

Tartalomjegyzék

TARTALOMJEGYZÉK	1
BEVEZETÉS	3
1. ELVÁRÁSOK A WEBLAPPAL SZEMBEN	4
1.1 Feladat specifikáció	4
1.2 Használati esetek	4
1.2.1 Felhasználók	4
1.2.2 Felhasználói eset diagramm	5
1.2.3 Felhasználói esetek	5
1.3 Felhasználói esetek részletezése	7
1.3.1 Felhasználó felvétele	7
1.3.2 Felhasználó törlése	7
1.3.3 Üzenetírás	7
1.3.4 Kérdéssor összeállítása	7
1.3.5 Dolgozat megoldás	7
1.3.6 Tantárgyfelvétele	8
1.3.7 Kérdés hozzáfűzése	8
1.3.8 Diák csoportba rendezés	8
2. ADATBÁZIS TERVEZÉS	9
2.1.1 Egyed – kapcsolat diagram	9
2.1.2 Relációs adatbázis séma	9
2.1.3 Az adatbázis táblái	10
2.1.4 Táblák részletezése	11
2.2 Navigáció	15
2.3 Osztályok, függvények és feladatai	16
2.3.1 A STATISZTIKA osztály	16
2.3.2 Az SQL osztály	17
2.3.3 A LOGIN osztály	18
2.4 Dolgozatírás folyamata	19
3. PROGRAMOZÁS, KÓDOLÁS	20
3.1 Fejlesztői környezet	20
3.2 A felhasználó nyomkövetése	20
3.3 Statisztikai kimutatás	21
4. TESZTELÉS	22
4.1 Üzembe helyezési teszt	22
4.2 Általános funkcionális teszt	22
4.3 Szélsőérték funkcionális teszt	26

5. ÜZEMBE HELYEZÉS, TELEPÍTÉS	27
5.1 Hardver, szoftver környezet.....	27
5.2 Appserv telepítése.....	27
5.3 A weblap telepítése	30
6. FELHASZNÁLÓI DOKUMENTÁCIÓ	32
6.1 A weblap használata.....	32
6.1.1 Az adminisztrátor lehetőségei	33
6.1.2 Az oktató lehetőségei.....	36
6.1.3 A diák lehetőségei	40
ÖSSZEFOGLALÁS.....	42
IRODALOMJEGYZÉK	43
CD MELLÉKLET	44

Bevezetés

Mára az informatika oly mértékben elterjedt, hogy minden iskolában, majd minden órán elérhetővé vált az internet ezáltal miért ne lehetne az órai dolgozatokat számítógép előtt, számítógép segítségével elkészíteni. Ennek kapcsán merült fel bennem, hogy kellene készíteni egy olyan programot, mely az oktató munkáját megkönnyítve a dolgozatírást interaktívvá tenné. Amennyiben ezt a programot teljes egészében elkészítenénk, úgy feleslegessé válna mindennemű papírmunka. Egy egyszerű szinkronizációval az oktató akár a saját palmtop-jára tölthetné az adatokat, mely segítségével a nap bármely szakában pontos információkkal rendelkezne a diákok tanulmányaival kapcsolatban. Természetesen a célkitűzésekben nem az osztálynapló megvalósítása áll, hanem maga a számonkérés, de nem sok módosítással könnyen megvalósítható egy olyan osztálynapló mely minden igényt kielégítene. Ennek a programnak mindenféleképpen hálózaton kellene működnie, és a legegyszerűbb lenne, ha mindez böngészőn keresztül lenne elérhető, ugyanis ekkor nem számít a kliens számítógép hardver és szoftver környezete. Egy régi számítógéppel is bárki hozzáférhetne, nem adódhat operációs rendszerekből származó kompatibilitási hiba. Viszont ha a másik oldalt vizsgáljuk, azaz a kiszolgált (szerver) és az adatátvitelt, akkor ott már bizony meg kell felelni bizonyos elvárásoknak. Mivel ez egy dolgozatírás itt nagyon számít az idő, tehát fontos, hogy a diák rögtön tudjon válaszolni, ne kelljen várni egy újabb kérdésre, mert ez befolyásolhatná a dolgozat végeredményét. Annak érdekében, hogy a diák ne szenvedjen hiányt olyan programozási eljárásokat igyekeztem megvalósítani, mely a legegyszerűbb és leghatékonyabb.

Az alkalmazás elkészítéséhez a PHP és XHTML nyelvet továbbá a MySQL adatbázis kezelő rendszert használok fel. Az ok, amiért eme a párosítás mellett vagyok az, hogy ez a programnyelv a legelterjedtebb a weboldalak világában, a legkönnyebben elérhető, nagyon sok ismeretanyag áll az érdeklődők rendelkezésére (néhányat kiemeltem az [1] – [4] – esnél) és nem utolsó sorban a fejlesztői eszközök ingyenesen állnak a rendelkezésre.

Az alkalmazást igyekszem úgy kialakítani, hogy egyszerűen bővíthető legyen, egy esetleges kérésre könnyen átalakítható, személyre szabható legyen. Az alkalmazásnak sok funkciója van, mely lehetővé teszi, hogy bármiféle módosítás nélkül bárhol – gondolok itt középiskolára és felsőoktatásra egyaránt – alkalmazható legyen.

1. Elvárások a Weblappal szemben

1.1 Feladat specifikáció

A weboldal elsődleges feladata, a tanár munkáját megkönnyítve a diákok tudásának mérése. Mindez alkalmazható a szakközépiskolákban és a felsőoktatásban egyaránt. Egy adott csoportvezető (oktató) képes a csoport számára összeállítani tetszőleges kérdéssorokat, melyeket a csoport tagjai megoldhatnak. Ennek eredményei tárolásra kerülnek. Továbbá a weblap lehetővé teszi, egy adott tantárgyat kiválasztva, a diákok gyakorolhassanak, így készülve a számonkérésre. Ezeken felül apróbb szolgáltatásokkal bővítettem ki a programot, mint például az üzenetküldés, összesítés stb.

Az ADMIN lehetőségei: Az adminisztrátor teljes körű hozzáférést élvez az egész adatbázishoz, de a program admin felületén egyelőre a következő funkciók érhetők el számára. Felhasználó felvétele, illetve törlése. Tantárgy felvétele, módosítása. Kérdések törlése az adatbázisból. Üzenetírás és olvasás.

Az OKTATÓ lehetőségei: Kérdés hozzáfűzése az adatbázishoz, kérdéssor készítése egy adott (saját) csoport számára, egy adott időpontra, egy adott tantárgyból. Csoportlétrehozás, illetve törlés. Képes csoportba rendezni diákokat, vagy onnan kivenni. Saját csoportjainak statisztikáinak megtekintésére, és képes üzenetet írni, olvasni.

A DIÁK lehetőségei: A diákok, természetesen belépés után képesek gyakorolni, mely témát maguk választanak meg. Képesek a tanár által összeállított kérdéssort megoldani, saját eredményeiket megtekinteni, illetve a hozzájuk szóló üzeneteket olvasni.

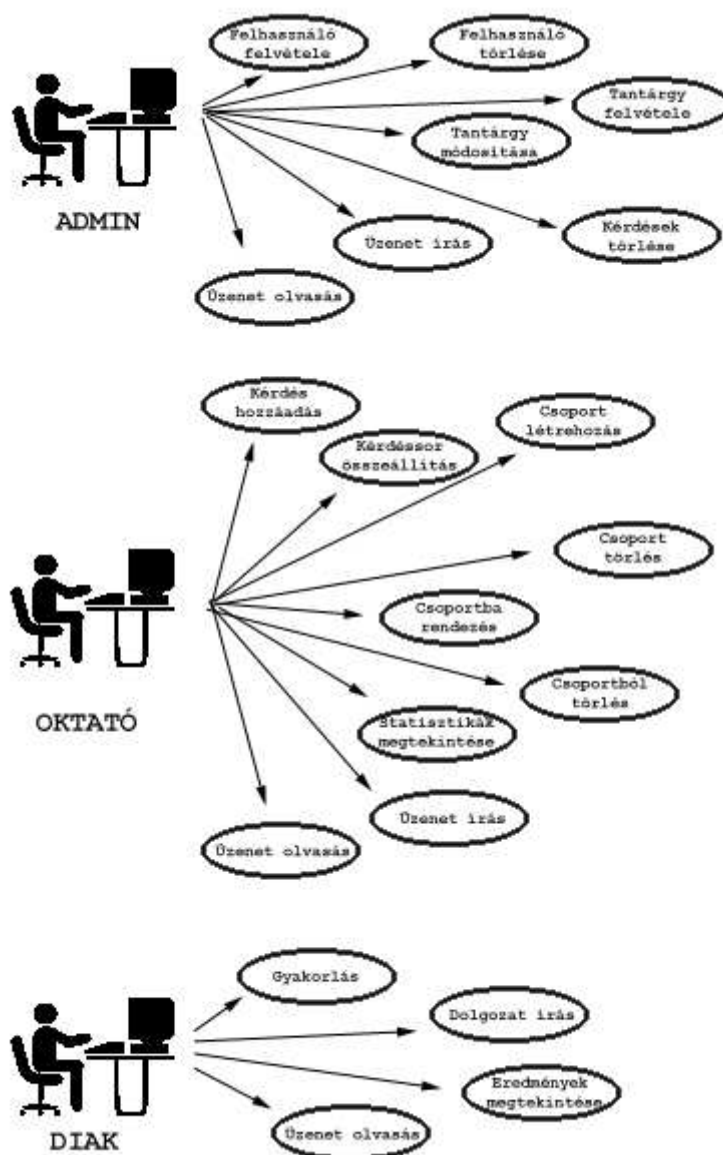
1.2 Használati esetek

Használati esetek tervezésénél a [5,14.-23.oldal] -es hivatkozásnál található irodalmat használtam fel.

1.2.1 Felhasználók

Ezen a Weboldalon 3 felhasználó fordul elő, mindegyiknek más és más lehetőségei vannak. A legfőbb aktor az *Admin*, akinek mindenhez hozzáférése van, valamint a többi felhasználó hozzáadását is ő végzi. A másik két felhasználó egymással kapcsolatban van, de alárendelt a viszony. A *Tanár* változtathat a *Diák*-okhoz tartozó adatokon, míg a *Diák* csak megtekintheti ezeket.

1.2.2 Felhasználói eset diagramm



1. 2.2.1. ábra Felhasználói eset diagram

1.2.3 Felhasználói esetek

Felhasználó felvétele:

Az Admin ebben a menüben hozhat létre új felhasználót, aki lehet oktató vagy diák.

Felhasználó törlése:

Itt törölhet felhasználót az admin.

Tantárgy felvétele:

Adminként lehet ebben a menüben tantárgyat hozzáfűzni az adatbázishoz.

Tantárgy módosítása:	Az admin itt tudja megváltoztatni a tantárgy jellemzőit.
Kérdés törlése:	Kérdéseket lehet törölni az adatbázisból.
Üzenetírás:	Ebben a menüben küldhetünk üzeneteket, változásokról.
Üzenetolvasás:	Az admin itt tudja elolvasni a hozzá érkező kéréseket.
--	
Kérdés hozzáadás:	Az oktató itt képes kérdést hozzáadni az adatbázishoz.
Kérdéssor összeállítás:	Ebben a menüben képes az oktató egy komplett kérdéssort összeállítani.
Csoport létrehozás:	Csoportot tud az oktató létrehozni, egy adott időpontra, helyre.
Csoport törlés:	A csoportokat képes megszüntetni is az oktató.
Csoportba rendezés:	Egy már létrehozott csoportba képes az oktató diákokat belerakni.
Csoportból törlés:	Az oktató egyes diákokat képes kivenni egy adott csoportból.
Statisztikák megtekintése:	Az oktató megtekintheti saját csoportjai eredményeit, átlagait.
Üzenetírás:	Üzenetet tud küldeni a rendszer minden felhasználójának.
Üzenetolvasás:	Az oktátónak címzett üzeneteket képes olvasni.
Gyakorlás:	A diák ebben a menüben egy kiválasztott tantárgyból készülhet egy esetleges dolgozatra.
Dolgozat megoldás:	Ebben a menüben a tanár által elkészített dolgozatot képes a diák megoldani.
Eredmények megtekintése:	Ebben a menüben a diák a saját eredményeit képes megtekinteni.
Üzenetolvasás:	A diák a hozzá, illetve a csoportjához szóló üzeneteket tudja itt olvasni.

1.3 Felhasználói esetek részletezése

1.3.1 Felhasználó felvétele

Admin tölti ki az űrlapot, ahol meg kell adni a felhasználó azonosítót, a felhasználó nevét, születési dátumot, e-mail-t és jelszót, amennyiben tanárról van szó, úgy a TANAR jelölő négyzetet kijelölve és a megjegyzés mezőt opcionálisan kitöltve elküldjük. Ezek után mind kliens oldalon (javascript), mind szerver oldalon (PHP) ellenőrizzük, hogy adott e meg adatot, illetve a felhasználó nem szerepel e már az adatbázisban. Majd a helyes adatok feltöltődnek a megfelelő táblákba.

1.3.2 Felhasználó törlése

Ebben az esetben az admin egy listát lát az adatbázisban szereplő felhasználókról. Mindenkihez tartozik egy törlés művelet, ha ezt választja az admin, akkor egy-egyszeri megerősítés kérés (javascript) után a felhasználó törlődik az adatbázisból.

1.3.3 Üzenetírás

Adminnak és oktátónak van joga üzenetet küldeni. Megadható, hogy konkrét személy, diák csoport, vagy mindenki megkapja az üzenetet. Az üzenetnek van tárgya, dátuma (automatikus), szövege és valamilyen formában címzettje. Ha mindez megvan, akkor a program tárolja az üzenetek táblában.

1.3.4 Kérdéssor összeállítása

Az oktató képes arra, hogy egy adott csoportnak, adott időpontra, saját kérdésekből álló kérdéssort készítsen. Fontos az időpont megadása, mert a diákok csak abban az időpontban tudja végigvinni a dolgozatot.

1.3.5 Dolgozat megoldás

A diák a dolgozatírás menüpontra kattintva választhat az éppen aktuális dolgozatok közül, aminek az eredményei tárolásra kerül. Egy dolgozatot csak egyszer lehet kiválasztani, elvégezni.

1.3.6 Tantárgyfelvétele

Az admin felületen keresztül lehet az adatbázisba új tantárgyat felvinni. Kötelező megadni a tantárgy nevet, a megjegyzés opcionális. Ezek után, amennyiben az adatok helyesek, a program egy azonosítót rendel a tantárgyhoz, és azzal együtt tárolja el a megfelelő táblákba.

1.3.7 Kérdés hozzáfűzése

Az oktató felületen a Kérdés hozzáadása menüpontban tehetünk új kérdéseket az adatbázisba. Először meg kell adnunk, hogy mely tantárgyhoz kapcsolódik a kérdés, ezt egy legördülő menüből választjuk ki. Majd magát a kérdést adjuk meg. Ezután a négy lehetséges választ adjuk meg, majd a helyes válasz sorszámát. Végül a helyes válasz esetén a pontértéket.

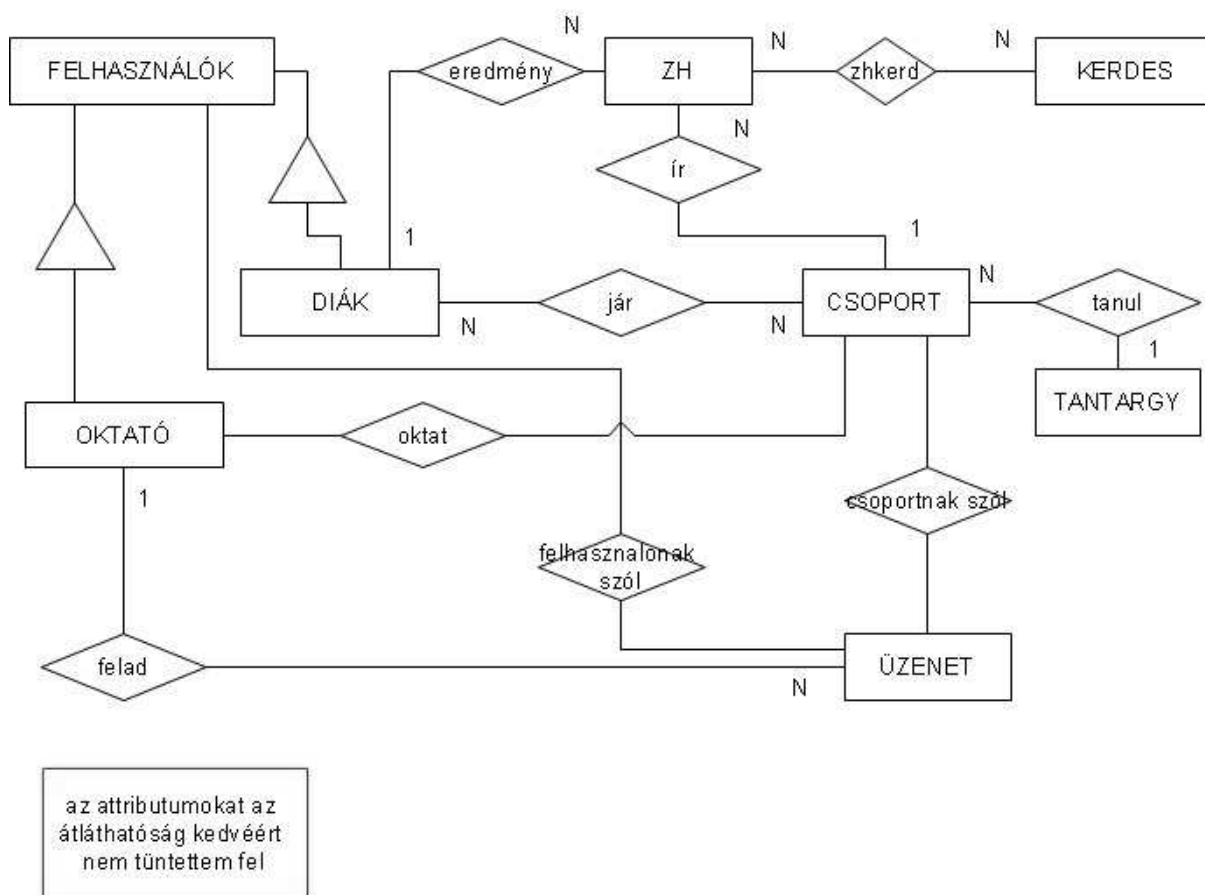
1.3.8 Diák csoportba rendezés

Az oktató teheti meg, hogy egy adott csoportba már létező diákot rendel. Ezt két legördülő menü segítségével hajthatja végre.

2. Adatbázis tervezés

A Tudásfelmérő Weblap a felhasználók, és a dolgozatírás minden egyes adatát egy relációs adatbázis fogja tartalmazni. Első lépésként az Egyed – kapcsolat diagramot (2.1.1.1 ábra) készítsük el, mely alapján a későbbi szisztematikus tervezés elvégezhető. Ez lesz az egész tervezés alapja. A szakdolgozat készítésnek ebben a szakaszában nagyon sokat segített előzetes tanulmányaim itt a Kecskeméti Főiskolán.

2.1.1 Egyed – kapcsolat diagram



2.1.1.1. ábra E-K diagram

2.1.2 Relációs adatbázis séma

USER (*useracc*, *userpass*, *usernev*, *userszuldate*, *useremail*, *tanar*);

EREDMENY (*useracc*, *zhazon*, *eredmeny*);

JAR (*useracc*, *csoportazon*);

OKTATO (*oktatoazon*, *useracc*, *megjegyzes*);

CSOPORT (*csoportazon*, *ideje*, *helye*, *tantargyazon*, *oktatoazon*);

ZH (*zhazon*, *csoportazon*, *megjegyzes*, *mikorra*);

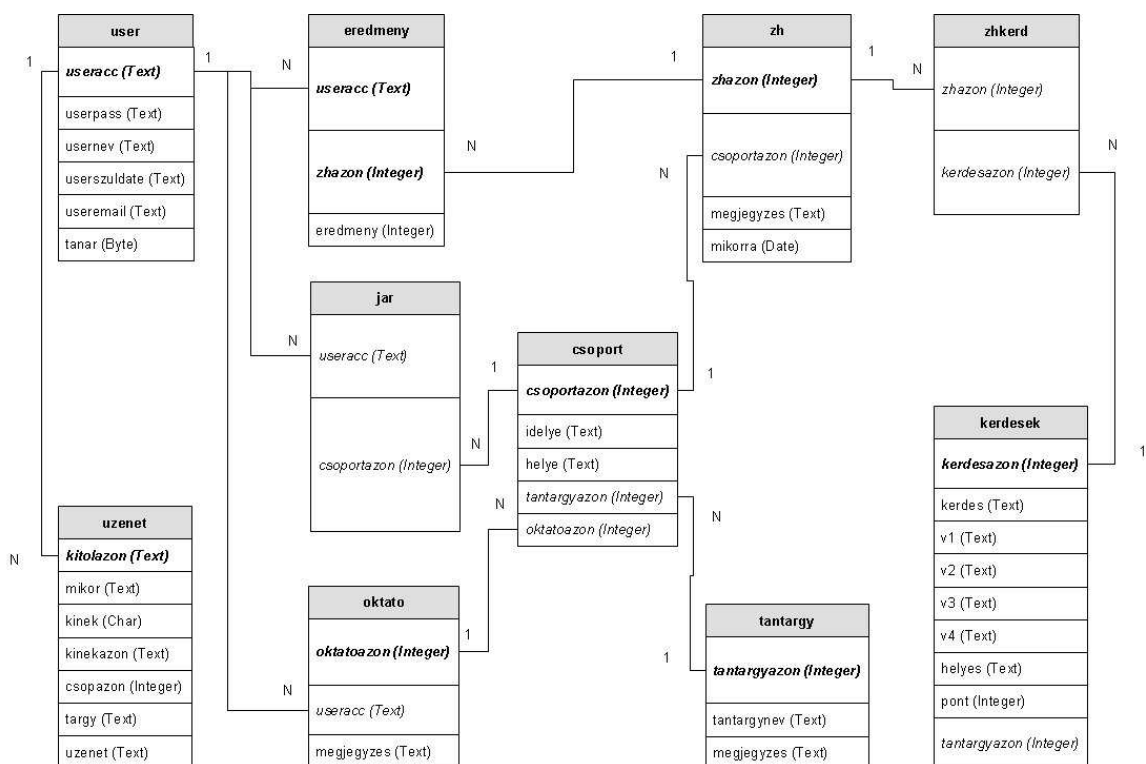
TANTARGY (*tantargyazon*, *tantargynev*, *megjegyzes*);

ZHKERD (*zhazon*, *kerdesazon*);

KERDESEK (*kerdesazon*, *kerdes*, *v1*, *v2*, *v3*, *v4*, *helyes*, *pont*, *tantargyazon*);

UZENET (*kitolazon*, *mikor*, *kinek*, *kinekazon*, *csoportazon*, *targy*, *üzenet*);

2.1.3 Az adatbázis táblái



2.1.3.1. ábra Az adatbázis táblái

2.1.4 Táblák részletezése

User tábla

Sorszám	Mező	Típus	Hossz	Leírás
<u>1</u>	<u>useracc</u>	varchar	50	A felhasználó egyedi azonosítója (primary key)
2	userpass	varchar	50	a felhasználó jelszava
3	usernev	varchar	50	a felhasználó teljes neve
4	userszuldate	varchar	19	a felhasználó születési dátuma
5	useremail	varchar	50	a felhasználó e-mail címe
6	tanar	tinyint	1	ha a felhasználó tanár, értéke 1 Ha admin értéke 3, ellenben 0.

Az USER tábla tartalmazza a felhasználók adatait.

Eredmény tábla

Sorszám	Mező	Típus	Hossz	Leírás
1	<u>useracc</u>	varchar	50	felhasználó azonosító
<u>2</u>	<u>zhazon</u>	int	11	a dolgozat egyedi azonosítója
3	eredmeny	int	11	a dolgozaton elért eredmény

EREDMENY tábla tartalmazza a diákok dolgozatok alkalmával elért eredményeit.

Kérdések tábla

Sorszám	Mező	Típus	Hossz	Leírás
<u>1</u>	<u>kerdesazon</u>	int	11	A kérdés egyedi azonosítója
2	kerdes	varchar	200	a kérdés
3	v1	varchar	50	Első lehetséges válasz
4	v2	varchar	50	Második lehetséges válasz
5	v3	varchar	50	Harmadik lehetséges válasz
6	v4	varchar	50	Negyedik lehetséges válasz
7	helyes	varchar	50	A helyes válasz sorszáma
8	pont	int	11	a kérdésre adható pontszám
9	<u>tantargyazon</u>	int	11	a kérdés tantárgyhoz kapcsolódó azonosítója

A KERDESEK tábla tartalmazza a kérdéseket.

Zhkerd tábla

Sorszám	Mező	Típus	Hossz	Leírás
<u>1</u>	<u>zhazon</u>	int	11	dolgozat azonosító
2	<u>kerdesazon</u>	int	11	kérdésazonosító

ZHKERD tábla tartalmazza egy adott dolgozat kérdéseit.

Jar tábla

Sorszám	Mező	Típus	Hossz	Leírás
<u>1</u>	<u>useracc</u>	varchar	50	felhasználó azonosító
2	<u>csoportazon</u>	int	11	Csoportazonosító

JAR tábla tartalmazza a csoport tagjait.

Csoport tábla

Sorszám	Mező	Típus	Hossz	Leírás
<u>1</u>	<u>csoportazon</u>	int	11	egyedi Csoportazonosító
2	ideje	varchar	50	A csoport melyik időpontban tartja az adott órát
3	helye	varchar	50	a csoport hol tartja az adott órát
4	tantargyazon	int	11	a csoport órája
5	oktatoazon	int	11	a csoportot oktató tanárazonosítója

CSOPORT tábla tartalmazza a csoportok adatait.

Oktato tábla

Sorszám	Mező	Típus	Hossz	Leírás
<u>1</u>	<u>oktatoazon</u>	Int	11	az oktató egyedi azonosítója
2	<u>useracc</u>	Varchar	50	mely felhasználó tanár.
3	megjegyzes	Varchar	200	az oktatóval kapcsolatos megjegyzés

OKTATO tábla tartalmazza az oktató adatait.

Tantargy tábla

Sorszám	Mező	Típus	Hossz	Leírás
<u>1</u>	<u>tantargyazon</u>	int	11	a tantárgy egyedi azonosítója
2	tantargynev	varchar	50	a tantárgy megnevezése
3	megjegyzes	varchar	200	tantárggyal kapcsolatos megjegyzés

TANTARGY tábla tartalmazza a tantárgy adatait.

Zh tábla

Sorszám	Mező	Típus	Hossz	Leírás
<u>1</u>	<u>zhazon</u>	int	11	a dolgozat egyedi azonosítója
2	<u>csoportazon</u>	int	11	megy csoportnak készült dolgozat
3	megjegyzes	varchar	100	dolgozattal kapcsolatos megjegyzés
4	mikorra	date		a dolgozatírás időpontja

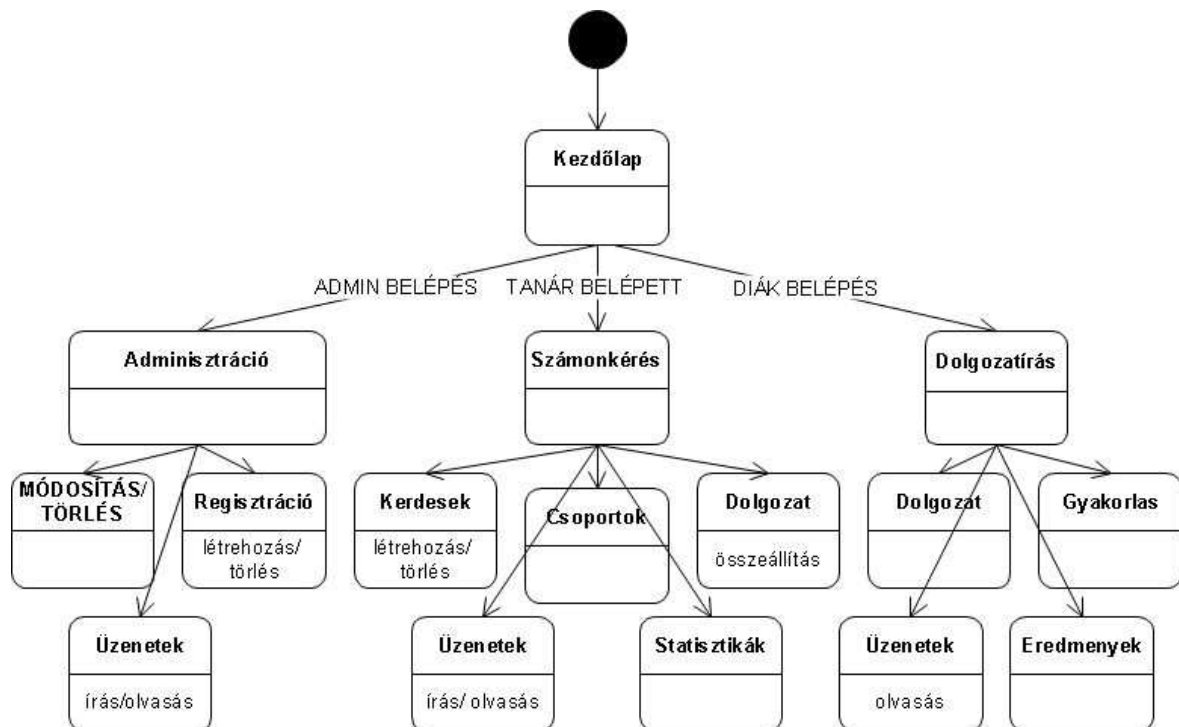
A ZH tábla tartalmazza a dolgozat adatait.

Üzenet tábla

Sorszám	Mező	Típus	Hossz	Leírás
<u>1</u>	<u>kitolazon</u>	varchar	50	a küldő azonosítója
2	mikor	datetime		dátum
3	kinek	char	1	megjelöli, hogy egy adott felhasználó, csoport vagy mindenki kapja meg. Értékei lehetnek : P(egy felhasználó), A (mindenki), T (csoport)
4	kinekazon	varchar	50	amennyiben a címzett egy felhasználó akkor itt tárolódik az azonosítója
5	csopazon	int	11	amennyiben a címzett egy csoport, itt tárolódik a csoportazonosító
6	targy	varchar	100	az üzenet tárgya
7	uzenet	varchar	255	Az üzenet

UZENET tábla tartalmazza az üzenteket.

2.2 Navigáció



2.2.1. ábra Navigáció

Ábrán látható a webportál *navigációs terve* [7]. Ez segít az oldal átláthatóságában. Ennek elkészítésében, és a szabvány elsajátításában igen nagy szerepet játszott egy előadás a PHP konferencián, aminek egyik témája ez volt [6]. Abban segít, hogy miként építsük fel az oldalt.

2.3 Osztályok, függvények és feladatai

Ezt a programot nem elsősorban objektum orientáltra készítettem el, de néhány osztály létrehozását tervezem, hogy segítse munkámat, egyszerűbbé tegye a felhasználó, illetve az adatbázis-kapcsolatok kezelését. Valójában nem határoztam meg, hogy melyik függvény privát és melyik publikus, viszont, hogy látható legyen az, hogy melyek azok a tagfüggvények, amiket csak maga az osztály használ, ezért azokat így jelöltem meg.

STATISZTIKA	SQL	LOGIN
+tabla = array(array()) -ideje -helye -userekszama -zhszama	+eredemeny -kapcsolat +sql(); +sql_close(); +sql_query(); +sql_fetch_array(); +sql_num_rows();	+teljesNev +loginNev +tanar +oktatoazon -voltProbalkozas -helyesBelepes +login(); +beleptet(); +kileptet(); +loginurlap();
+statisztika(\$csoplist, \$tantlist) +atlag() +osszeg() -eredmenycella(\$user, \$zh) -tablafej(\$csopazon) -tablauserek(\$csopazon)		

2.3.1 ábra Az osztály diagramok.

2.3.1 A STATISZTIKA osztály

Feladata

Egy csoport statisztikájának táblázatba rendezése, mely számol átlagot és összesített pontokat. Az átlagot az egyes zh-ra számol, összesítést pedig egy-egy felhasználó összes megírt zh-inak pontszámaira.

Attributomok

Tabla	egy kétdimenziós tömb, mely tartalmazza a csoport tagjainak dolgozat eredményeit.
Userekszama	egy csoporton belül a diákok száma
Zhszama	egy csoporthoz rendelt összes dolgozat darabszáma

Metódusok:

Statisztika()	ez a függvény építi fel a táblázatot
Atlag()	számolja a csoportátlagot
Osszeg()	számolja a diák által megírt zh eredmények összegét.
Eredmenycella()	ez számolja ki az egyes cellák pontos összegét, azaz hogy melyik diák, melyik zh-ra hány pontot kapott.
Tablafej()	a tábla felső sora, mely a zh-azonosítókat tartalmazza
Tablauserek()	az a táblázat bal szélső oszlopa, mely a csoport tagjainak azonosítóját tartalmazza.

2.3.2 Az SQL osztály**Feladata:**

Az adatbázis kezelés megkönnyítése.

Kapcsolat létrehozása.

Kapcsolat lezárása.

Kérés futtatása.

Kérés eredményének visszaadása.

Attributomok:

Private kapcsolat:	A kapcsolat tulajdonságai
Public eredmény:	A kérés eredménye

Metódusok:

sql();	Felépíti a kapcsolatot.
sql_close();	Zárja a kapcsolatot;
sql_query();	Kérést intéz az adatbázis szerver felé.
sql_fetch_array()	Az eredményt úgy adja vissza, hogy abból csak egy sort jelenít meg.
sql_num_rows();	Az eredmény sorainak számát adja vissza.

2.3.3 A LOGIN osztály

Feladata:

A belépett felhasználó azonosítása, nyomon követése.

Beléptetés

Kiléptetés

Attribútumok:

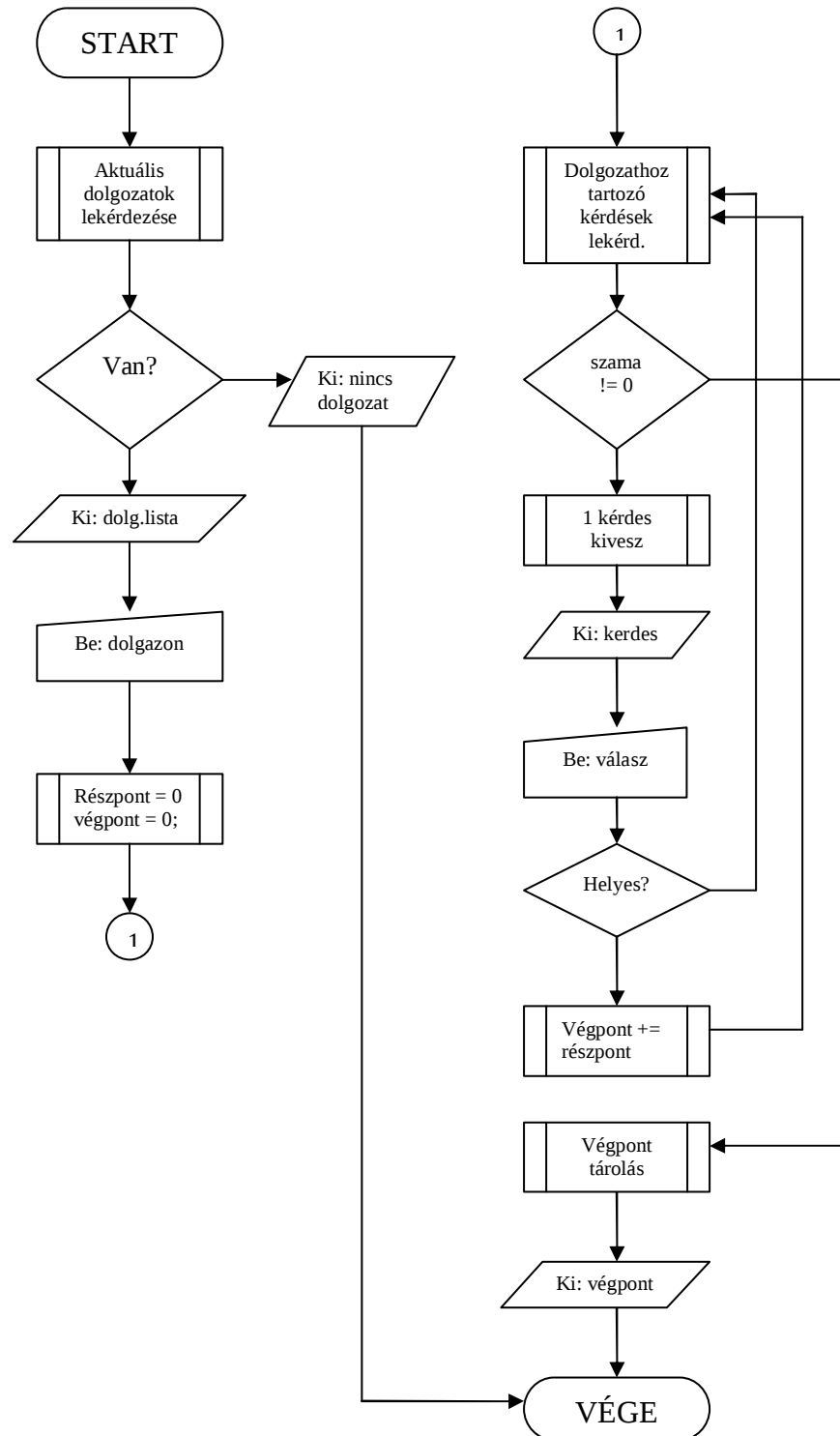
TeljesNev	A felhasználó teljes neve.
LoginNev	A felhasználó belépési azonosítója(egyedi)
Tanar	Értéke 1, ha a belépett felhasználó oktató.
Oktatoazon	Az oktatónak az azonosítója(egyedi).
voltProbalkozas	az osztály segédváltozója
helyesBelepes	az osztály segédváltozója

Metódusok:

Login();	Konstruktor, mely azonosítja a felhasználót, tárolja a szükséges adatait.
Beleptet();	Végzi az ellenőrzéseket, létezik e a felhasználó, megfelelő e a jelszó.
Kiléptet();	Kilépteti a felhasználót, törli a session változókat.
loginUrlap();	egy egyszerű beléptetési felület.

2.4 Dolgozatírás folyamata

Folyamatábra [7] segítségével megpróbálom felvázolni a dolgozatírás menetét. Sok-sok folyamatábra kellene az egész rendszer leírásához, de itt csak a legfontosabbat mutatom be (2.4.1. ábra).



2.4.1. ábra A dolgozatírás folyamatábrája

3. Programozás, kódolás

3.1 Fejlesztői környezet

A kódoláshoz a PHP Expert Editort (trial), 30 napig ingyenesen használható PHP szerkesztő programot használtam, amely segítségével a PHP kódban található fenntartott szavak, függvények jól láthatóan elkülönülnek, áttekinthetővé válnak. A program letölthető a <http://www.phpexperteditor.com> –ról.

Fejlesztés közben az igen közkedvelt és ingyenes AppServ v2.4.3. programot használtam helyi szervernek, amely letölthető az alábbi címről:

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

Appserv egy programcsomag, amely tartalmazza a következő elemeket:

AppServ 2.4.3:

- PHP 4.3.10
- Apache 1.3.33
- MySQL 4.1.8
- Zend Optimizer 2.5.7
- phpMyAdmin 2.6.1-rc2

Ezen program előnye, hogy minden összetevőt tartalmaz, amelyre a fejlesztés során szükségem volt, ingyenes és bárki számára hozzáférhető. Valamint telepítéskor nem kell a konfigurálással foglalkozni, ezért ha a Weblapot helyi szerveren működtetik, akkor könnyen üzembe helyezhető.

3.2 A felhasználó nyomkövetése

Legelső és a legnagyobb probléma programozás során, hogy a belépett felhasználót mindvégig figyelemmel tudjuk kísérni egészen a kilépés pillanatáig. Ezt a feladatot a PHP-ben jól ismert és könnyen használható `$PHP_SESSION_VARS[]` segítségével, azaz *session* kezeléssel oldottam meg. Egy nagyon jó ötlet alapján dolgoztam ki egy saját osztályt, ami ezt a feladatot maradéktalanul elvégzi. Ennek az ötletnek a forrása a [8]-nál található. A lényege, hogy mindaddig, amíg a felhasználó az oldalon tartózkodik, tehát be van jelentkezve, addig a saját változói (azonosító, név, tanár-e stb.) egy úgynevezett

session változóban vannak eltárolva szerver oldalon. Ez törlődik, amikor a felhasználó kilép.

Ennek az osztálynak a felépítése egyszerű. Két különböző belépési módot különböztet meg. Az egyik a tényleges belépés, amikor nincs belépve a felhasználó, tehát nincs *session* változója, ekkor egy belépési űrlapot kell kirakni, az űrlapból jövő adatokat – amennyiben azok helyesek – eltárolja *session* változóba. Viszont ugyanez a script fut le akkor is, ha már a felhasználó belépett, de egy olyan oldalt nyit meg amihez jogosultság kell, akkor viszont már nem az űrlapból veszi az adatokat, hanem a már belépéskor eltárolt változókból. Így mindig biztosak lehetünk abban, hogy mindenki csak a számára kijelölt oldalt nézheti meg.

3.3 Statisztikai kimutatás

3.3.1. Táblázat		Statisztikai kimutatás	
dolgozat	1	2	Össz.
User1	11	33	44
User2	34	22	56
User3	22	12	34
átlag	22,33	22,33	

Egy másik érdekes probléma, egy adott csoport eredményeinek kimutatása. Egy olyan táblázatot kell készíteni, mely az oktató számára minden lényeges információt tartalmaz. Programozás technikai eszközként egy újabb osztályt kell készíteni ennek a probléma megoldására, mely első hallásra könnyűnek is tűnik. A feladat, hogy egy olyan mátrixot hozzunk létre, melynek az oszlopai tartalmazza az egyes dolgozatok eredményeit, sorai pedig az egy diákok eredményeit. Azaz egy cella a táblázatban megmondja, hogy melyik diák melyik dolgozatra milyen eredményt produkált. A statisztika osztály konstruktora megkapja annak az osztálynak az azonosítóját, melynek szeretnének az eredményeit megtekinteni. Ekkor kikeresi, hogy abba a csoportba kik járnak és hányan, és a hozzájuk tartozó dolgozat azonosítókat. Ezekkel feltölti a mátrixot, tehát ekkor még csak az első sora és első oszlopa van feltöltve. Ezután két egymásba ágyazott ciklus és egy tagfüggvény segítségével a felhasználó azonosító és a hozzá tartozó dolgozat azonosító kiolvasásával lekérdezi az adatbázisból a hozzájuk tartozó eredményt. Így kapjuk meg azt a végleges mátrixot, melyet végül kiírunk a képernyőre, ami a **3.3.1-es táblázat** szemléltet.

4. Tesztelés

A weblap fejlesztés során fontos lépés a tesztelés, amelyet általában a program befejezésekor végeznek. Segítségképp kértem fel néhány ismerőszámot, arra, hogy segítsenek tesztelni az elkészült weblapot. Megkértem őket arra, hogy minden egyes esetet többféle módon hajtsanak végre és jelezzék számomra, ha valamit nem találnak rendbe, esetleg nem tartanak érthetőnek.

4.1 Üzembe helyezési teszt

Üzembe helyezéskor azt vizsgáltam, hogy amikor a böngészőben meghívtam az oldalt, akkor rendben betöltődött-e annak a beléptető és üdvözlő oldala. Amennyiben ez sikeres, akkor az üzembe helyezési tesztnek a következő lépése az adatbázis kapcsolat tesztelése. Ezt nagyon egyszerűen hajthatjuk végre, mégpedig be kell lépünk a teszt felhasználó azonosítójával (4.1.1. táblázat).

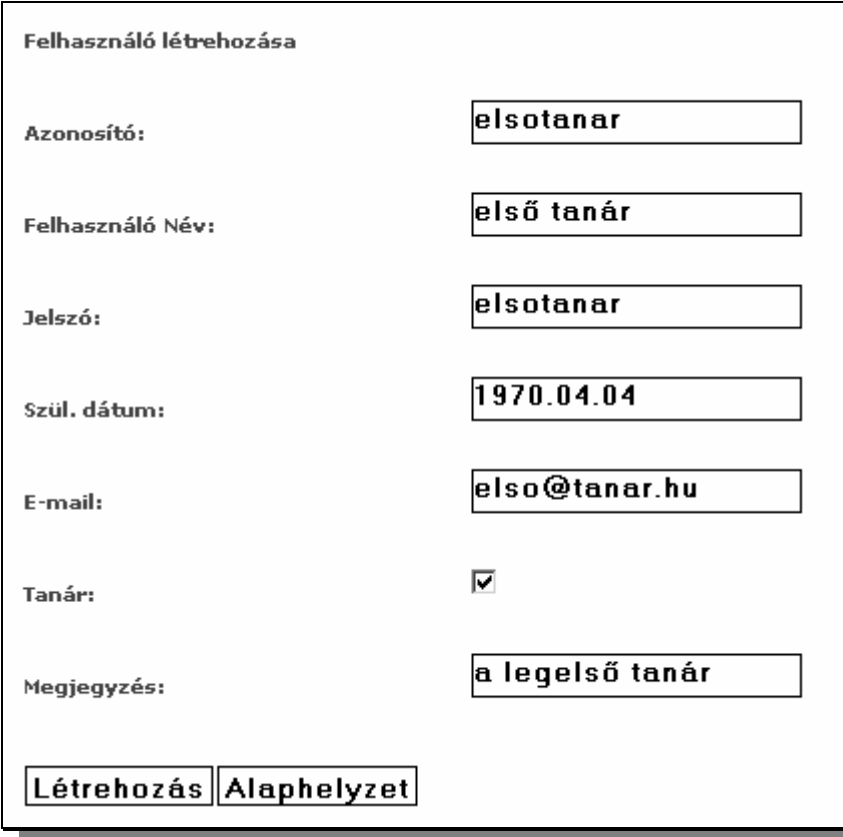
4.1.1. Táblázat

Tesztelési jogosultságok

<i>Tesztelés</i>	<i>Oktató</i>	<i>Diák</i>
<i>Azonosító</i>	<i>tanarteszt</i>	<i>diakteszt</i>
<i>Jelszó</i>	<i>tanarteszt</i>	<i>diakteszt</i>

4.2 Általános funkcionális teszt

Ebben az esetben ellenőrizzük, hogy a program normál keretek között helyesen működik e. Ez azt jelenti, hogy minden egyes beviteli mező, választó gomb stb. pontosan azt követi, ami a feladata. Ellenőrizzük az adatbevitelt az adatmódosítások helyes végrehajtását. Ilyenkor a következőket próbáltam ki.

Új felhasználó felvétele (admin)

Felhasználó létrehozása

Azonosító:

Felhasználó Név:

Jelszó:

Szül. dátum:

E-mail:

Tanár: ☒

Megjegyzés:

4.2.1. ábra Új felhasználó felvétele

Ezekkel az adatokkal sikeresen létrehoztam egy felhasználót.

Majd ezek után egy diákot is létre kell hozni, mégpedig a következőkkel:

- **Azonosító:** *elsodiak*
- **Név:** *első diák*
- **Jelszó:** *elsodiak*
- **Szül. dátum:** *1988.02.02*
- **E-mail:** *elso@diak.hu*
- **Tanár:** *NEM KIPIPÁLVA!*
- **Megjegyzés:** *NINCS!*

Az első diák is létrejött.

Új tantárgy felvétele (admin)

Új tantárgyat a *Tantárgyak* menüpont alatt, hozok létre a következő adatokkal:

- Név: elso tantargy
- Megjegyzés: ez az elso tantargy

Ezekkel sikeresen létrehoztam egy tantárgyat.

Üzenetírás az új felhasználónak (admin)

Üzenetet az *Üzenetírás* menüpont alatt küldök az új tanárnak, mégpedig a következő módon. Először is a *Személynek* legördülő listából kiválasztom az **első tanárt**-t, az üzenet *tárgyának* megadom az *első üzenet*-et, majd *üzenetnek* beírom a következőt: „Ez itt az első üzenet!”. Ezek után már csak a küldés gomb van hátra, és sikeresen elküldtem az első teszt üzenetet.

Kérdések feltöltése (oktató)

Ehhez váltani kell az admin jogosultságról egy oktatóéra. Tehát például, lépünk be az **elsotanar** jogosultsággal. Egy kérdést feltölteni akkor lehet, ha már az *admin* létrehozta azt a tantárgyat, amihez a kérdést hozzá szeretnénk rendelni. Ezt az ímént már sikeresen megtettem **elso tantargy** néven.

Első lépében a legördülő menüből válasszuk ki azt a tantárgyat, melyhez a kérdést szeretnénk, hogy tartozzon. Ebben az esetben csak egy lehetséges tantárgy van: **elso tantargy**. Majd adjuk meg a következő értékeket:

- **Kérdés:** *Mi legyen az első kérdés?*
- **Válasz 1:** *például ez*
- **Válasz 2:** *például az*
- **Válasz 3:** *egyik sem*
- **Válasz 4:** *mindkettő*
- **A helyes válasz:** *V1*
- **Pontérték:** *2*

Mindez sikeresnek bizonyult, felvettünk egy kérdést az adatbázisba.

Csoport létrehozása (oktató)

Csoportot a következő adatokkal hozzunk létre:

- **Ideje:** *kedd első óra*
- **Helye:** *főép. 211*
- **Tantárgy:** *eslotantargy*

Csoportba rendezés (oktató)

Csoportba rendezésnél két adatot kell megadnunk:

- **Csoport:** *1 – kedd első óra*
- **Felhasználó:** *elsodiak*

Kérdéssor létrehozása (oktató)

Ez a művelet két lépésből áll. Először a csoportot (tantárgy), majd az ahhoz tartozó kérdéseket kell megadni. Egy lehetséges változat:

- **Csoport:** *1. csoport*
- **Dátum:** *2005.05.16*
- **Megjegyzés:** *első dolgozat*

Tovább gomb után a kérdések listájából kiválasszuk a szükséges kérdéseket. Majd Ok-gomb.

Dolgozatírás (diák)

Lépünk be **elsodiak** azonosítóval. Ebben az esetben már az oktató létrehozott dolgozatot. Amennyiben a mai napra esedékes, úgy a Dolgozat menüpont alatt találunk egy listát az esedékes dolgozatokról. Mivel az előbb létrehoztunk egyet, ezért most ott kell, hogy legyen az alábbi dolgozat:

- 1 – első dolgozat

Kiválasztva meg kell, hogy kapjuk a hozzá tartozó kérdéseket. Ebben az esetben egy darab kérdést fogok kapni, mert még csak egy kérdés szerepel az adatbázisban. Erre válaszolok. Majd az Ok-gomb lenyomása után kiírja a szerzett pontszámot, azaz a 2 pontot.

4.3 Szélsőérték funkcionális teszt

Szélsőérték tesztelés azt jelenti, hogy olyan értékeket adunk meg a weblap tesztelése során, melyek nem megengedettek. Ekkor hiba üzeneteket kell, hogy kapjunk. Ez azért fontos, hogy ne kerülhessen hibás, nem megfelelő adat, feldolgozás alá.

Két példát részleteznék. Az egyik az *új felhasználó felvétele*, a másik a *kérdéssor készítése*.

Új felhasználó felvétele

A következő adatokat próbálom megadni:

- **Azonosító:** nem
- **Név:** jó
- **Jelszó:** els
- **Szül. dátum:** 1888.02.32
- **E-mail:** elsoiak.hu
- **Tanár:** NEM KIPIPÁLVA!
- **Megjegyzés:** NINCS!

Erre a következő hibaüzenetet kapom:

- Rövid azonosító
- Rövid jelszó
- Nem megfelelő dátum
- Nem megfelelő e-mail

Kérdéssor összeállítása

Szándékosan nem jelölöm meg a csoportot, és hibás dátumot írok be. Hiba üzenet érkezik, mert kötelező csoportot kiválasztani és pontos, helyes dátumot kell megadni. Majd egyetlen kérdést sem jelölök meg, ezért egy újabb hibaüzenetet kapok.

5. Üzembe helyezés, telepítés

5.1 Hardver, szoftver környezet

Szükséges megemlíteni, hogy a weblapnak a helyes működéséhez szükséges feltételeit két nagy részre kell osztani. Az egyik a kliens oldal, a másik és a fontosabb pedig a szerver oldal. Míg a kliens oldalon minimális hardver igények szerepelnek és csupán elég egy internet böngésző program, addig szerver oldalon komoly feltételeknek kell megfelelni amennyiben ezt a programot nagy nyilvánosság előtt, akarjuk helyesen használni.

Példa egy kliens számítógépre:

- Pentium II 400 MHz
- SVGA, ajánlott minimum 800*600-as felbontás
- Internet Explorer 4.0 vagy újabb

5.2 Appserv telepítése

Amennyiben a programot egy saját hálózaton belül szeretnénk használni, és előtte még nem volt számítógép erre a feladatra beállítva, úgy az **AppServ** nevű program segítségével bárki, bármilyen számítógépből, nagyon egyszerűen készíthet erre a célra kiszolgálót.

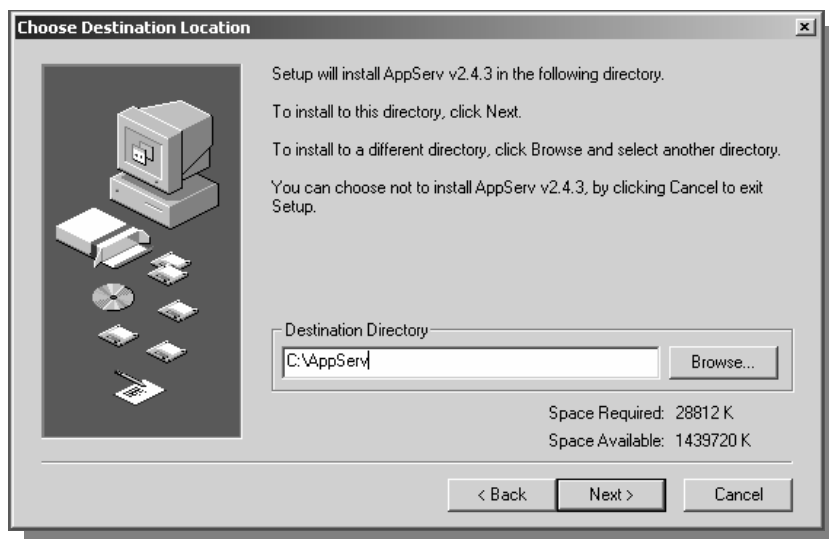
Az **AppServ** beszerzése:

Microsoft Windows platformra:

<http://mesh.dl.sourceforge.net/sourceforge/appserv/appserv-win32-2.5.3.exe>

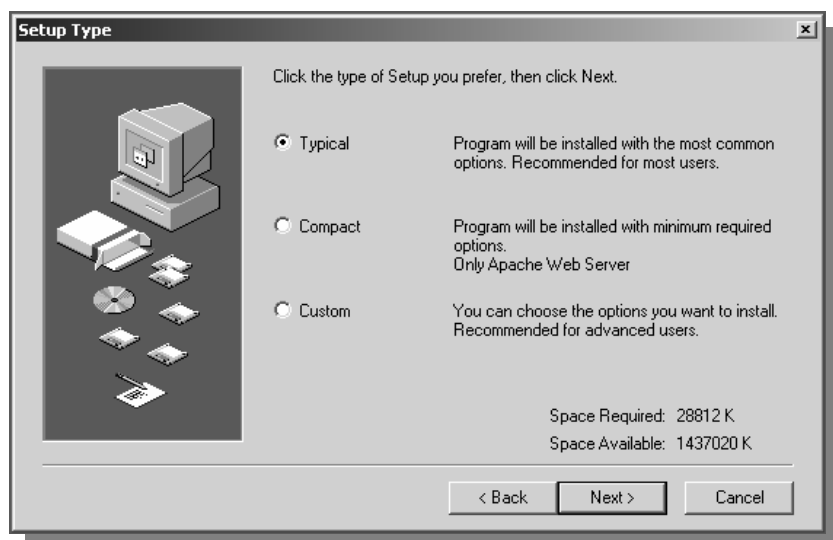
Az **AppServ** telepítésének lépései:

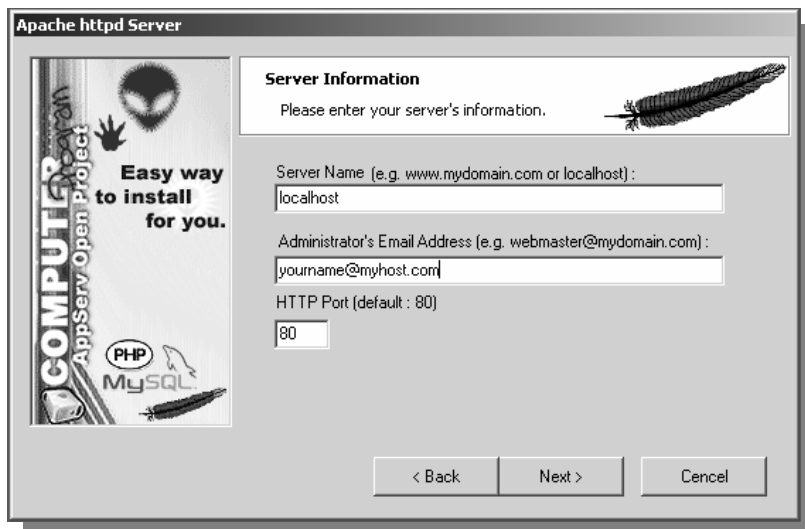
Az **appserv-win32-2.4.3.exe** indítása után közvetlenül egy üdvözlő párbeszédpanelt találunk. Miután elolvastuk, nyomjunk egy Next-et!



A következő párbeszédpanelen meg kell adnunk azt a könyvtárat, ahova az AppServ kerülni fog. Ebben a könyvtárban lesz majd a program, azaz a weblap is.

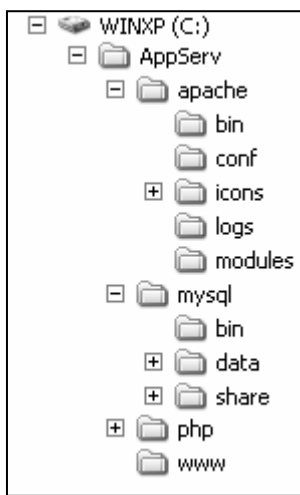
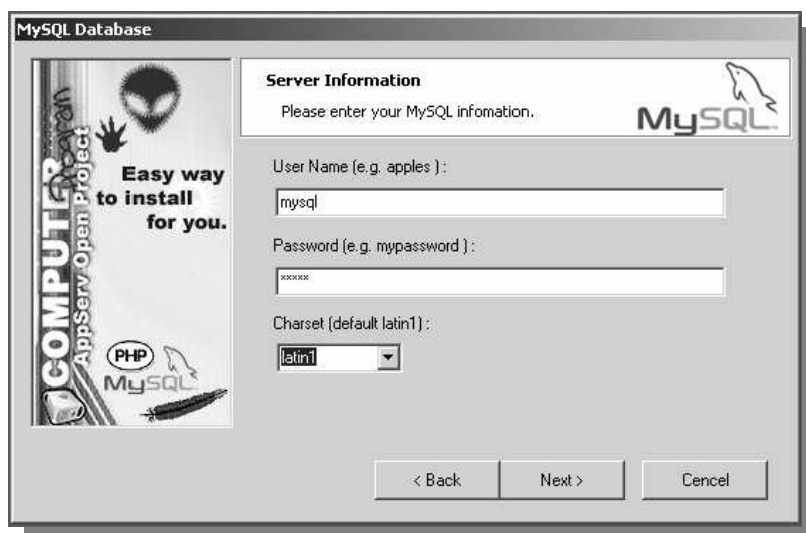
Harmadik lépésben választhatunk a telepítési módok közül. A *Custom*-ban magunk választhatjuk meg az AppServ csomagban lévő alprogramokat. Felhívom a figyelmet arra, hogy elegendő a *Typical* mód választása!





Ezen a párbeszédpanelen szerver beállításokat tehetünk meg. Az itt megadott **localhost** tökéletesen megfelel a beállításoknak, de az e-mail címet állítsuk be az adminisztrátor elérhetőségére.

A következő és egyben utolsó ablakban az adatbázis eléréséhez szükséges adatokat adjuk meg. Ezt is használhatja a weblap az adatbázis eléréséhez is. Állítsunk be például névnek: **mysql**, jelszónak: **mysql123**. Ezzel a jogosultsággal tudunk majd az adatbázishoz hozzáférni (lásd: **Weblap telepítése**).



5.2.1. ábra

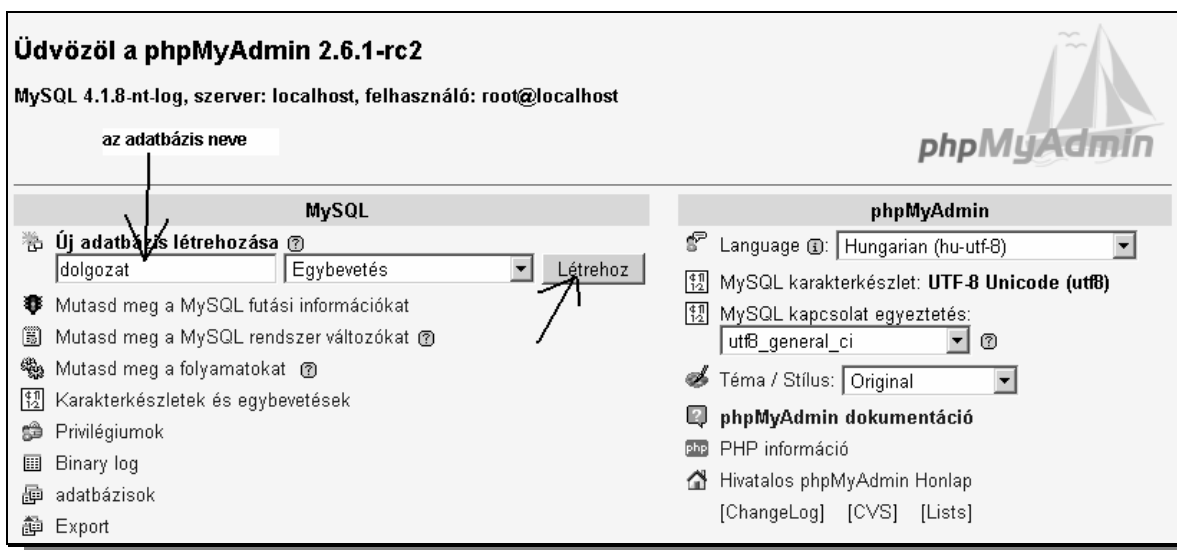
Miután itt is a következő gombra kattintottunk a telepítő fájlokat fog másolni a megadott helyre. A könyvtár struktúrája az **5.2.1.ábrán** látható. A **www** könyvtár tartalmazza a weblapokat, amikor a böngészőbe beírjuk a **localhost**-ot akkor ennek a tartalma jelenik meg. A **mysql/data** könyvtár tartalmazza magát az adatbázist.

Amennyiben sikeres volt a telepítés a tálca jobb szélén meg kell, hogy jelenjen egy lámpa, annak pedig zölden világítania.

5.3 A weblap telepítése

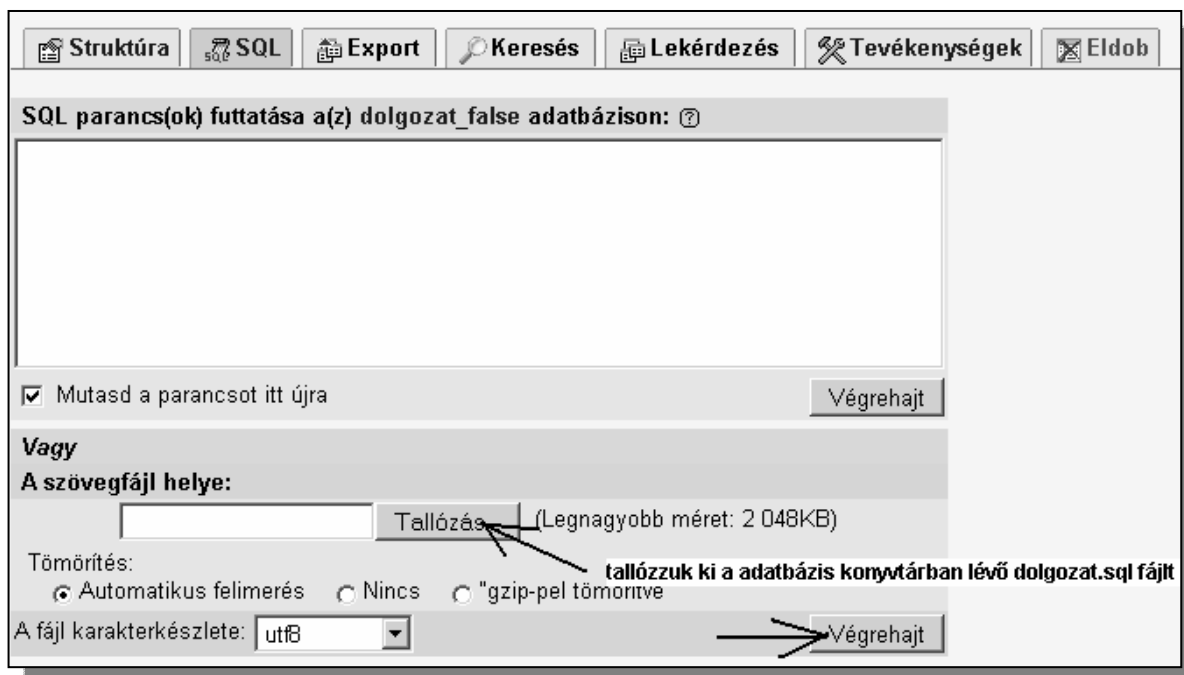
A sikeres szerver felépítése után következik a weblap felmásolása, és beállítása. Elsőként ellenőrizzük, hogy a **CD mellékletben** említett fájlok megtalálhatóak e telepítő cd-n.

Az adatbázis könyvtárban található *dolgozat.sql* nevű sql kéréseket tartalmazó dump fájl, maga az adatbázis. Ezt a fájlt kell lefuttatnunk a phpMyAdmin segítségével, mely az AppServ része. Első lépésben hozzuk létre a *dolgozat* adatbázist (5.3.1. ábra).



5.3.1. ábra Adatbázis létrehozása

Majd válasszuk a megjelenő felső sorban az SQL fület.



5.3.2. ábra SQL fájl lefuttatása

Nyomjunk egy *Tallózás* gombot. Keressük meg a **dolgozat.sql** fájlt a mellékelt cd-n lévő adatbázis könyvtárban. Majd a *Végrehajtás* gombbal érvényesítsük azt. Majd a **cd, www** könyvtárában lévő fájlokat tegyük az **appserv/www** könyvtárba.

Ezzel még nincs vége a telepítésnek, ugyanis személyre kell szabni a weboldal konfigurációs fájlját. Ez a fájl a **www** könyvtárban található meg, és a neve: **config.php**

config.php tartalma:

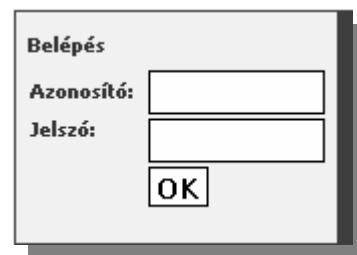
```
<?php
define("DBHOST", "localhost"); //adatbázis szerver neve
define("DBUSER", "mysql"); //felhasználó az adatbázis eléréséhez
define("DBPASS", "mysql123"); //jelszó az adatbázis eléréséhez
define("DBNAME", "dolgozat"); //a dolgozat adatbázis neve, állandó!
?>
```

6. Felhasználói dokumentáció

A tudásfelmérő weblap használatához elengedhetetlen egy adminisztrátor, aki létrehozza a felhasználókat. Kipróbáláshoz teszt felhasználók vannak az adatbázisban. Ezek a felhasználók a **tanarteszt** (jelszó: tanarteszt) és a **diakteszt** (jelszó: diakteszt). Az adminisztrátor alapértelmezett azonosítója: **admin** (jelszó: admin). A tudásfelmérő weboldalt bármilyen internetes böngészőn keresztül tudjuk használni. Címsorba írjuk be a már telepített tudásfelmérő elérhetőségét. Amennyiben erről nincs tudomásunk, akkor azt kérdezzük meg a rendszergazdától.

6.1 A weblap használata

Kezdjük megismerkedni a weblap kezelhetőségével. Böngészőbe írjuk be: <http://localhost> vagy ha nem a helyi gépen van telepítve, akkor kérdezzük meg a rendszergazdát a tudásfelmérő elérhetőségével kapcsolatban. Ekkor meg kell jelenjen egy beléptető oldal. Egy beléptető kis dialógos ablakkal, ami a **6.1.1. ábrán** látható. Itt kell megadnunk a belépéshez szükséges **azonosítót** és a **jelszót**. Majd az Ok-gombra kattintva beléptet a rendszer, amennyiben helyesek voltak az adatok, ha nem akkor ugyanez a beléptető fog újra megjelenni. Miután beléptünk a főoldalon a nekünk szóló üzeneteket olvashatjuk.



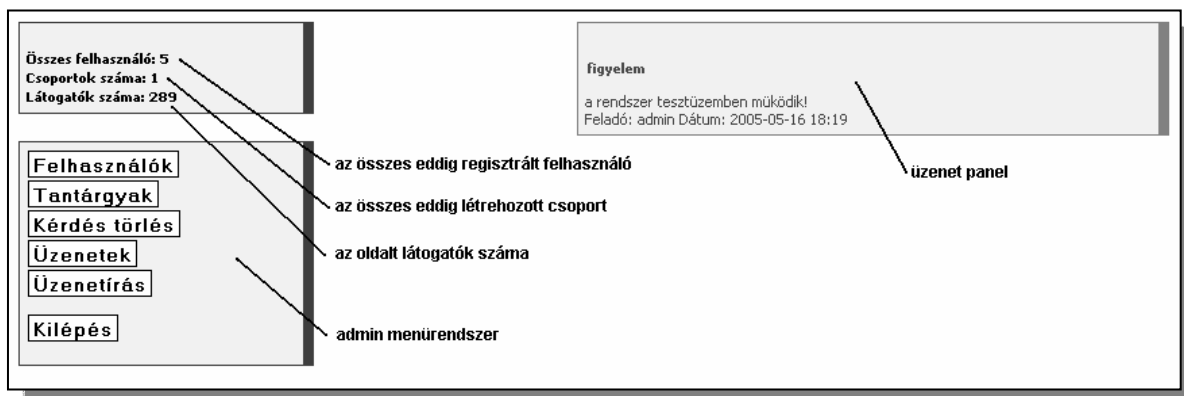
6.1.1 ábra Beléptető



Három belépési jogosultságot különböztetünk meg, melyeknek más-más lehetőségeik vannak az oldalon. Ezek lehetnek:

- Adminisztrátor
- Oktató
- Diák

6.1.1 Az adminisztrátor lehetőségei



6.1.1.1. ábra Az adminisztrátor kezelőfelülete

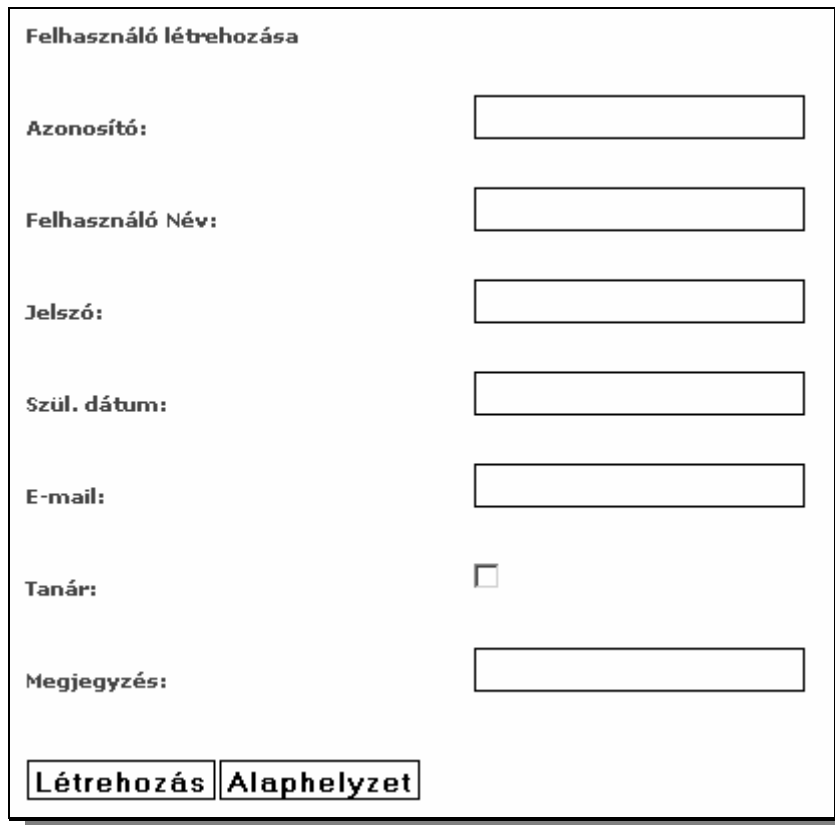
Menüpontok

- | | |
|------------------------|------------------------------------|
| • <i>Felhasználók</i> | Felhasználó regisztráció és törlés |
| • <i>Tantárgyak</i> | Tantárgy létrehozása és módosítása |
| • <i>Kérdés törlés</i> | Kérdés törlése az adatbázisból |
| • <i>Üzenetek</i> | Üzenet olvasása(alapértelmezett) |
| • <i>Üzenetírás</i> | Üzenetírás |

A szakdolgozat és a weboldal kiterjedése miatt csak a felhasználó regisztrációt és az új tantárgy létrehozását mutatom be. Amennyiben szükség volna a további funkciók leírásán, természetesen kérésre elkészítem.

Felhasználók

Kattintsunk a Felhasználók menüpontra és ott a 6.1.1.2. ábrán is látható dialógus ablakot láthatjuk.



Felhasználó létrehozása

Azonosító:

Felhasználó Név:

Jelszó:

Szül. dátum:

E-mail:

Tanár: ☐

Megjegyzés:

Létrehozás **Alaphelyzet**

6.1.1.2. ábra Új felhasználó létrehozása

Fontos szabályoknak kell megfelelni az új felhasználó párbeszédpanel kitöltése során. Azonosítónak és a jelszónak kötelezően minimum 4 karakter hosszúnak kell lennie. A felhasználónévnek nincs ilyen megkötése, de kitöltése kötelező! Hasonló módon a születési dátumának és az e-mail címnek kitöltése kötelező és a formátumoknak meg kell felelnie!

Követelmények a születési dátum formátumával kapcsolatban:

- Formája: **XXXX.XX.XX** (pl: 1982.06.06)
- Az évszám 1900 és 2005 között lehet
- A hónap 01 és 12 között lehet
- A nap 01 és 31 között vehet fel értéket

Ezek be nem tartására hibaüzenetet kapunk!

A párbeszédpanel további mezői a **'tanar'** checkbox, ami, azt jelenti, hogy ha ez be van jelölve, akkor a felhasználót oktatónak regisztráljuk. Ehhez írhatunk megjegyzést is. Ha minden rendben, akkor kattintsunk a Küldés gombra. Ha semmivel nem volt probléma, akkor újra ezt a párbeszédpanelt fogjuk látni hiba és egyéb üzenet nélkül. Az, hogy rögtön újra ezt a dialógus ablakot látjuk, azaz oka, ha sok diákot kell egyszerre regisztrálni, akkor így hatékonyabb és gyorsabb munkát lehet végezni.

Tantárgy létrehozása

Ez azért fontos, mert az oktatók csoportot, úgy tudnak létrehozni, hogy annak a csoportnak valamilyen tantárgyi besorolást adnak. Majd ezek után a dolgozat összeállításánál is fontos szerepet fog ez játszani. Kérdéseket is csak a már létrehozott tantárgyakhoz lehet rendelni. Kattintsunk a *Tantárgyak* menüpontra.

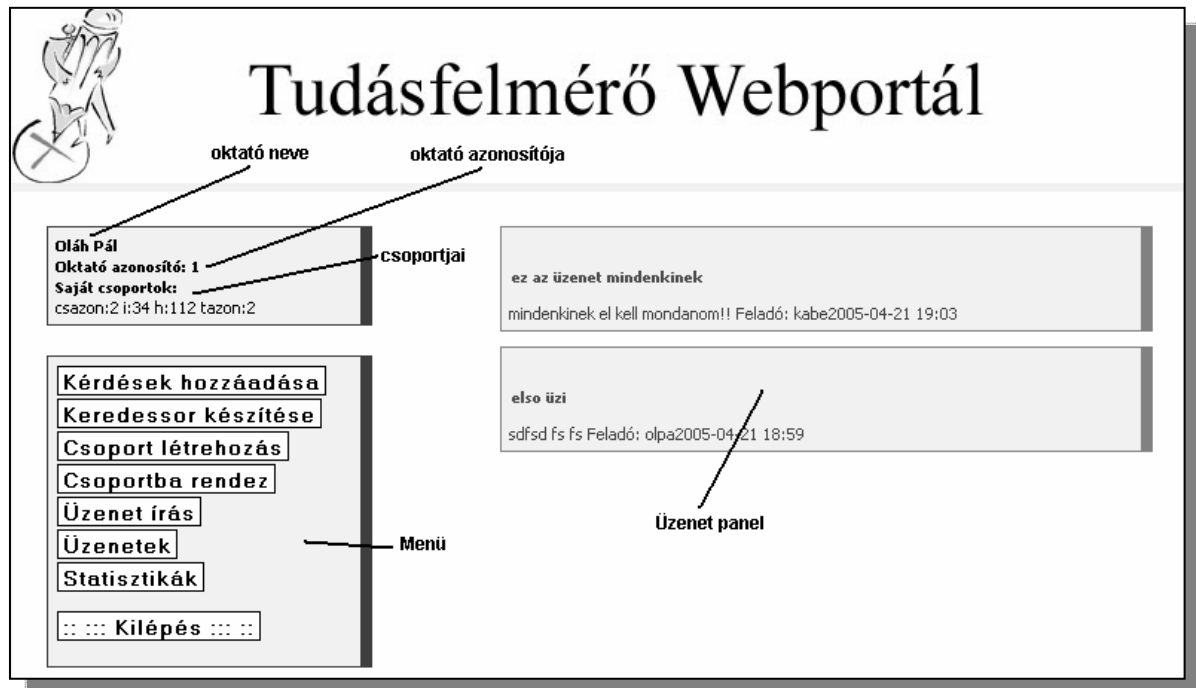
Tantárgy létrehozása:

Név: Megjegyzés:

6.1.1.3. ábra Tantárgy létrehozása

Egy tantárgy létrehozásánál (6.1.1.2. ábra) fontos, hogy nevet adjunk neki és ajánlott a megjegyzés megadása is, főleg akkor célszerű, ha több azonos tantárgyat akarunk létrehozni, de különböző évfolyam számára.

6.1.2 Az oktató lehetőségei



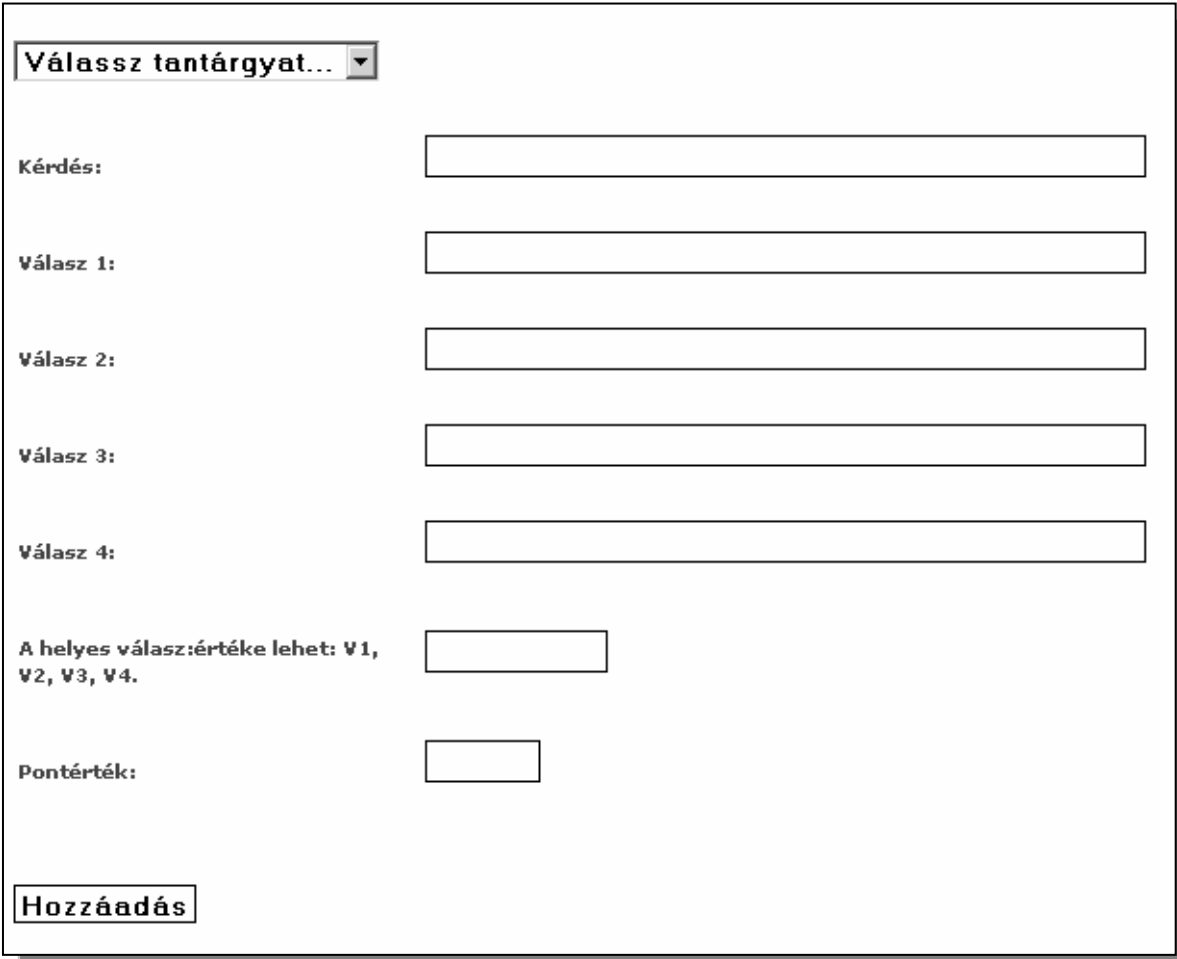
6.1.2.1. ábra Az oktató kezelőfelülete

Menüpontok:

- Kérdések hozzáadása
- Kérdéssor készítése
- Csoportba rendezés
- Üzenetírás
- Üzenetolvasás
- Statisztikák

Kérdések hozzáadása

Miután az admin létrehozott tantárgyat az adatbázisba, az oktató képes ezekhez kérdéseket feltölteni.



Válassz tantárgyat...

Kérdés:

Válasz 1:

Válasz 2:

Válasz 3:

Válasz 4:

A helyes válasz:értéke lehet: V1, V2, V3, V4.

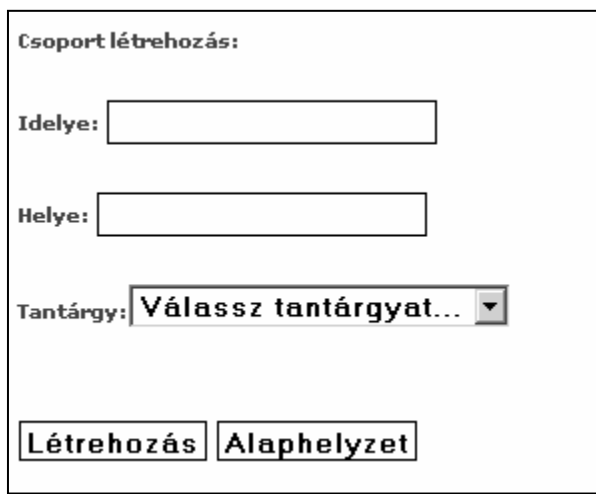
Pontérték:

Hozzáadás

6.1.2.2. ábra Kérdés feltöltése

Amint azt az 6.1.2.2-es ábrán látható először válasszuk ki azt a tantárgyat, melyhez hozzá szeretnénk rendelni a kérdést. Ez azért fontos, mert amikor a kérdéssort állítjuk össze annak a tantárgynak a kérdéseiből tudunk válogatni, melyet ott is megadunk. Majd a kérdés megadása következik, ami maximum 200 karakter hosszú lehet. A négy válasz lehetőség közül legalább kettőt meg kell adnunk, amik maximum 50 karakter hosszúak lehetnek. A helyes válasz megadásánál ügyelnünk kell arra, hogy nem a helyes választ kell megismételni, hanem a helyes válasz sorszámát. Vagyis lehetséges válaszok: V1, V2, V3, V4. Majd a pontérték, melynek értéke nagyobb legyen, mint nulla.

Csoport létrehozás

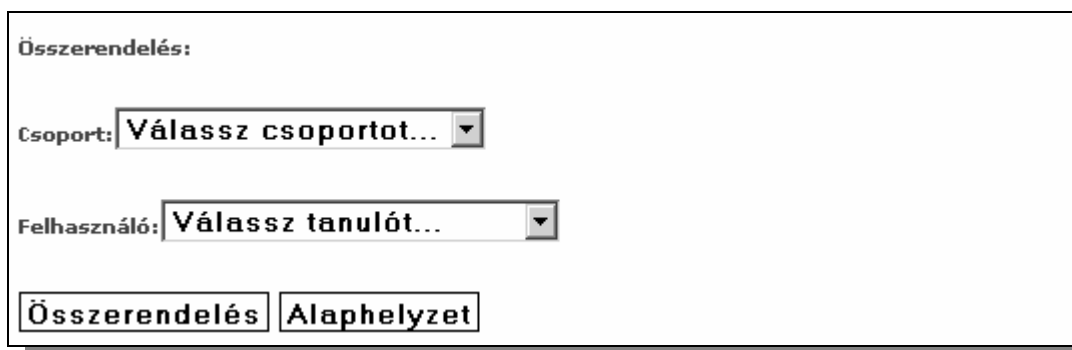


1. 6.1.2.4. ábra Csoportlétrehozás párbeszé

Csoportot kötelező létrehozni, ugyanis a diákoknak csak úgy tudunk dolgozatot készíteni, ha valamilyen csoportba vannak rendezve. Ezt a Csoportlétrehozás menüpont alatt tehetjük meg. Ekkor a 6.1.2.4-es ábrán látható párbeszédpanel jelenik meg előttünk. Bármilyen formátumban megadható az időpont és a hely, használhatunk vegyes karaktereket is. Továbbá meg kell adnunk, hogy a csoportnak milyen órát fogunk oktatni. Ez

azért fontos, mert ennek a csoportnak csak a kiválasztott tantárgyhoz rendelt kérésekből tudunk válogatni.

Csoportba rendezés



6.1.2.5. ábra összerendelés panel

Ezzel az opcióval tudunk diákokat csoportba rendezni. Egy diákot akár több csoportba is. Legördülő menüből válasszuk ki azt a csoportot, melybe a diákot szeretnénk belerakni, majd szintén legördülő menüből válasszuk ki azt a diákot, akit a csoportba szeretnénk rendezni.

Kérdéssor készítése

A dolgozat létrehozása (*Kérdéssor készítése*), ami már nem egy lépcsős művelet. Természetesen addig nem tudunk létrehozni dolgozatot, amíg nincs az adatbázisban tantárgy és hozzá rendelt kérdések halmaza. Továbbá fontos, hogy legyen csoport, akinek ezeket elkészítjük.

6.1.2.6. ábra Kérdéssor összeállítás

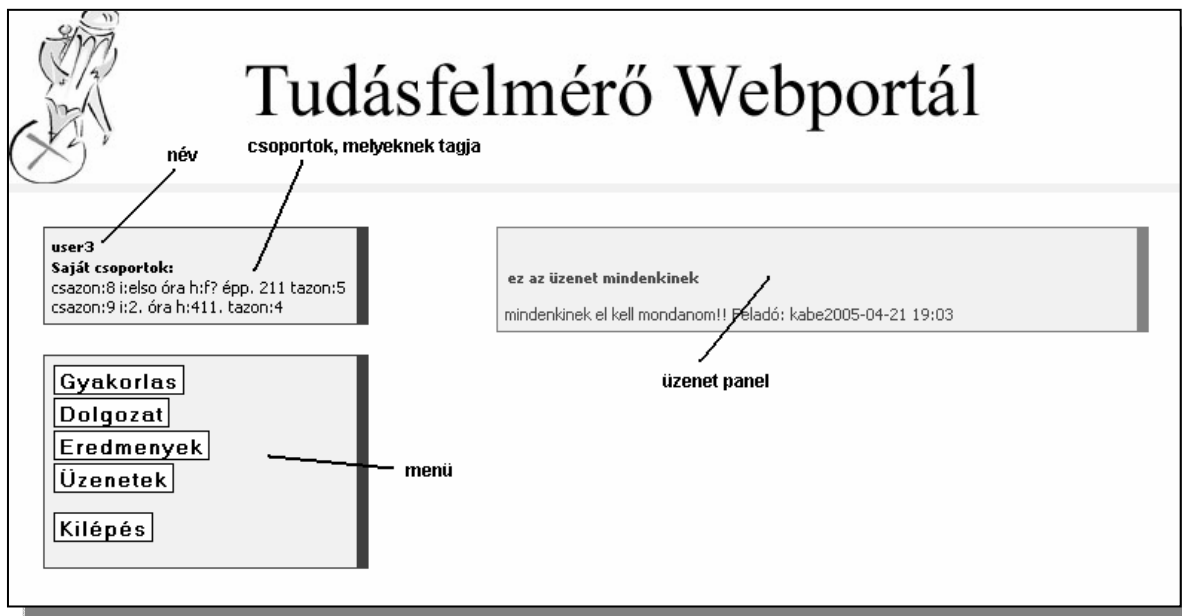
Első lépében meg kell adnunk, hogy melyik csoportnak készül a kérdéssor, és ez pontosan mikor lesz. Fontos a helyes dátum megadása, mert a diákok számára csak akkor válik elérhetővé a dolgozat. Elfogadott formátum dátumra: 2005.04.12
Megjegyzésként írhatunk bármit a diákok számára, hogy tudják, miről van szó. Nyomjunk egy Tovább gombot.

6.1.2.7. ábra Kérdések kiválasztása

A következő oldalon egy listát látunk, mely azokat a kérdéseket tartalmazza, mely a csoporthoz rendelt tantárgyhoz tartoznak. Amennyiben szeretnénk, hogy egy-egy kérdés szerepeljen a kérdéssorban, akkor az előtte lévő pipáló négyzetet jelöljük ki! Ha

készen vagyunk, klikkeljünk az Ok-gombra. Hiba nélküli kérdéssor összeállítás után a következő üzenetet kell, hogy kapjuk: **Kérdéssor összeállítva!**

6.1.3 A diák lehetőségei

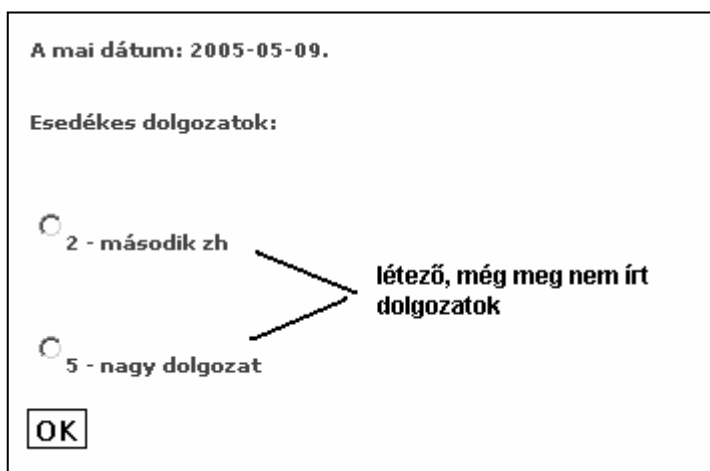


6.1.3.1. Diákok kezelőfelülete

Menüpontok:

- *Gyakorlás* egy választott tantárgyból kérdésekre lehet válaszolni
- *Dolgozat* ebben a menüben jelentkeznek az oktatók által összeállított kérdéssorok
- *Eredmenyek* saját eredmények megtekintése
- *Üzenetek* üzenetek olvasása(alapértelmezett)

A legfontosabbat emelném itt is ki, mégpedig a dolgozatírást. Ebben a menüpontban azokat az oktatók által már előre elkészített kérdéssorokat látjuk, amik épp az aktuális (mai) dátumra szólnak. Itt egy listát kapunk, melyek közül egyet kiválasztva elkezdjük azt megoldani. A szám jelöli a dolgozat azonosítóját.



5.4.3.2. ábra Dolgozat kiválasztása

The screenshot shows a quiz window with the following text and elements:

- Top status bar: "Kérdések száma: 5 Megválaszolt : 3"
- Question text: "Minek a rövidítése a ROM? (3pont)"
- Four radio button options:
 - ☒ véletlen hozzáférésű memória
 - ☐ csak olvasható memória
 - ☐ csak írható memória
 - ☐ írható és olvasható memória
- Bottom button: "OK"

Annotations with arrows point to the following parts:

- "kérdés" points to the question text.
- "lehetséges válaszok" points to the list of four radio button options.

1. 5.4.3.3. ábra Válaszadás

Miután az Ok-gombra kattintottunk, a kérdések egyesével jelennek meg. Amikre válaszolni kell. A fenti sorban láthatjuk, hogy összesen hány kérdésre kell válaszolni, és ebből hányra válaszoltunk már. Miután az összes kérdésre válaszoltunk a program kiírja, hogy hány pontból pontosan mennyit teljesítettünk. Ekkor történik a pontszám eltárolása.

FONTOS! Mindenkinél szerepel a kilépés gomb, mely segítségével ki tudjuk léptetni a felhasználót. Minden esetben léptessük ki a felhasználót, ugyanis ilyenkor töröljük a *session* változókat.

Összefoglalás

A szakdolgozat célja egy olyan webportál létrehozása volt, melynek elsődleges feladata az oktatók munkájának megkönnyítése, interaktívvá tétele. Ezen belül is egy olyan részterületre koncentráltam, ami nemcsak munkaigényes a tanár számára, hanem rengeteg papírmunkával is jár. Ilyen például a dolgozatírás, számonkérés, melyet úgy érzem a kitűzött céloknak megfelelően teljesítettem. Mégis szükségesnek éreztem olyan pluszszolgáltatásokkal kibővíteni a programot (pl.: üzenetküldés), melyek nem szerves részei az előbb említett problémának, de nagy hasznossággal bírnak.

A programmal megvalósítottam, hogy az oktató saját belátása szerint töltsse fel az adatbázist kérdésekkel egy-egy tantárgyon belül, képes legyen csoportokat létrehozni és ezeknek a csoportoknak a tagjait meghatározni. Továbbá a diák nem csak a tanár által elkészített dolgozatokat képes megoldani, hanem az adatbázisban szereplő kérdésekre – gyakorlás kedvéért – tud válaszolni. Ezeken a feladatokon kívül beépítettem egy olyan kimutatást (statisztika, saját eredmények), melyet mindenki a saját jogosultságának megfelelően tud használni.

A feladat elkezdésekor még nem voltam birtokában annak a programozástechnikaitudásnak, melyet a munka során sikerült elsajátítanom. Ez alatt az idő alatt sok olyan apróságra jöttem rá, melyeknek sok hasznát vehetem majd a jövőben. Mindenféleképp szeretném kibővíteni a programot, hogy az minden szempontból megfeleljen a felhasználók elvárásainak. Ezt a programot nagyon praktikus lenne kibővíteni például egy elektronikus ellenőrzővel, amihez a szülőnek külön hozzáférése lehetne. Véget nem érően sorolhatnám azokat a dolgokat, melyeket szerettem volna, illetve szeretnék megvalósítani ebben a témában.

Ami most ennél fontosabb, hogy mit is sikerült megvalósítani mindabból, amit az elején célkitűzésként tettem. A feladatot úgy próbáltam teljesíteni, hogy könnyedén bővíthető, bárki számára alakítható legyen. Egyszerűségről nem beszélhetünk ebben a témában, pont azért, mert nagyon kiterjedt a problémakör, és olyan elengedhetetlen részeket tartalmaz, melyek fontosak egy dolgozat összeállításához.

Tehát végértékelésként azt kell, hogy mondjam, hogy az alapkonceptciót sikerült teljesíteni.

Irodalomjegyzék

- [1] Stolnicki Gyula - SQL programozóknak
Computer Books Kft. 2003

- [2] (több szerző) - Webvilág - A PHP, a MySQL és az Apache használata
Panem Kiadó Kft. 2003

- [3] Peter Moulding - PHP Haladóknak - Fekete Könyv
Perfact-Pro Kft. 2002

- [4] Eric A. Meyer - CSS Pocket Reference
O'Reilly 2003

- [5] Majzik István - UML alapú rendszermodellezés.
BME Méréstechnikai és Információs Rendszerek Tanszék, 2004
<http://www.ing.mit.bme.hu/~varro/uml/slides/UML.ppt> (2005.05.03)

- [6] Nagy Gusztáv - A WSDM weboldaltervezési módszer a gyakorlatban, 2005
<http://www.phpconf.hu/2005/media/eloadasok/WSDM.pdf> (2005.05.08.)

- [7] Fábián Zoltán - Adatszerkezetek és Programozási tételek 1.1 kiadás.
Budapest, 2005
http://fz.szily.hu/download/Adatszerkezetek_es_Prog%20_tetelek.pdf (2005.04.10)

- [8] STS - Felhasználók beléptetése és követése 3.(2001.03.19)
<http://www.prog.hu/cikkek/427/Felhasznalok+beleptetese+es+kovetese+3.html>

CD melléklet

A CD melléklet tartalmazza a forráskódot és az adatbázis. Továbbá az üzembe helyezést megkönnyítő Microsoft Windows platformra készült ingyenes webszervert is csatoltam.

A CD-n a következő könyvtárak találhatóak meg:

- adatbázis
- eszközök
- www

Az *adatbázis* könyvtár tartalmazza az adatbázis dump fájlakat:

- dolgozat.sql (üres adatbázis)
- dolgozat_teszt.sql (néhány adatot tartalmaz a teszteléshez)

Az *eszközök* könyvtár tartalmazza az AppServ nevű webszerver programot.

- appserv-win32-2.4.3.exe

A *www* könyvtár tartalmazza a weboldal teljes forrását (30 darab fájl).