére. Számos nagy honlap is az Apache webszervert használja kiszolgálóként. Az Apache a mai napig a legnépszerűbb webszerver az interneten, 1996 áprilisa óta. [5.]

## 2.3. MySQL

A MySQL az egyik legnépszerűbb adatbázis-kezelő szoftver. A MySQL nyelv az angol nyelvtan alapján került kialakításra, mert a korábbi tervek szerint ezt a nyelvet nem számítástechnikai szakemberek részére fejlesztették ki. Ennek következtében használata egyszerű, könnyen elsajátítható. Számos platformon futtatható. Minden tulajdonsággal rendelkezik, amivel egy adatbázis-kezelő szoftvernek rendelkeznie kell, és számos más relációs adatbázisrendszerrel ellentétben még a következő képességekkel is: többféle tároló motor, natív tároló motorok, partnerek által fejlesztett tároló motorok, közösségi fejlesztésű tároló motorok, egyéni fejlesztésű tároló motorok, commitek csoportosítása, több tranzakció fogadása többféle kapcsolatról, melyek meggyorsítják a tranzakciók lefolyását. Egyedi illesztőfelületekkel az adatbázis-kezelő elérhető C, C++ , C#, Delphi, Eiffel, Smalltalk, Java, Lisp, Perl, PHP, Python, Ruby és Tcl programozási nyelvvel. [6.]

## 2.4. PHP

- **Története**

A nyelvet eredetileg Rasmus Lerdorf alkotta meg 1994-ben, de a ma létező egyetlen (és hivatalos specifikáció híján *de facto* szabvánnyá vált) PHP implementációt már a PHP Group tartja karban és fejleszti. A PHP a saját licence alatt kerül kiadásra, és a Free Software Foundation szabad szoftverként tartja számon. A PHP fejlődése kezdetén csak CGI-programok halmaza volt. Ezeket Lerdorf néhány Perl szkript lecserélésére írta, amelyeket honlapjának karbantartására (például önéletrajzának megjelenítésére és a látogatottság mérésére) használt. Később ezeket a programokat kombinálta a szintén általa írt Form Interpreter (űrlap-értelmező) alkalmazással - így jött létre a PHP/FI, ami már jóval szélesebb funkcionalitással bírt. Az új, C nyelven megírt változat képes volt adatbázisokhoz kapcsolódni és segítségével egyszerű dinamikus weboldalakat is létre lehetett hozni. Lerdorf 1995. június 8-án adta ki a PHP első nyilvános változatát, hogy külső segítséggel gyorsabbá tegye a hibák megtalálását és a kód további fejlesztését. Az új verziót PHP 2-nek nevezte el és már

megtalálhatóak voltak benne a mai PHP alapvető tulajdonságai: a Perl-éhez hasonló változók, az űrlapok kezelése és a HTML-kód beszúrásának lehetősége. A PHP szintaktikája is hasonló volt a Perl-éhez, de annál jóval korlátozottabb, egyszerűbb és kevésbé egységes volt. 1997-ben Zeev Suraski és Andi Gutmans, két izraeli fejlesztő újraírta az értelmezőt, ezzel megteremtve a PHP 3 alapját - ekkor született meg a PHP új neve, a PHP: Hypertext Preprocessor rekurzív rövidítés is. A fejlesztőcsapat 1997 novemberében, két hónapnyi béta-tesztelés után kiadta a PHP/FI 2-t, majd megkezdődhetett a PHP 3 tesztelése és végül megjelenése 1998 júniusában. Suraski és Gutmans ekkor megint a PHP-mag újraírása mellett döntött, amiből 1999-ben megszületett a Zend Engine. A két fejlesztő ekkor alapította meg a Zend Technologies-t is, ami máig aktívan ellenőrzi a PHP fejlesztését. A Zend Engine 1.0 által hajtott PHP 4 2000. május 4-én jelent meg. Ezt követte 2004. július 13-án a következő nagy mérföldkőnek számító, az új Zend Engine II-n alapuló PHP 5. Az ötös verzió sok újítást tartalmazott: fejlettebb objektumorientált programozási lehetőségeket, a PDO (PHP Data Objects) adatbázis-absztrakciós kiterjesztést, és sok teljesítményt növelő javítást is. A PHP 4 fejlesztése 2007 végén maradt abba, az utolsó megjelent verziója a 4.4.8 verziószámot viselte - a PHP Group ezekhez és a PHP 4 előző változataihoz 2008 augusztusáig biztosít biztonsági frissítéseket. Jelenleg a PHP 5 a nyelv egyetlen aktívan fejlesztett verziója. A hatos verzió bár már fejlesztés alatt áll, még nem jelent meg. 2008. február 4-étől a GoPHP5 kezdeményezés keretében több vezető PHP-alapú szabad-szoftver projekt döntött a PHP 4-támogatás megszüntetése mellett, hogy ezzel mozdítsák előre az ötös verzióra való áttérés folyamatát.

- **Működése**

A PHP az egyik leggyakrabban használt programozási eszköz webes alkalmazások készítéséhez. Könnyen illeszthető szinte bármelyik operációs rendszer kiszolgálójához. A programozási nyelv eszközei, a parancsok szintaktikája nagyban hasonlítanak a C programozási nyelvre, így akik C-ben már tudnak programozni, nem lesz nehéz dolguk a nyelv elsajátításában. A PHP szkriptnyelv, ami azt jelenti, hogy a programunk kódját szinte bármilyen szövegszerkesztővel meg tudjuk írni, a lényeg az, hogy a mentésnél egyszerű szövegfórmátumba, és .php kiterjesztéssel mentjük el az állományunkat a webkiszolgáló megfelelő könyvtárába. A legtöbb HTML szerkesztő program kezeli a PHP kódokat, és

számos olyan van köztük, ami különböző színekkel jelzi, hogy egy parancsot, vagy valamilyen programozási eszközt használunk-e. A szkriptünk fordítását minden alkalommal elvégzi a kiszolgáló szerver, amikor meghívjuk a szkriptet. Ez némileg lassabb futást eredményez, mintha egy korábban lefordított programot futtatnánk, de ezért cserébe hordozható kódot kapunk, ami azt jelenti, hogy a programjaink jelentős része fut majd például Linux és Windows alatt is. Ha a programjainkat körültekintően készítettük el, akkor azok teljes mértékben hordozhatók lesznek. A szkript típusú nyelv nagy hátránya, hogy a kódban megbújó hibára csak akkor derül fény, ha a program futása során arra a kódsorra lépünk, ahol a hiba van, emiatt a program tesztelése nehézkesé válhat. Ha azt szeretnénk, hogy a programunk logikailag is megfelelő módon működjön, akkor lehetőség szerint a program minden ágát le kell tesztelnünk. A böngészőnk számára a PHP kódjaink úgy jelennek meg, mintha azok HTML kódok lennének. A PHP kód átalakításáról a webkiszolgáló gondoskodik. Maga a PHP kód is tartalmazhat HTML elemeket, melyek segítségével tudjuk szövegünket formázni, illetve egyéb műveleteket végrehajtani rajta. Biztonsági szempontból nagyon fontos dolog, hogy a böngészőnk már csak a végeredményt kapja meg, mert így maga a kód nem látszik, és semmi sem utal arra, hogy ott korábban egy másik nyelv kódja állt. A PHP szkripteket, hasonlóan más HTML tag-hez, nyitó- és zárócímkék közé kell zárni. [1.][2.][4.]

## **2.5. phpMyAdmin**

A phpMyAdmin egy nyílt forrású eszköz, amit PHP-ban írtak a MySQL menedzselésére az interneten keresztül. Képes

- készíteni és eldobni adatbázisokat
- készíteni, eldobni, módosítani táblákat
- törölni, módosítani, hozzáadni mezőket
- SQL parancsokat futtatni és a mezőkön kulcsokat kezelni

Képes az egész MySQL szerver kezelésére épp úgy, mint egyetlen adatbázisra.

## **3. Basketball Manager**

### **3.1. A Basketball Manager célja**

Az alkalmazás egy kosárlabda bajnokság mérkőzéseinek adatait dolgozza fel. Ezt olyan módon teszi, hogy az adott mérkőzések után a hazai csapat egyik képviselője - akit az adminisztrátor korábban regisztrált – kitölt egy online adatlapot, és elküldi az alkalmazás számára. Ezt követően a megadott információk kiértékelődnek, az adatbázisok frissülnek, minden szükséges számítás automatikusan végrehajtódik. Az eredmény feltöltését követően azonnal láthatóak lesznek a megfelelő helyeken a frissített statisztikák, mint például a tabella, eredmények, dobólisták. Ebből következően az adminisztrátornak csak karbantartási feladatai lesznek, így jelentősen könnyebb lesz a feladata.

### **3.2. Követelmények**

- az adminisztrátornak teljes jogköre legyen
- az alkalmazás csak az arra jogosultnak engedje meg, hogy módosítson a tartalmi elemeken, ezen belül is csak a saját mérkőzésére vonatkozóakon
- kezelje a bevitt adatokban előforduló esetleges formai hibákat
- biztosítsa, hogy a feltöltött adatok minden szempontból helyesek legyenek
- egy már elküldött eredményt – az adminisztrátor kivételével – ne lehessen felülírni senkinek
- a tárolt adatok rendezett formában történő megjelenítése

### **3.3. Felhasználók, jogosultságok**

#### Felhasználók:

Három típusú felhasználó van:

- Adminisztrátor
- Felhasználó
- Vendég

Az felhasználókat az adminisztrátor hozza létre. Erre csak akkor van lehetőség, ha létre lett hozva az adott bajnokság, és fel lett töltve csapatokkal. Az egyszerű felhasználók neve minden esetben a csapatok neveivel egyezik meg.

#### Jogosultságok:

- Adminisztrátor

Teljes jogköre van, amelybe a következők tartoznak bele:

- Felhasználók létrehozása, törlése, listázása
- Bajnokságok létrehozása, törlése
- Csapatok létrehozása, törlése
- Bajnokságok rögzítése
- Játékosok hozzáadása, törlése, listázása
- Eredmény bevitele (bármelyik)
- Tabellák, eredmények, dobólisták megtekintése
- Jelszó megváltoztatása

- Egyszerű felhasználó
  - Eredmény bevitele (csak a saját csapatára vonatkozó)
  - Tabellák, eredmények, dobólisták megtekintése
  - Jelszó megváltoztatása
  
- Vendég
 

A vendég nem jelentkezik be, csak informálódhat a bajnokságokról.

  - Tabellák, eredmények, dobólisták megtekintése

## 4. Az alkalmazás részletes bemutatása

### 4.1. Telepítési útmutató

Az első teendő a telepítés során, hogy le kell tölteni, és telepíteni kell a 3. pontban említett XAMPP programcsomagot. Ezt legegyszerűbben a <http://www.apachefriends.org/en/xampp.html> weboldalról lehet megtenni, ahol a megfelelő operációs rendszer kiválasztása után le lehet tölteni a legfrissebb verziót. A letöltött fájl futtatásával egyszerűen végig kell menni a telepítésen. Miután ez sikeresen megtörtént, el kell indítani az XAMPP Control Panel-t, ahol kezelni lehet a programcsomag elemeit. Itt annyi a feladat, hogy el kell indítani az Apache-ot és a MySQL-t. A következő feladat az adatbázisok létrehozása és feltöltése. Ezt a phpMyAdmin segítségével lehet megtenni, amelyet egy böngészőt megnyitva lehet elérni el a <http://localhost/phpmyadmin> címen. Itt annyi a teendő, hogy az "Importálás" menüpont alatt az importálandó fájlnak meg kell adni az alkalmazás könyvtárában található db.sql fájlt, amelynek végrehajtása után létrejönnek a szükséges adatbázisok. Ezután már csak az marad hátra, hogy az alkalmazás könyvtárában található fájlokat be kell másolni a C:\XAMPP\htdocs könyvtárba, és a db.inc fájlt meg kell nyitni, és szükségyszerűen módosítani a benne lévő host nevet (ami az XAMPP esetén általában "localhost"), a MySQL felhasználói nevet és a jelszót. Ezt elvégezve készen van a telepítés.

Az alkalmazás elérése a <http://localhost> címen lehetséges. Alapértelmezetten egy adminisztrátor van létrehozva, akinek a következők a belépési adatai:

- Felhasználói név: admin
- Jelszó: admin

## 4.2. Adatbázisok

Az alkalmazás 5 adatbázist használ:

- bajnoksagok

Ez az adatbázis tartja nyilván az egyes bajnokságokon belül a csapatokat, és a tabella szempontjából fontos statisztikai elemeket: csapat neve, lejátszott mérkőzések száma, győzelmek száma, vereségek száma, dobott pontok, kapott pontok, ezeknek különbsége, és a százalékos mutató. Alapértelmezetten az adatbázis üres.

```
CREATE TABLE IF NOT EXISTS `megyei1` (  
  `csapat` varchar(30) NOT NULL,  
  `merkozes` int(11) DEFAULT NULL,  
  `gyozelem` int(11) DEFAULT NULL,  
  `vereseg` int(11) DEFAULT NULL,  
  `dobott` int(11) DEFAULT NULL,  
  `kapott` int(11) DEFAULT NULL,  
  `kulonbseg` int(11) DEFAULT NULL,  
  `szazalek` float DEFAULT NULL,  
  PRIMARY KEY (`csapat`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Egy tábla szerkezete az adatbázisban

- eredmények

Ez az adatbázis tárolja az adott bajnokságokhoz tartozó eredményeket, a mérkőzések adatait: csapatok nevei, dátum, játékvezetők, végeredmény, negyedek eredményei, pályára lépő játékosok nevei, a játékosok által dobott pontok és hárompontosok. Alapértelmezetten ez is üres.

```
CREATE TABLE IF NOT EXISTS `megyei1` (  
  `csapat1` varchar(30) DEFAULT NULL,  
  `csapat2` varchar(30) DEFAULT NULL,  
  `datum` date DEFAULT NULL,  
  `jv1` varchar(30) DEFAULT NULL,  
  `jv2` varchar(30) DEFAULT NULL,  
  `eredmenya` int(11) DEFAULT NULL,  
  `eredmenyb` int(11) DEFAULT NULL,  
  `n1a` int(11) DEFAULT NULL,  
  `n1b` int(11) DEFAULT NULL,  
  `n2a` int(11) DEFAULT NULL,  
  `n2b` int(11) DEFAULT NULL,  
  `n3a` int(11) DEFAULT NULL,  
  `n3b` int(11) DEFAULT NULL,  
  `n4a` int(11) DEFAULT NULL,  
  `n4b` int(11) DEFAULT NULL,  
  `nha` int(11) DEFAULT NULL,  
  `nhb` int(11) DEFAULT NULL,  
  `dobo1a` varchar(30) DEFAULT NULL,  
  `pont1a` int(11) DEFAULT NULL,  
  `hp1a` int(11) DEFAULT NULL,  
  `dobo2a` varchar(30) DEFAULT NULL,  
  `pont2a` int(11) DEFAULT NULL,  
  `hp2a` int(11) DEFAULT NULL,  
  
  ...  
  
  `dobo11b` varchar(30) DEFAULT NULL,  
  `pont11b` int(11) DEFAULT NULL,  
  `hp11b` int(11) DEFAULT NULL,  
  `dobo12b` varchar(30) DEFAULT NULL,  
  `pont12b` int(11) DEFAULT NULL,  
  `hp12b` int(11) DEFAULT NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Egy tábla szerkezete az adatbázisban

- játékosok

Ebben az adatbázisban található az adott bajnokságban szereplő játékosok egyéni statisztikái: név, csapat, összes dobott pont, hárompontosok, lejátszott mérkőzések. Alapértelmezés szerint ez az adatbázis is üres.

```
CREATE TABLE IF NOT EXISTS `megyei1` (  
  `jatekos` varchar(40) DEFAULT NULL,  
  `csapat` varchar(30) DEFAULT NULL,  
  `pont` int(11) DEFAULT NULL,  
  `hpont` int(11) DEFAULT NULL,  
  `meccs` int(11) DEFAULT NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Egy tábla szerkezete az adatbázisban

- lezart

Ennek az adatbázisnak az a funkciója, hogy nyilvántartsa azokat a bajnokságokat, amelyeket az adminisztrátor lezárt. Ez azt jelenti, hogy nem lehet hozzáadni több csapatot az adott bajnoksághoz, és elkezdődhet az eredmények feltöltése. E nélkül egy új csapat hozzáadása esetén hibásan jelenhetnének meg egyes adatok. Az adatbázis szerkezetének nincs különösebb fontossága, az alkalmazás csak a táblák neveit vizsgálja.

- login

Ez az adatbázis a felhasználók nyilvántartását végzi el. Egy táblája van, a "members". Ebben a táblában található meg az összes felhasználó, és a hozzájuk tartozó adatok: azonosító, felhasználónév, jelszó, e-mail cím, bajnokság. Alapértelmezetten egy felhasználó van benne, az adminisztrátor.

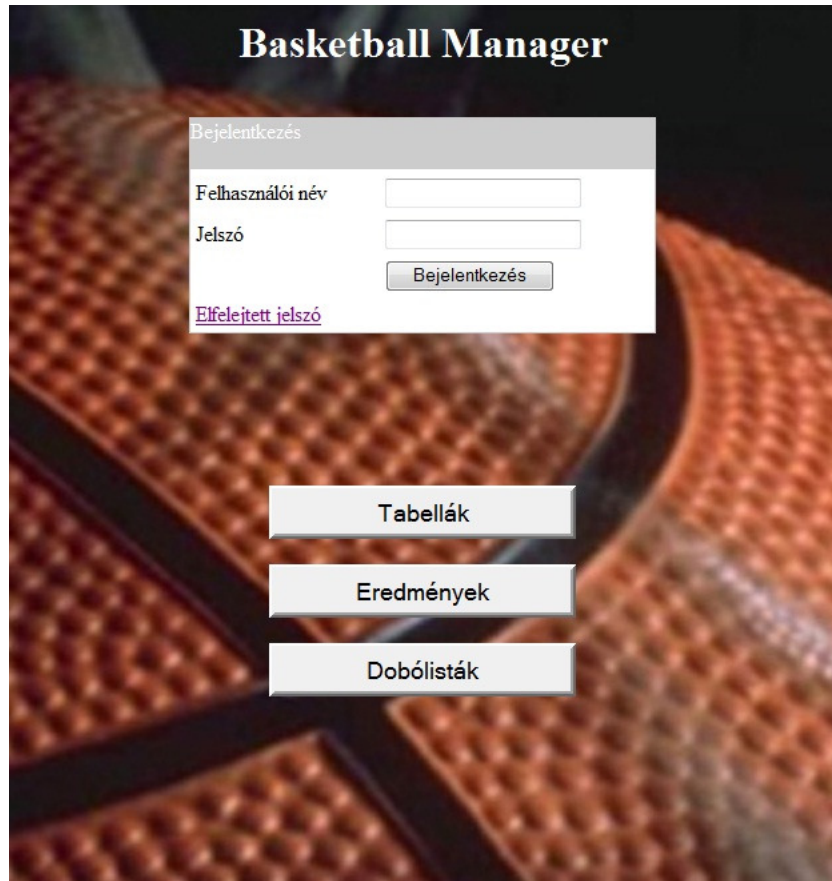
```
CREATE TABLE IF NOT EXISTS `members` (  
  `id` int(4) NOT NULL AUTO_INCREMENT,  
  `username` varchar(65) NOT NULL DEFAULT '',  
  `password` varchar(65) NOT NULL DEFAULT '',  
  `email` varchar(100) NOT NULL,  
  `bajnoksag` varchar(100) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=24 ;  
  
INSERT INTO `members` (`id`, `username`, `password`, `email`, `bajnoksag`) VALUES  
(1, 'ADMIN', 'admin', 'admin@localhost.com', 'admin');
```

**A tábla szerkezete és tartalma**

## 4.3. Az alkalmazás működése

### 4.3.1. Főoldal, bejelentkezés

Az alkalmazást igénybevevő felhasználót a következő oldal fogadja először:



Főoldal

Itt egy bejelentkezési ablak látható, és három menüpont. Mint a korábbiakban is szó volt róla, három típusú felhasználó van: az adminisztrátor, a felhasználó és a vendég. A vendégek számára a bejelentkezés alatt látható három menüpont érhető el. Itt informálódhatnak az egyes bajnokságok tabelláiról, eredményeiről és dobólistáiról. Az oldalra szabad regisztráció nincs, felhasználókat csak az adminisztrátor hozhat létre. A bejelentkezés során az alkalmazás ügyel arra, hogy csak az tudjon belépni, aki beírta a pontos felhasználói nevét és jelszavát. Minden más esetben átirányítja egy új oldalra, amin „Hibás felhasználói név vagy jelszó” üzenet jelenik meg, majd rögtön visszaküldi a bejelentkezési oldalra. Védelemre van szükség az

illetéktelen behatolók ellen is. Ebben a stripslashes() és a mysql\_real\_escape\_string() függvények segítenek, amelyek letisztítják a felhasználónévként vagy jelszóként megadott esetleges forráskódokat, amelyek manipulálhatják az alkalmazás működését vagy illetéktelen behatolást biztosíthatnak. Abban az esetben, ha valaki elfelejtette a jelszavát, a Bejelentkezés alatt lévő "Elfelejtett jelszó" linkre kattintva megadhatja a felhasználói nevét és a bajnokságot, amiben szerepel. Ezt követően az alkalmazás automatikusan kiküldi neki a jelszavát a felhasználói neve mellé regisztrált e-mail címre.

#### **4.3.2. Sikeres bejelentkezés**

A bejelentkezéskor megadott adatokat a checklogin.php fájl vizsgálja meg. Attól függően, irányítja tovább a bejelentkezőt, hogy adminisztrátor vagy felhasználó. Az első esetben az index1.php a cél, a másodikban az index2.php. Bejelentkezés után fontos a felhasználók nyomon követése, jogosultságaik betartatása. Ehhez session-ökre (munkafolyamat) van szükség. A sikeres bejelentkezés után az alkalmazás session-ök segítségével követi nyomon a felhasználót. Létrejön két session változó, a "user" és a "login". Ezek szuperglobális változók, amelyek a szkriptünk bármely részéből elérhetők. A session változók a szerveren tárolódnak mindaddig, amíg be nem zárjuk a böngészőt, vagy le nem jár az élettartamuk. Használatához először el kell indítani a munkamenet kezelést. A checklogin.php-ban létrehozott "user" session változó a bejelentkezett felhasználó nevét fogja megkapni, míg a "login" egy logikai true értéket. Ezeknek a kijelentkezésig meg fog maradni az értékük, és az egyes oldalak meg tudják vizsgálni ezeknek az értékeik alapján betöltéskor, hogy az adott felhasználó jogosult-e a használatukra. A sikeres bejelentkezés után tehát vagy az adminisztrátori felület, vagy a felhasználói felület fog megjelenni.

### 4.3.3. Adminisztrátori felület

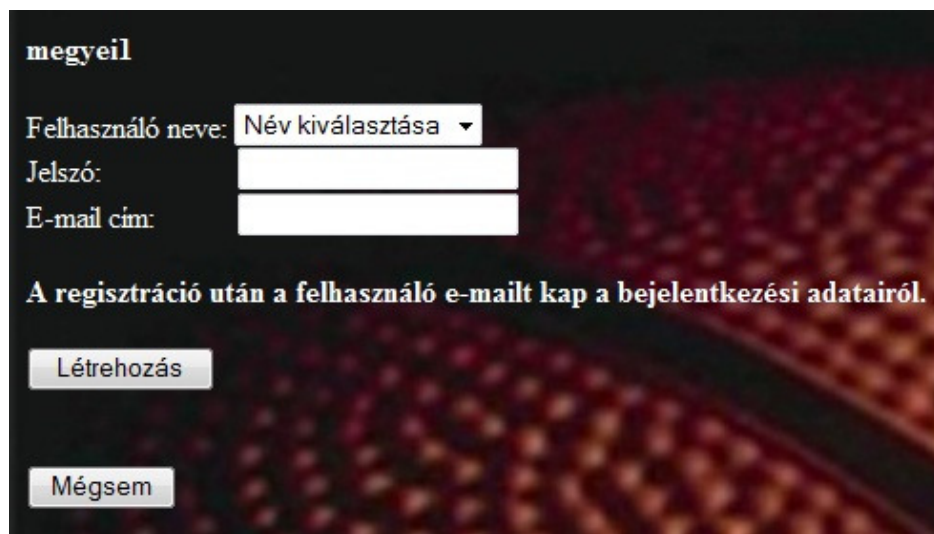
Az adminisztrátori bejelentkezés után az index1.php oldalra lesz irányítva az adminisztrátor, amelyet a következő ábra szemléltet:



Adminisztrátori felület

- **Új felhasználó**

Új felhasználó létrehozása csak abban az esetben lehetséges, ha van már létrehozva bajnokság, és azon belül csapatok. Ekkor az "Új felhasználó" menüpont kiválasztása után ki kell választani a bajnokságot, amelyben a felhasználó részt vesz, ezt követően pedig meg kell adni a név, jelszó és e-mail cím adatokat:



The screenshot shows a registration form with a dark background and a red pattern. At the top left, the word "megyei" is displayed. Below it, there are three input fields: "Felhasználó neve:" with a dropdown menu showing "Név kiválasztása", "Jelszó:", and "E-mail cím:". Below the input fields, there is a line of text: "A regisztráció után a felhasználó e-mailt kap a bejelentkezési adatairól." At the bottom, there are two buttons: "Létrehozás" and "Mégsem".

Felhasználó hozzáadása

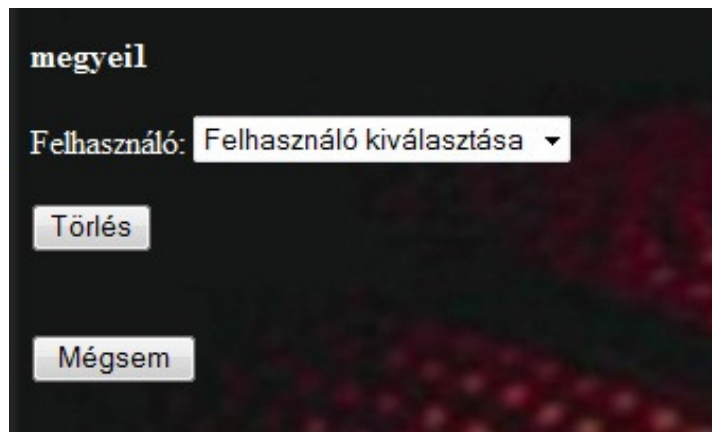
A felhasználó neve nem tetszőleges, csak az adott bajnokságban szereplő csapatok nevei lehetnek. Ez egy legördülő listából választható ki. Természetesen a megadott adatok ellenőrzésen mennek keresztül, és csak akkor jön létre az adott felhasználó, ha minden adat meg lett adva, és az e-mail cím is megfelelő formátumú. Ezt követően az alkalmazás automatikusan e-mailt küld a megadott e-mail címre a felhasználó adataival. A bejelentkezés ezekkel az adatokkal azonnal végrehajtható lesz. A folyamat során a "login" adatbázis "members" táblája bővül egy újabb elemmel.

- **Felhasználók listája**

Ennek a menüpontnak a segítségével a kiválasztott bajnokságban létrehozott felhasználók listája tekinthető meg, ami az adminisztrátor szempontjából fontos lehet.

- **Felhasználó törlése**

Itt a korábban létrehozott felhasználók távolíthatók el, amivel a jogosultságaik is megszűnnek az oldalon.

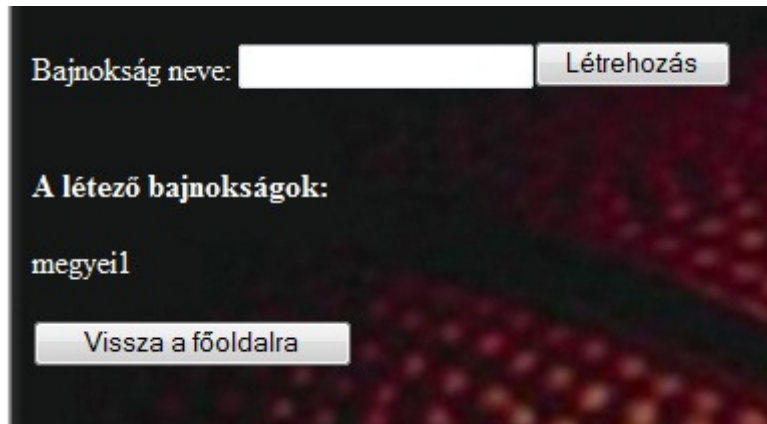


Felhasználó törlése

Szintén a bajnokság kiválasztása szükséges hozzá először, majd a legördülő listában megjelennek a létező felhasználók, akiket egy gombnyomással el lehet távolítani. Ez semmilyen tárolt adatra nem lesz befolyással, kizárólag a felhasználó szűnik meg.

- **Új bajnokság**

Ebben a menüpontban lehet létrehozni az új bajnokságokat. Ez a legelső feladat a bajnokságok kezelése során. Mint a korábbiakban is szó volt róla, erre épül többek között a felhasználók létrehozása is. Ez egyszerűen megtehető, csak egy nevet kell megadni.

The screenshot shows a dark-themed web interface. At the top, there is a text input field labeled "Bajnokság neve:" followed by a "Létrehozás" button. Below this, the text "A létező bajnokságok:" is displayed, followed by the word "megyei1". At the bottom of the visible area, there is a "Vissza a főoldalra" button.

Bajnokság létrehozása

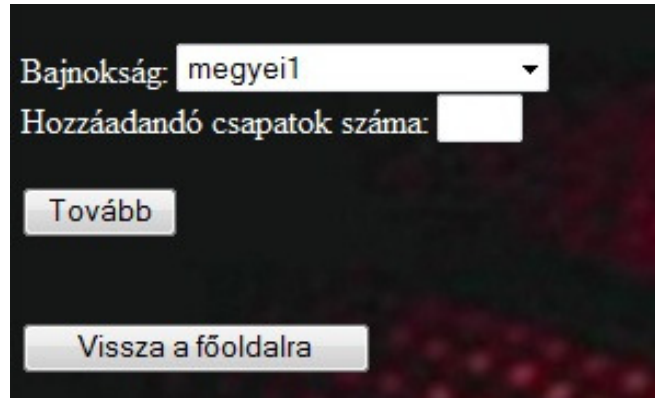
Arra ügyelni kell, hogy ezután mindenhol ez a név fog megjelenni. A megadás során látható az eddig létrehozott bajnokságok listája, ami egy új létrehozásakor azonnal bővülni fog. Ez több bajnokság esetén segíti a nyilvántartást. Létrehozása után a "bajnokságok" adatbázisban létrejön egy tábla a bajnokság nevével, valamint a "játékosok" adatbázisban is, hogy egy csapat létrehozása után azonnal fel lehessen tölteni azt játékosokkal.

- **Bajnokság törlése**

Ezzel a menüponttal nagyon óvatosan kell bánni, mert használata esetén azonnal törli minden adatbázisból az adott bajnokságra vonatkozó összes információt: a tabellát, az eredményeket és a játékosokat. Ez nagy bosszúságot okozhat, hiszen akár egy teljes szezon összes adatát semmisíthetjük meg pillanatok alatt. Használata során egyszerűen egy legördülő listából kell kiválasztani az eltávolítandó bajnokság nevét.

- **Csapat hozzáadása**

Ennek a folyamata két lépésből áll. Először ki kell választani a bajnokságot, amihez csapato(ka)t szeretnénk hozzáadni, és meg kell adni, hogy hány darabot: (Később van lehetőség további csapatok hozzáadására a bajnokság lezárásáig.)



Bajnokság: megyei1  
Hozzáadandó csapatok száma:   
Tovább  
Vissza a főoldalra

Csapat hozzáadása 1.

Másodszor meg kell adni a csapatok neveit:



Csapat-1:   
Csapat-2:   
Csapat-3:   
Csapat-4:   
Hozzáadás  
Mégsem

Csapat hozzáadása 2.

Ennek során a megadott csapatok bekerülnek a "bajnokságok" adatbázis kiválasztott bajnokságának táblájába, és megkapják az alapértelmezett értékeiket.

Pl.: ('csapat1',0,0,0,0,0,0,0)

- **Csapat törlése**

Csapat törlése csak akkor lehetséges, ha az adott bajnokság nem lett még lezárva, mert az ezt követően történő bővítés hatására az információk hibásan jelennének meg. Lezárt bajnokság esetén a csapat csak akkor távolítható el, ha a teljes bajnokságot töröljük.

- **Bajnokság rögzítése**

A bajnokság rögzítésének az a célja, hogy egy adott bajnokság elkezdése után ne legyen lehetőség a csapatok számának bővítésére, illetve a rögzítés előtt ne lehessen adatokat lekérdezni.

A rögzítés előtt először ellenőrzésre kerül, hogy nem volt-e már korábban rögzítve a bajnokság. A rögzítés végrehajtásával a következő folyamatok mennek végbe:

- "eredmenyek" adatbázis tartalmának bővítése a bajnokság nevével megegyező táblával

A táblának annyi rekordja lesz, ahány megrendezendő mérkőzés van a bajnokságban. Az rekordok első két eleme a csapatok lesznek. A többi elem az adott mérkőzésre vonatkozó adatok lesznek, amelyek alapértelmezetten NULL értéket kapnak. Ez mindaddig ilyen értékű marad, amíg fel nem tölti valaki a mérkőzés eredményét.

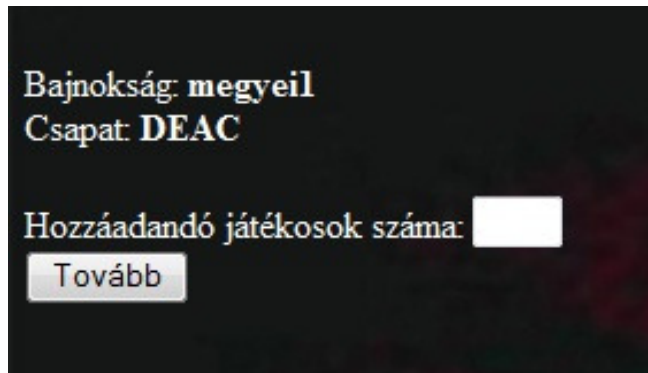
- "lezart" adatbázis tartalmának bővítése a bajnokság nevének megegyező táblával

A tábla tartalma nem lesz lényeges, mert csak azt a célt fogja szolgálni, hogy nyilvántartja a rögzített bajnokságok nevét, és a tábla nevének segítségével ezt lehet ellenőrizni.

A "játékosok" adatbázis már a csapatok létrehozásánál fel lett töltve a bajnoksághoz tartozó táblával. Ennek az az oka, hogy a bajnokságok lezárása előtt is lehetőség legyen már játékosokat adni a csapatokhoz.

- **Játékos hozzáadása**

Játékos hozzáadása előtt szükséges, hogy legyen létrehozva bajnokság, és ezen belül csapat. Ezek kiválasztása után meg kell adni a hozzáadandó játékosok számát. Majd ezt követően a játékosok nevét:

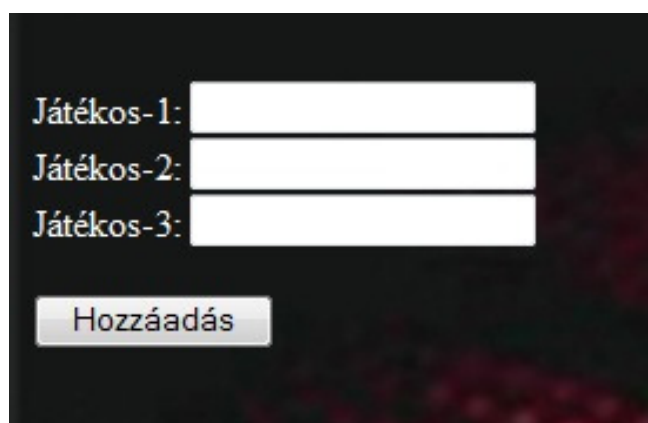


Bajnokság: megyei  
Csapat: DEAC

Hozzáadandó játékosok száma:

Tovább

Játékos hozzáadása 1.



Játékos-1:

Játékos-2:

Játékos-3:

Hozzáadás

Csapat hozzáadása 2.

A játékosok a "jatekosok" adatbázis megfelelő táblájában fognak létrejönni, ami a megadott bajnoksághoz tartozik. Itt van tárolva a játékos neve, csapata, az általa dobott pontok, hárompontosok, és a lejátszott mérkőzések száma.

- **Játékos törlése**

Egyszerűen végrehajtható a bajnokság és a csapat kiválasztása után bármelyik játékos törlése. A törléssel megsemmisülnek a játékos egyéni statisztikái is.

- **Játékosok listázása**

Adott bajnokságból kiválasztott csapat tagjai tekinthetők meg a menüpont segítségével.

A fentiekben ismertetett funkciók azok, amelyek használatára kizárólag az adminisztrátor jogosult. A továbbiakban a felhasználói felületet és a vendégek számára elérhető elemeket fogom bemutatni. Ezek szintén elérhetők az adminisztrátor számára, de rajta kívül már más is igénybe tudja venni.

#### 4.3.4. Felhasználói felület

A felhasználói felületen megjelenő menüpontok elérhetőek az adminisztrátori felületről is, az adminisztrátor számára. Bejelentkezés után az index2.php fog betöltődni. Az oldal bal felső sarkában megjelenik a bejelentkezett felhasználó neve, és az elérhető menüpontok:



Felhasználói felület

- **Eredmény bevitele**

Eredmények bevitele akkor lehetséges, ha létre lettek hozva az adott bajnokságban a csapatok és a játékosok, és a bajnokság le lett zárva. Eredmény beküldésénél az adott felhasználó csak a saját csapata által játszott mérkőzés eredményét tudja beküldeni, más mérkőzés eredményének bevételére nincs jogosultsága, így elkerülhető a jogosulatlan használat. Az adminisztrátor által történő használat során bármelyik eredmény beküldésére van lehetőség. A bevétel előtt először ki kell választani a bajnokságot, majd meg kell adni a hazai és a vendég csapat nevét. Ezt követően automatikusan megjelenik egy oldal, amely a mérkőzésre vonatkozó adatokat kéri be:

Bajnokság: megyei1

A csapat: DEAC      B csapat: DEAC U20

Dátum: (ÉÉÉÉ-HH-NN)

1. játékvezető:       2. játékvezető:

Végeredmény:  
A:  B:

Negyedek eredményei:

1.	<input type="text"/>	<input type="text"/>
2.	<input type="text"/>	<input type="text"/>
3.	<input type="text"/>	<input type="text"/>
4.	<input type="text"/>	<input type="text"/>

Hossz:

**Dobók:**

	Dobott pont	3 pontos		Dobott pont	3 pontos
A1:	<input type="text"/>	<input type="text"/>	B1:	<input type="text"/>	<input type="text"/>
A2:	<input type="text"/>	<input type="text"/>	B2:	<input type="text"/>	<input type="text"/>
A3:	<input type="text"/>	<input type="text"/>	B3:	<input type="text"/>	<input type="text"/>
A4:	<input type="text"/>	<input type="text"/>	B4:	<input type="text"/>	<input type="text"/>
A5:	<input type="text"/>	<input type="text"/>	B5:	<input type="text"/>	<input type="text"/>
A6:	<input type="text"/>	<input type="text"/>	B6:	<input type="text"/>	<input type="text"/>
A7:	<input type="text"/>	<input type="text"/>	B7:	<input type="text"/>	<input type="text"/>
A8:	<input type="text"/>	<input type="text"/>	B8:	<input type="text"/>	<input type="text"/>
A9:	<input type="text"/>	<input type="text"/>	B9:	<input type="text"/>	<input type="text"/>
A10:	<input type="text"/>	<input type="text"/>	B10:	<input type="text"/>	<input type="text"/>
A11:	<input type="text"/>	<input type="text"/>	B11:	<input type="text"/>	<input type="text"/>
A12:	<input type="text"/>	<input type="text"/>	B12:	<input type="text"/>	<input type="text"/>

**Eredmény bevitele**

A kitöltendő elemek:

- Dátum

Itt fontos, hogy megfelelő formátumban adja meg a felhasználó a dátumot, mert ez alapján lesz időrendbe téve a megjelenített eredmények listája. Hibás dátumformátum megadása esetén hibaüzenet jelenik meg.

- Játékvezetők

A mérkőzésen közreműködő játékvezetők nevét kell megadni. Ez szintén az eredmények menüpont alatt fog megjelenni, a mérkőzéssel kapcsolatos információként.

- Végeredmény

Végeredmény megadásánál ellenőrzésre kerül, hogy nem-e lett megadva egyenlő állás. Mivel a kosárlabdában nincs döntetlen, itt sem lehet ilyet megadni. Ebben az esetben is hibaüzenet jelenik meg.

- Negyedek eredményei

Ezek megadásánál fontos, hogy a negyedek eredményeinek összege (és az esetleges hosszabbítás) egyezzen a végeredménnyel. Ennek hiányában szintén hibaüzenet látható majd.

- Dobók

Egy csapatban maximum 12 játékos szerepelhet, így 12 játékos látható itt is. A játékosok legördülő listából választhatók ki, ami megkönnyíti a felhasználó dolgát, nem kell minden játékos nevét egyenként beírni. A kiválasztott játékosok után kötelező megadni a dobott pontok összegét, illetve az esetlegesen dobott hárompontosok összegét is. Az elküldés után ellenőrzésre kerül, hogy a játékosok által dobott pontok összege megegyezik-e a végeredménnyel, minden kiválasztott játékoshoz lett-e pont hozzárendelve, valamint lett-e pont hozzárendelve játékoshoz úgy, hogy nem lett a neve kiválasztva. Bármilyen eltérés esetén hibaüzenet lesz látható.

Fontos, hogy a felhasználó minden adatot megadjon, mert addig nem tudja beküldeni az eredményt a mérkőzésről. A sikeres beküldés után azonnal frissülnek a "bajnokság", "eredmények" és "játékosok" adatbázisok megfelelő táblái. A következő menüpontok ezeknek az adatbázisoknak a tartalmára fognak támaszkodni.

- **Tabellák**

A bajnokság kiválasztása után megjelenik a hozzátartozó tabella. Ebben megjelenítésre kerül a "bajnokságok" adatbázis adott bajnokságra vonatkozó táblájának tartalma táblázatba rendezve:

megyei								
Helyezés	Csapat	Mérkőzés	Győzelem	Vereség	Dobott	Kapott	Különbség	Arány
1.	csapat3	1	1	0	66	48	18	1
2.	csapat5	1	1	0	74	57	17	1
3.	csapat1	1	1	0	50	40	10	1
4.	csapat2	1	0	1	40	50	-10	0.5
5.	csapat6	1	0	1	57	74	-17	0.5
6.	csapat4	1	0	1	48	66	-18	0.5

Tabella

Ebben a rangsorolás három szempont szerint történik:

- 1. Arány

Az arány a következő képlet alapján kerül kiszámításra a kosárlabdában:  
$$\text{arány} = (2 * \text{nyert\_mérkőzés} + \text{vesztett\_mérkőzés}) / (2 * \text{összes\_mérkőzés})$$

- 2. Különbség

A dobott és a kapott pontok különbsége.

- 3. Dobott pont

A több dobott pont szerint.

- **Eredmények**

A kiválasztott bajnoksághoz tartozó eredmények jeleníthetők meg. Ez az "eredmények" adatbázis tartalmának segítségével megy végbe, ez jelenik meg rendezett formában dátum szerint rendezve az oldalon:

<b>2010-04-03</b>
<b>csapat5 - csapat6 74-57</b> (24-15, 18-18, 14-21, 18-3)
<b>Jv.:</b> xxx, yyy
<b>D.:</b> jatekos5 74/9 ill. jatekos6 57/3
<b>2010-04-02</b>
<b>csapat3 - csapat4 66-48</b> (15-15, 18-7, 24-18, 9-8)
<b>Jv.:</b> xxx, yyy
<b>D.:</b> jatekos3 66/12 ill. jatekos4 48/15
<b>2010-04-01</b>
<b>csapat1 - csapat2 50-40</b> (15-10, 15-10, 10-5, 10-15)
<b>Jv.:</b> xxx, yyy
<b>D.:</b> jatekos1 50/3 ill. jatekos2 40/6

Eredmények

- **Dobólisták**

Adott bajnokságban részt vevő játékosok rangsorolása az általuk dobott összes pont, valamint hárompontos alapján. Ez a "játékosok" adatbázis tartalmának segítségével történik.

Játékos	Csapat	Pont
jatekos5	csapat5	74
jatekos3	csapat3	66
jatekos6	csapat6	57
jatekos1	csapat1	50
jatekos4	csapat4	48
jatekos2	csapat2	40

Játékos	Csapat	Hárompontos
jatekos4	csapat4	15
jatekos3	csapat3	12
jatekos5	csapat5	9
jatekos2	csapat2	6
jatekos1	csapat1	3
jatekos6	csapat6	3

**Dobólisták**

A tabellák, az eredmények és a dobólisták menüpont természetesen az adminisztrátor számára is elérhető.

- **Jelszó megváltoztatása**

Ennek segítségével bármelyik felhasználó – beleértve az adminisztrátort is – megváltoztathatja a jelszavát. Ehhez az szüksége, hogy megadja a régi jelszavát, majd az új jelszavát kétszer. Ekkor ellenőrzésre kerül a régi jelszó helyessége, és amennyiben ez rendben van, az új jelszó felülírja a régit a "login" adatbázis "members" táblájának megfelelő rekordjában. A jelszó elfelejtése esetén lehetőség van kiküldésére e-mailben a bejelentkezéskor látható link segítségével.

- **Kilépés**

A kilépés során a felhasználó által használt session leáll, és megszűnnek a session változók. Így megszűnik a felhasználó jogosultsága a védett oldalak megtekintéséhez, használatához a következő sikeres bejelentkezésig. A kilépés után a bejelentkezési oldal fog betöltődni.

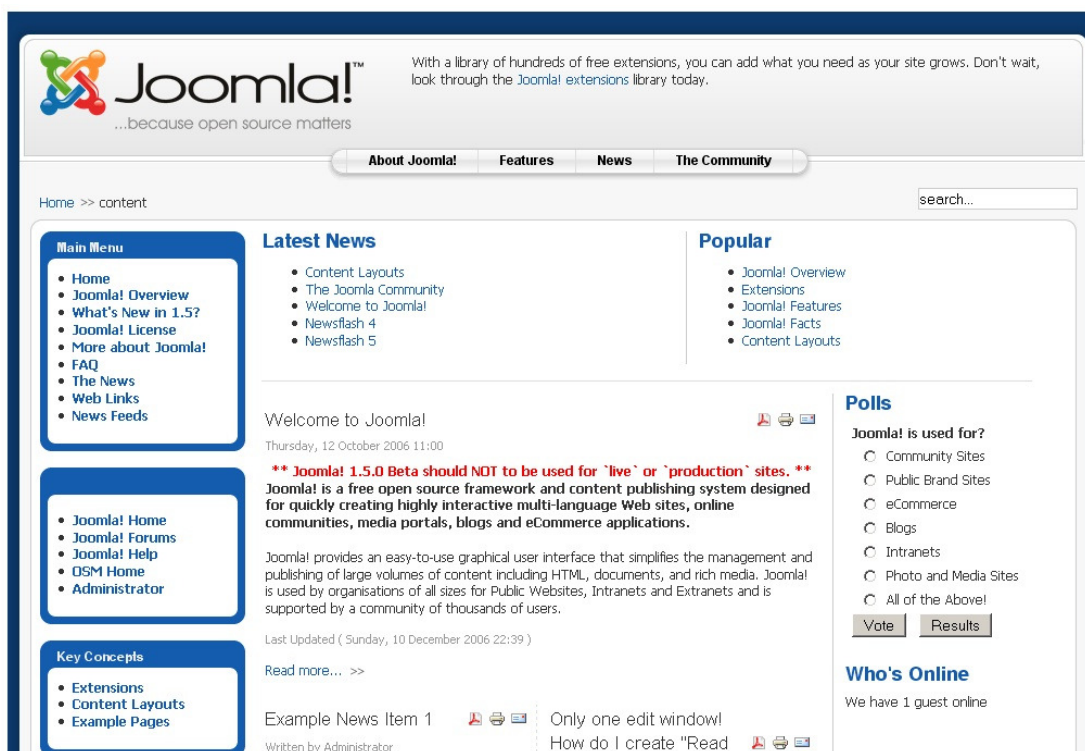
#### **4.3.5. Vendég**

A vendégek számára lehetőség van bejelentkezés nélkül is a főoldalon keresztül informálódni a bajnokságokról. Ez három menüponthoz való hozzáférést biztosít: Tabellák, Eredmények és Dobólisták.

## 5. Az alkalmazás továbbfejlesztésének lehetősége

Az alkalmazás ebben a formájában az elvárt funkciók végrehajtására teljes mértékben alkalmas. Felmerülhet az igény esetlegesen, hogy az alkalmazás más feladatok elvégzését is meg tudja valósítani. Ha a kosárlabda témájánál maradunk, elvárás lehet például egyes hírek, információk megjelenítése a nagyközönség számára, valamint a design fejlesztése is. Erre az egyik megoldás lehet az eddig alkalmazott mód, fejlesztés a fentebb említett, és egyéb eszközök segítségével. Másrészt viszont nagyban gyorsíthatja a feladat megoldását a tartalomkezelő rendszerek használata. Ezekről már a szakdolgozat 2. fejezetében is esett szó. A tartalomkezelő rendszerek segítségével rövid idő alatt felépíthetünk egy alapot, amelyet a széleskörű támogatás segítségével hamar továbbfejleszthetünk olyanra, amilyenre szeretnénk.

Ilyen tartalomkezelő rendszer például a Joomla! :



The screenshot shows the Joomla! website homepage. At the top, there is the Joomla! logo and the tagline "...because open source matters". Below the logo, there is a navigation menu with links for "About Joomla!", "Features", "News", and "The Community". The main content area is divided into several sections: "Main Menu" with links like "Home", "Joomla! Overview", and "What's New in 1.5?"; "Latest News" with a list of news items; "Popular" with a list of popular content; "Welcome to Joomla!" with a date and time; "Key Concepts" with links like "Extensions", "Content Layouts", and "Example Pages"; "Example News Item 1" with a date and time; and "Who's Online" with a count of online users. There is also a search bar and a "Polls" section with a question "Joomla! is used for?" and several radio button options.

Alapértelmezett Joomla! kezdőlap

Telepítése mindössze néhány egyszerű lépésből áll, az asztali programokhoz hasonlóan csak a varázsló utasításait kell követni. A Joomla!, mint keretrendszer a fejlesztői közösségnek lehetőséget ad az alapfunkciók kibővítésére, és könnyen telepíthető kiegészítőkkal lehet új funkciót adni a weblapnak. Például: fórumok és magánüzenetküldő komponensek, blogok, naptárak, képtárak, dokumentumkezelés, stb.

Ezek között a bővítmények között megtalálható olyan is, amely az általunk írt php kódot futtatja. Ennek segítségével például a Basketball Manager alkalmazás elemei beágyazhatók a tartalomkezelő rendszerbe, és ennek keretein belül tudjuk kihasználni az általa nyújtott lehetőségeket. [7.][8.]

Az alkalmazás továbbfejlesztésének természetesen számos további módja létezik.

## Összefoglalás

Az alkalmazás reményeim szerint nagy segítséget nyújt majd azoknak, akik a kosárlabdában bármilyen szerepet töltenek be, legyen akár játékos, szurkoló vagy adminisztrátor. Nyitott a lehetőség az alkalmazás továbbfejlesztésére is, amelyet a jelentkező igények szerint meg is lehet majd valósítani, akár a fent említett módon is. Manapság egyre több esetben alkalmazzák a dinamikus megjelenítés eszközeit, mert számos olyan lehetőséget biztosít, amelyet egy statikus weboldal nem tudna megadni. Az eszközök bemutatása során láthattuk, mi szükséges egy ilyen alkalmazás megvalósításához. Ennek elkészítése először nagy kihívásnak tűnt számomra, mivel alapszintről kezdtem el a munkát, sajátítottam el a szükséges tudásanyagot. Ennek során számos kérdés, kihívás merült fel számomra, de ezek teljesítésével egyre nagyobb gyakorlatot szereztem. Ebben nagy segítséget jelentettek az interneten lévő, programozással kapcsolatos fórumok, melyekben általában szinte azonnal megtalálható volt a keresett megoldás egy problémára, valamint a témához tartozó szakirodalom.

Úgy gondolom, az alkalmazás elkészítésével és bemutatásával elértem célomat, amely az volt, hogy szemléltessem a dinamikus megjelenítés előnyeit, és ennek segítségével egy adott témakörhöz tartozó feladatok megoldását leegyszerűsítsem.

## **Irodalomjegyzék**

**[1.] Matt Zandstra – Tanuljuk meg a PHP5 használatát 24 óra alatt**

Kiskapu Kiadó, 2005.

**[2.] Sági Gábor – Webes adatbázis-kezelés MySQL és PHP használatával**

BBS-INFO, 2005.

### **Internetes források:**

**[3.] Dinamikus weboldalak**

<http://prog.hu/cikkek/180/Dinamikus+weboldalak.html>

**[4.] PHP**

<http://www.php.net>

**[5.] Apache HTTP Server Project**

<http://httpd.apache.org/>

**[6.] MySQL**

<http://www.mysql.com/>

**[7.] Nagy Gusztáv – Webes tartalomkezelő rendszerek**

[http://nagygusztav.hu/sites/default/files/WebesTKRek\\_jegyzet\\_1.0.pdf](http://nagygusztav.hu/sites/default/files/WebesTKRek_jegyzet_1.0.pdf)

**[8.] Magyar Joomla! felhasználók Nemzetközi Egyesülete**

<http://www.joomla.org.hu/>

## **Köszönetnyilvánítás**

Ezúton szeretnék köszönetet mondani

- Dr. Bujdosó Gyöngyinek, aki vállalta szakdolgozatom témavezetését, és javaslataival segítette munkámat
- Szüleimnek, akik mindvégig támogattak egyetemi tanulmányaim során