

1.BEVEZETÉS	2
2.CMS RENDSZEREK RÖVID ÁTTEKINTÉSE	3
3.DRUPAL FELHASZNÁLÓI SZINTŰ BEMUTATÁSA	4
3.1. TELEPÍTÉS	4
3.2. ELSŐ LÉPÉSEK	4
3.3. BEÁLLÍTÁSOK	5
3.3.1. FELHASZNÁLÓ	5
3.3.2. ADMINISZTRÁTOR	5
3.4. TARTALOM BEKÜLDÉSE	9
4.FÁJL ÉS ADATBÁZIS SZERKEZET FELVÁZOLÁSA	10
4.1. A DRUPAL FÁJLSZERKEZETE	10
4.1.1. DATABASE	10
4.1.2. FILES	10
4.1.3. INCLUDES	11
4.1.4. MISC	13
4.1.5. MODULES	13
4.1.6. SCRIPTS	14
4.1.7. SITES/DEFAULT	14
4.1.8. THEMES	14
4.2. A DRUPAL ADATBÁZIS SZERKEZETE	15
4.2.1. BLOCKS	15
4.2.2. CACHE	16
4.2.3. COMMENTS	16
4.2.4. FILES	16
4.2.5. NODE	16
4.2.6. ROLE	16
4.2.7. SEARCH_INDEX	17
4.2.8. SYSTEM	17
4.2.9. USERS	17
4.2.10. USERS_ROLES	17
4.2.11. VARIABLE	17
4.2.12. WATCHDOG	17
5.FÜGGVÉNYEK FELTÉRKEPEZÉSE	19
5.1. FELHASZNÁLÓKAT KEZELŐ FÜGGVÉNYEK	19
5.1.1. USER_LOAD	19
5.1.2. USER_SAVE	19
5.2. TARTALMAKAT KEZELŐ FÜGGVÉNYEK	20
5.2.1. NODE_LOAD	20
5.2.2. NODE_SAVE	20
5.3. ADATBÁZIS KEZELŐ FÜGGVÉNYEK	20
5.3.1. DB_SET_ACTIVE	20
5.3.2. DB_CONNECT	20
5.4. MEGJELENÍTÉST KEZELŐ FÜGGVÉNYEK	21
5.4.1. INIT_THEME	21
5.4.2. THEME	21
6.ÚJ SMINK KÉSZÍTÉSE	22
7.ÖSSZEFOGLALÁS	24
8.IRODALOMJEGYZÉK	25

1. Bevezetés

A weboldal készítés, webprogramozás régóta érdekel, s úgy érzem ez az a témakör, amivel a későbbiekben hivatásszerűen foglalkozni szeretnék. Ezért kezdtem el iskolán kívül ismerkedni a HTML és PHP nyelvekkel, és ezért vettem fel az Adatbázis Alapú Webalkalmazások című tantárgyat. E tantárgy keretén belül kerültek megemlítésre a tartalomkezelő rendszerek (CMS), amelyek felkeltették az érdeklődésemet. Elhatároztam, hogy behatóbban szeretném tanulmányozni ezeket a rendszereket, mint amennyire a gyakorlatok szűkös időkerete alkalmat adott. Erre jó lehetőség kínálkozott a szakdolgozati téma kiválasztásakor, amikor is az Informatika Tanszék által javasolt témák között szerepelt egy szabadon választott tartalomkezelő rendszer felépítésének bemutatása. Úgy gondoltam ez jó alkalom a tartalomkezelő rendszerek közelebbi megismerésére, a HTML és PHP nyelvek terén szerzett eddigi ismereteim bővítésére, valamint egy konkrét tartalomkezelő rendszer ismerete előnyt jelenthet számomra a későbbiekben.

Rengeteg féle tartalomkezelő rendszer létezik. Ezeknek egy része „fizetős”, más részük ingyenesen elérhető. Én az ingyenesen letölthető, nyílt forráskódú, PHP nyelven íródott Drupal rendszert választottam bemutatásra. A Drupal egy széles körben használt rendszer, egy interneten talált lista szerint a második legnépszerűbb tartalomkezelő. Mellette szólt az is, hogy létezik magyar nyelvi kiegészítés hozzá, így itthon is sokan használják. Készült hozzá magyar nyelvű kézikönyv, és a magyar Drupal oldalon található fórumokon is segítséget kérhetek (<http://drupal.hu>).

Miközben saját ismereteimet bővitem, remélem sikerül egy mások számára is hasznos dokumentációt írnom a Drupalról, ami jól használható segítség lehet a rendszerrel ismerkedőknek.

2. CMS rendszerek rövid áttekintése

A CMS (Content Management System) magyarul tartalomkezelő rendszer. Ezek általában kevés szaktudást igénylő webes alkalmazások, melyek lehetővé teszik, hogy komoly szakértelem nélkül is bárki összetett weboldalt készítsen és karbantartson. Ennek előnye az, hogy a honlapon megjelenő új információt az tudja megjeleníteni, aki előállította és a téma szakértője. A weboldalak folyamatos karbantartásának legpraktikusabb módszere, ha azt a tartalom fejlesztésében, módosításában leginkább illetékes személyek végzik, ők azonban többnyire járatlanok a webes programozás terén. A tartalomkezelők lehetővé teszik honlapok változtatását, frissítését többnyire böngésző alapú, grafikus, felhasználóbarát felületen, weblapkészítési ismeretek nélkül is. Így jelentősen lerövidül a tartalom létrehozása és publikálása között eltelt idő.

Többféle tartalomkezelő rendszer létezik. Ezeknek egy része az internetről ingyenesen letölthető. Részletesebb bemutatásra én az egyik legnépszerűbbet [1], a Drupal nevű rendszer 4.6.3-as verzióját választottam, mely ingyenes és nyílt forráskódú. Használata elterjedt, Magyarországon is számos Drupal alapú weboldal létezik.

„A Drupal gyakorlatilag minden olyan rendszerre elérhető, ahol a PHP és valamely támogatott adatbázis rendszer működik. Nem csak Apache és Microsoft IIS alatt, hanem számos operációs rendszer alatt is futtatható, mint amilyen a Linux, a különböző BSD-k, a Solaris, a Windows vagy akár a Mac OS X platformok.

Bár a legtöbb Drupal alapú oldalt üzemeltető MySQL-t használ, ez nem mindenki számára kézenfekvő megoldás. A Drupal vékony adatbázis függetlenítő felülete segítségével PostgreSQL használata is lehetséges. Más adatbázisokhoz körülbelül egy tucat egyszerű függvény létrehozásával illeszthetjük oldalunkat.” [2]

3. Drupal felhasználói szintű bemutatása

3.1. Telepítés

A Drupal rendszert a <http://drupal.org> címről, a magyarítását a <http://drupal.hu> címről tölthetjük le. A telepítést, magyarítást és a konfigurálást a magyar Drupal oldalon található kézikönyv (<http://drupal.hu/kezikonyv>) útmutatása alapján tudjuk elvégezni. A Drupalt a webszerver dokumentumkönyvtárba kell feltelepítenünk.

3.2. Első lépések

Ha a rendszert telepítettük, és megfelelően konfiguráltuk, betölthetjük böngészőnkbe a Drupal kezdőlapját (1. ábra).

A Drupal kezdőlapja

1. ábra



Itt tudjuk regisztrálni magunkat, az első regisztráló lesz az adminisztrátor. Az adminisztrátor végezheti el a beállításokat, és küldhet be tartalmakat, melyek az oldalon megjelennek. Alapértelmezett állapotban a regisztrált látogatók olvashatják a tartalmakat, hozzászólásokat írhatnak és másokéra válaszolhatnak, míg a regisztrálatlan látogatók csak olvashatják a tartalmakat és hozzászólásokat. Ezeket a jogokat az adminisztrátor szűkítheti és bővítheti is, valamint újabb felhasználói csoportokat hozhat létre más jogosultságokkal.

3.3. Beállítások

3.3.1. Felhasználó

A regisztrált felhasználó a bejelentkezés után a menüből (2. ábra) megtekintheti, és módosíthatja a saját adatait. Kiválaszthatja az oldal nyelvét – ha engedélyezve van az alapértelmezettől eltérő nyelv –, megváltoztathatja felhasználónevét, jelszavát, új email címet adhat meg. Megadhat egy aláírást, mely minden hozzászólása alatt megjelenik, valamint beállíthatja a helyi időzónáját, és így az oldalon található időpontok ennek megfelelően jelennek meg.

Felhasználói menü 2. ábra



3.3.2. Adminisztrátor

Adminisztrátori menü 3. ábra



A rendszer adminisztrátora a bejelentkezés után egy bővebb menüt kap, mint a felhasználók (3. ábra). Az adminisztráció menüpont alatt egy listát kapunk a beállítható összetevőkről.

» **Beviteli formák:**

Tartalmak beküldésekor kiválaszthatjuk, hogy milyen jellegű a tartalom. Alap esetben három lehetőség közül választhatunk:

› „Filtered HTML”:

Egyszerű HTML kódot adhatunk itt meg, de a használható elemek korlátozva vannak. Ezekről eltérő elemeket a rendszer nem veszi figyelembe. Az engedélyezett elemek listáját az adminisztrátor bővítheti és szűkítheti. A rendszer automatikusan elhelyezi a sortöréseket a beírt szövegben (<p>,
). A felhasználók beküldött hozzászólásai ezt a beviteli formát használják.

› „PHP code”:

A beküldött tartalom PHP kódokat is tartalmazhat, ezek futtatását teszi lehetővé. Alapértelmezett esetben nincs benne HTML szűrés és automatikus sortörés. Biztonsági okokból célszerű csak az adminisztrátornak engedélyezni ezt a beviteli formát.

› „Full HTML”:

HTML elemeket használhatunk minden korlátozás nélkül. Alapértelmezetten itt is be van állítva az automatikus sortörés.

Ezekén kívül az adminisztrátor hozhat még létre beviteli formákat, ahol a három szűrőt (HTML szűrés, PHP kiértékelő, Sortörés-átalakító) tetszőlegesen kombinálhatja.

» **Beállítások:**

Itt el lehet végezni a webhely alapvető beállításait. Megadhatjuk a webhely nevét és esetleg a jelmondatát, melyek az oldal tetején jelennek meg hivatkozásként a főoldalra. Megadhatjuk a webhely hivatalos email címét, melyet a rendszer a felhasználóknak küldött levelekben használ. Leírhatjuk az oldal célját, és egy lábléc üzenetet, melyek az oldal tetején illetve legalján fognak megjelenni. Nevet adhatunk a

be nem lépett, névtelen felhasználóknak, ez alaphelyzetben „Anonymous”. Lehetőségünk van rövid webcímek használatára. Ebben az esetben a „?q=” rész nem jelenik meg az URL-ben. Ehhez szükség van a webszerver megfelelő beállítására is. Nem található, vagy nem hozzáférhető oldal kérése esetén saját hibajelentő oldalt használhatunk, ezek elérési útját itt megadhatjuk. Lehetőségünk van beállítani, hogy a hibaüzenetek csak a naplóba, vagy a képernyőre is kiíródjanak, valamint a naplóbejegyzések megtartásának időtartamát. Engedélyezhetjük az oldal gyorsítótár használatát. Ekkor a névtelen felhasználók által kért oldalakat egy adatbázisban tárolja a Drupal, így nem kell minden újabb kérésre előállítani, hanem csak az adatbázisból elővenni. Így jelentősen gyorsul a kiszolgálás, és csökken a szerver terhelése. Megadhatjuk annak a könyvtárnak a nevét, ahol a rendszer által kezelt fájlokat tároljuk és egy ideiglenes (temp) könyvtárat. A file letöltési módot privátra állíthatjuk, ha szabályozni akarjuk a jogosultságokat. Ezeken kívül még a dátumra, naptárra vonatkozó beállításokat tehetjük meg.

» **Blokkok:**

Az oldalon megjeleníteni kívánt blokkokat kiválaszthatjuk és beállíthatjuk. Ilyen blokkok például a belépés, navigáció vagy új felhasználók blokkok. A blokkok megjelenését engedélyezhetjük mindenhol, sehol, bizonyos oldalakon, vagy bizonyos jellegű oldalakon. Szabályozhatjuk, hogy a felhasználók testreszabhatják -e maguknak a blokkokat. A blokkoknak súly értéket is adhatunk, ezzel tudjuk szabályozni az egymáshoz viszonyított helyzetüket (a nagyobb súlyú blokkok lesüllyednek) [1-2 sz. melléklet].

» **Elérés szabályozás:**

Itt beállíthatjuk a regisztrált felhasználók és a látogatók jogosultságait, valamint újabb felhasználói csoportokat hozhatunk létre. Szabályokat vezethetünk be a regisztráció során megadandó felhasználónevekre és email címekre.

» **Felhasználók**

A regisztrált felhasználók adatait megtekinthetjük és módosíthatjuk, új felhasználókat adhatunk a rendszerhez. Itt sorolhatjuk a felhasználókat azokba a csoportokba, melyeket az elérés szabályozás oldalon létrehoztunk. Megadhatjuk a regisztrációs oldalon megjelenő „regisztrációs irányelvek” szöveget, beállíthatjuk az üdvözlő és új jelszót küldő email-ek tárgyát és szövegét.

» Hozzászólások

A felhasználók által beküldött hozzászólásokat megtekinthetjük, törölhetjük, vagy szerkeszthetjük. Módosíthatjuk a megjelenésüket, és beállíthatjuk a moderálási szabályokat.

» Kategóriák

A tartalmakat kategóriákba sorolhatjuk. Itt lehetőségünk van létrehozni kategóriákat.

» Modulok

A Drupal egy modulokból felépülő rendszer. A modulok használatát külön-külön engedélyezhetjük, vagy tilthatjuk.

» Napló

A rendszer által küldött üzenetek a naplóba íródnak, ezeket tekinthetjük meg ezen az oldalon. Lehetőségünk van kategóriánként megnézni őket.

» Nyelvek

A Drupal felületét másik nyelvre tudjuk állítani. Ehhez egy nyelvi állományt kell importálnunk, majd kiválasztani a listából az új nyelvet. Lehetőségünk van nyelvi fájlba exportálni is.

» Sminkek

A Drupal felületét teljes mértékben testre lehet szabni úgynevezett sminkekkel (theme). Ezek stíluslapokkal (CSS) változtatják meg az oldal kinézetét. A rendszer magában foglal négy sminket. Ezekhez tölthetünk le újabbakat az internetről, vagy készíthetünk magunknak is.

» Tartalom

Az eddig beküldött tartalmakat szerkeszthetjük. Lehetőségünk van a tartalmakat címlapra helyezni vagy eltávolítani, elrejteni, kiemelni, törölni. Korlátozhatjuk az egy oldalon megjelenő tartalmak számát.

3.4. Tartalom beküldése

A tartalmak beküldését adminisztrátorként bejelentkezve tudjuk elvégezni, a menüben a tartalom beküldése menüpontra kattintva. A tartalom típusát két lehetőség közül választhatjuk ki: oldal vagy írás. Bármelyiket is választjuk, folytatásként ugyanazt a felületet kapjuk. Megadhatjuk a tartalom szerzőjének nevét és a beküldés időpontját. Lehetőségünk van a hozzászólások írását és olvasását engedélyezni, vagy tiltani. Eldönthetjük, hogy a tartalmat közzé tesszük-e, a felhasználók csak ebben az esetben olvashatják azt. Ha közzétettük elhelyezhetjük a tartalom bevezetőjét a címlapon, ebben az esetben a tartalom címe linkként jelenik meg a tartalomra mutatva, alatta pedig egy bevezetőszöveg a tartalom törzséből, melynek a hosszát mi határozhatjuk meg. Lehetőségünk van a tartalmat a címlap tetejéhez „ragasztani”, ekkor az újabb tartalmak alatta jelennek meg (4. ábra). A törzs beviteli formáját a már fentebb tárgyalt három típus közül választhatjuk ki: szűrt HTML, teljes HTML vagy PHP kód.

„Ragadós” tartalom

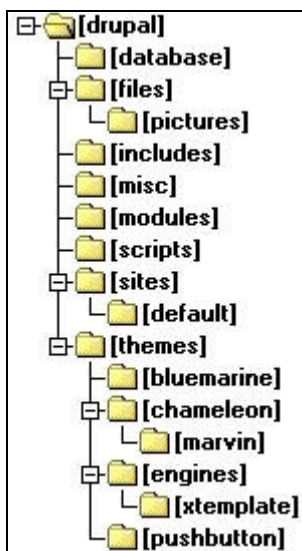
4. ábra



4. Fájl és adatbázis szerkezet felvázolása

4.1. A Drupal fájl szerkezete

Könyvtárszerkezet 5. ábra



4.1.1. Database

Ha belépünk a Drupal könyvtárba (5. ábra), az első alkönyvtár a *database*. Az első két file a *database.mysql* és a *database.pgsql*. Ezek a Drupal rendszer adatbázisának a dumpjai MySQL és PostgreSQL rendszerekhez. A harmadik file az *updates.inc*. Ez egy include állomány, mely az *update.php* állományba van beillesztve. Ez a file tartalmazza a Drupal rendszer újabb verzióra történő frissítésekor szükséges, adatbázis műveleteket végrehajtó PHP függvényeket.

4.1.2. Files

A rendszer által kezelt file-ok alapértelmezett tárolási helye. Ide kerülnek a különböző smink logóképei, a feltöltött egyéb állományok, és ha publikus letöltési módot választottunk a beállításoknál, akkor itt tartjuk a letölthető állományokat. A *pictures* alkönyvtárba kerülnek – ha engedélyeztük a használatát – a felhasználók képei, melyeket a smink megjeleníthetnek a beküldött hozzászólások, tartalmak mellett.

4.1.3. Includes

Ebben a könyvtárban tárolódnak az include állományok.

» **Bootstrap.inc:**

Az *index.php*, *cron.php* és az *xmlrpc.php* állományokba van beillesztve. Azokat a függvényeket tartalmazza, melyekre szükség van minden, a Drupalhoz érkezett kérés esetén.

» **Common.inc:**

Szintén az *index.php*, *cron.php* és *xmlrpc.php* állományokba van illesztve, olyan közös függvényeket definiál, melyeket sok modul használ. Azok a függvények, melyeknek rendelkezésre kell állni éppen akkor, mikor egy cache-elt oldalt szolgáltat a rendszer, a *bootstrap.inc* file-ban találhatók.

» **Database.inc**

A *bootstrap.inc* állományba van beillesztve. Az adatbázisszerver típusától független, adatbázis-műveleteket végrehajtó függvények definícióját tartalmazza. A MySQL vagy PostgreSQL specifikus adatbázis-kezelő függvények include állományból kerülnek beillesztésre. Az adatbázis szervernek megfelelően a *database.mysql.inc*, vagy a *database.pgsql.inc*.

» **File.inc**

A *common.inc* állományba van beillesztve. Ez tartalmazza a fájlkezeléshez szükséges függvények és konstansok definícióját.

» **Image.inc**

Szintén a *common.inc* állományba van beillesztve. Képek kezelését végző függvények definícióját tartalmazza.

» **Locale.inc**

A *locale.module* állományba van beillesztve. Különböző nyelvi file-ok kezelését, és a nyelvek kezelését lehetővé tévő oldalak – importálás, exportálás, új nyelv hozzáadása – létrehozását végző függvények találhatók benne.

» **Menu.inc**

A *common.inc* állományba van beillesztve. A Drupal menüjének működtetését végző függvények, és a menü állapotáról információt tároló konstansok (flag) definícióját tartalmazza.

» **Module.inc**

A *bootstrap.inc* állományba van illesztve. A Drupal moduljainak vizsgálatát, használatát végző függvények definícióit tartalmazza.

» **Pager.inc**

A *common.inc* állományba van beillesztve. Olyan függvények definícióit tartalmazza, melyek az adatbázis-műveletek eredményeit egymással kapcsolatban álló oldalak csoportjaként jelenítik meg (6. ábra).

Egymásra mutató táblák

6. ábra

felhasználó	2005-10-28 11:52	„admin” felhasználó bejelentkezett.	admin	részletek
felhasználó	2005-10-28 11:52	„admin” felhasználó kijelentkezett.	admin	részletek
első oldal előző oldal 1 2 3 következő oldal utolsó oldal				

» **Session.inc**

A *bootstrap.inc* állományba van beillesztve. A felhasználói munkamenetek (session) kezelését végző függvények definícióinak gyűjteménye.

» **Tablesort.inc**

A *common.inc* állományba van beillesztve. Olyan függvények definícióit tartalmazza, melyek segítik, hogy a felhasználó a megjelenített táblázatokat különböző oszlopok szerint rendezhesse sorba (7. ábra). Ehhez csak a táblázat címsorában a megfelelő helyre kell kattintani az egérrel.

Rendezett táblázat

7. ábra

ID	Felhasználói név	Állapot	Csoportok	Utolsó aktivitás	Műveletek
1	admin	aktív	authenticated user	2005-11-14 18:28	szerkesztés
3	szb	aktív	authenticated user	2005-11-14 18:28	szerkesztés
2	szbalint	aktív	authenticated user	2005-10-30 15:22	szerkesztés

ID	Felhasználói név	Állapot	Csoportok	Utolsó aktivitás	Műveletek
1	admin	aktív	authenticated user	2005-11-14 18:28	szerkesztés
2	szbalint	aktív	authenticated user	2005-10-30 15:22	szerkesztés
3	szb	aktív	authenticated user	2005-11-14 18:28	szerkesztés

» Theme.inc

A *common.inc* állományba van beillesztve. A sminkek (theme), ezen keresztül a Drupal megjelenésének, képernyőkimenetének a kezelését végző függvények definícióit tartalmazza.

» Xmlrpc.inc, xmlrpcs.inc

Az RPC protokoll távoli kiszolgálókon található függvények meghívását teszi lehetővé. [3] Ennek egy fajtája az XML-RPC, melyet a Drupal is használ. Az *xmlrpc.inc* állomány az xml-rpc kérésekhez, az *xmlrpcs.inc* a kérések kiszolgálásához szükséges függvényeket tartalmazza.

4.1.4. Misc

Képek és stíluslap állományok találhatók ebben a könyvtárban. Ezek a Drupal alapértelmezett megjelenéséhez szükségesek. A sminkekkel ezeket tudjuk felülbírálni.

4.1.5. Modules

A Drupal rendszer moduláris felépítésű. Minden funkciót egy önálló modul lát el, amely – ha nem alapfunkció – kikapcsolható vagy bekapcsolható, így a

rendszer teljesen testreszabható. Ezeknek a moduloknak a forrásállományai találhatóak ebben a könyvtárban.

Az egyes modulok funkciójának leírása a Drupal menü adminisztráció/modulok oldalán megtalálható.

4.1.6. Scripts

Az ebben a könyvtárban található állományok minták Perl script-ek és Linux shellscript-ek írásához.

4.1.7. Sites/default

Ebben a könyvtárban található a *settings.php* állomány. Ez a Drupal alapértelmezett, honlapunkra jellemző beállításokat tartalmazó konfigurációs állománya. Ebben az állományban adjuk meg az adatbázis szerver URL-jét (\$db_url), és lehetőségünk van beállítani egy előtagot az adatbázistábla-nevek elé (\$db_prefix). Meg kell adni a honlapunk kezdőlapjának URL-jét (\$base_url).

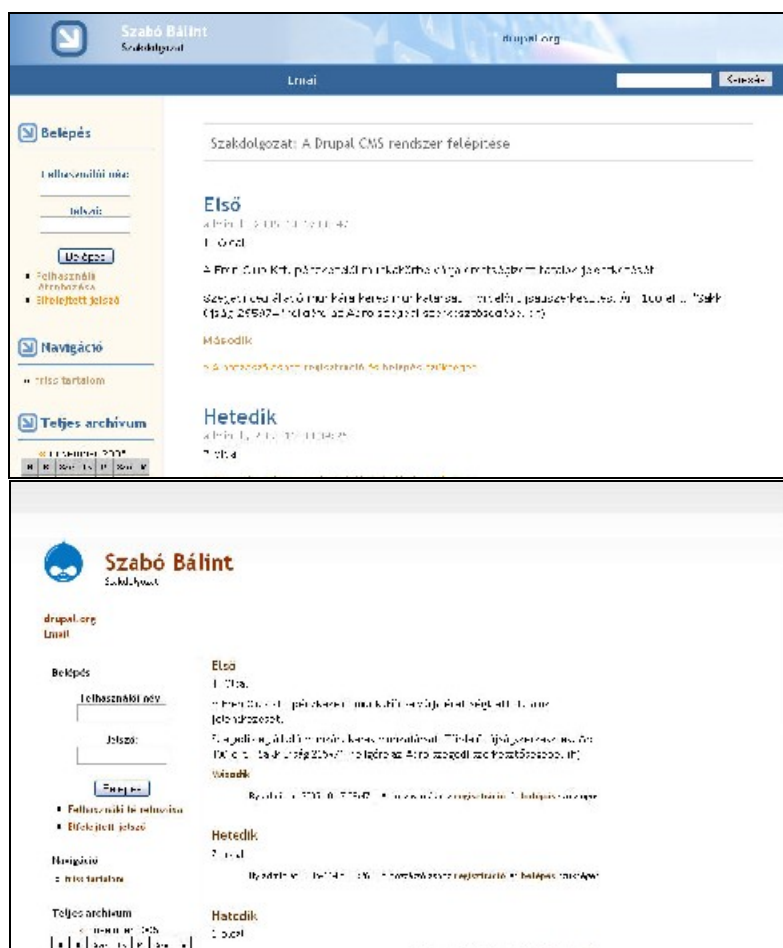
Bizonyos PHP beállításokat is eszközölhetünk. Lehetőségünk van bizonyos függvények viselkedésének futásidejű befolyásolására az *ini_set()* függvény segítségével.

4.1.8. Themes

Sminkekkel (theme) a Drupal megjelenése teljesen megváltoztatható (8. ábra). A sminkek könyvtárai ebben a könyvtárban találhatóak. A sminkek több állományból állnak. Az oldal megjelenését egy vagy több stíluslap (CSS) határozza meg. A sminkek lehetnek önállóak, ekkor egy *.theme* kiterjesztésű PHP állomány kezeli a megjelenítést. Lehetőségünk van a Drupal Xtemplate sablon kezelője segítségével, egy HTML állománnyal kezelni a kimenetet. Ekkor a fájl kiterjesztése *.xtmpl*. Ez a sablon a *themes/engines/xtemplate/* könyvtárban található, az *xtemplate.engine* állomány valósítja meg a kapcsolatot az xtemplate állományok, és a Drupal sminkkezelő rendszere (theme system) között.

Különböző smink

8. ábra



4.2. A Drupal adatbázis szerkezete

A Drupal 4.6.3. rendszer adatbázisa 55 adatbázis táblából áll. Ezek közül néhány fontosabbat az alábbiakban részletesebben is ismertetek.

4.2.1. Blocks

A Drupal bizonyos funkciói külön blokkban jelennek meg a felhasználói felületen. Ebben a táblában ezeknek a blokkoknak a tulajdonságai tárolódnak. Például a blokk státusza (engedélyezett-e?), a súlya, mely alapján sorba rendezi a rendszer a megjelenített blokkokat, és a terület azonosítója, ahol megjelenik az adott blokk az oldalon belül.

4.2.2. Cache

A rendszer a megjelenített összetevőket gyorsítótárazza, így azok megjelenítése kevesebb időt vesz igénybe. Ez a tábla szolgál gyorsítótárként.

4.2.3. Comments

A felhasználók által beküldött hozzászólások tárolódnak ebben a táblában. A hozzászólások azonosítót kapnak (cid), ez lesz az elsődleges kulcs. Tárolódik, hogy melyik tartalomhoz kapcsolódik (nid), ki a szerzője (uid), honnan érkezett (hostname) és mikor (timestamp), valamint az olvasóktól kapott pontszámok.

4.2.4. Files

A rendszer által kezelt állományokról tartalmaz információkat. Azonosítót rendel hozzájuk (fid), és tartalmazza, hogy melyik tartalomhoz kapcsolódnak (nid), a fájl nevét, elérési útját, típusát (filemime) és a méretét.

4.2.5. Node

A beküldött tartalmakat tárolja. A tábla elsődleges kulcsa a tartalom azonosítója, a nid mező (node id.), melyet több tábla is használ (pl.: comment, files). Tárolódik a tartalmak típusa (oldal vagy írás), a cím, a beküldő azonosítója (uid), az, hogy engedélyezett-e a megtekintése (status), a létrehozás és az utolsó változtatás dátuma. Ezenkívül engedélyezett-e a hozzászólások írása-olvasása (0-minden tiltott, 1-csak olvasható, 2-írható/olvasható), moderálható-e, milyen szűrőt alkalmaztak rá (format: 1-szűrt HTML, 2-PHP kód, 3-teljes HTML), címlapon megtalálható-e (promote=0/1), valamint a címlap tetején van-e (sticky=0/1). A teaser mezőben a bevezető, a body mezőben a hozzászólás törzse tárolódik.

4.2.6. Role

A felhasználói csoportokat tartalmazza, melyekhez azonosítót rendel. Alap esetben csak két érték van benne, anonymous user (rid=1) és authenticated user (rid=2), de az adminisztrátor bővítheti a listát.

4.2.7. Search_index

A rendszer indexeli a tartalmakat a keresés meggyorsításához, ezek az adatok itt tárolódnak.

4.2.8. System

A rendszer fő összetevőinek (modulok és sminkek) elérési útja és rövid leírása található ebben a táblában. A modulok és a rövid leírások a Drupal adminisztráció/modulok oldalán jelennek meg.

4.2.9. Users

A tábla elsődleges kulcsa a felhasználó azonosító, az uid mező (user id.), melyre több tábla is hivatkozik (comments, node, users_roles). A rendszer felhasználóiról tartalmaz információkat, például a felhasználó jelszavát kódolt formában, az e-mail címét, a hozzászólások alatt megjelenő aláírását, az oldalon használt nyelvet, az alapértelmezett időzónát. Ha a felhasználó töltött fel magáról képet, akkor ennek az elérési útja is megtalálható, valamint a felhasználó létrehozásának és adatai utolsó módosításának dátuma időbélyeg formában.

4.2.10. Users_roles

A regisztrált felhasználókhoz hozzárendel egy csoportazonosító számot (rid) a role táblában meghatározottak közül, annak megfelelően, hogy a felhasználó melyik csoportba tartozik.

4.2.11. Variable

A rendszerről információt szolgáltató változók nevét és értékét tartalmazza. Például az utolsó frissítés dátuma (update_start), alapértelmezett smink (theme_default), az oldal neve (site_name), az oldal e-mail címe (site_mail), alapértelmezett időzóna (date_default_timezone), stb...

4.2.12. Watchdog

A Watchdog a Drupal naplózó rendszere, ebbe a táblába írja a rendszerüzeneteket. Rögzíti az üzenet szövegét, típusát, az üzenethez tartozó

felhasználó azonosítóját (uid) és a host nevét, az üzenet súlyosságát, az oldalon belüli helyet, ahol az üzenet generálódott, valamint a létrejöttének idejét időbélyeg formában.

5. Függvények feltérképezése

A Drupal 4.6.3.-as verziója PHP4-ben íródott. Ebben a fejezetben a rendszer működéséhez legszükségesebb függvényeket fogom kiemelni. Az alap függvények definíciói a 4.1.3. fejezetben tárgyalt include állományokban találhatók. Ezeken kívül megemlítek néhányat a fontosabb modulok függvényei közül. Ezek definíciói a *modules* könyvtárban, *.module* kiterjesztésű állományokban találhatók.

5.1. Felhasználókat kezelő függvények

Ezek a függvények a *modules/user.module* állományban vannak definiálva. A Drupal regisztrált felhasználóinak adatait kezelik, a *users*, *roles* és *users_roles* adatbázis táblákkal dolgoznak.

5.1.1. User_load

Paraméterként egy tömböt kap, melyben a felhasználó bizonyos tulajdonságai vannak. Ezek alapján megkeresi a felhasználót az adatbázisban, és onnan az értékeket a *\$user* objektumba tölti. Ez az objektum a függvény visszatérési értéke.

5.1.2. User_save

Paraméterként a *\$user* (a *user_load* függvény visszatérési értéke) objektumot kapja. A felhasználói adatok módosítását vagy létrehozását végzi el. A felhasználót az adatbázis táblában, az objektumban található *uid* (user id) alapján keresi meg, ha az nem nulla. Ekkor elvégzi a változtatásokat, majd újratölti az adatokat a *user_load* függvény meghívásával. Ha az *uid* nulla, akkor létrehozza az új felhasználót, utána hasonlóképpen jár el mint az előzőekben. A függvény visszatérési értéke a *\$user* objektum.

5.2. Tartalmakat kezelő függvények

A `modules/node.module` állományban vannak definiálva. A `node` adatbázis táblát kezelik.

5.2.1. `Node_load`

A tartalmat a paraméterként kapott `$conditions` tömb alapján keresi meg, mely a tartalom jellemzőit tartalmazza. Általában csak a `nid` (node id) tulajdonság található benne. A tartalom adatait egy `$node` objektumba töltik be, ez lesz a függvény visszatérési értéke.

5.2.2. `Node_save`

Paraméterként a `$node` objektumot kapja. Ez alapján megállapítja, hogy létezik-e már a tartalom. Ha nem, akkor az aktuális időpontot hozzáadja a `created` és `changed` mezőkhöz, és az adatbázisba menti a tartalom adatait az objektumból. Ha létezik már a tartalom az adatbázisban, akkor csak a `changed` mezőt írja felül, és frissíti az adatbázis táblát az új adatokkal. Visszatérési értéke a tartalom azonosítója, a `nid` mező értéke.

5.3. Adatbázis kezelő függvények

5.3.1. `Db_set_active`

Aktivál egy adatbázist a későbbi kérések számára. Megállapítja az adatbázis típusát, és ez alapján beilleszti a típusnak megfelelő `include` állományt (`includes/database.$db_type.inc`), ha létezik. Ha nem létezik, akkor hibaüzenetet küld. Visszatérési értéke az aktív adatbázis URL-je.

5.3.2. `Db_connect`

Létrehozza a kapcsolatot egy adatbázissal. A függvény lehetővé teszi, hogy ne a szokásos MySQL portot használjuk. Paraméterként az adatbázis URL-jét kapja, amit a PHP `parse_url` függvényének segítségével egy tömbbé alakít. Ha az URL-ben a portszám is meg volt adva, azt szöveggént, egy kettősponttal elválasztva hozzáfűzi az

URL-hez. Ezután létrehozza a kapcsolatot a MySQL szerverrel (`mysql_connect`), és kiválasztja az adatbázist (`mysql_select_db`). A visszatérési értéke, a `mysql_connect` által visszaadott kapcsolatazonosító.

5.4. Megjelenítést kezelő függvények

A Drupal képernyőkimenetét a sminkrendszer határozza meg. Minden megjelenítésre kerülő részletet a sminknek megfelelően alakít át. A függvények definíciói az `includes/theme.inc` állományban találhatók.

5.4.1. `Init_theme`

Ez a függvény inicializálja a sminkrendszert. Paramétere nincs. Megvizsgálja, hogy a kiválasztott smink rajta van-e az engedélyezett sminkek listáján. Ha nincs, akkor az alapértelmezett „Bluemarine” sminket állítja be. Ha a kiválasztott smink egy önálló smink, akkor beilleszti, ha pedig PHP- vagy Xtemplate smink, akkor a template motort illeszti be. Megkeresi a sminkhez használt stíluslapot, és azt is betölti. Visszatérési értéke az aktuális smink neve.

5.4.2. `Theme`

Ez a függvény végzi a Drupal megjelenítendő összetevőinek átalakítását a sminknek megfelelően. Paramétere nincs. Ha még nincs inicializálva a smink, akkor meghívja az `init_theme` függvényt. Ezután, ha a smink egy önálló smink, akkor annak a függvényeit, ha valamelyik template-et használja, akkor a template motor függvényeit, ha pedig egyik sem, akkor a Drupal függvényeit hívja meg. Visszatérési értéke a meghívott függvények által generált HTML kód.

6. Új smink készítése

A Drupal felületét egy sminkkel teljesen megváltoztathatjuk. Smink írására többféle lehetőségünk is van. Írhatunk teljesen önálló sminket, ekkor egy PHP állományt kell készítenünk *.theme* kiterjesztéssel. Használhatjuk azonban a Drupal valamelyik template motorját is. Ebben a dolgozatban a legegyszerűbb módszerre nézünk egy példát, XTemplate segítségével hozunk létre egy új sminket. Ehhez csak a HTML/XHTML nyelvet kell ismerni.

Első lépésként létre kell hoznunk egy alkönyvtárat a *themes* könyvtárban. Ennek a könyvtárnak a neve lesz az új smink neve, ez fog megjelenni a Drupal adminisztrációs felületén. Ebben a könyvtárban kötelezően csak egy állományt kell elhelyezni, egy *xtemplate.xtpl* nevű, HTML/XHTML állományt. Ezeknek a nyelveknek az elemeit használja, illetve néhány speciális XTemplate tag is található benne. Ezenkívül opcionálisan elhelyezhetünk még a könyvtárban stíluslapokat, kép vagy JavaScript állományokat. Ha stíluslapot is elhelyezünk a könyvtárban, annak a neve legyen *style.css*, ekkor a Drupal automatikusan alkalmazni fogja, és nem kell nekünk beillesztenünk az XTemplate állományunkba. Ha a sminkünk könyvtárán belül létrehozunk egy újabb könyvtárat, és ebben is elhelyezünk egy *style.css* nevű állományt, akkor az alkönyvtár nevével egy újabb sminket kapunk. Ez ugyanazt az XTemplate állományt fogja használni, mint az előző, csak az új stílussal.

Az XTemplate állományokban két különleges elemmel bővülnek a HTML/XHTML elemek. Az egyik a területet jelöli ki, ahol a megadott összetevő meg fog jelenni. Ezek között fognak elhelyezkedni az adott elemhez tartozó HTML elemek.

Pl.:

```
<!-- BEGIN: title -->
```

```
<!-- END: title -->
```

A másik az adott összetevő konkrét helyét jelöli a HTML elemeken belül. Ezeknek a formája:

Pl.:

```
{title}
```

Ezeknek megfelelően a teljes szakasz így jelenik meg:

```
<!-- BEGIN: title -->
.
.
<h1 class="title">{title}</h1>
.
.
<!-- END: title -->
```

A kipontozott területre egyéb HTML elemek kerülhetnek, vagy más összetevők ehhez hasonló keretben, mivel ezek egymásba ágyazhatók.

Pl.:

```
<!-- BEGIN: node -->

  <!-- BEGIN: title -->

  <!-- END: title -->

<!-- END: node -->
```

Legegyszerűbben úgy készíthetünk saját sminket, ha egy már létező sminket veszünk alapul, és azt kezdjük módosítgatni. Én a Drupal 4.6.3-as verziójában is megtalálható Pushbutton nevű sminket választottam ki átalakításra.

7. Összefoglalás

Szakdolgozatom célja az volt, hogy általános képet adjon a tartalomkezelő rendszerekről azoknak, akik weblapkészítési, webprogramozói ismeretekkel rendelkeznek, de a CMS-ek témájában járatlanok. Ehhez a Drupal jó eszköznek bizonyult, megfelel azoknak az elvárásoknak, melyeket egy tartalomkezelővel szemben támaszthatunk. Használata egyszerűen tanulható, felépítése logikus. Úgy gondolom, hogy sikerült egy olyan dolgozatot írnom, amely megfelelő segítség lehet a tartalomkezelő rendszerekkel és a Drupallal ismerkedők számára.

A dolgozat írása közben magam is sok új ismerettel gazdagodtam. Elsősorban a Drupalról és a tartalomkezelőkről tanultam sokat, de a függvények vizsgálata és a smink írása által a PHP, HTML és CSS nyelvek terén is szereztem tapasztalatokat.

8. Irodalomjegyzék

- [1]. OpenSourceCMS
http://opensourcecms.com/index.php?option=com_reporter&queryId=32&menuGroup=OSCMS
- [2]. Lehetőségek, szolgáltatások | drupal.hu
<http://drupal.hu/kezikonyv/lehetosegek>
- [3]. Együttműködés távoli szolgáltatásokkal | Weblabor
<http://weblabor.hu/cikkek/xmlrpc>
- [4]. <http://cms.lap.hu>
- [5]. Tartalomkezelő rendszerek, tartalom menedzsment
http://www.fixaidea.hu/tartalomkezelo_rendszerek_cms_tartalom_menedzsment
- [6]. Netrix CMS
<http://www.netrix.hu/cms/>
- [7]. Netrix CMS
http://www.netrix.hu/cms/cms_reszletesen.php
- [8]. ACT International - DirACTor 2.0 tartalomkezelő rendszer (CMS)
<http://www.act-int.hu/megoldasaink/interaktiv/cms.php>
- [9]. Content management system - Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/Content_Management_System
- [10]. drupal.org | Community plumbing
<http://drupal.org>
- [11]. drupal.hu | A Drupal tartalomkezelő magyar honlapja
<http://drupal.hu>
- [12]. functions | drupaldocs.org
<http://drupaldocs.org/api/4.6/function>
- [13]. Blake Schwendiman – PHP 4 Referenciakönyv - 2. kötet
Panem Kiadó, Budapest 2002

[14]. Theme developer's guide

<http://drupal.org/book/export/html/509>

[15]. Rimár Miklós - Honlapok formázási lehetőségeinek bővítése a CSS nyelv segítségével – Szakdolgozat

Szegedi Tudományegyetem Bölcsészettudományi Kar - Szeged

[16]. [H T M L i n f o v2.0] - online agytáp webmestereknek és műkedvelőknek

http://htmlinfo.polyhistor.hu/html_ref/main.htm