

# **Oktatásszervező rendszer adatbázis-kezeléssel PHP-ben**

Témavezető:  
Dr. Kósa Márk  
egyetemi tanársegéd

Készítette:  
Balogh Tamás  
programtervező informatikus

Debrecen  
2011

## Tartalomjegyzék

Tartalomjegyzék.....	2
Bevezetés .....	4
Fejlesztői és tesztelői környezet .....	5
A fejlesztés során használt eszközök és technológiák.....	6
Apache HTTP Server .....	6
MySQL.....	7
PHP .....	10
jQuery.....	14
JSON .....	17
Fogalomjegyzék .....	19
Adatbázisséma.....	20
Tárgyak tábla .....	21
Vizsgák tábla .....	21
Kurzusok tábla.....	21
Felhasználók tábla.....	22
Üzenetek tábla .....	22
Session tábla .....	23
Folyamatok .....	24
Bejelentkezés .....	24
Üzenetek olvasása .....	24
Üzenet írás .....	24
Tárgyak listázása.....	26
Kurszus létrehozása, módosítása, törlése.....	26
Kurszusfelvétel.....	27
Vizsga létrehozása, módosítása, törlése .....	28
Vizsgajelentkezés.....	28
Vizsga leadás .....	29
Időszakok listázása.....	29
Időszakok kezelése.....	30
Hallgató jelentkeztetése kurzusra .....	30
Hallgatói jogkör.....	32
Oktatói jogkör .....	33
Adminisztrátori jogkör .....	34
Biztonság .....	36
Összefoglalás .....	39

Irodalomjegyzék.....	40
Köszönetnyilvánítás .....	41

## Bevezetés

Már középiskolásként is érdeklődtem az internet és a webes technológiák iránt és egyetemi tanulmányaimmal párhuzamosan sikeresen el is végeztem egy webprogramozó képzést. Ezek után elég kézenfekvőnek tűnt, hogy a szakdolgozatom is webes témájú lesz. A szakdolgozat célja, hogy egy projekt keretei között elkészítsünk egy webes oktatásszervező rendszert, illetve hogy a dolgozat címében megadott alkalmazás fejlesztőjeként megismerkedhessenek az alkalmazás elkészítése során alkalmazott technikákkal. A cél egy gyors, könnyen tanulható illetve bővíthető rendszer megalkotása volt. A rendszerben megtalálható funkciók elkészítéséhez a Debreceni Egyetem Neptun tanulmányi rendszere szolgált mintaként.

A fejlesztés során többek között megismerkedtem a jQuery, és a JSON keretrendszerekkel, valamint a PHP nyelvet is sikerült még jobban elsajátítanom.

## Fejlesztői és tesztelői környezet

A fejlesztés során az Eclipse PDT-t használtam, amely számos hasznos kényelmi funkciójának köszönhetően nagyban megkönnyítette a fejlesztést.

Adatbázis-kezelőnek (továbbiakban DBMS) a MySQL-t használtam, mely a mai napig a legelterjedtebb webes DBMS. A MySQL adatbázis adminisztrálására pedig az ingyenes HeidiSQL nevű programot, amely ingyenesen letölthető innen: <http://www.heidisql.com/>.

A fejlesztés megkönnyítésének érdekében a WampServer 2.1-es verzióját választottam webfejlesztő-keretrendszerként. A Wamp négy szoftver nevéből alkotott mozaikszó. A Wamp egy Windowson futtatható alkalmazás-kiszolgáló programcsomag, amely fő elemeit az alábbi négy program alkotja:

- Windows, a Microsoft által gyártott operációsrendszer
- Apache HTTP Server, a jelenleg legnépszerűbb szabad szoftver/nyílt forráskódú webszerver
- MySQL, egy többszálú, többfelhasználós SQL adatbázis-kezelő rendszer (DBMS)
- PHP

Az alkalmazásomat Mozilla Firefox 3.6, Google Chrome, Opera 10 és Microsoft Internet Explorer 8 alatt teszteltem. A JavaScriptek tesztelését nagyban megkönnyítették a Firefox Firebug, Firecookie és FireQuery kiegészítői.

## A fejlesztés során használt eszközök és technológiák

### **Apache HTTP Server**

Az Apache HTTP Server (röviden Apache) nyílt forráskódú webkiszolgáló alkalmazás, szabad szoftver, mely kulcsfontosságú szerepet játszott a World Wide Web elterjedésében. A készítőinek a célja olyan webszerver program létrehozása, karbantartása és fejlesztése, amely megfelel a gyorsan változó Internet követelményeinek, biztonságos, üzleti, vállalati felhasználásra is megfelelő és szabadon használható. Az Apache a régi NCSA (National Center for Supercomputing Applications) HTTPd szerverre épül, az Apache Szoftver Licenc feltételei alatt terjesztik.

Az Apache robusztus, erőteljes és rugalmas webszerver, amely kompatibilis a HTTP/1.1 (RFC2616) protokollal. Az Apache projekt koordinálását az Apache Software Foundation végzi.

Az Apache volt az első használható alternatíva a Netscape Communications Corporation webszerverrel szemben (melynek mai neve Sun Java System Web Server). A későbbiekben továbbfejlődött, és más Unix-alapú webszerverekkel is felvette a versenyt funkcionalitás és teljesítmény tekintetében.

Többek között a következő operációs rendszerekhez készítették el az Apache-ot: Unix, FreeBSD, Linux, Solaris, Novell NetWare, Mac OS X és Microsoft Windows.

### Szolgáltatások és tulajdonságok

Az Apache sok szabványt támogat, melyeknek nagy része lefordított modulok formájában áll rendelkezésre a mag kiegészítéseként. Ezek a modulok sok területet lefednek a kiszolgáló oldali programnyelv-támogatástól kezdve hitelesítési sémákig. Az ismertebb, támogatott programnyelv modulok a mod\_perl, a mod\_python, a Tcl és a PHP. A népszerűbb hitelesítési modulok a mod\_access, mod\_auth és a mod\_digest. További modulokban megvalósított tulajdonságokra példa még az SSL és TLS támogatást nyújtó mod\_ssl, a proxy modul, egy hasznos URL átíró a mod\_rewrite modulban, testreszabható logolás a mod\_log\_config modulban és szűrési támogatás a mod\_include és mod\_ext\_filter modulok segítségével.

A mod\_gzip modul tömöríteni tud, segítségével csökkenthető a weboldalak mérete. Az Apache logok elemezhetőek böngészővel az AWStats/W3Perl vagy a Visitors programok segítségével.

A virtual hosting segítségével egyetlen Apache kiszolgáló ki tud szolgálni több különböző webhelyet is. Például a jelenlegi munkahelyemen egy Apache webservert szolgál ki több mint száz weboldalt.

### Felhasználás

Statikus és dinamikus weboldalak közzétételére egyaránt használják. Sok webalkalmazást az Apache által nyújtott környezethez és szolgáltatásokhoz terveznek. Az Apache alkotja a webservert komponens az általam is használt WAMP alkalmazáscsomagban, melynek további komponensei a MySQL adatbázisszerver és a PHP/Perl/Python programozási nyelvek.

Az Apache webservert különböző kereskedelmi szoftvercsomagok (például az Oracle Database és az IBM WebSphere) részeként is terjesztik. A Mac OS X is magába foglalja az Apache-ot beépített webserverként, valamint a Novell NetWare 6.5 alapértelmezett webservere is az Apache.

Az Apache nem csak weboldalak, hanem egyéb tartalom publikálására is használható, például tetszőleges fájlok megosztására is. Ha egy felhasználó telepíti az Apache-ot a személyi számítógépén, akkor tetszőleges fájlokat meg tud osztani, ha bemásolja őket az Apache document root könyvtárába.<sup>1</sup> Népszerűségét az is bizonyítja, hogy számos nagy és ismert honlap is Apache webserveren fut, például a Google keresőmotor felülete egy módosított Apache webserveren fut, melyet a Google Web Server (röviden GWS) névre kereszteltek.

### **MySQL**

A MySQL használatának főbb okai:

- gyorsaság,
- hordozhatóság,
- bármely más nyelven írt felület illeszthetősége,
- kis méretű,
- alacsony erőforrásigény,
- ingyenesség.

A MySQL egy többszálú végrehajtást támogató kiszolgáló, ami azt jelenti, hogy minden egyes új kapcsolat létesítésekor elindul egy új kiszolgálófolyamat. A MySQL-hez kötődő kapcsolatok nem osztják meg egymással a folyamataikat, így amikor egy folyamat valamilyen hiba következtében leáll, vagy valamilyen módon túlterheli a kiszolgálót, csak az adott

---

<sup>1</sup> Forrás: [http://hu.wikipedia.org/wiki/Apache\\_HTTP\\_Server](http://hu.wikipedia.org/wiki/Apache_HTTP_Server)

folyamat áll le, és nem az egész kiszolgáló. A MySQL ezen tulajdonsága az erőforrás használat optimalizálása mellett jótékony hatással van a sebességre is.

A MySQL-t az is népszerűvé teszi a fejlesztők körében, hogy gyakorlatilag minden nyelv – a PHP, a Perl, a C, illetve a C++, a Delphi, a Java, a Python és a Tcl – alkalmas arra, hogy felhasználói felületet írjanak hozzá. Az operációs rendszerek cserélgetése is gyerekjáték: ha bármikor operációs rendszert vagy nyelvet kellene váltanunk, használhatjuk a `mysqldump` parancsot az adataink kinyerésére és a `mysqlimport` parancsot arra, hogy az adatainkat újra bevigyük a rendszerbe.

A MySQL a használt operációs rendszerek döntő többségében, így bármilyen Unix-változaton, illetve a Windows 95/98/NT/2000/XP/Vista/7 változatain futtatható, ráadásul nemrégiben megjelent a MySQL kiszolgáló 64 bites változata is, mely a napjainkban egyre gyakoribb 64 bites processzorok teljesítményének jobb kihasználását teszi lehetővé. Ha letöltés útján kell a MySQL-re szert tennünk – a Windows-felhasználók majdnem mindig ebbe a csoportba esnek –, első lépésként a program honlapjára érdemes ellátogatni: <http://www.mysql.com/>. A letölthető Windows-változat telepítését varázsló segíti. Amennyiben Linux-munkaállomást vagy -kiszolgálót futtatunk, az operációs rendszer telepítőlemezén nagy valószínűséggel megtalálható a programcsomag, és az operációs rendszer telepítésekor beállítható ennek a telepítése is.

### Adminisztráció

A MySQL adatbázisok adminisztrációjára a mellékelt parancssori eszközöket (`mysql` és `mysqladmin`) használjuk. A MySQL honlapjáról grafikus felületű adminisztráló eszközök is letölthetőek: MySQL Workbench (korábban MySQL Administrator) és MySQL Query Browser.

Széles körben elterjedt és népszerű adminisztrációs eszköz a PHP nyelven írt, nyílt forráskódú phpMyAdmin, mely letölthető a <http://www.phpmyadmin.net/> címen. Személyes tapasztalataim szerint egyszerűbb adatbázisok kezelésére tökéletesen megfelelő eszköz a phpMyAdmin, de egy összetettebb, esetleg tárolt eljárásokat és triggereket is tartalmazó adatbázis adminisztrációja meglehetősen körülményessé válhat, ezért én az ingyenesen használható HeidiSQL nevű grafikus felületű adminisztrációs eszközt használom.

### Használat

A MySQL, mint minden adatbázis-kezelő, a kliens-szerver architektúrát követi. Ebből az következik, hogy a szerver folyamat általában egy dedikált, kitüntetett gépen szokott futni, és ehhez a géphez kapcsolódnak az egyes kliensek.

Kliens lehet bármi, ami képes kommunikálni a szerver folyamattal: például a MySQL parancssori kezelőfelülete, vagy egy Delphiben könyvelőprogram, egy C++-ban írt, QT-s, vagy GTK-s program, vagy a PHP.

Ahhoz, hogy csatlakozni tudjunk egy MySQL szerverhez szükségünk van egy felhasználónév jelszó párosra. Az érdemi munkához ezek mellett szükség van egy a felhasználó által elérhető adatbázisra is.

Egy MySQL rendszer vázlatos felépítése tehát a következő: egyfelől egy MySQL szerverfolyamat és kliensek, másrészt felhasználók és adatbázisok. A felhasználók és adatbázisok között sok-sok kapcsolat van, tehát egy felhasználó használhat több adatbázist, és egy adatbázist is használhat több felhasználó.

A valóságban nemcsak adatbázisszinten lehet szabályozni a felhasználók hozzáférését, hanem táblaszinten is, de általában az egy felhasználó egy adatbázis a jellemző hozzáállás, ahogy az a szakdolgozatomban is szerepel.

### Parancssor

Nézzük, hogy lehet a legegyszerűbben kapcsolódni a MySQL-hez! A lenti példa akkor működik, ha a MySQL kiszolgáló ugyanazon a gépen fut, mint ahová beléptünk. Ha másik géphez akarunk kapcsolódni, akkor a `-h` kapcsoló után meg kell adnunk a gép nevét.

```
C:\>mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.1.53-community-log MySQL Community Server (GPL)
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>
```

A parancssori kliens neve tehát `mysql`. A `-u` kapcsoló azt a felhasználói nevet adja meg, amivel kapcsolódni akarok, a `-p` kapcsoló pedig jelszó bekérésre utasítja a MySQL-t. Ha az adott felhasználóhoz van hozzárendelve jelszó, de mégsem adtuk meg a `-p` kapcsolót, akkor egy ehhez hasonló hibaüzenetet kapunk:

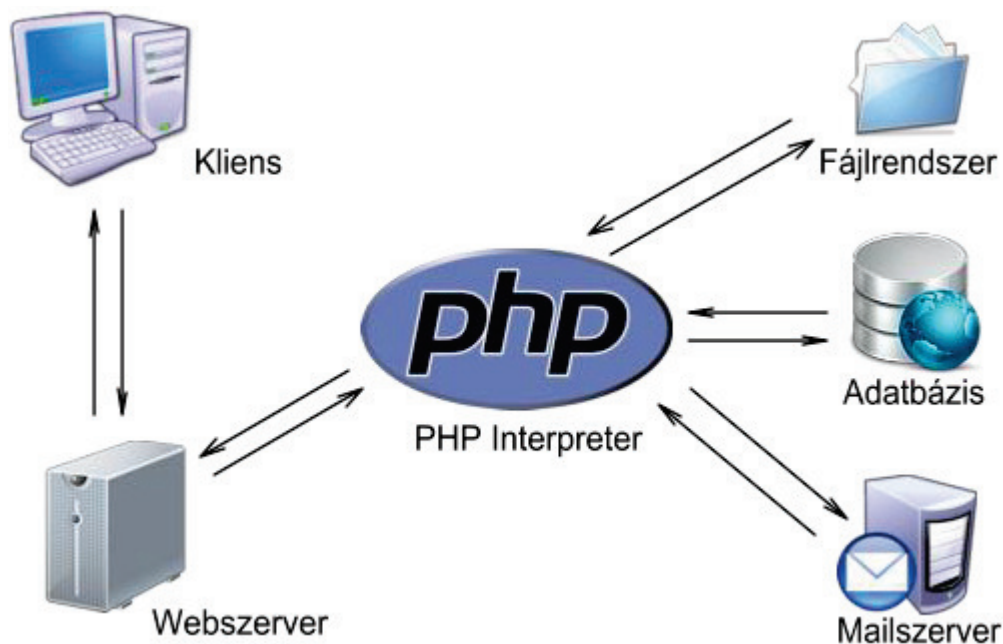
```
C:\> mysql -u root
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using
password: NO)
```

## PHP

A PHP (PHP: Hypertext Preprocessor) nyílt forráskódú, számítógépes szkriptnyelv, legfőbb felhasználási területe a dinamikus weboldalak készítése. Emiatt a PHP-t jórészt szerveroldalon használják, bár létezik parancssori interfésze is, illetve önálló, grafikus felületű alkalmazások is létrehozhatóak vele. A PHP a legtöbb webserverre, operációs rendszerre és platformra ingyenesen telepíthető. Manapság több mint 20 millió weboldal és egymillió szerver futtat PHP-t, bár a nyelvet használó weboldalak száma 2005 augusztusától kezdve folyamatosan csökken. A PHP emellett az Apache webservert egyik legnépszerűbb beépülő modulja.

### A PHP működése leegyszerűsítve

A PHP oldalak elkészítésénél a HTML-t gyakorlatilag csak mint formázást használjuk, ugyanis ezek a lapok teljes funkcionalitása a PHP-re épül. Amikor egy PHP-ben megírt oldalt akarunk elérni, a kiszolgáló először feldolgozza a PHP utasításokat, és csak a kész (HTML) kimenetet küldi el a böngészőnek, így a programkód nem is látható a kliens oldalon. Ehhez egy úgynevezett interpretert (értelmezőt) használ, amely általában egy küldő modulja a webservernek.



A PHP nyelv lényegében nagymértékű kiegészítése a HTML-nek, ugyanis rengeteg olyan feladat végezhető el vele, amelyre az ügyféloldali szkriptek nem képesek. Ilyen például a bejelentkezés, az adatbáziskezelés, fájlkezelés, kódolás, adategyeztetés, kapcsolatok létrehozása, e-mail küldése, adatfeldolgozás, dinamikus listakészítés stb. Minden olyan esetben, ahol nagyszámú ismétlődő feladatsort kell végrehajtani, ott ez a programnyelv nagyszerű segítség.

### Szintaxis

A PHP utasításokat mindig egy meghatározott karaktersorral kell kezdeni és bezárni. Sok hibát kizárhatunk, ha programjainkban a „<?php” nyitó és a „?>” záró karaktersorozatokat használjuk. Ugyanis az érvényes nyitó és záró elemek a php.ini konfigurációs fájlban vannak deklarálva, amelyek szerverenként eltérhetnek egymástól, így előfordulhat, hogy egy szerveren nem értelmezi az interpreter a PHP programunkat, hanem közvetlenül a kimenetre küldi a forráskódot.

<b>karaktersorozat</b>	<b>megjegyzés</b>
<? ?>	ha a php.ini-ben engedélyezve van, akkor a short_open_tag
<?php ?>	alapértelmezett nyitó és záró karaktersorozat
<?=\$változo?>	a <?php echo \$változo; ?> kód rövidítése

Minden változót és összetett adatszerkezetet (objektumot, tömböt) \$ jellel kezdünk.

```
$változo="érték";
echo $változo;
```

Kivételt képeznek ez alól a konstansok.

```
define("KONSTANS", "123456");
echo KONSTANS;
```

Fontos megemlíteni, hogy a PHP-ben string típus jelölésénél az aposztrófban ( ' ) és az idézőjelben ( " ) lévő karaktersorozat más-más jelentéssel bír. Az idézőjelek közé zárt szöveg esetében az értelmező a szövegben beszúrt változókat is keres – ezért a kiértékelése minimálisan ugyan, de lassabb –, ugyanakkor aposztróffal jelölt string típusnál ez nem valósul meg.

```
<?php
    $változo = 'szöveg';
    $változo1 = 'példa $változo';
    $változo2 = "példa $változo";
    echo $változo1.'  

```

Az eredmény pedig:

```
példa $változo
példa szöveg
```

A tömb indexelése – hasonlóan más C-alapú nyelvekhez – 0-tól kezdődik, tehát a `$tomb[1]` eleme valójában a tömb 2. eleme.

Az utasításokat pontosvessző (;) zárja, hasonlóan például a C nyelvhez.

## Adattípusok

A PHP az egész számokat platformfüggő tartományban tárolja. Ez a tartomány jellemzően 32-bites előjeles ábrázolás. Eltérően más programnyelvektől bizonyos esetekben az előjel nélküli számok is előjelessé konvertálódnak. Egész változók értékadásánál használható az előjeles decimális, oktális és hexadecimális jelölés is. A lebegőpontos számok tárolási tartománya szintén platformfüggő, megadhatóak lebegőpontosan és kétféle tartományos jelöléssel is. A PHP a string adattípust is támogatja, stringeket megadni aposztrófok és idézőjelek között valamint az úgynevezett heredoc szintaxissal egyaránt lehet.

A szabványos PHP programkönyvtár (Standard PHP Library, SPL) igyekszik megoldani a szabványos adatkezelési feladatokat és hatékony adatelérési interfészeket és osztályokat implementál.

A PHP gyenge típusossága – helyesen használva – az egyik legnagyobb előnye és a leggyakoribb hibaforrása is egyben, mivel a nyelvre jellemző az úgynevezett Type Juggling, ami lehetővé teszi, hogy a változó típusát a környezete határozza meg.

```
<?php
    $size = "0";                // $size egy string (ASCII 48)
    $size += 2;                 // $size most egész (2)
    $size = $size + 1.3;        // $size most lebegőpontos (3.3)
    $size = 5 + "10 Kis Pingvin"; // $size egész (15)
    $size = 5 + "10 Kismalac";  // $size egész (15)
    $size = "almafa1";          // $size egy string (almafa1)
    $size++;                    // $size egy string (almafa2)
?>
```

Lehetőségünk van a C-hez hasonló a típus megváltoztatására.

```
<?php
    $foo = 10;                  // $foo most szám típusú
    $bar = (boolean) $foo;      // $bar logikai típusú
?>
```

A leggyakoribb ellenőrző módszer egy változó, objektum vagy tömb aktuális értékének és típusának meghatározására, ha a megfelelő sorban a `var_dump` függvényt használjuk, mely strukturált információt szolgáltat, tömb és objektum esetében a függvény rekurzív.

```
<?php
    $a = array(1, 2, array('a', 'b', 'c'));
    echo '<pre>';
    var_dump($a);
    echo '</pre>';
?>
```

Az eredmény:

```
array(3) {
  [0]=> int(1)
  [1]=> int(2)
  [2]=> array(3) {
    [0]=>string(1) "a"
    [1]=>string(1) "b"
    [2]=>string(1) "c"
  }
}
```

## jQuery

A jQuery egy JavaScript könyvtár (Keretrendszer, Eszköztár), amivel egyszerűsödnek a JavaScript feladatok a HTML dokumentumokon, mivel számtalan megoldást készen kapunk benne. De a legnagyobb előnye, hogy a jól megírt produktum nagy valószínűséggel az összes böngészőben fog futni.

Milyen eszközöket ad a kezünkbe a jQuery:

- Eseménykezelések
- Effektek
- Módosítók
- Kijelölők
- Attribútum módosítók
- AJAX kérések kezdeményezéséhez és lekezeléséhez
- Felhasználói interfész
- stb.

A lista nem teljes, mivel az elérhető pluginekkal szinte minden feladatra találunk megoldást pl.: \*.csv kezelés, audió és videó lejátszáshoz stb. <sup>2</sup>

A jQuery keretrendszer 2006 januárjában látott először napvilágot a BarCamp elnevezést viselő rendezvényen, New Yorkban. Jelenleg a legelterjedtebb JavaScript framework. A keretrendszer ingyenes és nyílt forráskódú.

Talán a legfőbb oka, amiért ennyire elterjedt az, hogy nagyon könnyen elérhetünk egy vagy több elemet a DOM-ban CSS selectorokat használva. Ez rengeteg getElementById és társai függvénymeghívástól megkíméli a programozókat. Ez a fajta lekérés mód a HTML5-ben már alaptól támogatva lesz, nem kell majd hozzá jQuery.

A jQuery szerkezeti felépítése lehetőséget nyújt annak bővítésére is, ezért számtalan hasznos plugin íródott már eddig is hozzá.

Feltétlen megemlíteném, hogy a jQuery hasznos társprojektje a jQueryUI (<http://jqueryui.com/>), ami a főként a vizuális dolgokért felel, például animációk és widgetek. A sablonozható jQueryUI talán legfontosabb elemei a drag and drop, újraméretezés és kiválasztás, illetve az effektusok. A widgetek segítségével könnyedén remek web applikációkat hozhatunk létre, használva a „füles” felhasználói felületet, a görgetőket, a gombokat, naptárakat és sok egyebet.

Egy újabb jQuery szülemény a jQuery Mobile (<http://jquerymobile.com/>), ami jelen írás idején még Alpha 2 verziónál tart. A jQuery Mobile háta mögött húzódó ötlet az, hogy az egyre növekvő okostelefon piacon is tiszteletre méltó részt tudjon magának, megkönnyítve a fejlesztők munkáját.

### Selectorok

A selectorok a jQuery könyvtár legfontosabb elemei közé tartoznak. A selectorok használatával gyorsan és egyszerűen azonosíthatóak az oldal egyes elemei, hogy aztán azok együttműködjenek további jQuery metódusokkal.

Négy alapvető módja van az oldal egyes elemeinek azonosítására selectorok használatával:

- tag neve által,
- ID által,
- osztálynév által,
- XPath kifejezés által.

Az XPath kifejezések tűnnek talán a legbonyolultabbnak, így ahhoz, hogy azokat megértsük, először tisztában kell lennünk az első három módszer lépéseivel.

---

<sup>2</sup> Forrás: [http://opendir.hu/?freedom=/tartalom/jquery/jquery\\_alapozo](http://opendir.hu/?freedom=/tartalom/jquery/jquery_alapozo)

Talán a legnépszerűbb módja egy elem elérésének JavaScript használatával az ID alapján történő azonosítás:

```
var változo;  
változo = document.getElementById("elemId");
```

Ugyanezt érhetjük el jQuery-ben az alábbi kód használatával:

```
var változo;  
változo = $("#elemId");
```

Ha jQuery segítségével ID alapján akarunk elérni egy elemet, akkor az ID előtt mindig a '#' prefixet kell használni.

Adott típusú tagek csoportjának elérésére is lehetőség van JavaScript segítségével. Például, ha az összes link taget (anchor tag) el akarod érni és rajtuk végezni egy műveletet, valami ehhez hasonló kódot kéne írunk:

```
var aTags;  
aTags = document.getElementsByTagName("a");  
var aIndex;  
for(aIndex = 0; aIndex < aTags.length; aIndex++){  
    /* műveletek */  
}
```

jQuery használatával ugyanez:

```
$("#a").elvezendoFuggveny();
```

Az osztálynév segítségével történő kiválasztás is hasonlóan egyszerűen megvalósítható jQuery segítségével.

```
$(".osztalyNev")
```

A fenti kódrészlet eredményül visszaadja az összes olyan elemet, amelynek a class attribútumában szerepel az osztalyNev.

Az eddigi példákból is jól látható, hogy mennyire egyszerű a DOM elemek elérése jQuery segítségével. XPath kifejezést eddig csak abban az esetben használtam, amikor nem volt más

lehetőségem. Ilyen eset például, amikor egy űrlapon meg kell „fogni” az összes text input mezőt, de az űrlap tartalmaz még rádiógombokat és reset gombot is. Ebben az esetben a legegyszerűbb megoldás, ha az alábbi XPath kifejezést használjuk:

```
$ (input[type=text])
```

## JSON

A JSON (JavaScript Object Notation, JavaScript objektumjelölés) emberek számára is olvasható-írható, programozottan pedig könnyen feldolgozható és előállítható, pehelysúlyú adatsere-formátum. A JSON a JavaScript programozási nyelv ([ECMA-262 szabvány harmadik kiadás, 1999. december](#)) egy részén alapul. A JSON noha programozási nyelvtől független szöveg-formátum, a C családú nyelvekben – C, C++, C#, Java, JavaScript, Perl, Python stb. – jártas programozó számára ismert konvenciókhoz igazodik. Ezek a tulajdonságok teszik a JSON-t ideális nyelvvé adatszeréhez.

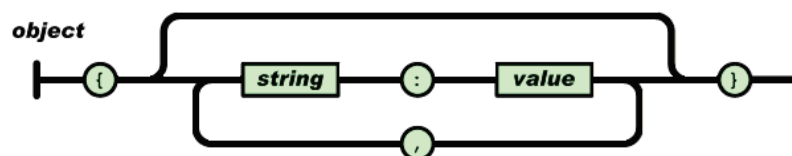
A JSON-nak két típusú felépítése lehetséges:

- *Név-érték párok halmaza:* ennek megvalósításai az objektum, rekord, struktúra, szótár, hasítótábla, kulcsos lista vagy asszociatív tömb lehetnek a különféle nyelvekben.
- *Értékek rendezett listája:* a legtöbb nyelvben ez tömb, vektor, lista vagy szekvencia.

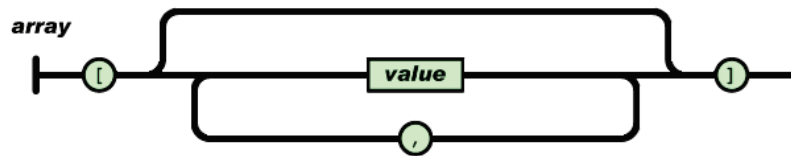
Ezek olyan univerzális adatszerkezetek, amiket gyakorlatilag minden programozási nyelv támogat ilyen vagy olyan formában. Ésszerűnek tűnik tehát, hogy az az adatformátum, amelyet programozási nyelvek kommunikációjához kívánunk használni, szintén ezekre a szerkezetekre épüljön.

JSON-ban ezek a szerkezetek az alábbi formákban nyilvánulnak meg:

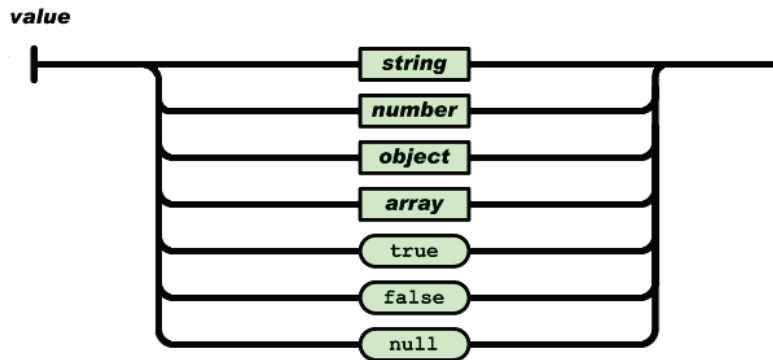
Az objektum név-érték párok rendezetlen halmaza. Egy objektum { (nyitó kapcsos zárójel) karakterrel kezdődik és } (záró kapcsos zárójel) karakterrel zárul. Minden nevet kettőspont követ. A név-érték párok vesszővel tagoltak.



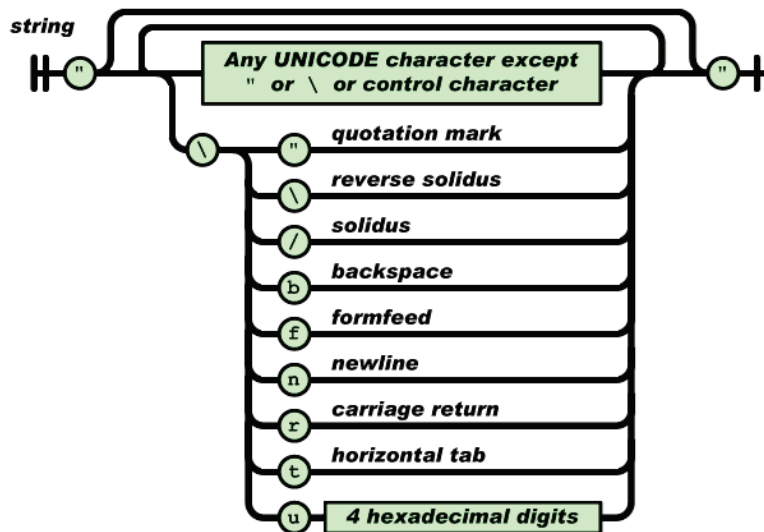
A tömb értékek rendezett halmaza. A tömb [ (nyitó szögletes zárójel) karakterrel kezdődik és ] (záró szögletes zárójel) karakterrel zárul. Az értékeket vesszővel választjuk el egymástól.



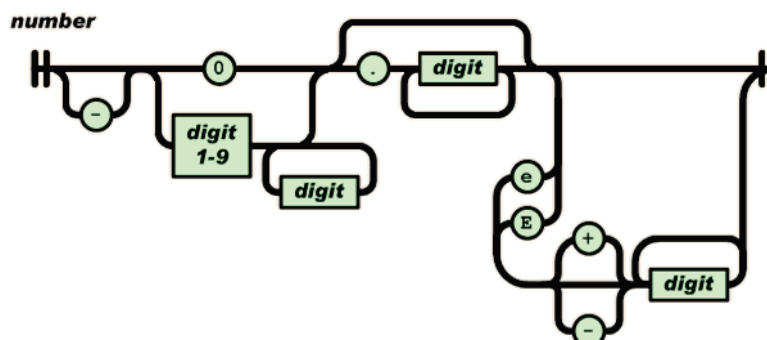
Érték lehet idézőjelek közé írt karakterlánc, szám, logikai igaz, logikai hamis, null, objektum vagy tömb. A struktúrák egymásba ágyazhatók.



A karakterlánc nulla vagy több, idézőjelek közé zárt Unicode karakter, szükség szerint \ karakterrel kivédve. A karakter egy 1 karakter hosszúságú karakterláncnak felel meg. A karakterlánc nagyban hasonlít a C vagy Java karakterláncaihoz.



A szám a C és Java számaihoz hasonló. A különbség az, hogy oktális és hexadecimális formátum itt nem használható.



Whitespace karakter bármely két JSON elem között megadható. Kivétel néhány kódolási részlet, amely egészében a nyelvet írja le.<sup>3</sup>

## Fogalomjegyzék

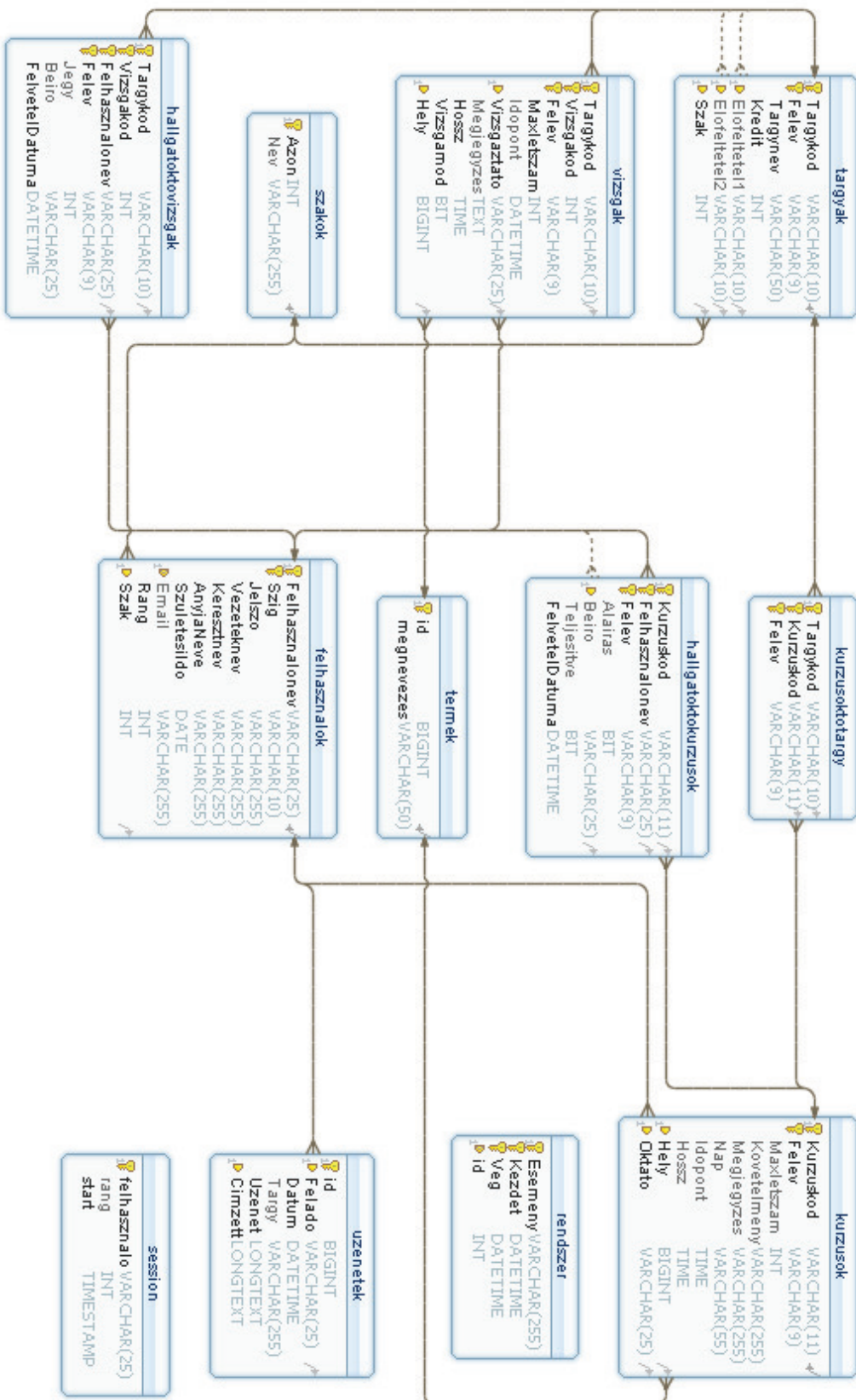
<b>Fogalom</b>	<b>Definíció</b>
felhasználó	Olyan személy, akinek legalább felhasználónevét és jelszavát tartalmazza a rendszer és be tud lépni vele.
hallgató	Olyan felhasználó, aki kurzusokat, vizsgákat vehet fel, illetve adhat le.
oktató	Olyan felhasználó, aki tantárgyat, vizsgát hirdethet meg, módosíthatja és törölheti, valamint aláírást és vizsgajegyet adhat.
adminisztrátor	Olyan felhasználó, aki új felhasználókat rögzíthet a rendszerben, illetve a már rögzített felhasználók adatait módosíthatja.
vizsga <sup>4</sup>	A vizsga (kollokvium) valamely tantárgy – általában egy félévet átfogó – anyagának számonkérése, amelynek sikeres letétele a tantárgyi követelményekben előírt kredit megszerzését jelenti. Értékelése ötfokozatú minősítéssel történik.

---

<sup>3</sup> Forrás: <http://www.json.org/json-hu.html>

<sup>4</sup> A Debreceni Egyetem Tanulmányi és Vizsgaszabályzata alapján.

# Adatbázisséma



## **Tárgyak tábla**

A tárgyak táblában tárolja a rendszer egy-egy tárgy fontosabb adatait. Ilyen adatok:

- a tárgykód,
- a tárgy neve,
- az a félév, amelyikben meg lett hirdetve,
- a tárgy kreditértéke,
- a szak, amin meghirdetésre került az adott tárgykóddal
- maximálisan két előfeltétel.

Az előfeltételek ugyanebben a táblában tárolt más tárgyakat hivatkoznak. Alapértelmezett értékük NULL.

## **Vizsgák tábla**

A vizsgák táblában tárolja a rendszer egy-egy vizsga adatait. Egy vizsga esetén ezek az adatok a következők:

- a tárgy kódja; amiből a vizsga történik,
- a vizsgakód; ami az adott félévben az adott vizsgához tartozó vizsga sorszáma,
- a félév, amelyikben a vizsga meghirdetése történt,
- a vizsgára maximálisan jelentkezhető hallgatók száma,
- vizsgáztató,
- a vizsga időtartama óra:perc formátumban,
- a vizsga módja (1 – írásbeli, 2 – szóbeli, 3 – írásbeli és szóbeli, 4 - gyakorlati),
- a vizsga helye,
- opcionálisan egy megjegyzés mező, ahol a vizsgát meghirdető felhasználó egyéb információt közölhet a vizsgára jelentkező hallgatókkal.

## **Kurzusok tábla**

A kurzusok tábla egy tárgy kurzusainak adatait tárolja. Egy-egy kurzus esetén nyilvántartja

- a félévet, amelyben az adott kurzus meg lett hirdetve,
- a kurzust felvehető hallgatók számát,
- a követelményt (aláírás vagy kollokvium),
- a kurzus időpontját,
- a kurzus hosszát óra:perc formátumban,
- oktatót,
- a kurzus helyét,

- opcionálisan egy megjegyzés mezőt, ahol a kurzusra jelentkező felhasználóknak lehet egyéb a kurzussal kapcsolatos információt megadni.

### **Felhasználók tábla**

Ebben a táblában tárolja a rendszer az összes rendszerben szereplő felhasználó adatait.

Egy-egy felhasználóról tartalmazza:

- a felhasználónevet, amivel bejelentkezhetsz a rendszerbe,
- a felhasználó jelszavának SHA1 algoritmus szerinti hash értékét,
- a felhasználó személyi igazolvány számát,
- a felhasználó vezeték és keresztnévét,
- a felhasználó anyja nevét,
- a felhasználó születési idejét,
- a felhasználó e-mail címét,
- a felhasználó szakjának kódját,
- a felhasználó rangját (1 – hallgató, 2 – oktató, 3 - adminisztrátor).

A rendszerben van egy „system” felhasználónevű felhasználó, akit semmilyen felhasználói lista nem jelöl. Erre a felhasználóra azért volt szükség, mert a rendszer az ő nevében küldi ki az automatikusan generálódó rendszerüzeneteket. Természetesen ezzel a felhasználóval nem lehet bejelentkezni.

### **Üzenetek tábla**

Ebben a táblában tárolja a rendszer az összes üzenetet. Egy üzenetről nyilvántartja a következőket:

- az üzenet azonosítóját,
- az üzenet feladójának azonosítóját,
- az üzenet feladásának idejét,
- az üzenet tárgyát,
- az üzenet szövegét,
- és az üzenet címzettjeinek azonosítóját vesszővel elválasztva.

Az első azonosító előtt is van egy vessző, amely megkönnyíti a keresést az adott mezőben. Amikor egy felhasználó törli az üzenetet, akkor a rendszer törli a felhasználó azonosítóját a címzettek közül. Ha az update művelet után, a címzett mező már csak egy vesszőt tartalmaz, akkor a rendszer ténylegesen törli az üzenetet.

### **Session tábla**

A rendszer ebben a táblában tárolja a felhasználók belépési adatait, amíg a rendszerben vannak. Három dolgot tárol:

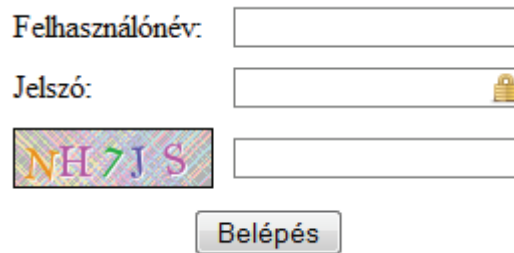
- a felhasználónevet,
- a felhasználó rangját,
- és a belépés időbélyegét.

A rendszer minden hallgató jogkörrel rendelkező felhasználót maximum 15 percig enged tevékenykedni. Ha lejárt a 15 perces időkorlát, a hallgató csak az első kattintása után kerül át a bejelentkező oldalra.

## Folyamatok


### Bejelentkezés

Egy adminisztrátor által rögzített felhasználó megadja a felhasználónevét és jelszavát és a rendszer által generált CAPTCHA kódot. Ha a megadott kód megegyezik a képen láthatóval, akkor megtörténik a hitelesítés, és ha a megadott adatok érvényesek, akkor létrejön a session. Ezzel párhuzamosan a felhasználó adatai bekerülnek a session nevű táblába.



Felhasználónév:

Jelszó:



1. ábra

Kényelmi szempontokból a CAPTCHA kódot akkor is érvényesnek tekinti a rendszer, ha az csak annyiban különbözik a képen láthatótól, hogy a benne lévő betűket kisbetűvel írtuk.

### Üzenetek olvasása

A felhasználó a sikeres bejelentkezést követően a kezdő képernyőn látja a neki küldött üzeneteket. Az üzenet előtti „Olvas” feliratra kattintva megjelenik a kiválasztott üzenet.

	Feladó ▾	Tárgy ▾	Dátum ▾	Töröl
<input type="checkbox"/> Olvas	Kósa Márk	A jövő heti óta elmarad!	2011-03-20 11:43:33	<input type="checkbox"/>

**A jövő heti óta elmarad!** ×

Tisztelt Hallgatók!

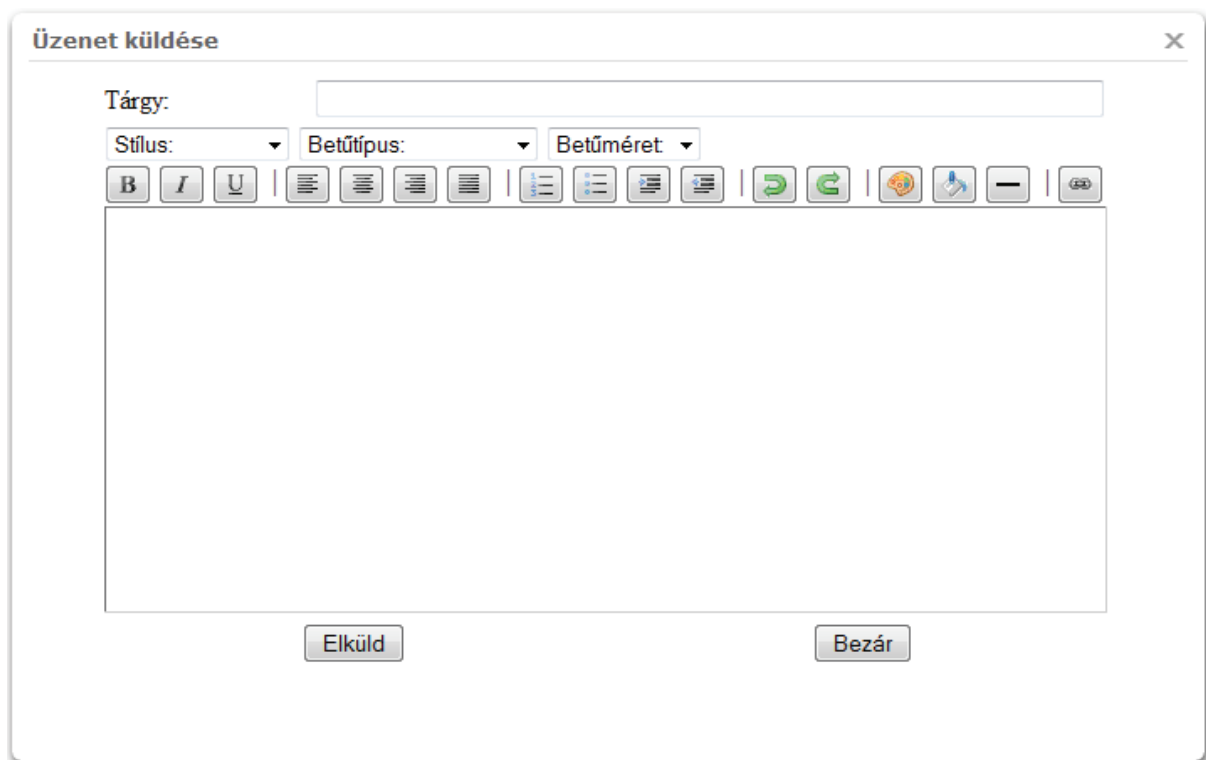
A jövő heti óra nem lesz megtartva!

Kósa Márk

2. ábra

### Üzenet írása

Mivel a rendszer jellegéből adódóan nem egy közösségi oldal, korlátoztam az üzenetek küldésének módját. A hallgatók egyáltalán nem tudnak üzenetet küldeni, csak fogadni. A tanár vagy adminisztrátor jogkörrel rendelkező felhasználók küldhetnek üzenetet egy-egy kurzus hallgatóinak.



3. ábra

Üzenet küldésekor megadható a küldendő üzenet tárgya és maga az üzenet szövege. Az üzenet formázható, de kényelmi szempontokat figyelembe véve lehetőség van külső szövegszerkesztőben megírt és formázott szöveg beillesztésére is a vágólapról. Mivel az Internet Explorer érzékeny a vágólapról beszúrt és formázott szövegben lévő megjegyzésekre, ezért az üzenet adatbázisba történő tárolása előtt ezeket a következő PHP kódrészlet segítségével eltávolítom.

```
$uzenet = utf8_decode($uzenet);  
$options = array('hide-comments' => true, 'clean'=> true, 'join-  
classes'=> false, 'join-styles'=> false, 'show-body-only' =>  
true);  
$tidy = tidy_parse_string($uzenet, $options);  
$tidy->cleanRepair();  
$uzenet = tidy_get_output($tidy);
```

## Tárgyak listázása





Hallgatói jogkörben a hallgató szakjához tartozó összes olyan tárgy listázása, amely meghirdetésre került az adott félévben.

Tárgykód	Tárgynév	Kredit	Teljesítve
INDK-101	Diszkrét matematika 1	5	<input type="checkbox"/>
INDK-111	Kalkulus 1	5	<input type="checkbox"/>
INDK-211	Operációs rendszerek 1	5	<input type="checkbox"/>
INDK-421	Adatszerkezetek és algoritmusok	5	<input checked="" type="checkbox"/>
INDK-441	A mesterséges intelligencia alapjai	5	<input type="checkbox"/>
INDK-501	Adatbázisrendszerek	5	<input type="checkbox"/>





4. ábra

A rendszer a „Teljesítve” oszlopban jelöli azokat a tárgyakat, amelyeket az adott hallgató sikeresen teljesített. Ezeket a tárgyakat a hallgató már nem tudja felvenni.

Oktatói jogkörben az adott oktató által tartott tárgyak/kurzusok listázása.

Kurzus kód	Kurzus név	Fő/Limit	Idő	Hely	
INDK-421E	Adatszerkezetek és algoritmusok	1/100	Hétfő 08:00	M418	   

5. ábra

Ahogy az 5. ábrán is látható, az oktató rögtön látja a saját kurzusait, illetve azoknak néhány főbb adatát. A megjelenő táblázat rendezhető az első öt oszlopa szerint. Az utolsó oszlopban pedig az oktató számára elérhető műveletek ikonjai szerepelnek. Az első (  ) ikon a kurzus hallgatóinak listáját mutatja meg, a második (  ) ikon segítségével tudja az oktató módosítani a kurzus adatait. Ez akkor lehet hasznos, ha a félév elején meg kell növelni a kurzus létszámát, ekkor az oktató egyszerűen átírja a létszámot és esetlegesen a kurzus helyét és időpontját is megváltozathatja. A harmadik (  ) ikon az üzenetküldő panelt jeleníti meg, míg a negyedik (  ) a kurzus hallgatóinak az értékeléséhez kell.

## Kurzus létrehozása, módosítása, törlése

Egy adminisztrátori jogkörrel rendelkező felhasználó egy adott tárgyhöz létrehozhat, módosíthat és törölhet kurzust. A kurzus adatait a kurzushoz rendelt oktató is módosíthatja a Tárgyak listázása pontban leírt módon. Egy kurzus meghirdetéséhez a 6. ábrán látható űrlapot kell kitölteni.

Félév:

Tárgynév:

Típus:  Előadás  Labor  Gyakorlat  Szeminárium

Max. létszám:

Követelmény:  Aláírás  Kollokvium

Időpont:  óra

Hossz:  (óra:perc)

Hely:

Oktató:

Megjegyzés:

6. ábra

### Kurzusfelvétel

Egy hallgatói jogkörrel rendelkező felhasználó jelentkezhet és leadhat tárgyakat a tárgyhoz tartozó kurzusok felvételével és leadásával. A kurzusfelvétel csak abban az esetben jár sikerrel, ha a következő feltételek mindegyike teljesül:

- a hallgató még nem teljesítette a tárgyat,
- tárgyfelvételi időszak van,
- az adott tárgyhoz tartozó összes előfeltételként szereplő tárgy teljesítve lett,
- az adott kurzus még nem telt be,
- a hallgatónak nincs az adott félévben ugyanezen tárgyhoz tartozó kurzusa.

Kurzus kód	Kurzus típusa	Fő/Limit	Hely	Órarend info.	Oktató	Megjegyzés	Felvesz
INDK-501E	Előadás	0/100	M419	Hétfő 10:00-12:00	Kósa Márk		<input type="checkbox"/>
INDK-501G01	Gyakorlat	0/20	M113	Kedd 08:00-10:00	Kósa Márk		<input type="checkbox"/>

7. ábra

Természetesen a hallgató egyszerre veheti fel az ugyanazon tárgyhoz tartozó előadást és gyakorlatot is.

## Vizsga létrehozása, módosítása, törlése

Oktatói vagy adminisztrátori jogkörrel rendelkező felhasználó egy adott tárgyhoz létrehozhat, módosíthat és törölhet vizsgát. Az oktató csak a saját vizsgáit módosíthatja, míg az adminisztrátor bármely vizsgát szerkeszthet. Egy vizsga meghirdetéséhez a 8. ábrán látható űrlapot kell kitölteni.

Tárgy:	<input type="text"/>
Időpont:	<input type="text"/> <input type="text" value="08:00"/>
Hossz:	<input type="text" value="02:00"/> (óra:perc)
Limit:	<input type="text"/> fő
Hely:	<input type="text"/>
Vizsgamód:	<input type="text" value="Írásbeli"/>
Vizsgáztató:	<input type="text" value="Kósa Márk"/>
Megjegyzés:	<input type="text"/>

8. ábra

A fenti képen az adminisztrátori vizsgahirdető form szerepel. Az oktatói űrlap ettől annyiban tér el, hogy a vizsgáztató mező nem módosítható.

## Vizsgajelentkezés

Egy hallgatói jogkörrel rendelkező felhasználó csak akkor jelentkezhet, ha a következő feltételek mindegyike teljesül:

- a hallgató felvette az adott félévben a tárgyat,
- a vizsgajelentkezés legalább 24 órával a vizsga kezdetét megelőzően történik,
- az adott vizsga még nem telt be,
- a hallgató teljesítette a vizsgára való jelentkezés feltételeit (pl.: megkapta az aláírást).

## Vizsga leadás

Egy hallgatói jogkörrel rendelkező felhasználó lejelentkezhet egy vizsgáról, ha a következő feltételek mindegyike teljesül:

- a hallgató előzőleg jelentkezett az adott vizsgára,
- a vizsga leadás legalább 24 órával a vizsga kezdetét megelőzően történik.









Tárgy ▾	Tárgykód ▾	Típus ▾	Kezdés ▾	Fő/Limit ▾	Terem ▾	Oktató ▾	Megjegyzés ▾	
Adatszerkezetek és algoritmusok	INDK-421	Írásbeli	2011-04-28 08:00:00	1/15	M418	Kósa Márk		Lead

9. ábra

Az utolsó oszlopban a művelet attól függően változik, hogy az adott hallgató jelentkezett-e a vizsgára, illetve hogy felveheti-e/leadhatja-e még.

## Időszakok listázása

Egy adminisztrátori jogkörrel rendelkező felhasználó, kilistáztathatja az összes, a rendszerben szereplő tantárgyfelvételi és vizsgajelentkezési időszakot. A lista alapértelmezetten az időszakok rögzítési sorrendjében van rendezve, de a táblázat fejlécére kattintva a rendezés módosítható.

Kezdet ▾	Vég ▾	Esemény ▾	
2010-02-19 10:00:00	2010-03-20 12:11:00	Tárgyfelvétel	 
2010-01-01 16:25:55	2010-06-30 16:26:04	Vizsgajelentkezés	 
2011-03-01 11:40:28	2011-04-30 11:40:37	Tárgyfelvétel	 
2011-03-02 11:40:44	2011-04-30 11:40:47	Vizsgajelentkezés	 

Új időszak

10. ábra

Az „Új időszak” gombra kattintva új időszakot lehet megadni.

## Időszakok kezelése

Egy adminisztrátori jogkörrel rendelkező felhasználó megadhat új tárgyfelvételi vagy vizsgajelentkezési időpontot, valamint szerkesztheti a már meglévőket. Egy új időszak rögzítéséhez az alábbi űrlapot kell kitölteni:

**Időszak módosítása**

**Kezdet:**

**Vég:**

**Esemény:** Tárgyfeltétel


Felvesz

2011	Április					
<b>Április 2011</b>						
Hé	Ke	Sz	Cs	Pé	Sz	Va
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

11 : 29 : 20

11. ábra

Az új időszak kezdetét és végét is egy-egy dátumválasztó mezővel lehet megadni.

A rendszerben lévő időszakokat a mellettük lévő  ikonra kattintva lehet módosítani, és a  ikon segítségével törölni.

## Hallgató jelentkeztetése kurzusra

Egy adminisztrátori jogkörrel rendelkező felhasználó hozzáadhat egy hallgató jogkörű felhasználót bármely kurzushoz, függetlenül attól, hogy tárgyfelvételi időszak van-e, és hogy az adott kurzus betelt-e. A hozzárendelés egy többlépcsős folyamat. Első lépésben az adminisztrátornak ki kell választania azt a szakot, amilyen szakon a kurzus meg lett hirdetve, és ki kell választania a hallgatót is. A könnyebb kezelhetőség érdekében a szakok név szerint ábécésorrendben vannak, a hallgatók kiválasztását pedig automatikus kiegészítő segíti, ahogy az a 12. ábrán is látható.


**Szak:** Programtervező informatikus

**Felhasználónév:** p

Kiválaszt

- proba
- proba123
- proba2

12. ábra

A „Kiválaszt” gombra való kattintás után jobb oldalon meg fog jelenni azon tárgyak listája, amelyek az aktuális félévben a kiválasztott szakon meghirdetésre kerültek. A tárgylista rendezhető, ezzel is segítve a felhasználót. Ha megtaláltuk a tárgyat, akkor a mellette lévő „Hozzárendel” felírra kattintva, megjelenik a tárgyhoz tartozó kurzusok listája. Végezetül a 13. ábrán látható táblázat utolsó oszlopában lévő  ikonra kattintva megtörténik a hozzárendelés.

Adatszerkezetek és algoritmusok kurzusai							
Kód ▾	Típus ▾	Fő/Limit ▾	Nap ▾	Időpont ▾	Oktató ▾	Hely ▾	
INDK-421E	Előadás	1/100	Hétfő	08:00	Kósa Márk	12	

13. ábra

## Hallgatói jogkör

A rendszerben a hallgatóknak van a legkevesebb szolgáltatáshoz hozzáférésük. Bejelentkezés után a főképernyőn megjelennek a hallgatónak címzett üzenetek, melyeket az előtűk lévő „Olvas” linkre kattintva el is olvashat. Sokáig gondolkoztam, hogy lehessen-e hallgatói szerepkörben is üzenetet írni, de ezt végül elvettem, mert idővel nagyon leterhelné a rendszert, valamint az én elképzeléseim szerint a hallgató csak a ténylegesen fontos dolgokról kap üzenetet pl.: oktatói üzenetek, vizsgaeredmény beírása, stb.

A Tárgyak menüpont alatt jelenik meg a hallgató szakjának megfelelő, az adott félévben meghirdetett tárgyak listája. Az egy-egy tárgyhoz tartozó kurzusokat a kurzuskódra kattintva érhetjük el. A tárgyfelvételnél esetlegesen előforduló túljelentkezés elkerülésének érdekében a jelentkezés rögzítését egy tárolt eljárás végzi. A hallgató jelentkezése többek között akkor sikeres az adott kurzusra vagy vizsgára, ha az még nem érte el a maximális létszámot.

A Vizsgák menüpont alatt az aktuális hallgató által felvett tárgyakhoz meghirdetett vizsgaidőpontok jelennek meg.

A Leckekönyv menüpont alatt a hallgató leckekönyvének egy digitális változata található. Benne szemeszterenként, tárgyanként megtalálhatóak a hallgató vizsgaeredményei.

Ezen funkciók mellett a hallgatónak lehetősége van még egyes személyes adatait módosítani. Ilyen adatok a vezetéknev, keresztnév, személyi igazolvány száma és e-mail címe. Továbbá ugyanitt lehetősége van a hallgatónak megváltoztatni a bejelentkezéshez szükséges jelszavát.

## Oktatói jogkör

Az oktatók számára elérhető funkciók: üzenetek olvasása és írása, saját kurzusok adatainak listázása és módosítása, saját vizsgák listázása és adatainak módosítása, aláírás megadása, jegybeírás.

Az oktatók a Tárgyak menüpont alatt az általuk tartott tárgyakról találhatnak információt. Ugyanitt tudják módosítani is ezeket az adatokat, üzenetet küldeni egy-egy tárgy vagy kurzus hallgatói számára.


A Vizsgák menüpont alatt lehet az aktuális oktató vizsgáit listázni, az egyes vizsgák eredményeit rögzíteni a rendszerbe. Miután az oktató sikeresen rögzítette az adott vizsga eredményeit, a rendszer minden, az adott vizsgára jelentkezett hallgató számára küld egy sablonüzenetet, amely tartalmazza, hogy ki, mikor, milyen tárgyból milyen érdemjegyet adott a hallgatónak. Ezenkívül az oktatónak lehetősége van az általa az adott félévben tartott tárgyakból vizsgaalkalmak meghirdetésére. Egy vizsga meghirdetéséhez az oktatónak meg kell adnia a következő adatokat:

- a tárgyat, amiből a vizsgaalkalmat meghirdeti,
- a vizsga kezdetének időpontját,
- a vizsga várható időtartamát,
- a maximális hallgatólétszámot,
- a vizsga helyét,
- a vizsga módját,
- esetlegesen megjegyzést.

## Adminisztrátori jogkör

A rendszerben az adminisztrátori jogkör a legmagasabb, ebből kifolyólag nekik biztosítja a rendszer a legtöbb szolgáltatást. A rendszer adminisztrátorainak lehetőségük van a rendszerben lévő összes felhasználó listázására és bizonyos adatainak módosítására. Lehetősége van új felhasználót rögzíteni a rendszerbe a 14. ábrán szereplő űrlap kitöltésével.

**Új felhasználó felvétele**

<b>Vezetéknév:</b>	<input type="text"/>
<b>Keresztnév:</b>	<input type="text"/>
<b>Anyja neve:</b>	<input type="text"/>
<b>Születési dátum:</b>	<input type="text"/> 
<b>Személyi igazolvány száma:</b>	<input type="text"/>
<b>E-mail:</b>	<input type="text"/>
<b>Szak:</b>	<input type="text"/>
<b>Rang:</b>	<input checked="" type="radio"/> hallgató <input type="radio"/> oktató <input type="radio"/> TO

14. ábra

Az újonnan felvett felhasználó jelszava a fenti űrlapban megadott születési dátum lesz *ééééhhnn* formátumban. Ezt a jelszót minden felhasználó bármikor megváltoztathatja.

Egy adminisztrátornak továbbá lehetősége van tárgyakat meghirdetni, hallgatókat hozzárendelni kurzusokhoz – még akkor is, ha nincs kurzusfelvételi időszak –, tárgyakat meghirdetni, adatait módosítani, vizsgát meghirdetni, vizsga eredményt rögzíteni, valamint időszakokat kezelni.

A hallgató kurzusokhoz való hozzárendelését egy tárolt eljárás végzi, melynek forrása:

```
BEGIN
DECLARE felvettek INT;
DECLARE maxfo INT;
DECLARE l_kezdet DATETIME;
DECLARE l_veg DATETIME;
SELECT count(*) INTO felvettek FROM hallgatoctokurzusok WHERE
Kurzuskod=kurzus;
SELECT Maxletszam INTO maxfo FROM kurzusok WHERE Kurzuskod=kurzus;
```

```

SELECT Kezdet, Veg INTO l_kezdet, l_veg FROM rendszer WHERE
Kezdet<NOW() AND
      NOW()<Veg AND Esemeny='Targyfelvetel';
IF (felvettek<maxfo) AND ((rang!=1) OR (rang=1 AND l_kezdet<NOW()
AND NOW()<l_veg)) THEN
INSERT INTO hallgatoktokurzusok (Kurzuskod, Felhasznamev, Alairas,
Beiro, Felev, FelvetelDatuma) VALUES (kurzus, hallgato, NULL, NULL,
felev, NOW());
END IF;
END

```

Az eljárás paraméterül kapja kurzus kódját, a hallgató azonosítóját, a hívó fél rangját és az aktuális félév azonosítóját.

## Biztonság

Mint minden rendszer esetében, itt is fontos szerepe van a biztonságnak, főleg azért, mert az elkészült rendszer webes környezetben fog futni, ahol állandóan ki lesz téve különböző támadó botok, scriptek, hackerek és unatkozó, programozói beállítottságú személyek támadásának. Éppen ezért a kritikus műveletek esetében többrétegű azonosítást használtam. A webes botok elleni védelem érdekében a bejelentkező képernyőre egy CAPTCHA tesztet is tettem. A CAPTCHA a számítástechnikában olyan automatikus teszt, ami képes megkülönböztetni az emberi felhasználót a számítógéptől. A kifejezés a "Completely Automated Public Turing test to tell Computers and Humans Apart" (teljesen automatizált nyilvános Turing-teszt a számítógép és az ember megkülönböztetésére) rövidítése. A teszt során a számítógép generál egy feladványt, amit csak egy ember tud helyesen megválaszolni, de a válasz helyességét a gép is könnyedén el tudja dönteni.<sup>5</sup> Bár ez a megoldás a botok jelentős részét kiszűri, de az emberi támadások ellen nem véd, ezért az egyes oldalak megjelenítését jogosultságokhoz kötöttem. Ha valaki meg akar tekinteni egy oldalt, de nincs bejelentkezve, vagy be van jelentkezve, de nem megfelelő a jogosultsági szintje, esetleg nem a bejelentkező oldalon keresztül akarja elérni a rendszert, akkor a rendszer automatikusan átirányítja a felhasználót a bejelentkező oldalra, és egyben megszünteti a felhasználó munkamenetét. A lentebb látható kód mutatja be az oldalak megtekintéséhez szükséges jogosultság ellenőrző függvényt.

```
function auth($rang = null){
    sessionCheck();
    if(!isset($_SESSION['felhasznalonev']))||
!isset($_SESSION['loginimg'])){
        session_destroy();
        header("location: ../index.php");
    }
    if(($rang != null) && (intval($_SESSION['Rang'])<intval($rang))){
        session_destroy();
        header("location: ../index.php");
    }
    if($_SESSION['felhasznalonev'] == ""){
        session_destroy();
    }
}
```

---

<sup>5</sup> Forrás: Wikipédia

```
        header("location: ../index.php");
    }
}
```

Mint látható, az auth függvény első lépésben meghívja a sessionCheck nevű függvényt, melynek forráskódja:

```
function sessionCheck() {
    if ($_SESSION['Rang']==HALLGATO) {
        $ok=mysql_query("SELECT * FROM `session` WHERE
ADDDATE(start,INTERVAL 15 MINUTE)>NOW() AND rang=1 AND felhasznalo =
'{"$_SESSION['felhasznalonev']}'");
        if(mysql_numrows($ok)==0) {
            ?>
            <script type="text/javascript">
                window.top.location.href="../module/logout.php";
            </script>
            <?php
            }
        }
    }
}
```

A függvény megnézi, hogy hallgató kérvényezte-e a műveletet. Erre azért volt szükség, hogy az oktatókat vagy az adminisztrátorokat ne érintse a 15 perces szabály, mert nekik könnyen elképzelhető a 15 percnél hosszabb munkamenet. Ilyen lehet például nagy mennyiségű vizsgaeredmény rögzítése, vagy felhasználók felvétele. Ezekben az esetekben nagyban rontaná a felhasználhatóságot, ha a felhasználónak még az órára is figyelnie kell, nehogy kiléptesse a rendszer.

A munkamenetek kezelése úgy működik, hogy belépéskor a session nevű táblába bekerül a felhasználónév, a rang és az aktuális időbélyeg. Innentől kezdve, minden kérésnél le fog futni a sessionCheck függvény, és ha a függvény harmadik sorában lévő SQL lekérdezés nulla sorral tér vissza, akkor átirányítja a hallgatót a kijelentkeztető oldalra, amely automatikusan kitörli a felhasználót a session táblából, valamint felszabadítja a munkamenethez tartozó változókat.

Ha a felhasználó, nem a hagyományos módon – a Kilépés menüpontra kattintva – lép ki a rendszerből, akkor természetesen nem törlődik a megkezdett munkamenete a session nevű

táblából. Ez azért jelent problémát, mert a következő bejelentkezéskor a rendszer érzékelni fogja, hogy neki lejárt a munkamenete és a sikeres bejelentkezés után rögtön ki is lépteti. Azért, hogy ilyen helyzet ne fordulhasson elő, a bejelentkező folyamat automatikusan törli a felhasználót a session táblából, majd beszúrja az aktuális időbélyeggel.

Ezenkívül bekerült a rendszerbe egy harmadik biztonsági ellenőrzés is, bár ez csak néhány, a rendszer szempontjából kritikus műveletének futását ellenőrzi. Ilyen műveletek például az aláírás megadása vagy a vizsgajegy beírás. Ekkor a MySQL adatbázisban lévő megfelelő tárolt eljárás megvizsgálja, hogy őt milyen ranggal és melyik felhasználó hívta meg, és ha az nem egyezik a felhasználók táblájában lévő adatokkal, a tranzakciót nem hajtja végre.

A rendszer további védelme érdekében a felhasználótól érkező adatokkal meghívom a „tisztít” nevű függvényt, aminek a forrása:

```
function tisztit($mit, $hossz=-1) {
    if($hossz>0) {
        $mit = substr ($mit, 0, $hossz);
    }
    return trim(escapeshellcmd(mysql_escape_string($mit)));
}
```

A függvénynek a tisztítandó érték mellett opcionálisan meg lehet adni egy számot is, és ilyenkor a függvény visszatérési értékének a hossza maximum a paraméterül kapott szám lesz. A függvény alkalmazásával elkerülhető az SQL injection és a shell scriptek végrehajtása.

## Összefoglalás

Bár a fejlesztés során használt technikák egy részét már korábban is találkoztam, mégis sok újdonságot tapasztaltam a fejlesztés során, melyeket a későbbiekben is hasznosítani tudok. Az adatbázis alaposan átgondolt tervezésének köszönhetően a rendszer bővítése nem eredményezné a rendszer teljesítmények számottevő romlását.

Többek között az idő hiánya miatt néhány funkció megvalósítására nem került sor. Az első terveim között szerepelt egy pénzügyi és egy órarendszerkesztő modul megírása is. Sajnos mire az implementáció arra a pontjára ért, hogy a pénzügyi modul integrálása megtörténhessen, az OTP Bank regisztrációhoz kötötte annak használatát. A rendszer tovább bővíthető még index nyilvántartási modullal, valamint egy demonstrátori jogkörrel.

## Irodalomjegyzék

PHP dokumentáció, mely letölthető innen: <http://php.net/>

MySQL dokumentáció és maga az kiszolgáló: <http://www.mysql.com/>

jQuery alap keretrendszer és dokumentáció: <http://jquery.com/>

A jQuery-hez fejlesztett grafikus környezet és dokumentációja: <http://jqueryui.com/>

Highslide dokumentáció és keretrendszer: <http://highslide.com/>

## Köszönetnyilvánítás

Szeretnék köszönetet mondani mindazoknak, akik tudásukkal, tapasztalatukkal segítették ennek a dolgozatnak a létrejöttét.

Külön szeretném megköszönni:

- Kósa Márk tanár úrnak a sok hasznos tanácsot és segítséget, amit témavezetőként és az általa oktatott tárgyak formájában adott át.
- Pánovics János tanár úrnak, az Adatbázisrendszerek megvalósítása 1-2 című tárgyakon tanultakért.