

TDK dolgozat

Nemes András

Eszterházy Károly Főiskola
Tanárképzési és Tudástechnológiai Kar
Mozgóképkultúra Tanszék

Arculatot mindenkinek

generálj.hu

Készítette: Nemes András, III. évfolyam

Konzulens: Kvaszingerné Prantner Csilla, tanársegéd

2014

Tartalomjegyzék

1 fejezet – Elősző.....	1. oldal
2. fejezet – A generálj!hu bemutatása.....	2. oldal
2.1. Milyen eszközöket használok a projekt során?.....	2. oldal
2.1.2. A fontosabb alkalmazások bemutatása.....	3. oldal
3. fejezet – A generálj!hu felhasználói szemszögből.....	5. oldal
3.1. Menürendszer.....	6. oldal
3.1.1. Címlap.....	6. oldal
3.1.2. Sablonok.....	7. oldal
3.1.3. Gépház.....	7. oldal
3.1.4. Kisokos.....	7. oldal
3.1.5. Betűtípusok.....	7. oldal
3.2. Felhasználói funkciók.....	8. oldal
3.2.1. Logó generálás.....	8. oldal
3.2.2. Weboldal sablonok.....	9. oldal
3.2.3. Névjegykártya készítés.....	11. oldal
3.2.4. Közösségi média.....	13. oldal
3.2.5. Kereső modul.....	13. oldal
4. fejezet – A generálj!hu programozói szemszögből.....	13. oldal
4.1. Az oldal szerkezeti felépítése.....	14. oldal
4.2. Biztonsági megoldások.....	14. oldal
4.3. Fontosabb függvények, szkriptek.....	15. oldal
4.3.1. MySQL csatlakozás.....	15. oldal
4.3.2. Tartalom kezelés.....	15. oldal
4.4. Sablonok felépítése.....	16. oldal
4.4.1. Logó generálás.....	16. oldal
4.4.2. Weboldal sablonok felépítése.....	16. oldal
5. fejezet – Összegzés.....	17. oldal
6. fejezet – Felhasznált külső források.....	18. oldal
7. fejezet - Melléklet.....	19. oldal
7.1. A "fuggvenyek.php" forráskódja.....	19. oldal

1. Előszó

Manapság, ha az ember szeretne igénybe venni valamilyen egyszerűbb szolgáltatást, például valamilyen szerelést, vagy egyszerűbb háztartás körüli dolgot, sokszor kis-, vagy közepes cégeket kér fel az adott munka elvégzésére.

Ezeknek a cégeknek vagy nincs arculatuk, vagy, ha van, akkor is nem jól átgondoltak, illetve nem szépen kivitelezettek. Sok esetben nincs névjegykártya a céget képviselő személyeknek, vagy nincs internetes oldala a cégnek. Tehát sajnos, nem adnak magukra a külvilág felé. Ilyenkor adódik a kérdés, hogy, ha egy cég ennyire nem ad a külsőségekre, nem ad magára, akkor milyen lehet a munka, amit végez? Persze egy jó arculattól még nem lesz minden tökéletes, de a cég, vagy egyéni vállalkozó, már sokkal jobban eladható lesz, ha van egy átgondolt arculata a célközönség felé.

Saját tapasztalataim alapján, ennek az "arculatnélküliségnek" két oka van alapvetően. Az egyik az, hogy sok kis cég még abban a hitben él, hogy majd a jó munka eladja magát. A másik ok pedig a pénz hiány. Az a cég, amelyik pénz hiány miatt nem épít maga köré megfelelő arculatot, alapvetően azért teszi, mert nincs tisztában a mostanság elterjedt nyílt forrás kódú, ingyenes eszközök használhatóságával. Abban a hitben élnek, hogy egy weboldal üzemeltetése "úri hóbot", az sokba kerül és amúgy sem biztos, hogy megtalálják. Ingyen persze nem mindent lehet megcsinálni, de, ha valaki veszi a fáradságot és kitalál pár jó domain nevet, ami az ő cégére, vagy az ő tevékenységére utal, akkor már csinál magának, vagy a cégének egy elég masszív alapot, már ami a megjelenést illeti. Egy ".hu"-s domain költsége kb. bruttó 3500,- Ft. három évre. Ehhez még hozzájön egy jó tárhely, aminek az éves költsége bruttó 10.000 - 15.000,- Ft. között mozog. Tehát, ha csinál magának valaki például három "domain"-t, akkor az éves költsége legrosszabb esetben is kb. bruttó 16.200,- Ft. Hogy egy élő példát nézzünk, ez az ár egyenlő egy új bojler beüzemelésének az árával. Magyarán éves szinten egy kisvállalkozó, vagy egy pár fős vállalat is képes egy munkával kitermelni a weboldal költségét, ami, ha megfelelően van elkészítve képes több munkát is hozni a tulajdonos számára.

Nyilván egy weboldal önmagában nem tesz csodát. Egy weboldal is csak az arculat egyik eleme.

Még mielőtt azonban valaki belevágna egy weboldal építésébe erősen ajánlott az alapvető grafikai elemek megtervezése. Gondolok itt elsősorban a logóra, illetve a színvilágra. Ezen

a ponton adódik a kérdés, hogy jó-jó, de mivel? Az egyik irány az, hogy fogja a cég, vagy vállalkozó és megtervezteti egy profi szakemberrel, viszont ez vélhetően magas költségekkel jár. A másik út, hogy megpróbálja saját maga elkészíteni ezeket.

Az én szakmám alapvetően fényképész. Amikor az osztálytársaimmal a szakiskolában másodéves végzős diákok voltunk alapvető problémát jelentett, hogy hol tudunk magunknak publicitást szerezni. Mik azok a felületek, ahol tudjuk magunkat reklámozni. A dolgozatom szempontjából ez azért érdekes, mert én személyesen is beleestem abba a problémába, amelybe beleesik, beleesett több kis-, vagy közép vállalkozás is. Én ekkor elkezdtem foglalkozni a nyílt, ingyenes alkalmazások felkutatásával.

Így ismerkedtem meg több nyílt forráskódú oldalmotorral, illetve szerkesztő programmal. Emiatt is fogalmazódott meg bennem a gondolat hozzávetőlegesen egy évvel ezelőtt, hogy kéne csinálni egy olyan alkalmazást, amely bárki számára elérhető, illetve bárki számára felhasználható is saját arculati elemeinek (logó, névjegykártya, weboldal). kialakítására.

2. A „generálj.hu” bemutatása

Az oldal alapfilozófiája az, hogy olyan emberek számára legyen egy eszköz, akiknek nincsen hasonló területen előképzettségük, vagy, ha van, akkor is csak minimális.

Az oldal egy olyan alkalmazás elsősorban, amely segít azoknak a cégeknek kialakítani egy arculatot, akiknek nincs módjuk magasabb összeget kifizetni profi szakemberek felé.

Kifejezetten elvárás az, hogy az egész projektnek legyen egy olyan üzenete, hogy az ember képes minimális befektetéssel is értékes dolgokat előteremteni. Cél az, hogy felhívja ez a projekt a figyelmet a nyílt forráskódú, bárki számára elérhető eszközök használatára.

2.1. Milyen alkalmazásokat használok a projekt során?

A készítés során törekedtem a nyílt eszközök használatára. Ez alól kivételt az operációs rendszer, illetve az Adobe Photoshop Elements használata jelentett.

Bár próbálkoztam ezek kiváltásával, de sajnos nem találtam olyan eszközt amely ezeket minden szempontból megfelelően helyettesítené. (A két tesztalany a Linux és a Gimp volt, de például a Linux esetében több verziót kipróbálva sem jutottam eredményre.)

Az említett két programon kívül a következő eszközöket használtam:

- Notepad++
- WAMP szerver a teszteléshez
- jQuery alapú bővítmények
 - Colorpicker
 - Camera slider
- FPDF Library PDF generator

Alapvetően az oldal PHP, mysql, html, css és jQuery alapokra épül. Azért választottam ezeket, mert egyfelől mindegyik egyszerűen tanulható és magas fokú programozói tudás nélkül is használható, másfelől pedig igazán más lehetőség nem volt. Bizonyos felületeken használhattam volna flash-t is, de meglátásom szerint ettől az oldal használhatósága csak csökkent volna. Elég, ha arra gondolunk, hogy a flash android-os eszközökön nem futtatható.

2.1.2. A fontosabb felhasznált alkalmazások bemutatása

Notepad++

A Notepad++ egy olyan alkalmazás, amellyel képesek vagyunk többféle programnyelvben is dolgozni. Sokan használják HTML szerkesztésre is, mint például én is a sablon, illetve az oldalmotor kialakítása során. Előnye, hogy rendkívül sok kiegészítő van hozzá, valamint az is, hogy ingyenesen frissíthető. Az egyik legfontosabb kiegészítője számomra az NPPftp volt, amely segítségével a szerveren lévő fájlokat is tudom szerkeszteni.

Használható text, C C++ Java C# XML HTML, PHP CSS makefile ASCII art (.nfo) doxygen ini file, batch file Javascript ASP VB/VBS SQL Objective-C, RC resource file Pascal Perl Python Lua TeX adatokhoz. (Origo Szoftverbázis/2011)

WAMP Server

A fejlesztés során a WAMP Server-t használtam, mint fejlesztői környezet.

Ennek a programnak a segítségével a saját gépemem, helyben is tudok PHP-t, illetve MySQL-t futtatni.

Az azonban elég hamar kiderült, hogy mekkora eltérések lehetnek szerver és szerver között is. Emiatt sajnos magán a külső szerveren is elég sokat kellett változtatnom az oldal motoron, sőt magán az adatbázis szerkezeten is.

PHP

A PHP egy olyan szerver oldali szkriptnyelv, amely képes fájl feldolgozásra, illetve HTML alapú oldalak generálására is többek között. Fontos tulajdonsága még, hogy a PHP képes adatbázis kezelésre is, erre általában MySQL-t szoktak használni, mint például én is. A PHP alapvetően nem objektum orientált, magyarul a kód lefutása sorról sorra zajlik. Születésekor csupán egy makrókészlet volt, amely személyes honlapok karbantartására készült. Innen ered a neve is: Personal Home Page Tools. (Wikipedia/2004).

A 7.1. A "függvények.php" forráskódja c. fejezetben mellékeltem a forráskódját a függvény gyűjteménynek, ami a lelke az oldalmotornak.

jQuery

A jQuery egy olyan nyílt forráskódú Javascript felhasználó oldali keretrendszer, amelynek segítségével különféle módon összekapcsolhatjuk a HTML-t a „Javascript”-tel. Például különféle „slider”-eket, galériákat hozhatunk vele létre, illetve képes különféle DOM manipulációkat is végrehajtani.

A jQuery 2006-ban jelent meg, megalkotója John Resig.

FPDF Library PDF generator

Mint az a nevéből is kiderült ez egy PDF generáló. Maga az alkalmazás egy PHP osztály, amit rendkívül egyszerűen, bonyolultabb szerver oldali telepítés nélkül is lehet használni.

Ezt az alkalmazást főként a névjegykártya generáláshoz fogom használni, de a későbbiekben is nagyon hasznos lehet más sablonokhoz is.

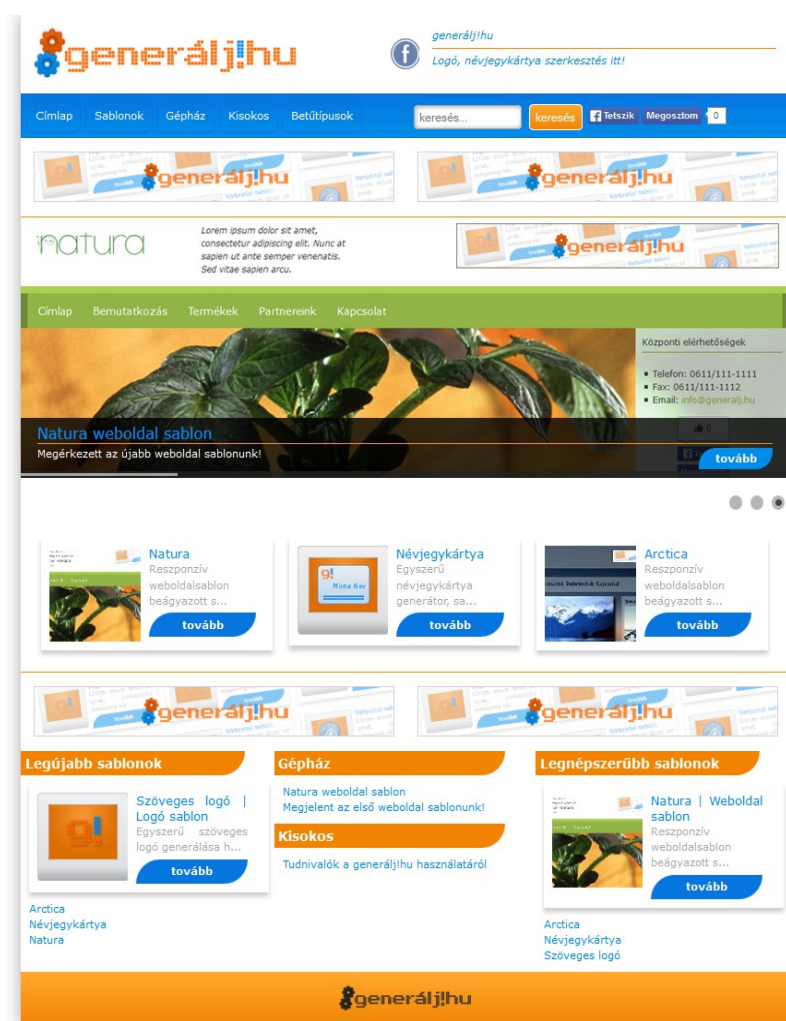
Legfontosabb tulajdonságai, hogy képesek vagyunk vele képeket, illetve szövegeket is beilleszteni közvetlenül egy HTML/PHP alapú űrlapból, valamint képes nyomdai

felhasználásra is alkalmas PDF generálására, erre egy kiegészítőt kell használni:
<http://fpdf.de/downloads/addons/44/>.

Az alkalmazás képes True Type fontok kezelésére is. Ez azért kiemelten fontos, mert így képesek vagyunk az oldalhoz felhasznált bármilyen True Type fontot beépíteni az alkalmazásba. Ezt később egy adminisztrációs felületen lehet megtenni, közvetlenül a betűtípus feltöltésekor, jelenleg azonban ezt használom a fontok előkészítéséhez:
<http://www.fpdf.org/makefont/>.

3. A generálj.hu felhasználói szemszögből

Mint azt már az oldal bemutatásánál mondtam próbáltam minél egyszerűbbre, felhasználó barátabbra megépíteni az oldalt.



1. ábra - A címlap

3.1. Menü rendszer

Az oldal jelenlegi állapotában öt darab menü van. Igazából ennél többre nagyon nincs is szükség, majd lesznek almenük is, amennyiben már elég sablon összegyűlt. (Meglátásom szerint a kategorizálás jelenleg nem indokolt még.)

3.1.1. Címlap

Hivatkozva az 1. ábrára a Címlap felépítése fentről lefelé haladva a következő:

- *Fejléc:* ez tartalmazza a logót, egy linket a hivatalos Facebook oldalra, valamint egy szöveges reklám felületet, amely jelenleg még saját tartalomra mutat, de ez később értékesíthető.
- *Navigációs sáv:* ide kerültek a menük, illetve a kereső felület, valamint a Facebook modul.
- *Kiemelt hirdetések:* ez a sáv a legértékesebb hirdetési felület szerintem, jelenleg itt is saját, belső tartalomra mutató hirdetések vannak, de természetesen ez a felület is értékesíthető.
- *Kiemelt bejegyzés váltó:* egy modern oldal ilyen jellegű eszköz nélkül már nem létezhet. Ide olyan bejegyzésekre mutató linkek, képek kerülnek amelyek a szerintem fontosabb bejegyzésekre mutatnak. (Például új sablonról, vagy új fejlesztésről szóló bejegyzés.)
- *Kiemelt sablonok:* gyakorlatilag ez hasonlóan fontos, mint az imént említett rész. Ide mindig az oldal készítője szerinti három legfontosabb sablonra mutató hivatkozás kerül ki.
- *Középső hirdetés sáv:* hasonló a többi hirdetési felülethez.
- *Információs sáv:* ez a sáv négy külön álló blokkra van osztva: legújabb sablonok, gépház bejegyzések, kisokos bejegyzések, legnépszerűbb sablonok. Itt is megjelennek a frissen felkerülő bejegyzések és sablonok, valamint van egy rész, ami önmagától is változik, ez pedig nem más, mint a legnépszerűbb sablonok lista. Felhasználói szempontból nincs nagy jelentősége, viszont a kereső optimalizálás szempontjából igen. Ennek oka, hogy ez a rész folyamatosan változik, ha pörög az oldal, ami ugye tartalom változás, ezt pedig a kereső robotok figyelik.

3.1.2. Sablonok

Mondhatjuk azt, hogy a címlap után gyakorlatilag ez a menü az oldal lelke. Ide kerülnek be azok a tartalmak, amelyek miatt ez az oldal létre lett hozva, ide kerültek a sablonok.

Szerkezeti szempontból fontos különbség, hogy itt nem jelenik meg a kiemelt bejegyzés váltó, ennek oka, hogy túlságosan nagy, hangsúlyos területet foglalna el a sablonok szerkesztő-, vagy tartalmi felületétől.

3.1.3. Gépház

A címválasztás egy kicsi indoklásra szorul, ezért ezzel kezdeném a menü bemutatását.

Az oldal készítése során fontos irányelv volt, hogy ez egy vidám hangulatú weboldal legyen. Ez a szándék azt gondolom, hogy a színvilágtól kezdve minden fontosabb tartalmi részben visszatükröződik. Lehetett volna aktualitások, vagy hírek is a neve, de szerintem ezzel a név választással egy kicsit sikerült a kliséktől elszakadni.

Ide kerülnek azok a bejegyzések, amik közvetlenül az oldalhoz kapcsolódó dolgokról szólnak. Például megjelent egy új sablon, vagy, hogy milyen karbantartások, fejlesztések várhatók. Gyakorlatilag egy üzenőfal a felhasználók felé az oldal fejlesztőjétől.

3.1.4. Kisokos

Címválasztás szempontjából hasonló a helyzet, mint a gépháznál, így ezzel nem is foglalkoznék különösebben.

Ide kerülnek azok a bejegyzések, amelyek az oldal használatához fontosak, illetve jövőbeni cél az, hogy ez a szekció olyan tartalmakkal legyen feltöltve, amik egy arculatépítés során hasznos szakmai információkat tartalmaznak. Más oldalakhoz hasonlóan ez a GyIK (Gyakran Ismételt Kérdések) menünek felel meg.

3.1.5. Betűtípusok

Sok esetben, ha az ember ingyenes alkalmazásokat, eszközöket használ, akkor meg kell jelölni a készítőt.

Ilyenkor a legtöbb esetben elvárás a készítő részéről, hogy webes megjelenés esetén legyen egy jól látható direkt, aktív hivatkozás a készítő weboldalára.

Az általam használt külső eszközöket próbálom jól láthatóan feltüntetni, általában a háttértől eltérő színű linkként az oldalak láblécében.

Azonban annyira sok betűtípust kell megjelenítenem, hogy indokolt volt a saját menü.

A felhasználó szempontjából azért érdekes ez a menü, mert itt végig tudja böngészni az oldalon használt betűtípusokat, mielőtt magukat a sablonokat használná, illetve ha megtetszik neki egy betűtípus, akkor azt a megadott linken akár magának is beszerezheti.

3.2. Felhasználói funkciók

Ez a fejezet azoknak a főként interaktív eszközöknek a bemutatásáról szól, amelyeket a felhasználó is elér az oldalon. Ilyen például a logó készítő, vagy a kétféle közösségi média modul, illetve a kereső modul.

3.2.1. Logó generálás

Jelenleg csak egy szöveges logó generáló van az oldalon, de fejlesztés alatt van ennek a kibővített verziója is, amely már képes több sort, illetve különféle grafikai elemeket is lekezelni.

Ha egy felhasználó rákattint a „Szöveges logó” feliratú linkre a sablonoknál, akkor a következő űrlap/szerkesztő felület fogja fogadni.

Szöveges logó

Hozzáadva: 2014-03-30 21:17:39, frissítve: , megtekintve: 202x.

Egyszerű szöveges logó generálása honlapon található betűtípusok segítségével.

Előnézet:

Ide kerül majd a logó!

Szöveg: ide írd a szöveget...

Betűtípus: AdventPro-Bold Betűméret: 6

Betűszín:

R	255	G	0	B	0	#	FF0000
G	0	R	255	B	0	#	00FF00
B	0	G	0	R	255	#	0000FF
R	255	G	255	B	255	#	FFFFFF
R	255	G	0	B	255	#	FF00FF
R	0	G	255	B	255	#	00FFFF
R	255	G	255	B	0	#	FFFF00
R	255	G	0	B	0	#	FF0000
R	0	G	255	B	0	#	00FF00
R	0	G	0	B	255	#	0000FF
R	255	G	0	B	255	#	FF00FF
R	0	G	255	B	255	#	00FFFF
R	255	G	255	B	255	#	FFFFFF

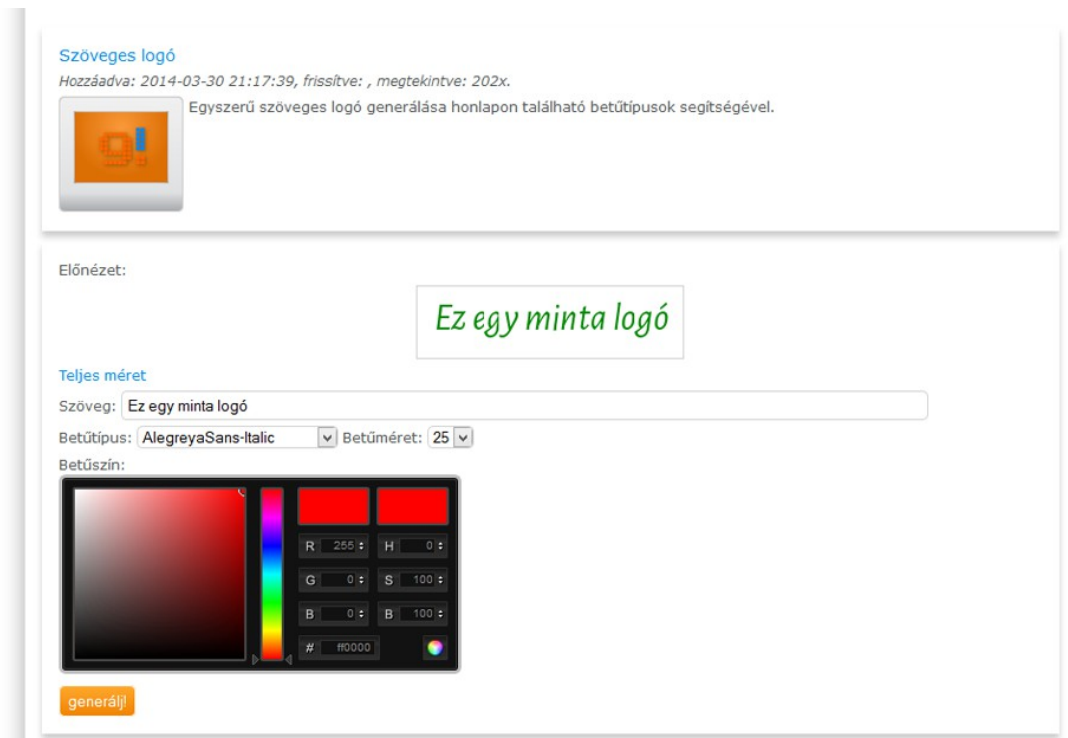
generálj

2. ábra - Szöveges logó szerkesztő

Ennél a sablonnál a szerkesztőben három dolgot lehet megadni. Fentről lefelé haladva van egy szövegbeviteli mező, ahol magát a legenerálandó szöveget lehet megadni, ezek után pedig a felhasználandó fontot, valamint a betűméretét tudja kiválasztani a felhasználó.

A színek kiválasztásához egy jQuery alapú Color Picker nevű kiegészítőt használtam, aminek segítségével, akár egér használatával, akár kézzel beírt értékekkel is tudja a felhasználó módosítani a logó színét.

Fontos megjegyezni, hogy a bevitt adatok csak ideiglenesen tárolódnak!



3. ábra – Kész szöveges logó

Miután a felhasználó megnyomja a „generálj!” feliratú gombot, a fenti ábrán látható kép fogja fogadni.

Az elkészült „png” típusú logót, a képre, vagy a „Teljes méret” feliratú linkre kattintva lehet letölteni.

3.2.2. Weboldal sablonok

Az oldalon megtalálható sablonok mindegyike a kor elvárásainak megfelelően reszponzív technológiával készült, így azok mobil- és/vagy kis felbontású eszközökön is normálisan

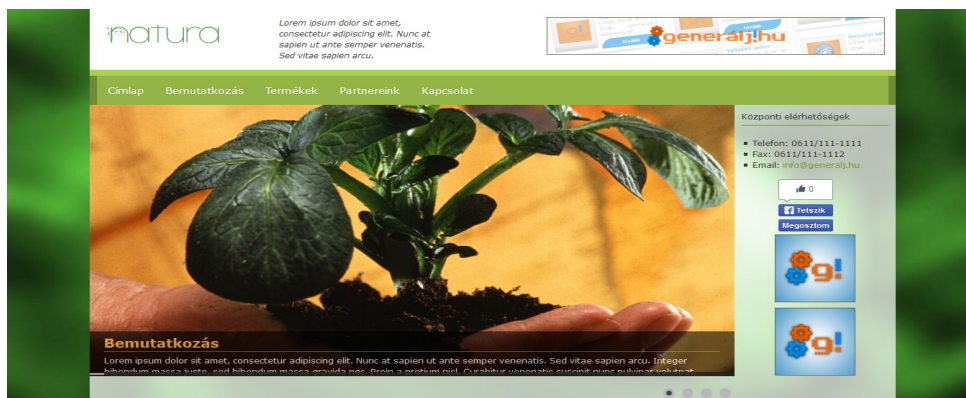
megtekinthetők. Mindegyik sablon a HTML5/CSS3 szabványt követi ennek köszönhető például, hogy reszponzív a szerkezetük.

A kiszemelt weboldal sablont többféleképpen találja meg a felhasználó az oldalon. Az egyik lehetősége, hogy, ha a sablon kiemelt, akkor a kiemelt sablonok sávjában rákattint a megfelelő linkre, de megtalálja a sablonok menüjében is, illetve adott esetben alul az információs sávjában is. Plusz opció, ha készül a sablonhoz „gépház” bejegyzés is, mint például a Natura nevű sablon esetében.



4. ábra – Natura weboldal sablon letöltési oldal

A sablont a letöltés ikonra, vagy a „Letöltés” feliratra kattintva tudja ZIP formátumban letölteni a felhasználó. A beillesztett ábrákon is látszik, hogy a sablonokhoz tartozik Facebook modul is. Ezzel nincs a felhasználónak tennivalója, mert egy PHP kód segítségével önmagát állítja be a modul. A logó is cserélhető természetesen, ha a felhasználónak tetszik a beágyazott font, akkor elég csak a szöveget átírnia a logó modulban, de kép is beilleszthető.



5. ábra – Natura sablon, normál felbontáson (képernyő részlet)



6. ábra – Natura sablon, maximum 1080px széles felbontáson (képernyő részlet)



7. ábra – Natura sablon, maximum 720px-, illetve 640px széles felbontáson (képernyő részlet)

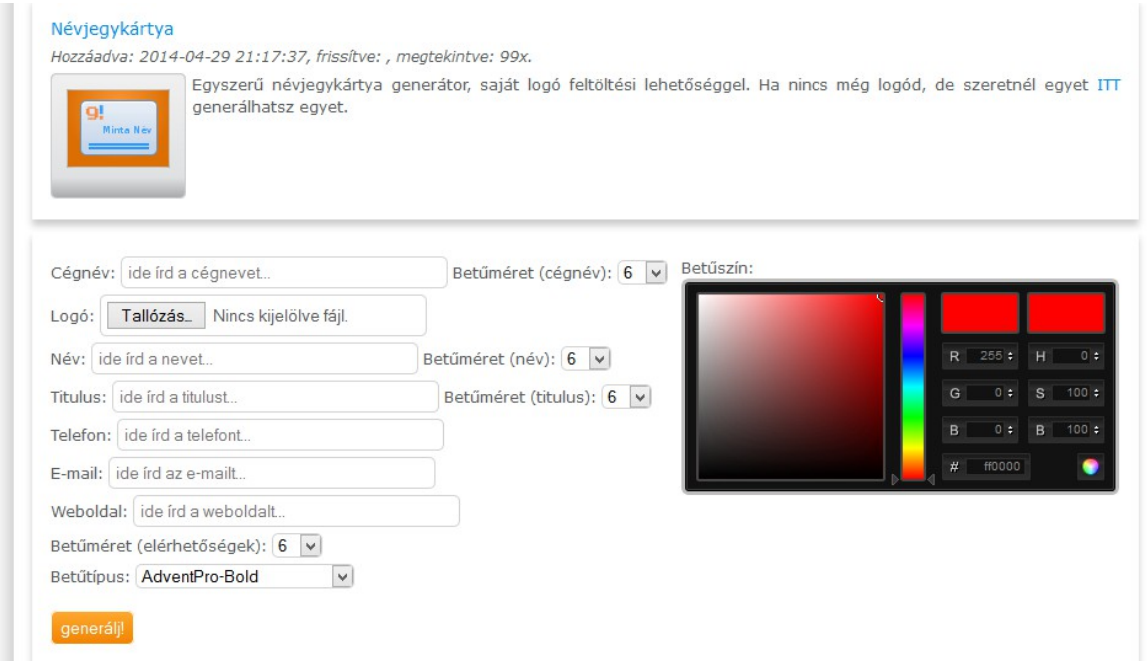
A letöltött sablonok használatához szükséges valamilyen szerkesztő program. Erre alkalmas például a Notepad++, amit én is használok a fejlesztés során.

3.2.3. Névjegykártya készítés

Ez a funkció még fejlesztés alatt van, de ettől függetlenül nem kerülhetjük meg az oldal elemzése során, hiszen ez egy eléggé egyedülálló funkció. Kutatásaim során elég kevés

ehhez hasonló online megoldást találtam, főként nyomdák oldalán találtam hasonló eszközöket. (Például a <http://www.interprint.eu> oldalon van hasonló eszköz.)

A terv szerint a megfelelő sablon linkjére klikkelve érné el a felhasználó a megfelelő sablont, ahol a terv szerint a következő szerkesztő felület segítségével tudja a saját adatait bevinni a felhasználó.



The screenshot shows a web application titled "Névjegykártya" (Business Card). It includes a header with a date and time stamp, and a small preview of a business card. The main area contains several input fields for user data: "Cégnév:" (Company Name), "Logó:" (Logo), "Név:" (Name), "Titulus:" (Title), "Telefon:" (Phone), "E-mail:", and "Weboldal:" (Website). Each text field is followed by a "Betűméret" (Font Size) dropdown menu. There is also a "Betűszín" (Font Color) color picker. At the bottom, there is a "generálj!" (Generate!) button.

8. ábra – névjegykártya szerkesztő felület

A fejlesztés alatt álló sablon használata roppant egyszerű. Elég, ha a szükséges adatok beírása, valamint a szín és a betűméret kiválasztása után a felhasználó a „generálj!” feliratú gombra kattint.

Ezután egy új lapon megnyílik a kész PDF állomány, amit az FPDF PHP osztály segítségével generál az oldalmotor. A PDF szabvány A/4 méretű, tartalmazza a vágójeleket, illetve a színeket CMYK színrendszerben tárolja. A konverzióra nem kell a felhasználónak figyelnie, azt végrehajtja az oldalmotor automatikusan.

A sablon lehetőséget kínál saját logó felhasználására is, ez esetben azonban csak maximum 1000px X 1000px felbontású, valamint PNG, JPG, vagy GIF típusú képeket lehet használni, amelyeknek a mérete nem haladhatja meg a 4MB-ot. Ha ennek nem felel meg a feltöltendő kép, akkor a PDF generálás nem történik meg, a felhasználót pedig értesíti az oldal, hogy milyen kritériumoknak nem felel meg a kiválasztott kép.

Ezeknek a funkcióknak a tudatában elmondható, hogy ha minden adat és a formázás is megfelel a felhasználó számára, akkor hozzávetőleg 10 perc alatt kézben van a látogató számára a nyomdakész PDF állomány.

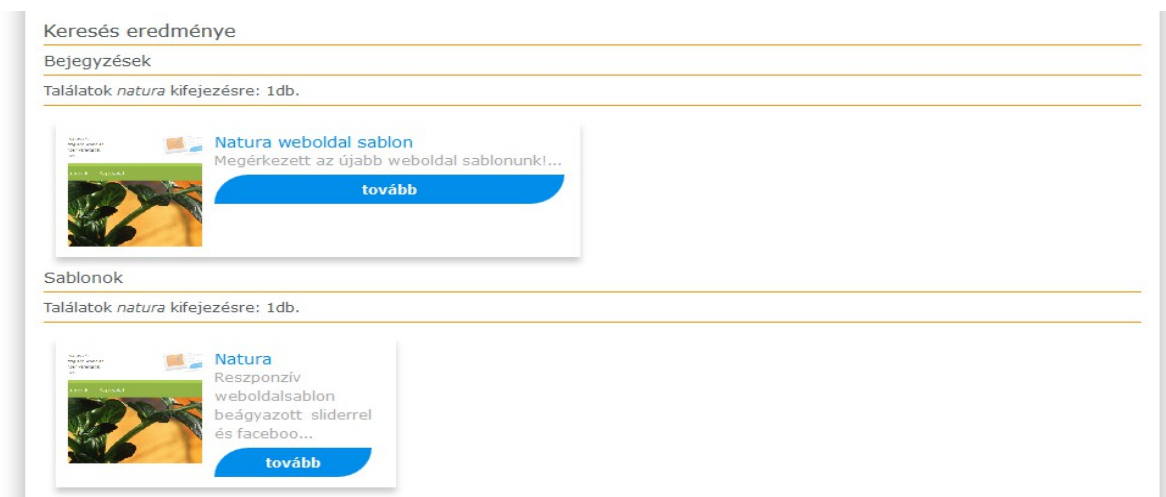
A felhasználó számára fontos információ, hogy az oldalmotor az adatokat csak ideiglenesen tárolja, így ha bezárja a megnyitott lapokat, vagy a böngészőt, akkor újra ki kell töltenie a sablon szerkesztő felületén található űrlapot.

3.2.4. Közösségi média

Az oldal rendelkezik beépített Facebook modullal, ami megjelenik a navigációs sávban minden oldalon. Ennek segítségével megoszthatja ismerőseivel, vagy kedvelheti az adott URL-hez tartozó tartalmat. Az oldalnak van saját Facebook oldala is, amit a fejlécben található ikonra kattintva lehet elérni.

3.2.5. Kereső modul

A kereső modul szintén a navigációs sávban található meg. Ennek segítségével egyidejűleg tud keresni a felhasználó a sablonok és a bejegyzések között is.



9. ábra – Keresés eredménye

4. A generálj!hu programozói szemszögből

Először Wordpress tartalomkezelő rendszert szerettem volna használni, mint oldalmotort, de olyan mértékben lassította volna a sablon kialakítása a projektet, hogy az már a gyakorlati megvalósítás rovására ment volna. Nem maradt volna vélhetően elég idő a belső alkalmazások, funkciók kialakítására.

4.1. Az oldal szerkezeti felépítése

Az oldalmotor kialakításánál törekedtem a minél egyszerűbb és átláthatóbb struktúrára. Ennek megfelelően moduláris felépítése van az oldálnak. Amit lehetett függvényekbe rendeztem, illetve a főbb oldalelemeket külön PHP fájlokba raktam. (Például fejléc, menü, stb.)

Ennek ellenére a névjegykártya generálást leszámítva, minden az index.php-n belül fut le.

A névjegykártya generálásnál azért szükséges külön PHP fájlt használni, mert enélkül az FPDF PHP osztály képtelen a PDF-et legenerálni.

A mai kor követelményeinek megfelelően HTML5, illetve CSS3 szabványt alkalmaztam a HTML alapok felépítésekor. Ennek köszönhetően egyszerűen tudtam olyan grafikai elemeket használni, mint például a színátmenetes háttér, árnyékolások, lekerekített sarkak, stb. A színátmenetek készítésekor nagy segítségemre volt az Ultimate CSS Gradient Generator, aminek segítségével roppant egyszerűen sikerült a színátmenetek CSS kódjának létrehozása.

4.2. Biztonsági megoldások

Fontos szempont volt az oldalmotor készítése közben a biztonság. Ezért az oldal minden modulja, illetve a függvényeket tartalmazó PHP fájl is a public_html mappán felülre került. Ez egy bevett szokás a saját motorral rendelkező oldalaknál. Ennek az eljárásnak a lényege, hogy így meg tudjuk védeni például a MySQL csatlakozáshoz szükséges adatokat, valamint ezzel nagy mértékben megakadályozható az oldal másolása is.

Adat védelmi szempontból fontos megoldás, hogy maga az oldal csak ideiglenesen tárolja a felhasználók által generált, vagy bevitt tartalmakat.

Ezt úgy oldottam meg, hogy a szerveren lefut 30 percenként ez a parancs: 'rm' home/sndowg/public_html/tartalom/ideiglenes/*.

Ez egy UNIX parancs, aminek értelmében például, ha valaki generál magának egy logót, akkor az bekerül egy ideiglenes nevű mappába, amit a szerver 30 percenként ürít. Ez egyébként a tárhely kihasználtság szempontjából is szükséges eljárás.

Egy sablon használata során a folyamat külön munkamenetben zajlik, saját azonosítóval, ezért nincs arra mód, hogy egy felhasználó egy másik felhasználó által bevitt adatokhoz férjen hozzá.

4.3. Fontosabb függvények, szkriptek

4.3.1. MySQL csatlakozás

Az oldal tárhelyét üzemeltető szerver a legújabb PHP-t, illetve MySQL-t futtatja. Emiatt a MySQL adatbázishoz mysqli függvényeket kell használnom. Ez elég nagy különbség az általam fejlesztői környezetként használt WAMP szerverhez képest, ahol normál mysql-t használtam.

A leglényegesebb különbség fejlesztés szempontjából a függvények között az, hogy a mysqli függvények használatakor, minden egyes blokkban meg kell hívni a csatlakozást biztosító PHP fájlt, illetve minden mysql utasításkor egy \$mysqli változó formájában deklarálnom kellett a csatlakozást.

4.3.2. Tartalom kezelés

Minden tartalom típus (pl.: bejegyzések, sablonok, felhasznált fontok ki listázása külön menüben) saját PHP fájlal rendelkezik (moduláris felépítés). Azt, hogy éppen melyik PHP modulfájlt használja az oldal egy tartalom, illetve sablonok esetében egy sablon nevű függvény dönti el, amit a függvény gyűjteményben (fuggvenyek.php) helyeztem el. Ezek a függvények az alapján döntenek el, hogy mi jelenjen meg az oldalon, hogy megvizsgálják az URL-ben található GET típusú változókat. Egy időben az URL csak két GET típusú változót tartalmazhat. Az első mindig a tartalom típusa a második pedig a hozzátartozó azonosító.

Az oldal csak azokat a tartalmakat, sablonokat jeleníti meg, amik aktívak. Ezt úgy tudtam megoldani, hogy az adatbázisban létezik egy „aktív” nevű mező, melynek minden sorban

az értéke 1, vagy üres. Ennek köszönhetően elkerülhető, hogy olyan tartalmak is megjelenjenek az oldalon, amik még szerkesztés alatt, vagy sablonok esetében fejlesztés alatt állnak. Azt, hogy mi minősül kiemelt tartalomnak, vagy sablonnak hasonlóan oldottam meg, csak itt a mező neve az adatbázisban „kiemelt”. (Egyidejűleg maximum hat darab kiemelt bejegyzés, illetve három darab kiemelt sablon lehet.) Ha a felhasználó olyan linket hív meg mondjuk az előzményekből, ami már nem aktív a „tartalom” függvény a 404.php nevű fájlt hívja meg. Ugyanez történik, ha megpróbálja módosítani például az azonosító számot tartalmazó változót.

4.4. Sablonok felépítése

4.4.1. Logó generálás

Ehhez a funkcióhoz a PHP GD Library nevű függvény családját használtam fel. Ezeknek a függvényeknek a segítségével a PHP képes képeket kezelni. Számos lehetőség van ezeknek a függvényeknek a segítségével a különféle képmanipulációkra, ezek közül számunkra a logó generálás során a legérdekesebb az „imagecreatetruecolor”, az „imagefttext”, az „imageftbbox”, illetve az „imagepng” nevű függvények.

Az „imagecreatetruecolor” nevű függvénnyel létrehozunk egy „truecolor” képet, amibe később az „imagefttext”- és az „imageftbbox” nevű függvényekkel behelyezzük a szöveget, az egészet pedig az „imagepng” függvénnyel mentjük el.

A logó készítésekor egy „logo_generalas” nevű függvényt hív meg az oldalmotor. A generáláshoz szükséges összes adat egy POST tömbben érkezik meg a függvénybe.

A betűtípus kiválasztásához szükségünk van adatbázis csatlakozásra is, hiszen a függvény csak ez alapján tudja megállapítani, hogy hol találja meg a szükséges fontot a szerveren.

4.4.2. Weboldal sablonok felépítése

Bár maga az oldal nem reszponzív (ez egy későbbi fejlesztés során majd változni fog), az oldalra fel kerülő weboldal sablonoknál elvárás volt magammal szemben, hogy azok mobil- és/vagy alacsony felbontású eszközökön is normálisan megjelenjenek.

Ehhez kellett, hogy a főbb oldal elemek méretezésekor százalékban adtam meg a szélességet, illetve szükség esetén pixelben a maximális szélességet. Önmagában ez azonban még kevés a reszponzivitáshoz. Szükségem volt még a CSS3 szabvány azon tulajdonságára, ami lehetővé teszi a maximális felbontás figyelését is. Erre a „media” függvényt használtam, amivel meg tudom mondani, hogy bizonyos maximális felbontás esetén a HTML elemek tulajdonságai miként változzanak (szélesség, elrendezés, látszik-e vagy sem az adott felbontáson).

Minden sablon moduláris felépítésű. Erre azért van szükség, hogy a felhasználó a fontosabb oldal elemeket könnyen kezelhesse. Alapvetően a sablonok használatakor minden egyes oldal külön PHP fájlba kell, hogy kerüljön, ha csak valaki nem épít mögé egy adatbázist, illetve az ahhoz szükséges oldalmotort.

5. Összegzés

Az oldal még messze nincs kész. Bár remélem, hogy soha nem is lesz az, hiszen ez azt jelenti, hogy mindig sikerül valamivel bővíteni, valamit újítani rajta.

A távlati célok között szerepel a grafikus logó, illetve névjegykártya szerkesztő elkészítése is, valamint szeretnék a későbbiek során Wordpress alapú sablonokat is készíteni.

Fontos újítás lesz még a weboldal szerkesztő elkészítése is, amivel a kiválasztott sablont már magán az oldalon fel tudja tölteni a felhasználó a saját adataival. Ez a funkció Magyarországon egyedülálló, bár külföldön már van rá példa (<http://www.wix.com>).

A fejlesztés során sok akadályt kellett leküzdeni. Vélhetően a további fejlesztések során is fognak akadni technikai jellegű problémák, de ezeket inkább érdekes kihívásként, mint sem nehézségként kell kezelni. Ezek a problémák vezetnek olyan megoldásokhoz, amivel a fejlesztés során új technológiákat tanulhatok meg.

Remélem, sőt hiszem, hogy ezzel az oldallal segíteni tudok másoknak a saját vállalkozásuk arculatának kialakításában.

6. Felhasznált külső források

Szöveges források:

Agyhullám: PHP & MySQL - Lynn Beighley, Michael Morrison/2009

Notepad++ – Origo Szoftverbázis/2011, <http://www.szoftverbazis.hu/szoftver/notepad---v5.html>

PHP – Wikipedia/2004, <http://hu.wikipedia.org/wiki/PHP>

Weblapszerkesztés - Kavaszingerné Prantner Csilla - Nagy Dénes/2014

Alkalmazások:

Camera Slider - <http://www.pixedelic.com/plugins/camera>

Color Picker - <http://www.eyecon.ro/colorpicker>

FPDF Pdf Generator Library – <http://www.fpdf.org>

jQuery - <http://www.jquery.org>

Notepad++ - <http://www.notepad-plus-plus.org>

Ultimate CSS Gradient Generator - <http://www.colorzilla.com/gradient-editor>

WAMP Server - <http://www.wampserver.com>

Fontok:

AlegreyaSans - Juan Pablo del Peral/2014 - juan@huertatipografica.com.ar

Akronim - Grzegorz Klimczewski/2014 - <http://www.fonty.pl>

AdventPro - Andreas Kalpakides/2014 - hello@inderesting.com

New gardern - Billy Argel/2014 - <http://billyargel.blogspot.com>

Grand Hotel - Brian J. Bonislowsky, Jim Lyles DBA Astigmatic (AOETI)/2012 - astigma@astigmatic.com

7. Melléklet

7.1 A "fuggvények.php" forráskódja

```
<?php
require('mysql.php');
session_start();

//Oldal függvények

function tartalom(){
    require('mysql.php');
    if(($_GET['menu'] == 'gephaz') || ($_GET['menu'] == 'kisokos') and
isset($_GET['bejegyzes'])){
        $sql_tomb = 'SELECT `aktiv` FROM `bejegyzesek` WHERE `id`="'.
$_GET['bejegyzes'].'"';
        $sql_tomb = mysqli_query($mysqli, $sql_tomb);
        $sql = mysqli_fetch_array($sql_tomb);
        if($sql['aktiv'] === '1'){
            include('../modulok/bejegyzes.php');
        }
        else{
            include('../modulok/404.php');
        }
    }
    if(($_GET['menu'] == 'gephaz') || ($_GET['menu'] == 'kisokos')){
        include('../modulok/bejegyzesek.php');
    }
    if($_GET['menu'] == 'sablonok'){
        include('../modulok/sablonok.php');
    }
    if($_GET['menu'] == 'betutipusok'){
        include('../modulok/betutipusok.php');
```

```

    }
}

function lapozo(){
    require('mysql.php');
    if(($_GET['menu'] == 'gephaz') || ($_GET['menu'] == 'kisokos')){
        $abc_tomb = 'SELECT COUNT(*) FROM `bejegyzesek` WHERE
`tipus`="'.$_GET['menu'].'" AND `aktiv`="1"';
        $abc = mysqli_fetch_array(mysqli_query($mysqli,$abc_tomb));
        $abc['0'] = $abc['0']/30;
        $abc['0'] = ceil($abc['0']);
        for($i=1; $i<=$abc['0']; $i++){
            if($i == 1){
                $limit = 0;
            }
            else{
                $limit = ($i * 30) - 30;
            }
            echo ' <a href="index.php?menu='.$_GET['menu'].'&limit='.$limit.'">'.
$i.'</a>';
        }
    }
    else{
        if(!isset($_GET['limit'])){
            header('Location: index.php?menu='.$_GET['menu'].'&limit=0');
        }
        $abc_tomb = 'SELECT COUNT(*) FROM `'.$_GET['menu'].'` WHERE
`aktiv`="1"';
        $abc_tomb = mysqli_query($mysqli,$abc_tomb);
        $abc = mysqli_fetch_array($abc_tomb);
        $abc['0'] = $abc['0']/30;
        $abc['0'] = ceil($abc['0']);
    }
}

```

```

        for($i=1; $i<=$abc['0']; $i++){
            if($i == 1){
                $limit = 0;
            }
            else{
                $limit = ($i * 30) - 30;
            }
            echo      '<a      href="index.php?menu='.$_GET['menu'].'&limit='.$
$limit.'">'. $i.'</a>';
        }
    }
}
//Védelem

```

```

function anti_troll(){
    if(!isset($_GET['menu'])){
        header('Location: index.php?menu=cimlap');
    }
    else{
        $abc_tomb = 'SELECT `url` FROM `menuk` WHERE `url`="'.
$_GET['menu'].'"';
        $abc = mysqli_fetch_array(mysqli_query($mysqli,$abc_tomb));
        if(empty($abc['url'])){
            header('Location: index.php?menu=cimlap');
        }
    }
    if(isset($_GET['id']) && (($_GET['menu'] == 'gephaz') || ($_GET['menu'] ==
'kisokos'))){
        $abc_tomb = 'SELECT `id`, `aktiv` FROM `bejegyzesek` WHERE `id`="'.
$_GET['id'].'"';
        $abc = mysqli_fetch_array(mysqli_query($mysqli,$abc_tomb));
        if(empty($abc['id']) or $abc['aktiv'] !== '1'){

```

```

        header('Location: index.php?menu=cimlap');
    }
}
}

```

//Fejléc adatok

```

function fejllec(){
    $abc = 'SELECT * FROM `oldal_beallitasok` WHERE 1';
    $abc = mysqli_query($mysqli, $abc);
    $abc = mysqli_fetch_row($abc);
    if($abc['uzemmod'] == 2){
        header('Location: offline.php');
    }
    $_SESSION['fejllec'] = $abc;
}

```

//Keresés

```

function kereses(){
    if(isset($_POST['kereses']) and !empty($_POST['kereses_szoveg'])) {
        include('../modulok/kereses.php');
    }
}

```

//Bejegyzések

```

function kiemelt_kepek(){
    require('mysql.php');
    $kiemelt_kep_tomb = 'SELECT `eleresi_utvonal` FROM `kepek` WHERE `id`="'.
    $_SESSION['kiemelt_kep'].'"';
}

```

```

$kiemelt_kep_tomb = mysqli_query($mysqli,$kiemelt_kep_tomb);
$kiemelt_kep = mysqli_fetch_array($kiemelt_kep_tomb);
$_SESSION['kiemelt_kep'] = $kiemelt_kep['elesi_utvonal'];
}

```

//Betűtípusok

```

function betutipusok(){
    echo 'betutipusok';
}

```

//Sablonok

```

function sablon(){
    if(($_GET['menu'] === 'sablonok') && isset($_GET['sablon'])){
        require('mysql.php');
        $sql_tomb = 'SELECT `aktiv` FROM `sablonok` WHERE `id`="'.
$_GET['sablon'].'"';
        $sql_tomb = mysqli_query($mysqli, $sql_tomb);
        $sql = mysqli_fetch_array($sql_tomb);
        if($sql['aktiv'] === '1'){
            $sql_tomb = 'SELECT `tipus` FROM `sablonok` WHERE `id`="'.
$_GET['sablon'].'"';
            $sql_tomb = mysqli_query($mysqli,$sql_tomb);
            $sql = mysqli_fetch_array($sql_tomb);
            if($sql['tipus'] == 'logo'){
                include('../modulok/sablonok/logo.php');
            }
            if($sql['tipus'] == 'weboldal'){
                include('../modulok/sablonok/weboldal.php');
            }
            if($sql['tipus'] == 'nevjegykartya'){

```

```

            include('../modulok/sablonok/nevjegykartya.php');
        }
    }
    else{
        include('../modulok/404.php');
    }
}
}

```

```

function betumeret(){
    for($i=6; $i<=72; $i++){
        if($_POST['betumeret'] == $i){
            $kivalasztott = 'selected = "selected"';
        }
        else{
            $kivalasztott = "";
        }
        echo '<option value="'. $i. '" '. $kivalasztott. '>' . $i. '</option>';
    }
}

```

```

function betutipus(){
    require('mysql.php');
    $abc = 'SELECT * FROM `betutipusok` WHERE `aktiv`="1" ORDER BY `nev`';
    $abc_tomb = mysqli_query($mysqli,$abc);
    while($abc = mysqli_fetch_array($abc_tomb)){
        if($_POST['betutipus'] == $abc['id']){
            $kivalasztott = 'selected = "selected"';
        }
        else{
            $kivalasztott = $_POST['betutipus'];
        }
    }
}

```

```

        echo '<option value="'. $abc['id']. ' "' . $kivalasztott . '>' . $abc['nev'] . '</option>';
    }
}

```

```

function logo_generalas() {
    require('mysql.php');
    if(isset($_POST['generalj']) and !empty($_POST['logo_szoveg'])){
        $file = session_id();
        $file = 'tartalom/ideiglenes/' . $file . '.png';
        $_SESSION['logo'] = $file;
        $text = $_POST['logo_szoveg'];
        $betumeret = $_POST['betumeret'];

        //Betűtípus kikeresése adatbázisból
        $abc = 'SELECT `eleresi_utvonal`, `felhasznalva` FROM `betutipusok` WHERE
`id`="'. $_POST['betutipus'] . '";
        $abc = mysqli_query($mysqli,$abc);
        $abc = mysqli_fetch_array($abc);
        $font = '../modulok/fontok/' . $abc['eleresi_utvonal'];
        $felhasznalva = $abc['felhasznalva']+1;
        $abc = 'UPDATE `betutipusok` SET `felhasznalva`="'. $felhasznalva . ' WHERE
`id`="'. $_POST['betutipus'] . '";
        $abc = mysqli_query($mysqli,$abc);

        //Boundingbox beállítása
        $szovegmezo = imagettfbbox($betumeret, 0, $font, $text);

        //Képméreteinek beállítása
        $padding = $_POST['betumeret']*0.8;
        $kep_szelessege = abs($szovegmezo[4] - $szovegmezo[0]) + $padding;
        $kep_magassaga = abs($szovegmezo[5] - $szovegmezo[1]) + $padding;
    }
}

```

```

//Logó elkészítése
$logo = imagecreatetruecolor($kep_szelessege, $kep_magassaga);
imagealphablending($logo, false);
imagesavealpha($logo, true);

//Színek beállítása
$szoveg_szine = imagecolorallocate($logo, $_POST['colorpicker_voros'],
$_POST['colorpicker_zold'], $_POST['colorpicker_kek']);
$shatter_szine = imagecolorallocatealpha($logo, 0, 0, 0, 127);

//Képkitöltése:
imagefill($logo, 0, 0, $shatter_szine);
$x = $_POST['betumeret']*0.4;
if(preg_match('/j/', $_POST['logo_szoveg']) || preg_match('/g/',
$_POST['logo_szoveg'])) {
    $y = $kep_magassaga - $_POST['betumeret']*0.8;
}
else {
    $y = $kep_magassaga - $_POST['betumeret']*0.5;
}

// Szöveghozzáadása a képhez
imagefttext($logo, $betumeret, 0, $x, $y, $szoveg_szine, $font, $text);

// Logó legenerálása
imagepng($logo, $file);
}

function sablon_css(){
    if(isset($_GET['sablon'])) {
        include('../modulok/js/colorpicker/colorpicker_css.php');
    }
}

```

```
    }  
}  
  
function sablon_js(){  
    if(isset($_GET['sablon'])){  
        include('../modulok/js/colorpicker/colorpicker_js.php');  
    }  
}  
?>
```

Nyilatkozat:

A dolgozat és a dolgozat témájául szolgáló weboldal a saját munkám.

Budapest, 2014. május 5.

Nemes András