

**Debreceni Egyetem
Informatika Kar**

DINAMIKUS ISKOLAI HONLAPOK KÉSZÍTÉSÉNEK MÓDSZERTANA

**ESETTANULMÁNY A VÁCI MIHÁLY GIMNÁZIUM
HONLAPJÁNAK KÉSZÍTÉSÉNEK ALAPJÁN**

Témavezető:

Dr. Papp Zoltán
egyetemi adjunktus

Készítette:

Orémusz Bertalan
informatika tanár szakos hallgató

**D E B R E C E N
2 0 0 9**

T a r t a l o m j e g y z é k

Bevezetés	2
I. Iskolai honlapok tervezése	3
I. 1. Igények, elvárások megismerése.....	3
I. 2. Lehetőségek felmérése	3
I. 3. Eszközök kiválasztása	4
I. 4. Honlap tervének elkészítése.....	5
II. Honlap elhelyezése	8
II. 1. Hardver kiválasztás, konfigurálás.....	8
II. 2. Operációs rendszer telepítése, beállítása, tesztelése	8
II. 3. Webszerver telepítése, beállítása, tesztelése.....	9
II. 4. Adatbázis-szerver telepítése, beállítása, tesztelése	12
II. 5. Dinamikus környezet telepítése, beállítása, tesztelése	17
II. 6. Kiegészítők kiválasztása, telepítése.....	19
III. Honlap létrehozása	20
III. 1. Adatbázisok létrehozása, feltöltése	20
III. 2. Vizuális elemek (képek, animációk) elkészítése.....	24
III. 3. A honlap tárhelyének strukturálása	25
III. 4. Stílusinformációk létrehozása	27
III. 5. Fő oldalak létrehozása	28
III. 6. Kiegészítők létrehozása.....	29
IV. Honlap tesztelése, karbantartása, bővítése	32
Összefoglalás	34
Irodalomjegyzék	35
Függelékek.....	36
1. függelék.....	36
2. függelék.....	37
3. függelék.....	38
4. függelék.....	39
5. függelék.....	40

“Nem akkor alkottál tökéleteset, ha már nem tudsz mit hozzátenni, hanem ha már nem tudsz mit elvenni belőle.”

Antoine de Saint-Exupéry

Bevezetés

A központi forrásokból finanszírozott Irisz-Sulinet program, illetve az egyes intézmények önerős fejlesztései hatására az ország általános és középiskoláiban lehetővé vált az Internet elérése. Természetes igény napjainkban a szülők és a tanulók részéről, hogy az iskolák ne csak „egyirányban” használják az Internet áldásait, hanem saját maguk is jelenjenek meg a világhálón. Így manapság (majdnem) minden iskola rendelkezik saját honlappal. Sajnos a honlapok között kevés a színvonalas, a kor igényeinek megfelelő és rendszeresen karbantartott. Az okok kézenfekvőek: nagyon kevés iskolának vannak kellő anyagi forrásai arra, hogy profi cégekkel készíttessék el, és tartassák karban oldalait; illetve kevés az olyan iskola, ahol van olyan tanár (esetleg tanuló), aki rendelkezik kellő szaktudással (és sok szabadidővel!) egy korszerű honlap elkészítéséhez.

Munkahelyemen, a Tiszavasvári Váci Mihály Gimnáziumban, saját honlap elkészítésének igénye még 2001-ben megjelent. Akkor készítettem el iskolán honlapjának első változatát, amit saját webszerverünkön helyeztem el (a rendszergazda teendőit is én látom el). 2009 elején készült el a honlap legfrissebb, 4-ik változata, amiben már dinamikusan generálódnak az oldalak, a fontosabb adatokat adatbázis-szerveren tárolja és tartalmaz interaktív elemeket is (pl. szavazás, üzenetek küldése). E többéves munka tapasztalatait és tanúságait szeretném megosztani a szakdolgozatomban, valamint felhívni a figyelmet a lehetséges hibákra és buktatókra. A honlapot a www.vacim-tvasvari.sulinet.hu címen lehet elérni és megtekinteni.

I. Iskolai honlapok tervezése

I. 1. Igények, elvárások megismerése

Minden komoly munka a tervezéssel kezdődik. Ennek a munkának az első lépése a honlappal kapcsolatos igények és elvárások összegyűjtése. Fontos megkérdezni az iskolavezetést, az iskola dolgozóit, a tanulókat, a szülőket (esetleg még a fenntartót is) milyen iskolai honlapot képelnének el, milyen információkat szeretnének kapni az oldalakról, milyen rendszerességgel várnák a frissítéseket.

A Váci Mihály Gimnáziumban az iskolavezetés kérte, hogy a honlapon jelenjenek meg az intézmény adatai (pl OM azonosító, postacím, telefon és fax szám), legyenek elérhetőek a fontosabb iskolai dokumentumok (pl SzMSz, Pedagógiai program, Házi rend, éves munkaterv), valamint a leendő tanulók kapjanak információkat az induló osztályokról, azok tantervéről (beiskolázás).

Iskolánk dolgozói javasolták, hogy legyen egy oldal a névadónkról, Váci Mihályról, továbbá néhány oldal, ahol megismerhető az iskola múltja és jelene.

A tanulók véleménynyilvánítási lehetőséget (szavazás, vendégkönyv) és sok-sok képet kértek.

A szülők az aktualitásokról vártak tájékoztatást, valamit kérték, hogy a honlapon minél könnyebben lehessen eligazodni, tájékozódni.

I. 2. Lehetőségek felmérése

Az elvárások és igények ismeretében, a következő teendő a lehetőségeink felmérése. Fontos tisztában lenni azzal, hogy egy színvonalas honlap elkészítése komoly szaktudást, sok időt, megfelelő hardvert és szoftvert igényel. Valamint nem árt néhány megbízható kolléga sem, akik segítenek a honlapra szánt anyagok, adatok összegyűjtésében, rendszerezésében.

Vegyük számba mire lesz szükség:

1. Szükség van 1 számítógépre (szerverre), amin a honlapot el lehet helyezni
2. A szerverre operációs rendszer kell telepíteni, amit megfelelően konfigurálni kell
3. A http kérések kiszolgálására telepíteni kell egy webszervert.

4. A dinamikus oldalak létrehozásához szükség van (webes) programok (scriptek) futtatására alkalmas környezetre
5. Az adatok kezelésére kell egy megfelelő adatbázis-rendszer
6. Az oldalak elkészítéséhez megfelelő fejlesztőeszközök is kellenek
7. Objektív szükségleteken kívül szubjektív tényezőkre is gondolni kell! Például az oldalak elkészítésére szánt idő (ami legalább napi 2-3 óra), vagy programozási és adatbázis-kezelési ismeretek, és kreativitásból is kell egy jó adag.

Fontos felmérni, hogy a szükséges eszközökre milyen anyagi forrásai vannak az iskolának, illetve az oldalak elkészítéséhez, adatbázisok feltöltéséhez, scriptek megírásához ki és milyen segítséget tud adni.

I. 3. Eszközök kiválasztása

Az igények és lehetőségek ismeretében a következő lépés az eszközök kiválasztása.

Szervernek célszerű (ha van rá anyagi fedezet) egy olyan számítógépet választani, aminek teljesítménye, bővítésekkel, elegendő legalább 3-5 évre. Én nem javaslok speciális hardvert, mert meghibásodás esetén, illetve bővítéskor, a megfelelő alkatrészek ára sokszorosan meghaladja egy hagyományos egységek árát. A mi iskolánkban már 5 éve stabilan üzemel szerverként egy hagyományos PC, igaz ez a számítógép minőségi, válogatott alkatrészekből készült.

Szerver operációs rendszerként a mi iskolánkban Windows működik, de lehet választani akár a Linux valamelyik változatát is. Nálunk a Windows mellett a több éves tapasztalat (megszokás) szól, ami még az NT4-es időben kezdődött.

A http (HiperText Transver Protokol) kérések kiszolgálására alkalmas programok alapvető funkcióik azonosak. Minden webszerver HTTP kéréseket fogad a hálózatról, és HTTP válaszokat küld vissza. A HTTP válasz az esetek többségében egy HTML dokumentum, de lehet még egyszerű szöveges fájl, kép, vagy más típusú fájl is. A legnépszerűbb HTTP webszerverek a következők:

- Apache HTTP Server az Apache Software Foundation terméke.
- Internet Information Services (IIS) a Microsoft terméke.
- lighttpd

- Zeus Web Server a Zeus Technology terméke.
- Sun ONE a Sun Microsystems terméke.
- Cherokee HTTP Server – egy gyors és rugalmas webszerver, szabad szoftver.
- Xerver egy nyílt forráskódú, ingyenes, GPL licencelt szoftver.
- IBM Lotus Domino.
- MyServer egy gyors, biztonságos, sokoldalú webszerver, a MyServer Project terméke.
- SimpleW egy nagyon egyszerű és ingyenes webszerver, valamint JSP motor.

Dinamikus környezetet gimnáziumunkban a PHP biztosítja, ami egy ingyenes, gyors és népszerű programozói környezet. Alternatíva lehet az ASP (Application Service Provider), JSP (Java Server Pages), Perl, CGI (Common Gateway Interface), stb.

Adatbázis-rendszerként használni lehet SQLite-ot, MS SQL szervert vagy MySQL-t. Az MS SQL szerver elég erőforrás-igényes és egy iskolában nem lehet kihasználni az általa nyújtott szolgáltatásokat. Iskolánkban a MySQL szerverre esett a választás, mert oktatási intézményekben ingyenesen lehet használni, a PHP hatékonyan támogatja, gyors, biztonságos és kevés erőforrással is beéri.

Az oldalak elkészítéséhez szükség van fejlesztő-eszközökre. Az adatbázisok létrehozásában és karbantartásában hatékony segítséget adnak a MySQLCC (MySQL Control Center) és a winMySQLadmin programok. A weboldalak kódjának elkészítéséhez PHP Coder Pro-t (ingyenes), Front Page-t vagy PHP Designer-t (fizetős) lehet használni. A képek átalakításához, konvertálásához ideális eszköz a FastStone Image Viewer, a szerkesztésükhöz legjobb a Photoshop valamelyik változata.

I. 4. Honlap tervének elkészítése

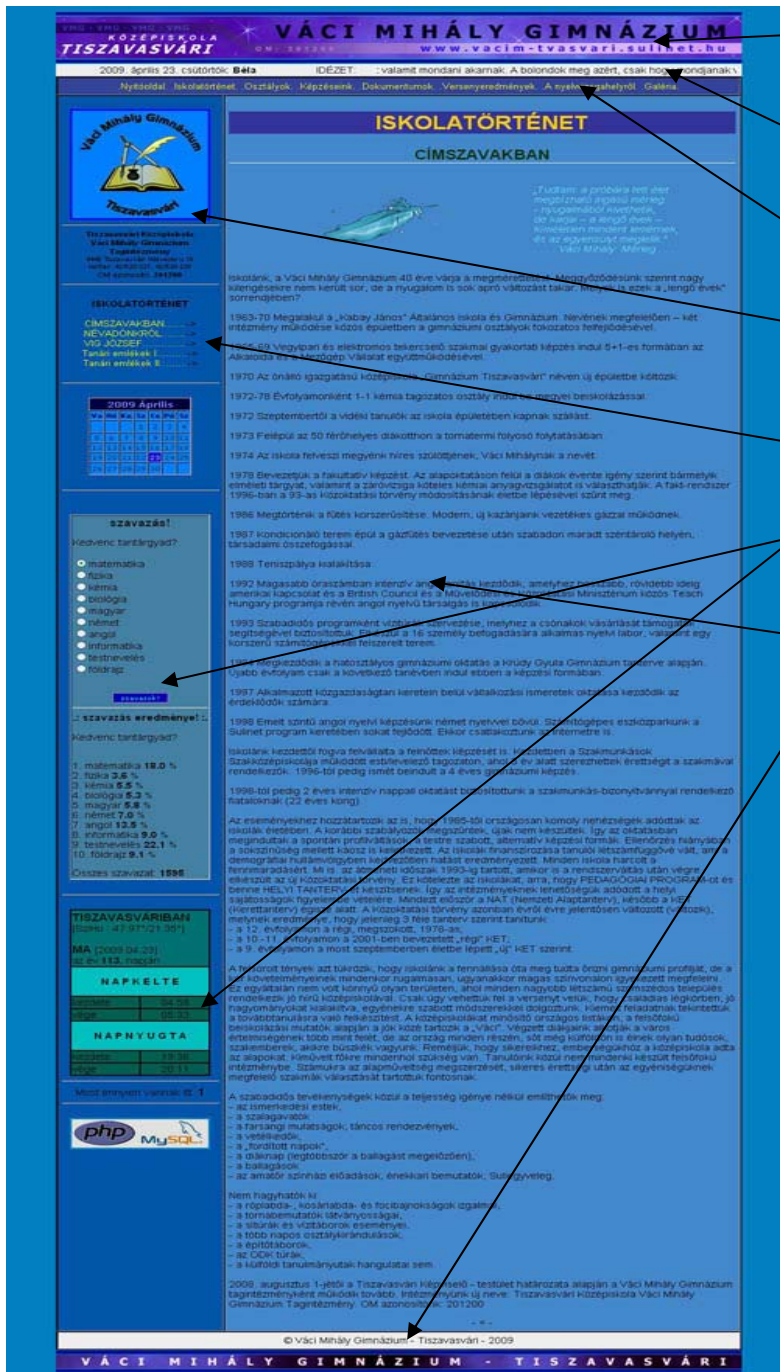
Következő teendő a honlap látványtervének elkészítése és szerkezetének megtervezése. Itt sokféle elrendezés között lehet választani, nálunk az **1. ábrán** látható szerkezet került kialakításra, ez lett az oldalak „sablonja”.

Kép (iskola neve) Dátum, idézet	
Globális navigációs rész	
Iskola logója, adatai, aktualitások	Oldal tartalma
Helyi navigáció	
Érdekessegek Szavazás	
Lábléc	

1. ábra

A legfelső sáv „Kép” részébe az iskola adatait tartalmazó kép kerülhet. Alá, dinamikus tartalomként, az aktuális dátum és névnap, mellé egy idézet, ami véletlenszerűen változik az oldal minden betöltésekor. A „globális navigációs rész” a honlap fő témaköreit tartalmazza, ez minden oldalon azonos. Az „iskola logója, adatai, aktualitások” rész tartalma a nevében benne van. A „helyi navigáció” területen a fő témakörön belül lehet navigálni, tartalma az aktuális „globális navigációs résztől” függ. Az „érdekessegek, szavazás” rész minden oldalon azonos, tartalma értelemszerű. Az „oldal tartalma” terület a kiválasztott téma fontos információit jeleníti meg. A „lábléc” az oldal készítőjéről és tulajdonosáról tartalmaz adatokat és tartalma minden oldalon azonos.

A gyakorlatban az elkészített sablont a **2. ábra** szemlélteti.



Kép (iskola neve)

Dátum, névnap, idézet

Globális menü

Iskola logoja, adatai, aktualitások

Helyi (lokális) menü

Szavazás, érdekességek

A helyi oldal tartalma

Lábléc

2. ábra

II. Honlap elhelyezése

Az iskola honlapját legegyszerűbb egy megfelelő bérelt tárhelyen elhelyezni, sajnos kevés a minden szempontból megfelelőt megtalálni. Sokszor alkalmazkodni kell a szolgáltató szoftvereihez és azok korlátozásaihoz, valamint az oldalak frissítése is körülményes. Ezért célszerű helyben, saját web-szerveren, megoldani az oldalak elhelyezését, publikálását.

II. 1. Hardver kiválasztás, konfigurálás

A honlapnak helyet adó számítógép, ahogy azt a I. 3 részben leírtam, lehet egy (viszonylag) olcsó PC felszerelve megfelelő kiegészítő hardver elemekkel. Arról, hogy a számítógép megfeleljen a választott operációs rendszer „normális” (nem minimális) követelményeinek mindenképpen gondoskodni kell. Ezért célszerű megfelelő mennyiségű kiegészítő RAM (memória) beszerelése, ami (Windows esetén) minimálisan 2 GB, további internetes alkalmazások futtatásához (pl. levelező rendszer, tűzfal, proxy) még nagyobb méretű memóriára van szükség (kb 4 GB). Az adatok tárolására javasolt 2 db merevlemez beépítése, ezek kapacitása a lehető legnagyobb legyen (legalább 200 GB). Ha a szerver helyettesítő-kiszolgálóként is működik (proxy pl. ISA) két hálózati kártyára is szükség van. Folyamatos (24 órás) üzem miatt, gondoskodni kell az egységek megfelelő hűtéséről (pl. merevlemez, ház)! Egy külső hardver, a szünetmentes tápegység (UPS) használata szintén nélkülözhetetlen.

II. 2. Operációs rendszer telepítése, beállítása, tesztelése

A legtöbb időt az operációs rendszer telepítése és beállítása igényli, mert sok és hosszadalmas lépésekből áll. Mivel a telepítés során óhatatlanul előfordulnak hibák, okos dolog felkészülni ezekre. Rendszeres mentésekkel (pl. ghost, partition backup) mindig rögzíteni kell az operációs rendszer aktuális működő változatát, amit probléma esetén bármikor vissza lehet állítani. Ezzel a módszerrel, tapasztalataim szerint, nagyon sok időt meg lehet takarítani, mert nem kell a teljes telepítést (beállítást) ellőről kezdeni.

Az operációs rendszer sikeres telepítése után ne feledkezzünk meg a kiegészítők (pl. DNS szerver) és az aktuális frissítések telepítéséről sem! Gyakran ez a művelet (Windows esetén) több időt igényel, mint az operációs rendszer telepítése.

A rendszer fontosabb beállításait már a telepítés során el kell végezni. Ilyenek (például) IP címek és a DNS név megadása. Az operációs rendszer stabil működését néhány napon keresztül érdemes tesztelni a megfelelő szolgáltatások elindításával, illetve speciális diagnosztikai programokkal.

II. 3. Webszerver telepítése, beállítása, tesztelése

A stabilan működő kiszolgálóra immár telepíthető az interneten történő megjelenéshez nélkülözhetetlen webszerver. A I.3 fejezetben megemlített webkiszolgálók közül a gyakorlatban két népszerű program közül érdemes választani: Apache vagy IIS. Nézzük kicsit részletesebben ezeket a webszervereket.

Az **Apache HTTP Server** (röviden **Apache**) egy nyílt forráskódú webkiszolgáló alkalmazás, szabad szoftver, mely kulcsfontosságú szerepet játszott a World Wide Web elterjedésében. A projekt célja egy 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 httpd szerverre épül, az Apache Szoftver Licenc feltételei alatt terjesztik. Az Apache egy robusztus, erőteljes és rugalmas webszerver, amely kompatibilis a HTTP/1.1 (RFC2616) protokollal. Az Apache projekt koordinálását az Apache Software Foundation végzi. Néhány vezető és több száz fejlesztő van e projekt mögött. Többek között a következő operációs rendszerekhez készítették el az Apache-ot: Unix, FreeBSD, Linux, Solaris, Novell NetWare, Mac OS X és Microsoft Windows. Az Apache sok szabványt támogat, melyeknek nagy része fordított modulok formájában áll rendelkezésre a mag kiegészítéseként. Ezek a modulok sok területet lefednek a kiszolgáló-oldali programnyelv-támogatástól kezdve a 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

loggolás a mod_log_config modulban és szűrési támogatás a mod_include és mod_ext_filter modulok segítségével. Az Apache-t statikus és dinamikus weboldalak közzétételére egyaránt használják. Sok webalkalmazást az Apache által nyújtott környezethez és szolgáltatásokhoz terveznek. Számos nagy honlap Apache webserveren lakik. A Google keresőmotor felülete egy módosított Apache-on fut, melyet a Google Web Server (röviden GWS) névre kereszteltek.

Iskolánkban a választás az IIS-re esett. Az **IIS** az **Internet Information Services** rövidítése. Ez a Microsoft cég terméke. Feladata hálózati szolgáltatások megvalósítása. Felépítése szerint rendszer szinten regisztrált szolgáltatásokból és beállító-vezérlő konzol részekből áll. A szolgáltatások egyenként látnak el egy-egy szolgáltatást, a konzol pedig ezeket kezeli. A szolgáltatások között található http vagyis a web kiszolgáló, FTP kiszolgáló, illetve levelezőszerver. Az IIS a Microsoft szerver rendszereinek alapja. A szoftver egy egyszerűbb, szolgáltatásokban korlátozott verziója a Windows XP Professional kiadásában megtalálható a Windows összetevők között. Illetve több egyéb szoftver csomag részét is képezi, többek között a Microsoft BackOffice Server kiszolgálócsomagban is megtalálható. Alapvetően támogatja a virtuális szerver módot, vagyis a névfeloldás szerint szétválasztott többszörös kiszolgálást. Röviden több web vagy FTP hely egy szerveren. A teljes beállítás grafikus felületen keresztül történik. Lehetőséget ad a kliens számítógépeknek, hogy az egyes helyek elérhetőek legyenek az alapvető Windows azonosítással (hitelesítéssel).

Az Internet Information Services szolgáltatás telepítése alapértelmezésben a rendszert tartalmazó kötet \inetpub mappájába történik. Sokan követik azt a taktikát, hogy az alkalmazásokat egy kötetre telepítik, az adatokat, pedig egy másikra. Így nagyon leegyszerűsödik a biztonsági mentés, mert egyszerűen le kell menteni az adat kötetet, az alkalmazások, pedig bármikor újratelepíthetők.

Az IIS beállítása előtt fontos, hogy megértsük azokat a problémákat, amelyekkel a használat során találkozhatunk. A problémák megértésével könnyebbé válik a beállítási folyamat. Ezek a problémák a következők lehetnek:

- Az internetkapcsolat tűzfala

- Távoli felügyelet
- IP-cím és tartománynév regisztrációja
- Engedélyek és a hozzáférés szabályozása

A Windows Server termékcsalád a távoli számítógépek a kiszolgálóhoz való engedély nélküli kapcsolódását egy szoftver alapú tűzfalal akadályozza meg. Az internetkapcsolat tűzfala alapértelmezés szerint nincs bekapcsolva, ha viszont a Windows Server termékcsalád valamely tagjának telepítése után, de még az IIS telepítése előtt a tűzfal be lett kapcsolva az alapértelmezett beállításokkal, akkor az ügyfelek nem kapcsolódhatnak a kiszolgálóhoz. Ahhoz, hogy az ügyfelek webes és egyéb IIS alapú kapcsolatokat kezdeményezhessenek a kiszolgálón, a megfelelő beállításokat el kell végezni az internetkapcsolat tűzfalán.

A hálózati rendszergazdák gyakran távoli eléréssel állítják be a kiszolgálókat, ami azt jelenti, hogy a kiszolgáló beállítása egy másik számítógépről történik. Amikor az IIS szolgáltatást telepíti, be kell állítania a távoli felügyeletet, ha a kiszolgálót távoli eléréssel kívánja felügyelni és konfigurálni.

Ha egy helyi hálózat tagjaként vagy egy útválasztó mögül konfiguráljuk az IIS szolgáltatást a kiszolgálón, a hálózati rendszergazdától IP-címet kell kérni, illetve szükség szerint azt, hogy az útválasztót és/vagy tűzfalat állítsa be az IIS adatforgalmának engedélyezéséhez. Ilyenkor a tartománynév regisztrációja általában nem szükséges, mivel az IIS-kiszolgálókat a helyi hálózatokon belül általában csak az intranethez használják.

Ha az IIS más kiszolgálókkal vagy hálózati komponensekkel, például útválasztókkal együttműködve fog tartalmat szolgáltatni, az IIS konfigurálásának megkezdése előtt meg kell győződni arról, hogy a többi kiszolgáló vagy hálózati összetevő konfigurációja engedélyezi az IIS kiszolgáló elérését, és hogy megfelelő hitelesítéssel rendelkezik.

A telepített és beállított szerver teszteléséhez el kell helyezni egy weboldalt a /wwwroot mappában, amit el kell tudni érni úgy az iskola belső hálózatából, mint „kívülről”.

II. 4. Adatbázis-szerver telepítése, beállítása, tesztelése

A dinamikus tartalmak létrehozásához, tárolásához nélkülözhetetlen egy adatbázis-szerver telepítése.

Az **adatbázis-kezelő rendszer** (angolul *Database Management System, DBMS*) többfelhasználós, hálózatos környezetben, adatbázisokhoz való hozzáférést, rendszeres és a felhasználói folyamatok zavartalan működést biztosító szoftveralkalmazás. Eredetileg csak nagy cégek alkalmaztak adatbázis-kezelőket, melyek nagy mennyiségű adatot kezeltek nagy teljesítményű számítógépekkel, de mára ezek elterjedt komponensei a cégek számítástechnikai rendszereinek.

Maga az adatbázis-kezelő különféle felhasználói folyamatok és a fizikai adatbázis – tipikusan egy fájlrendszer vagy más állománykezelő – között teremt kapcsolatot. A felhasználók különféle szándékkal és szemlélettel fordulhatnak a rendszerhez (például segédalkalmazás kapcsolódik a rendszerhez, lekérdezést akar futtatni, sémát hoz létre) – éppen ezért a rendszerhez kapcsolódó felhasználói folyamatokat egységesen nézetnek szokás nevezni.

A nézeteket élesen megkülönböztetjük aszerint, hogy azok az adatbázisban tárolt adategységek lekérdezésével („manipulálásával”) vagy az adatszerkezetek változtatásával, kialakításával („definiálásával”) foglalkoznak. A két nézettípusnak eltérő nyelvezete van, illetve lehet, előbbit általában az adatlekérdező DML (Data Manipulation Language) nyelv, utóbbit az adatszerkezetet kialakító DDL (Data Definition Language) vezérli. A legtöbb adatbázis-kezelő nyelve mindkét elemet integráltan tartalmazza, de tipikusan más nyelvi szabályok érvényesek az egyikre és másokra is (lásd például SQL nyelv).

Mindkét nyelv feldolgozását, értelmezését maga az adatbázis-kezelő végzi – a két nyelvhez pedig két értelmezést végző szoftverkomponens tartozik, amelyeket rendre lekérdezés-feldolgozónak, illetve sémafordítónak nevezünk (az adatbázis-kezelő típusától függetlenül). A DDL és DML nyelvek kialakításának egyik célja az volt, hogy a felhasználónak elegendő legyen kizárólag az adatok logikai adatszerkezetével tisztában lenniük, ne kelljen ismerniük az adatok fizikai tárolásának formáját, mikéntjét. Ha ezekhez a nyelvekhez külön fordító tartozik, akkor értelemszerűen több DDL és DML nyelven is kapcsolódhatunk akár ugyanahhoz az

adatbázishoz is – tehát egyfajta függetlenséget, nevezetesen logikai adatfüggetlenséget tudunk általa biztosítani. Például ma tipikusan objektumorientált DDL-en keresztül is hozzáférhetünk egy relációs adatbázis-kezelőhöz.

A felhasználói utasítások értelmezése után annak konkrét végrehajtására kerül sor, amely az adatbázis-kezelő motor (angolul: database engine) egyik feladata. Az adatbázis-kezelő motor ezt egyrészt úgy teszi meg, hogy a felhasználói folyamatok egyidejű futásából eredő potenciális problémák elkerülése (például közösen írnak és olvasnak egy adataegységet, egyik folyamat elöl mindig elveszik az erőforrásokat, folyamatok egymásra várnak), illetve zavartalan futásuk biztosítása érdekében mindenféle szabályozást eszközöl, más szóval: tranzakció-kezelést végez. Másrészt pedig, magát a fizikai adatbázist kezelő rendszert kell megfelelő utasításokkal vezérelni úgy, hogy az a kapott utasításokat optimalizálva a felhasználói folyamatot a lehető legrövidebb idő alatt szolgálja ki.

Sokak szerint az adatbázis-kezelő tartalmaz egy állománykezelőt is, de ez nem feltétlenül igaz. A legtöbb kereskedelmi termék önálló, az adatbázis-kezelő magját képező adatbázis-kezelő motortól független állományszervezési rendszert dolgozott ki elsősorban amiatt, hogy a háttértár szervezése, illetve eszközfüggőségéből (például máshogy viselkedik egy optikai, egy lyuk-, egy mágnesszalagos és egy merevlemezes háttértár is, hogy a memóriaadatbázis-okról ne is beszéljünk) eredő sajátosságai ne befolyásolják jelentősen az adatbázis-kezelő motor kialakítását, illetve hatékonyságát. A szétválasztás azt is lehetővé teszi, hogy a háttér optimalizálását (például töredezettség-mentesítését) az adatbázis-kezelő motortól függetlenül, azt nem befolyásolva is végrehajthassuk; de arra is lehetőséget teremt, hogy elosztott, akár több önálló (idegen szóval: autonóm) számítógépből rendszereket is kezelni tudjunk egyetlen adatbázis-kezelő motor segítségével. A szétválasztás tehát egy másik fajta adatfüggetlenséget biztosít, vagyis az ún. fizikai adatfüggetlenséget garantálja.

Jelenleg elérhető termékek

- Oracle Corp.: Oracle Database, BerkeleyDB
- IBM: DB/2, Informix, Illustra
- Microsoft: MS-SQL
- SyBase: SyBase (korábban System NN – NN helyén egy verzió állt)

- Borland: FireBird (korábban, illetve Borland termékeknél InterBase)
- Computer Associates (CA): Ingres, Jasmine
- Progress: ObjectStore
- SAP: MaxDB
- Novell: FLAIM

Nyílt forráskódban elérhető adatbázis-kezelők

- BerkeleyDB
- Derby (ugyanaz, mint IBM Cloudscape, Sun Java DB)
- FireBird (ugyanaz, mint InterBase)
- FLAIM
- Ingres (ugyanaz, mint openIngres)
- MaxDB (ugyanaz, mint SAPDB)
- MySQL
- PostgreSQL
- SQLite

Iskolánkban a választás a MySQL-re esett.

A **MySQL** egy többfelhasználós, többszálú, SQL-alapú relációs adatbázis-kezelő szerver. A szoftver fejlesztője a svéd MySQL AB cég, amely kettős licenccel teszi elérhetővé a MySQL-t; választható módon vagy a GPL, vagy egy kereskedelmi licenc érvényes a felhasználásra.

Az MySQL az egyik legelterjedtebb adatbázis-kezelő, aminek egyik oka lehet, hogy a teljesen nyílt forráskódú LAMP (Linux–Apache–**MySQL**–PHP) összeállítás részeként költséghatékony és egyszerűen beállítható megoldást ad dinamikus webhelyek szolgáltatására.

Egyedi illesztőfelületekkel az adatbázis-kezelő elérhető C, C++, C#, Delphi, Eiffel, Smalltalk, Java, Lisp, Perl, PHP, Python, Ruby és Tcl programozási nyelvvel. Egy *MyODBC* nevű ODBC interfész további, ODBC-t kezelő nyelvek számára is hozzáférhetővé teszi az adatbázis-kezelőt. A MySQL számára az ANSI C a natív nyelv.

A MySQL adatbázisok adminisztrációjára a mellékelt parancssori eszközöket (*mysql* és *mysqladmin*) használhatjuk. A MySQL honlapjáról grafikus felületű

adminisztráló eszközök is letölthetők: *MySQL Administrator* és *MySQL Query Browser*.

Széles körben elterjedt és népszerű adminisztrációs eszköz a PHP nyelven írt, nyitott forráskódú *phpMyAdmin*. A *phpMyBackupPro* (amelyet szintén PHP-ban írtak) adatbázisok (akár időzített, ismétlődő) mentésére szolgál eszközzül.

A MySQL különböző platformokon futtatható: AIX, BSDi, FreeBSD, HP-UX, Linux, Mac OS X, NetBSD, Netware, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP és a Windows frissebb verziói. Már MacOSX 10.4-re is elérhető. Egy OpenVMS átírat elérhető.

A MySQL képességei:

- ANSI SQL 99, számos kiegészítéssel
- Keresztplatformos elérhetőség
- Tárolt eljárások
- Adatbázis triggerek
- Kurzor adatbázisok
- "View" adatbázisok
- Valódi VARCHAR támogatás
- INFORMATION_SCHEMA támogatás
- "Strict" (szigorú) mód
- X/Open XA elosztott tranzakció-feldolgozás (DTP) támogatása; az Oracle InnoDB motorjának használata
- Különálló tároló motorok,(MyISAM olvasási sebességért, InnoDB a tranzakciókhoz és a referenciális integrációhoz, MySQL Archive az elavult adatok kevés helyen történő tárolására
- Tranzakciók az InnoDB, BDB és Cluster tároló motorokkal
- SSL támogatás
- Lekérdezés gyorsítótár (cache)
- Egymásba ágyazott SELECT -ek
- Szöveges indexelés és keresés a MyISAM motorral

- Beágyazott adatbázis-könyvtár
- Részleges Unicode támogatás
- ACID megfelelés az InnoDB-vel, BDB-vel és Cluster-rel
- Továbbfejlesztett MySQL Cluster
- "Példányosítás"

MySQL szerver telepítése Windows operációs rendszeren.

A telepítéshez az Internetről le kell tölteni a program valamelyik verzióját. Többféle csomag letöltésére van módunk, válasszuk azt, amelyikben a szerver és a kliens is benne van. Keressük meg a <http://www.mysql.com> címről kiindulva valamelyik magyar tükörszervert.

A letöltött fájl kicsomagolása után futtassuk a SETUP.EXE programot. Elindul egy telepítő varázsló. Sok választási lehetőségünk nincs, lényegében csak a célmappát tudjuk módosítani. Tegyük fel, hogy itt is megmaradtunk az alapértelmezett beállításoknál és a C:\MYSQL könyvtárba tettük az adatbázis kiszolgálót. Mappaszerkezetünk ebben az esetben a következő:

C:\MYSQL\BIN - Itt található a futtatható állományok melyek magát a szervert és a klienseket is jelentik. Illetve néhány kisebb alkalmazás-konfigurálási, adminisztrációs céllal.

C:\MYSQL\DATA - Az adatbázisok mappája. Ha belépünk két további mappát találunk "mysql" és "test" néven. Mindkettő mintaadatbázist takar. Tegyük fel, hogy létrehozunk egy adatbázist "Sales" néven, néhány táblával. Ebben az esetben létrejön a C:\MYSQL\DATA\SALES mappa és benne lévő fájlok nevei megegyeznek a táblák neveivel. Legegyszerűbben tehát úgy tudjuk meg, hogy milyen adatbázisok találhatóak a kiszolgálón, hogy megnézzük a C:\MYSQL\DATA mappa almappáit.

C:\MYSQL\DOCS - Dokumentációkat tárol. Mindenkinek ajánljuk figyelmébe a több mint 1,5 MB-os MANUAL.HTML-t, melyben részletes leírás található az SQL nyelvről és a MySQL kiszolgálóról.

C:\MYSQL\LIB - Itt alapértelmezésben egyetlen fájl található CYGWINB19.DLL néven. Kétféle parancssori segédprogram létezik a kiszolgáló közvetlen elérésére: MYSQL és MYSQLC. Utóbbi a Cygnus GNU fordítójával lett

fordítva és használata előtt a DLL-t be kell másolni a %systemroot%\system32 mappába.

C:\MYSQL\SHARE - Különböző nyelvekre fordított hibaüzenetek. Megtalálható a "hungarian" mappában a magyar nyelv is. Alapértelmezett az angol.

A regisztrált verziók létrehoznak egy C:\MY.CNF fájlt a kiszolgáló alapbeállításainak tárolására. Ha nem a C:\MYSQL mappába kérjük a telepítést és nem rendelkezünk regisztrált verzióval, akkor létre kell hoznunk a C:\MY.CNF fájlt. Legegyszerűbb, ha átmásoljuk a C:\MYSQL\MY-EXAMPLE.CNF mintafájlt és átszerkesztjük. Tegyük fel, hogy az E:\MySQL mappát választottuk, ekkor minimum a következő sornak szerepelnie kell a konfigurációs állományban:

```
[mysqld]
basedir=E:/MySQL/
```

Kilépve a varázslóból még nem vehetjük igénybe a szerveret. Elsőként telepíteni, majd futtatni kell a kiszolgáló szolgáltatást.

Annak érdekében, hogy ne kelljen a szolgáltatások kezelőfelületét meglátogatni a szolgáltatás elindítása és leállítása érdekében a WinMySQLadmin nevű eszközt érdemes használni, ami jelentősen megkönnyíti a dolgunkat. A WinMySQLadmin a MySQL könyvtár /bin alkönyvtára alól indítható a telepítés után - ajánlott egy indítóikon elhelyezése az asztalon vagy a gyorsindító eszközsorban! Amikor először indítjuk, nem sok mindent látunk: az eredmény mindössze egyetlen apró, közlekedési lámpát ábrázoló ikon a képernyő alján, az óra mellett. Ha a lámpa piros, a MySQL kiszolgáló szolgáltatás nem fut, zöld lámpa esetén fut.

II. 5. Dinamikus környezet telepítése, beállítása, tesztelése

Ha PHP nyelven kívánunk webes alkalmazásokat készíteni, akkor megvan a lehetőségünk arra, hogy mindezt IIS szerveren is megvalósítsuk, vagyis a PHP telepíthető az IIS-re.

Az első teendő, hogy látogassunk el www.php.net weboldalra, és töltsük le a PHP legfrissebb stabil telepíthető verzióját. Válasszuk a Windows Binaries verziók közül a zip változatot, amely minden szükséges komponenst tartalmaz. (Ha CGI módban szeretnénk telepíteni a PHP-t, akkor töltsük le a futtatható verziót, és a

varázsló segítségével a CGI módú telepítés automatikus, azonban a kiterjesztéseket a varázsló nem telepíti.) Csomagoljuk ki a letöltött PHP verziót (pl. a C:\PHP mappába).

PHP telepítése CGI futtatható módban

Ha CGI módban szeretnénk a PHP-t használni, akkor a telepítést egyszerűen úgy is elvégezhetjük, ha a futtatható verziót töltjük le a zip helyett, és a telepítést a varázsló automatikusan elvégzi, azonban a PHP kiterjesztéseket nem telepíti.

- Nyissuk meg az IIS kezelőt az Administrative Tools -> Internet Information Services (IIS) Manager elindításával.
- Válasszuk ki a kívánt számítógépet, majd kattintsunk a Web Services Extensions-ra, majd válasszuk az Add a new Web service extension...-t.
- A kiterjesztés nevének adjuk meg azt, hogy PHP, és a szükséges fájlok listájába vegyük fel a PHP telepítési könyvtárából a php.exe állományt.
- Hozzunk létre egy mappát a c:\inetpub\wwwroot mappa alá, és legyen a neve phpscript.
- Az IIS kezelőben válasszuk ki az alapértelmezett weboldalt, és az újonnan létrehozott mappának nézzük meg a tulajdonságait a jobb egérgomb menüből a Properties-t választva.
- Az új dialógusablakban kattintsunk a Create-re, majd a Configuration-ra, és ha nem létezik php megfeleltetés, akkor az Add... segítségével hozzunk létre egyet. Az Executable mezőbe írjuk azt, hogy C:\PHP\php.exe, és az Extensions-hoz pedig azt, hogy ".php".
- Fogadjuk el a beállításokat OK-kal.

PHP telepítése ISAPI módban

Az ISAPI mód viszonylag új a PHP-ben, kevésbé tartják stabilnak, de ez az újdonságából fakad. Teljesítményben azonban messze eredményesebb, és biztonságosabb is a CGI változatnál.

- Másoljuk a telepített mappából a php.ini-recommended állományt a %WINDIR% mappába php.ini néven.
- Másoljuk a php4ts.dll állományt a system32 mappába.
- Az előbbieken ismertetett módhoz hasonlóan az IIS kezelőben vegyünk fel új Web Service Extension-t, csak a neve most legyen PHP-ISAPI, és a szükséges fájl, a C:\PHP\sapi\php4isapi.dll legyen.

- A CGI változatnál ismertetett módon hozzunk létre kiterjesztés megfeleltetést a ".php" kiterjesztéshez, de a megfeleltetett állomány a C:\PHP\sapi\php4isapi.dll legyen.

Ha szeretnénk PHP kiterjesztett függvényeket használni, akkor a %WINDIR% mappában elhelyezett php.ini állományban az extension_dir beállításban adjuk meg a C:\PHP\extensions mappát, és az adott kiterjesztés dll-ek bekapcsolásához távolítsuk el a megjegyzés jelet előlük.

A PHP működését tesztelhetjük egy kis script fájl futtatásával, amit a \wwwroot mappába kell elhelyezni. A fájl neve lehet *teszt.php*, tartalma:

```
<? phpinfo(); ?>
```

Ha a böngészőben megnyitjuk a sajátomain/teszt.php oldalt, akkor a PHP környezet összes adatát megkapjuk.

II. 6. Kiegészítők kiválasztása, telepítése

A megfelelő szerver környezet kialakítása után már csak az oldalak elkészítéséhez szükséges szoftvereket kell telepíteni.

A statikus (html) oldalak elkészítéséhez használjunk weboldalkészítő programot. Én erre a célra a FrontPage-et alkalmazom.

A dinamikus (php kódok) tartalmak létrehozásához használjunk php-szerkesztőt, ezekből igen nagy a kínálat. Az ingyenes szerkesztők közül én a PHP Coder Pro-t használom, de komoly segítség lehet egy profi (fizetős) szerkesztő is, pl. a Php Designer valamelyik változata.

Az adatbázisok menedzseléséhez kényelmes és nagy tudású eszköz a MySQLCC, amiben grafikus felületen lehet elvégezni minden adatbázis-kezelési műveletet.

A weboldalakon megjelenő grafikák, képek elkészítéséhez megfelelő eszköz lehet a Gimp nevű ingyenes képszerkesztő, ami persze nem vetekedhet a Photoshop-al. Amikor sok képet kell átméretezni, átnevezni, átalakítani nagyon hasznos segédeszköz a FastStone Image Viewer, ami képes a képek kötegelte feldolgozására.

III. Honlap létrehozása

III. 1. Adatbázisok létrehozása, feltöltése

Új adatbázis a következő paranccsal hozhatunk létre:

```
CREATE DATABASE adatbázis_név;
```

Például a *CREATE DATABASE test_db*; parancs egy *test_db* nevű, üres adatbázist eredményez.

Ha törölni szeretnénk egy már létező adatbázist, akkor használjuk a következő parancsot:

```
DROP DATABASE adatbázis_név;
```

Most törölni akarom az előbb létrehozott adatbázist, így a *DROP DATABASE test_db*; parancsot használom. Ha a törölni kívánt adatbázis nem létezik, akkor a parancs lefuttatásakor egy hibaüzenetet kapunk, miszerint a MySQL képtelen eldobni az adatbázis, hiszen az nem létezik.

Hogy elkerüljük az esetleges kellemetlenségeket - hibaüzeneteket - használjuk az *IF EXISTS* opciót adatbázis törlése esetén és az *IF NOT EXISTS* opciót adatbázis létrehozásakor:

```
CREATE DATABASE IF NOT EXISTS adatbázis_név;
```

```
DROP DATABASE IF EXISTS adatbázis_név;
```

Néhány tipp és dolgok tisztázása:

Az SQL parancsokat kis és nagybetűvel egyaránt lehet írni, azonban javasolt a nagybetű az áttekinthetőség érdekében

Az SQL parancs végét a “;” jelzi, de általában működik nélküle is

Az adatbázisok és a táblák nevei lehetőleg egyetlen szóból vagy alulvonással összekötött szavakból álljanak, amelyek csak az angol ábécé kis betűit tartalmazzák és lehetőleg tükrözzék a tartalmukat

Az adatbázisokon belül több táblát is készíthetünk, amelyek majd adatainkat fogják tartalmazni.

Egy táblát úgy képzelhetünk el, mint egy oszlopokból és sorokból álló egyszerű táblázatot, ahol az egyes oszlopokban szereplő adatok különböző típusúak lehetnek. A táblák egyes sorait rekordoknak nevezzük.

Mielőtt elkészítenénk első táblánkat, vessünk egy pillantást az MySQL adattípusokra. Három fő csoportba sorolhatjuk őket: numerikus, dátum és idő, valamint szöveg típusokat különböztetünk meg. Ezen típusok, mindegyikének altípusai vannak, amelyek méretben különböznek egymástól, dátum és idő típusoknál, pedig formátumban.

Numerikus típusok:

A numerikus adattípusoknál az nagyon fontos, hogy megértsük, bármelyik *INTEGER* (egész) kétféle lehet: *SIGNED* vagy *UNSIGNED* (előjeles vagy előjel nélküli). Különbség a kettő között az, hogy ha egy adattípusunk előjeles, akkor negatív számokat is tárolhatunk benne, ha pedig előjel nélküli, akkor kétszer akkora számokat tárolhatunk benne, mint előjeles változatában. Ha a táblák készítésekor nem mondjuk meg, hogy milyen legyen, előjeles lesz.

A leggyakrabban használt numerikus adattípusok és az extrém értékek, amiket bennük tárolhatunk:

TINYINT - 1 byte

SIGNED - minimum: -128 maximum: 127

UNSIGNED - minimum: 0 maximum: 255

SMALLINT - 2 byte

SIGNED - minimum: -32768 maximum: 32767

UNSIGNED - minimum: 0 maximum: 65535

MEDIUMINT - 3 byte

SIGNED - minimum: -8388608 maximum: 8388607

UNSIGNED - minimum: 0 maximum: 16777215

INT - 4 byte

SIGNED - minimum: -2147483648 maximum: 2147483647

UNSIGNED - minimum: 0 maximum: 4294967295

BIGINT - 8 byte

SIGNED - minimum: -9223372036854775808 maximum:

9223372036854775807

UNSIGNED - minimum: 0 maximum: 18446744073709551615

Ha tizedes számokat szeretnénk tárolni, akkor használjuk a *FLOAT* adattípust vagy ha ennél nagyobb tárolóra van szükségünk, akkor a *DOUBLE()* lesz a tuti választás.

A szövegtípusok:

Talán a leggyakrabban használt típus a *VARCHAR()*, amelynek maximális hossza 255 karakter lehet. Előnye más szövegtípussal szemben abban rejlik, hogy a táblák készítésekor megadhatjuk hosszúságát. Egy másik, gyakran felbukkanó szövegtípus a *TEXT*, amelyben 65,535 karakternyi szöveget tárolhatunk. Hosszát nem állíthatjuk. Abban az esetben, ha a *VARCHAR()* típus hosszának nagyobb értéket adunk, mint 255, akkor az automatikusan *TEXT* típusúvá konvertálódik. Ha ennél többre van szükségünk, akkor használjuk a *LONGTEXT* típust, amely maximális hossza 4,294,967,295. Ezekon kívül talán még a *MEDIUMTEXT* típust érdemes megemlíteni, amely maximális hossza 16,777,215 lehet.

Dátum és idő típusok

DATE: YYYY-MM-DD

DATETIME: YYYY-MM-DD HH:MM:SS

TIMESTAMP: YYYY-MM-DD HH:MM:SS

TIME: HH:MM:SS

YEAR: YYYY

A MySQL sok adattípust tesz elérhetővé a felhasználók számára, nekünk csak annyi a dolgunk, hogy a számunkra megfelelőt válasszuk ki. Ezen a ponton kell elgondolkodni, hogy projektünkben miért használunk adatbázist, milyen adatokat fogunk benne tárolni, hiszen a projekt sikere nagy részben a helyes adattípus kiválasztásán áll vagy bukik. Viszonylag gyakran történik meg, hogy elkészül egy projekt, működik is, ahogyan kell, azonban borzalmasan lassú. Ilyenkor általában vissza kell térni ahhoz a fázishoz, amikor az adattípusokat vagy a táblák struktúráját választottuk ki...

Táblák létrehozásánál és törlésénél is használjuk a már említett *IF EXISTS* és *IF NOT EXISTS* opciókat az esetleges hibák elkerülése érdekében. Így hozhatunk létre új táblát:

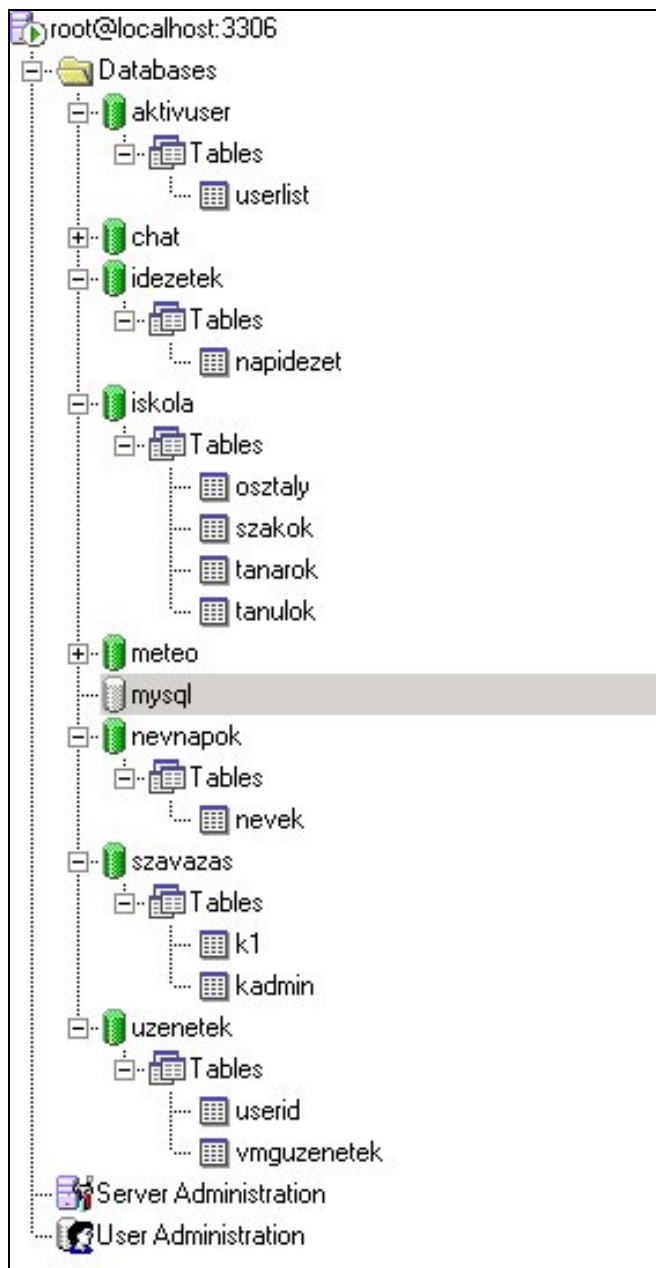
CREATE TABLE IF NOT EXISTS *tábla_név* (oszlop1 típus, oszlop2 típus, ..., oszlop3 típus);

A létrehozott táblákat ugyanolyan könnyedséggel távolíthatjuk el, mint ahogyan az adatbázisokat:

DROP TABLE IF EXISTS *tábla_név*;

Ha sikerült elkészíteni az adatbázisunkat és tábláinkat, akkor nincs más hátra, mint tábláink feltöltése adatokkal. Ezt az *INSERT INTO* paranccsal tehetjük meg.

Iskolánkban használt adatbázisokat a 3. ábra mutatja.



3. ábra

III. 2. Vizuális elemek (képek, animációk) elkészítése

Egy szép honlapot nem lehet elképzelni képek és animációk nélkül. Persze ésszerű kompromisszumokat kell kötni a képi elemek mennyisége és az oldalak mérete között. Minél több képet helyezünk egy oldalra, annál hosszabb ideig tart a letöltése.

A színek jelentéssel bírnak. A marketingben ez már régóta ismert tény. Azonban amikor színt választunk, lehet, hogy magunk sem tudjuk, de egyben látható rezgéseket is küldünk a reménybeli honlaplátogató felé.

Melyik alapszín mit jelent?

- Piros - energia, dinamizmus, a vér színe, erotikus, a tűz színe, cselekvésre készlet, negatív megközelítésben a harag színe
- Sárga - jókedv, nyár, tettekkészség, felszabadultság, negatív megközelítésben az irigység színe
- Zöld - természetesség, megnyugvás, remény, termékenység, negatív megközelítésben a féltékenység színe
- Kék - igazság, békesség, bölcsesség, az égbolt színe, higgadtság, biztonság, hűség, bizalom, tekintély
- Bordó - elegancia, drágaság
- Barna - férfiasság, stabilitás, a föld színe
- Narancs - büszkeség, önérzetesség, segítőkészség, ambíció, kreativitás, életerő
- Lila - lelkeség, szellem, királyi, misztikus
- Fehér - tisztaság, szűziesség, béke, teljes egyensúly, semlegesség
- Fekete - elegancia, méltóság, formális, szigorú, negatív megközelítésben gyász, depresszió

A színek együtt

A grafikai tervezőnek tisztában kell lennie azzal, hogy melyik színt melyikkel együtt alkalmazhatja, vagy melyek azok, amelyek nemcsak hogy nem illenek össze, de egyenesen **olvashatatlaná teszik a szöveget**, ha a háttér az egyik, a betű a másik színben készül. Ilyen "végzetes kombináció" a kék és a piros.

Általában harmonikus kombinációkat adnak azok a színek, amelyek a természetben is megtalálhatók egymás mellett.

Milyen kép (fotó) kerüljön a weboldalra?

- **Kis méretű!** Elektronikus értelemben a kép nagyságát a kilobyte méret határozza meg. Honlapra 100 kB alatti elegendő, sőt a galériaoldalak esetében minimális méretű indexképeket jelenítsünk meg, melyekre rákattintva nyílik meg a nagyméretű kép.
- Módosított, vagyis ha kell világosított, átméretezett és 72 DPI felbontású.
- A képfájl típusa JPG, GIF, esetleg PNG legyen, bár ez utóbbi némelyik böngészőn nem megfelelően jelenik meg.
- Legyen **figyelemfelkeltő**.
- Összhangban legyen a többi képpel vagy a szöveggel.

Például a Váci Mihály Gimnázium honlapján a **4. ábrán** látható kép került minden oldalra.

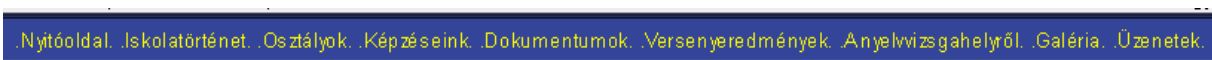


4. ábra

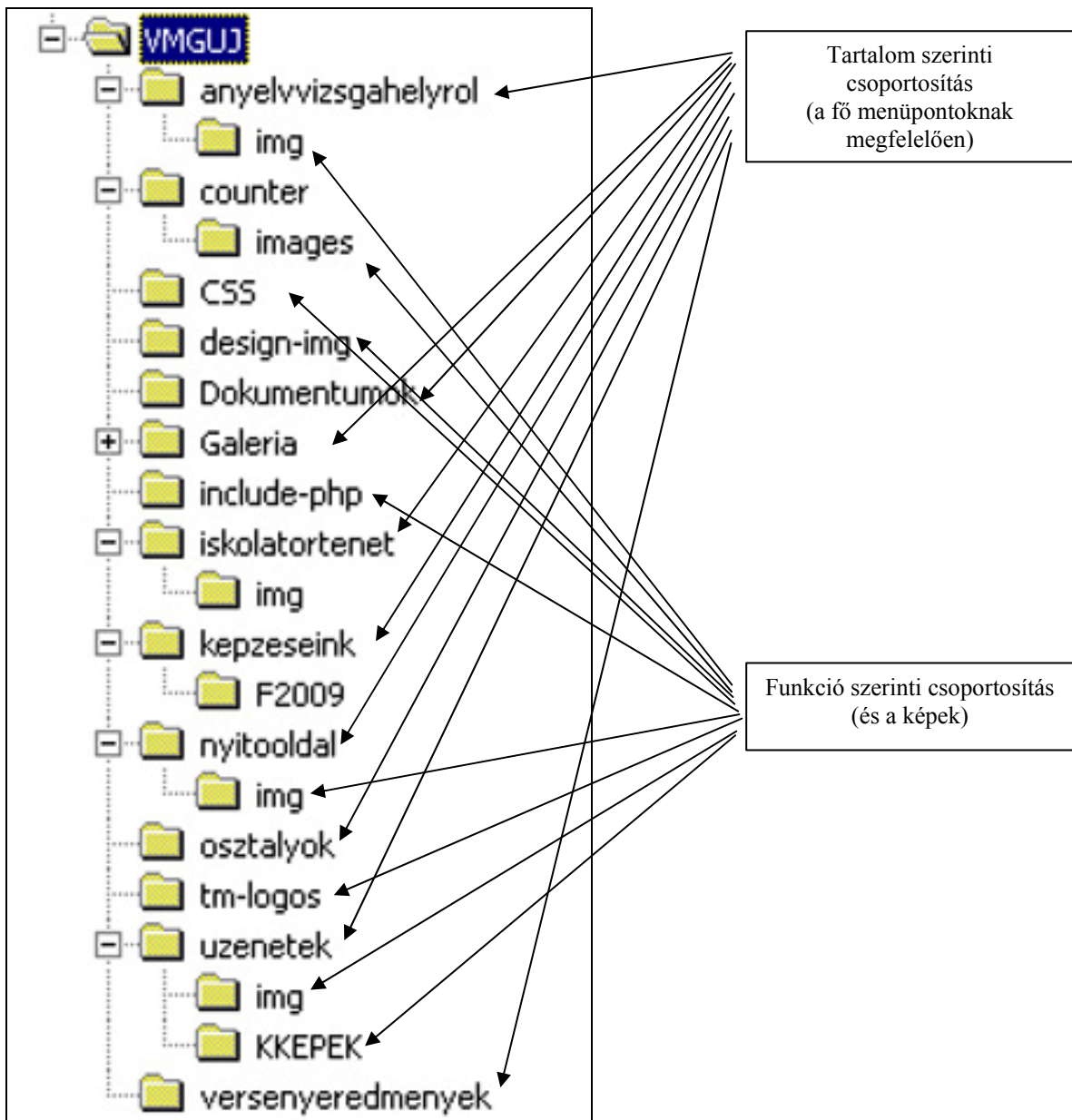
III. 3. A honlap tárhelyének strukturálása

Rend a lelke mindennek – tartja a mondás és ez így van! Már egy kisebb honlap is 30-40 weboldalt tartalmaz, plusz akár több száz képet, ezeket a fájlokat, a könnyebb kezelhetőség miatt, rendszerezve kell tárolni. A rendszerezés eszköze a mappák létrehozása, persze megfelelő szempontok alapján. A mi iskolánk honlapján két szempont szerint van rendszerezés: 1. tartalom szerint, 2. funkció szerint.

A tartalom szerint minden fő menüpontnak megfelel egy mappa, funkció szerint a közösen használt fájloknak, illetve a képeknek külön mappa van létrehozva. Iskolánk honlapja esetén a fő menüpontokat az **5. ábra**, a mappaszerkezetet **6. ábra** a szemlélteti.



5. ábra



6. ábra

Néhány magyarázat az ábrához. Az **img** mappa mindig az adott témához tartozó képeket tárolja, „anyelvvizsgahelyrol” mappa (például) a „Anyelvvizsgahelyről” fő menüponthoz tartozó fájlokat tartalmazza, az „include-php” a minden oldalon megtalálható közös elemeknek ad helyet.

A „CSS” mappa az oldalak egységes megjelenéséhez szükséges stílusinformációkat tartalmazza (*vmgcss.css* fájl, **1.függelék**), erről bővebben a következő fejezetben esik szó.

III. 4. Stílusinformációk létrehozása

A CSS jelentése Cascading Style Sheets, azaz egymásba ágyazott stíluslapok. A HTML oldalaink megjelenését befolyásoló egyszerű nyelvről van szó, mely segítségével meghatározhatjuk, hogy hogyan (és hol) jelenjenek meg az egyes HTML elemek (paragrafusok, címsorok, stb.), többek között befolyásolhatjuk a színüket, méretüket, elhelyezkedésüket, margóikat, stb. Az egymásba ágyazhatóság (kaszkádozás) arra utal, hogy több stíluslapot, meghatározást is megadhatunk egyszerre, illetve egy stílus lehet több elemre is érvényes, amit egy másik stílussal felüldefiniálhatunk. A stílusok öröklődnek az oldal hierarchiája szerint, ha például a gyökér elemre definiálunk egy stílust, akkor az többnyire az oldal összes elemére érvényes (a tulajdonságok örökölhetőségétől függően).

A CSS kód elsajátítása komoly lépés egy profi honlap elkészítése felé. A CSS előnye, hogy külön kezeli a külsínyt meghatározó formázásokat a szöveges tartalmat rögzítő HTML-kódtól, így a HTML-dokumentumot nem kell annyira teletűzdelni rengeteg, gyakran ismétlődő formázási elemmel. Korábban, a CSS bevezetése előtt a dokumentumok megjelenését meghatározó elemek a HTML kódon belül voltak - a betűtípusok, színek, háttér stílusok, elrendezések, dobozok, keretek és méretek külön meg voltak adva. A CSS használatával a webfejlesztők ezeket az információkat áthelyezhetik a stíluslapra, így a HTML kód egyszerűbb, világosabb lesz. A HTML dokumentumok kisebbé válnak, a lapok letöltési sebessége is javulhat.

A CSS abból a szempontból is kedvező, hogy nagyon egyszerűen lehet módosítani a stílusbeállításokat. Ha eldöntjük, hogy a szöveg ezentúl más színű vagy más méretű legyen, nem kell a teljes oldalt végigbogarászni a kódokat keresve, elég a CSS lapon átírni egy-két kódot, ami azonnal módosítja a teljes HTML dokumentumban az adott szövegtípus megjelenését.

A CSS stílusinformációkat beépíthetjük a HTML dokumentum fejlécébe, de csatolhatjuk külső fájlként is.

Több lapból álló honlapstruktúra esetén célszerűbb külön CSS fájlba elhelyezni a formázásokat, mert így mindegyik HTML lapról erre a CSS-re tudunk hivatkozni, így a honlapunk különböző oldalai teljesen egységes formázással fognak megjelenni. Ha pedig úgy döntünk, hogy változtatunk a honlapunk megjelenésén, elég a CSS fájlt

módosítani, ez egyszerre hatással lesz az összes oldalra. További előny, hogy mivel a webböngészők gyakran eltárolják az egyszer már letöltött CSS stíluslapokat a gyorsítótárban, a honlap oldalainak meglátogatása során a böngésző elég egyszer letöltse a CSS fájlt, így a letöltési sebesség gyorsul.

A gimnázium honlapján alkalmazott stíluslap definícióját az **1. függelék** tartalmazza.

III. 5. Fő oldalak létrehozása

Fő oldalak azok, amelyeket a honlap látogatói láthatnak. Ezek létrehozása azzal kezdődik, hogy minden globális navigációs pontnak (fő menüpont) a megfelelő mappájába bemásoljuk a „sablon” oldat, aminek a forrása megtalálható a **2. függelékben**.

Itt látható, hogy az oldal bizonyos részei **include** utasítással kerülnek beillesztésre. Ez nagyon hatékony és kényelmes módszer php kódok létrehozására, mert megfelelő tervezéssel a közös részeket nem kell minden lapon módosítani, elég egy helyen, továbbá a változó részek (lokális menü) is egyszerűen kezelhetők. Például a következő kérés:

`http://www.vacim-tvasvari.sulinet.hu/vmguj/nyitoldal/default.php?go=0`

azt jelenti, hogy a *nyitoldal default.php* sablonjába a változó (tartalom) részre a *page0.php* fájl kerüljön elhelyezésre. Ezt a *default.php* sablon (**2.függelék**) következő része végzi:

```
<?php $megnyit_lap="page".$go.".php"; include($megnyit_lap); ?>
```

A *\$go* (amit a kérés tartalmaz) határozza meg, hogy a *pageX.php* (X helyett egy szám állhat) fájlok közül melyik kerüljön beillesztésre.

Tehát a honlap fő oldalainak kialakítása a *default.php* fájl megfelelő mappákba való feltöltéséből és *pageX.php* fájlok (tartalom) elkészítéséből áll.

Példaként az Iskolatörténet menüpont kapcsolatait a **7. ábra** és **8. ábra** szemlélteti.

	Fájlmappa
img	
default	3 KB PHP Coder File
local-menu	1 KB PHP Coder File
page0	7 KB PHP Coder File
page1	5 KB PHP Coder File
page2	6 KB PHP Coder File
page3	4 KB PHP Coder File
page4	4 KB PHP Coder File

7. ábra



8. ábra

Magyarázat:

- *default.php* a sablon
- *local-menu.php* az Iskolatörténet helyi menüje
- *page0.php* a CÍMSZAVAKBAN (helyi) menüpont sablonba illesztendő része
- *page1.php* a NÉVADÓRÓL (helyi) menüpont sablonba illesztendő része
- *page2.php* a VIG JÓZSEF (helyi) menüpont sablonba illesztendő része
- *page3.php* a Tanári emlékek I (helyi) menüpont sablonba illesztendő része
- *page4.php* a Tanári emlékek II (helyi) menüpont sablonba illesztendő része

III. 6. Kiegészítők létrehozása

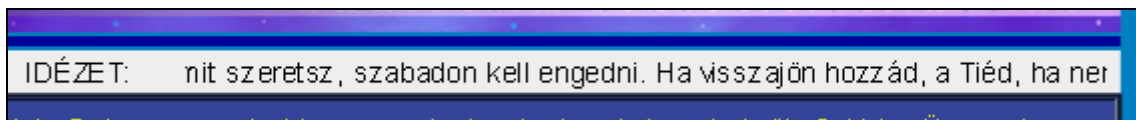
A kiegészítő elemek (részek) tesznek érdekesebbé és látványosabbá egy honlapot. Iskolánk honlapján a következő kiegészítők színesítik a tartalmat:

- A belépő oldalon van látogatottság-számláló
- Névnepok – Itt az aktuális dátumot és egy nevet látni
- Idézet – Itt egy véletlenszerűen megjelenő „okosságot” találunk
- Szavazás – Egy kérdésben lehet véleményt nyilvánítani
- Üzenetek – Rövid szövegek küldését teszi lehetővé
- Képküldés – Képeket lehet feltölteni a honlapra
- Napkelte - Napnyugta – Itt megtalálható hogy az adott napon Tiszavasváriban mikor kel és nyugszik le a Nap.

A következőkben néhány dinamikus kiegészítő felépítését részletesebben szeretném bemutatni.

Idézet

Az idézet részt a weboldalon a **9. ábra** mutatja.



9. ábra

Ennek a résznek a sablonban (*default.php 2.függelék*) a következő kód felel meg:


```
<marquee bgcolor="#EEEEEE" border="" align="middle"
scrollamount="2" scrolldelay="20" direction="left" behavior="scroll"
width="98%" style="font-family:Arial; color:0033FF; font-size:10">
<?php include('../include-php/idezet.php'); ?> </marquee>
```

A `<marquee...>` `</marquee>` között elhelyezett szöveg fényreklámszerűen jelenik meg az oldalon. Magát az idézetet egy adatbázisból az `idezet.php`-ban található kód keresi ki. Ezt a kódot láthatjuk a **10. ábrán**.

```
<?php
$user = "root"; $pass = ""; $host = "localhost";
$dbname = "idezetek";
$conid = mysql_connect( $host, $user, $pass );
$veletlen_szam = ( integer ) rand(1,230);
if ( ! $conid ) die ( "Hiba a kapcsolatban!" );
mysql_select_db ( $dbname ) or die ( "Nem lehet megnyitni az adatbázist!" );
$napiidezet = mysql_query( "SELECT napi FROM napidezet WHERE pkey =
$veletlen_szam" );
$idzet = mysql_fetch_row( $napiidezet );
foreach ( $idzet as $idzet )
print $idzet;
mysql_close( $conid );
?>
```

10. ábra

A fenti kód egy *idezetek* nevű MySQL adatbázis *napidezet* táblájából választ ki egy véletlenszerű rekordot. Az adattábla szerkezetét a **11. ábra** mutatja.

Field	Type	Null	Key	Default	Extra
 pkey	int(250)		PRI		auto_increment
napi	varchar(250)				

11. ábra

A *pkey* mező egész típusú (1-től 230-ig változhat), *napi* mező szöveg típusú és az idézet szövegét tartalmazza.

Képküldés

Érdekes eleme honlapunknak az a része, ahol a látogatók által feltöltött képeket lehet nézegetni. Ez a rész az „Üzenetek” menüpont „Ti küldtétek” lapján található. Az *uzenetek* mappa mappá tartalmát a 12. ábra mutatja

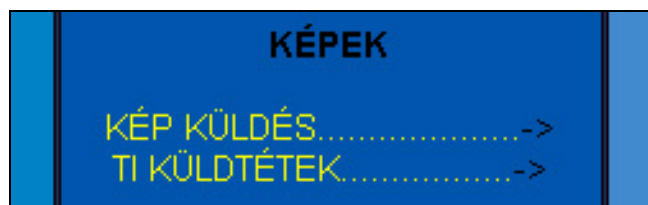
Név ▲	Méret	Típus
img		Fájlmappa
KKEPEK		Fájlmappa
default	3 KB	PHP Coder File
local-menu	1 KB	PHP Coder File
page0	4 KB	PHP Coder File
page1	3 KB	PHP Coder File
page2	5 KB	PHP Coder File
page3	2 KB	PHP Coder File
page4	2 KB	PHP Coder File
page5	1 KB	PHP Coder File

12. ábra

A fájlok funkciója:

- *default.php* – Sablon
- *local-menu.php* – Az üzenetek helyi menüje
- *page0.php* – Üzenetek listázása
- *page1.php* – Üzenet beírása
- *page2.php* – Üzenet rögzítés
- *page3.php* – Kép feltöltése
- *page4.php* – Feltöltött kép ellenőrzése, rögzítése
- *page5.php* – Képek megjelenítése

A képküldés helyi menüjét a 13. ábra mutatja:



13. ábra

A „KÉP KÜLDÉS” hivatkozás a *page3.php* fájlra mutat. A *page3.php* tartalmát a 3. függelék-ben lehet látni, képét pedig a következő 14. ábrán:

14. ábrán

Az ábrán látható, hogy a feltöltéshez ki kell választani a megfelelő képet (Tallózás), majd begépelni a jelszót (ez valójában egy ellenőrzés, hogy ember van-e a gépnél?), végül egy Feltöltés gombot kell megnyomni. Biztonsági okokból csak képeket lehet elküldeni (.jpg, .gif, .png, .bmp), a 150 Kb méretkorlát pedig, a helytakarékosság miatt.

Az elküldött képet a *page4.php* (4. függelék) ellenőrzi és rögzíti. Itt olvasható például, hogy a feltöltött képek átnevezés után az *uzenetek/KKEPEK* mappába kerülnek. Átnevezni azért kell, hogy ne legyenek egyforma nevű képfájlok.

A feltöltött képeket a *page5.php* (5. függelék) jeleníti meg. Valójában ez a programkód egy adott mappában lévő összes képet megjeleníti (bejárja) és ha kell átméretezi (max szélesség 500px) a képernyőkép védelme érdekében.

IV. Honlap tesztelése, karbantartása, bővítése

Az elkészült honlapot alaposan tesztelni kell különböző böngészőkkel. Mi az, amit meg kell vizsgálni?

- Azonosan jelennek-e meg az oldalak a különböző böngészőkben
- Jók-e az oldalakon, menükben elhelyezett hivatkozások (linkek)

- Milyen gyorsan töltődnek le az oldalak (ha kell mentsük az oldal képeit gyengébb minőségben!)
- Az oldalak dinamikus elemei helyesen működnek-e (php kódok), próbáljuk ki őket a „buta felhasználó” szemszögéből
- Ellenőrizzük a tartalmi elemeket (pl. helyesírás)

A működő honlapot rendszeresen karban kell tartani. Milyen teendőket kell elvégeznünk?

- Elavult tartalmak aktualizálása, törlése, frissítése
- Adatbázisok frissítése, tartalmuk törlése (pl. régi üzenetek)
- Biztonsági hiányosságok orvoslása (pl. sebezhető php kódok átírása, kiegészítése)
- Tárhely foglaltságának figyelése (ha kell törlés vagy bővítés)

A már működő és aktualizált honlapot (elegendő szabadidő esetén) érdemes folyamatosan bővíteni, fejleszteni. Ötleteket (ha sajátunk már nincs) lehet meríteni más oldalak példájából, vagy honlapunk látogatóinak javaslataiból. Arra azonban érdemes figyelni, hogy oldalaink ne legyenek túlzsúfoltak!

Összefoglalás

Szakedolgozatomban bemutattam hogyan jön létre egy teljes, dinamikus iskolai honlap a „semmitől”. Felvázoltam a honlaptervezés menetét a honlappal szembeni elvárások megismerésétől, a saját objektív és szubjektív lehetőségeink felmérésén keresztül, az eszközök kiválasztásán át, egészen a honlap tervének elkészítéséig. Ebből kiderül, hogy egy komoly honlap elkészítésébe nem szabad a „közepén” belevágni, alaposan fel kell rá készülni. Így már a tervezéskor eldőlhet mennyire lesz honlapunk látványos, könnyen áttekinthető és karbantartható.

Munkámból látható, hogy aki önállóan, komoly tartalommal szeretne megjelenni a világhálón - az informatika rengetek területével kapcsolatba fog kerülni, így hardverrel, operációs rendszerrel, webserverral, adatbázis-kezelővel, web-programozási környezettel, képszerkesztőkkel és nagyon sok kiegészítő programmal találkozik. Ezek beállításának, kezelésének, programozásának elsajátítása igen komoly, de nagyon izgalmas feladat.

A dolgozatban konkrét segítséget, útmutatást lehet találni a megfelelő dinamikus környezet kialakításával és programozásával kapcsolatban. Megtalálható néhány példa, amelyekből (remélhetőleg) sok hasznos tudás nyerhető (függelékek).

Munkámban igyekeztem megosztani közel tíz éves iskolai honlapkészítési tapasztalatomat és biztatok minden informatika tanár kollégát, vágjon bele bátran az iskolai honlap elkészítésébe, kellő kitartással biztos sikerben lehet része!

Irodalomjegyzék

1. <http://www.standard-team.hu/asp-php-html.html>
2. http://hu.wikipedia.org/wiki/Internet_Information_Services
3. <http://weblabor.hu/cikkek/mysqltelepites>
4. <http://weblabor.hu/cikkek/apachehttptelepites>
5. <http://szoftvertipp.info/programozas/mysql-tutorial-adattipusok/>
6. Christian Wenz:
PHP zsebkönyv
Kiskapu, Budapest 2007
7. George Reese, Randy Jay Yarger, Tim King
A MySQL kezelése és használata
Kossuth kiadó, Budapest 2003

Függelék

1. függelék

Stílusdefiníciós fájl (vmgcss.css)

```
body {
background-color: #0080C0;
color: #000000;
font-family: Arial, Helvetica, sans-serif;
font-size: 14px;
}
a {
color: #FFFF00;
font: bold;
font-size: 13px;
text-decoration: none
}
a:visited {
color: #FFFF00;
font: bold;
}
a:hover {
color: #FFFFFF;
font: bold;
background: #000066;
border-bottom-style: solid;
border-bottom-width: thin;
border-bottom-color: #FFCC33;
border-top-style: solid;
border-top-width: thin;
border-top-color: #FFCC33;
}
h1 {
text-align: center;
font-size: 30px;
background: #333399 repeat-x;
color: #FFCC00
}
h2 {
text-align: center;
font-size: 22px;
color: #003300;
border-bottom-style: solid;
border-bottom-width: thin;
border-bottom-color: #FFFFFF
}
h3 {
text-align: left;
font-size: 18px
}
```

2. függelék

Sablon fájl (default.php)

```
<?php session_start(); ?>
<html>
<head>
<title>V&acute;ci Mih&acute;ly Gimn&acute;gium - Tiszavasv&acute;ri - 2009 -
DOKUMENTUMOK</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link href="../CSS/vmgcss.css" rel="stylesheet" type="text/css">
</head>
<body>
<div align="center">


<br>
<table width="880" border="0" align="center" cellpadding="1" cellspacing="1"
bgcolor="#EEEEEE">
<tr> <td width="310" align="center">
<?php include('../include-php/nevnapok.php'); ?> </td>
<td width="90" align="center">
<div align="center"> ID&acute;ZET: </div>
</div></td>
<td width="470" align="center"> <marquee bgcolor="#EEEEEE" border="" align="middle"
scrollamount="2" scrolldelay="20" direction="left" behavior="scroll"
width="98%" style="font-family:Arial; color:0033FF; font-size:10">
<?php include('../include-php/idezet.php'); ?>
</marquee> </td> </tr>
</table>
<table width="880" border="1" cellpadding="5" cellspacing="1" bordercolor="#110011"
bgcolor="#223355">
<tr bgcolor="#334499">
<td colspan="2"><div align="center">
<?php include('../include-php/global-menu.php'); ?>
</div></td> </tr>
<tr> <td width="200" height="400" align="center" valign="top" bgcolor="#0055AA">
<?php include('../include-php/kep-in.php'); ?>
<?php include('local-menu.php'); ?><br>
<?php include('../include-php/calendar.php'); ?><br><br>
<?php include('../include-php/szavaz.php'); ?><br>
<?php include('../include-php/plusz.php'); ?> </td>
<td width="645" valign="top" bgcolor="#4488CC"><div align="center">
<?php $megnyit_lap="page".$go.".php"; include($megnyit_lap); ?>
</div></td> </tr>
<tr bgcolor="#EEEEEE">
<td colspan="2"><div align="center"><font color=#000000'>
&copy; V&acute;ci Mih&acute;ly Gimn&acute;gium - Tiszavasv&acute;ri - 2009
</font></div></td> </tr>
</table>


 </div>
</body>
</html>
```

3. függelék

Kép küldés (page3.php)

```
<?php
function randomString($len) {
    srand(date("s"));
    $possible="ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    $str="";
    while(strlen($str)<$len) {
        $str.=substr($possible,(rand()%(strlen($possible))),1);
    }
    return($str);
}
?>

<?php
if (! isset($kep_jelszo)) { session_register("kep_jelszo"); }
$kep_jelszo = "".randomString(8);
?>
<br><br>
<form enctype="multipart/form-data" action="default.php?go=4" method="POST">
<table border="1" width="95%" bgcolor="#004080">
<tr>
<td width="100%" colspan="3" bgcolor="#000000" height="70"><p align="center"><big><font
color="#FFFFFF"><big><big><strong>KÉP
KÜLDÉS!</strong></big></big></font></big><br><br>
<font color="#0080FF"><strong><small>FIGYELEM! CSAK KÉPEKET LEHET FELTÖLTENI
(jpg, gif,
png, bmp) MAX 150 Kb-ig!</small></strong></font></td>
</tr>
<tr>
<td width="50%" bgcolor="#000040" height="100"><font
color="#8080FF"><small><strong>KÉP
KIVÁLASZTÁSA:</strong></small></font><br><br>
<input type="hidden" name="MAX_FILE_SIZE" value="150000">
<input type="file" name="fajl"></td>
<td width="30%" bgcolor="#000040">
<font color="#8080FF"><strong><small>JELSZÓ:</small></strong></font>
<font color="#CCFFFF"><big><?php print "".$kep_jelszo;?></big></font><br><br>
<font color="#8080FF"><strong><small>JELSZÓ:</small></strong></font>
<input type="text" name="jelszo" size="9">
</td>
<td width="20%" bgcolor="#000040"><font color="#8080FF"><p
align="center"><strong><small>KÉP FELTÖLTÉSE!</small></strong></font>
<br><br><input type="submit" value="Feltöltés!">
</td>
</tr>
<tr>
<td width="100%" colspan="3" bgcolor="#000000" height="30"><p align="center"><small><font
color="#0080FF"><strong>TISZTELETTEL KÉREK MINDENKIT IZLÉSTELEN KÉPEKET NE
KÜLDJÖN!</strong></font></small></td>
</tr>
</table>
</form>
```

4. függelék

Kép küldés (page4.php)

```
<?php
$feltoltes_konyvtar = "F:/www-root/vmg/uzenetek/kkepek";
$ok_teszt = 0;
print "<META HTTP-EQUIV = 'Refresh' Content = '5; URL =default.php?go=5'> ";
/*----- FÁJL ELLENŐRZÉS!(LÉTEZIK-E?) -----
if ( isset($fajl) )
{
/* <<<<- - LÉTEZIK!!!
/*----- JELSZÓ ELLENŐRZÉS! -----
$jelszo_kuldve = $_POST['jelszo'];
$jisz_teszt = 0;
if ($kep_jelszo==$jelszo_kuldve)
{ $jisz_teszt = 1; }
else
{ $jisz_teszt = 0; }
/*-----

/*----- KÉP FORMÁTUM ELLENŐRZÉS + NÉV VÁLTÁS! -----
$fajl_name = strtolower($fajl_name);
$k_list = array("jpg", "gif", "png", "bmp");
$kep_teszt = 0; $kf_hosz = strlen($fajl_name);
for ($i=0; $i<=3; $i++)
{
if (substr($fajl_name,-3) == $k_list[$i])
{
$kep_teszt = 1;
$uj_nev = date("U").".".$k_list[$i];
}
}
/*-----

if ( $jisz_teszt*$kep_teszt == 1)
{
/*----- HA MINDEN RENDBEN!!! -----
copy($fajl, "$feltoltes_konyvtar/$uj_nev") or die("Nem lehet másolni?!");
print "<br><big><big>A KÉP TÁROLVA!!!</big></big>";
}
else
{
print "<br><big><b>HIBA!!!</b></big><br><br>";
print "A KÜLDENI KÍVÁNT FÁJL NEM KÉP!!!<br>";
print "VAGY HIBÁS JELSZÓ!!!";
}
}
/* <<<<- - NEM LÉTEZIK!!!
else
{
print "HIBA!!! A FÁJL NEM LETT ELKÜLDVE! (max méret 350 Kb!!!)";
}
?>
```

5. függelék

Kép küldés (page5.php)

```
<big><big><b>TI KÜLDTÉTEK!!!</b></big></big>

<?php
$path = "kpek";
// Mappa megnyitása
$dir_handle = @opendir($path) or die("Unable to open $path");
// A mappam fejljainak bejárása
while ($file = readdir($dir_handle)) {
if($file == "." || $file == ".." || $file == "index.php" )
continue;
$meret = GetImageSize ($path.'/'.$file);
if ($meret[0] > 500)
{ $modosit=' width="500"; }
else
{ $modosit=""; }

print '<p align="center">';
print '<br>';
print $file."<br>";
print "<small>eredeti képméret: ".$meret[0]."x".$meret[1]."</small>";
print '</p>';
}
// Mappa bezárása
closedir($dir_handle);
?>
```