

SZAKDOLGOZAT



Debreceni Egyetem
Informatikai Kar

Témavezető:

Dr. Kuki Attila
egyetemi adjunktus

Készítette:

Szabó Balázs
mérnök informatikus
hallgató

Debrecen,
2011

Tartalomjegyzék

I.	BEVEZETÉS.....	4
I.1.	Előszó	5
I.2.	Betekintés	6
I.3.	Statikus és dinamikus honlapok.....	6
I.3.1.	A statikus oldalak	6
I.3.2.	A statikus weblapok korlátai	7
I.3.3.	A statikus weblapok előállítása (informálisan)	7
I.3.4.	A dinamikus oldalak.....	8
I.3.5.	Összefoglalás	9
I.4.	Az adatbázisokról röviden	9
I.4.1.	Az adatbázisok tervezésének lépései.....	10
II.	TÉMAVÁLASZTÁS	11
II.1.	A megalkotott webalkalmazás.....	11
II.2.	A témaválasztásról.....	11
II.3.	Kezdőlépések.....	12
III.	SZOFTVEREK.....	13
III.1.	Az alkalmazott szoftverek.....	13
IV.	A TERVEZÉS ELSŐ LÉPÉSEI.....	15
IV.1.	Képernyőképek, felülettervezés.....	15
IV.1.1.	A CSS-ről röviden	15
IV.1.2.	Követelmények a felületekkel szemben	17
IV.1.3.	Az oldal fő funkciói.....	18
V.	A SZOFTVERFEJLESZTÉS.....	20
V.1.	Szoftverfejlesztési modellek	20
V.1.1.	Vizesés modell.....	20
V.1.2.	Evolúciós fejlesztési modell	21
V.1.3.	Spirális modell.....	23
V.1.4.	Inkrementális fejlesztési modell	24
V.2.	Szoftverfejlesztési modell választása.....	25
VI.	AZ ADATBÁZIS TERVEZÉS.....	26
VI.1.	Az ER modell	26
VI.1.1.	Az egyedtípusokról.....	26
VI.1.2.	Az egyedtípusok közti kapcsolatok szemléltetése.....	27

VI.1.3.	A ER modell megalkotása	28
VI.2.	A normalizálásról	29
VI.2.1.	A normálformák ismertetése	29
VI.3.	Az ingatlan adatbázis	33
VI.4.	A rendszerrel szemben támasztott követelmények	38
VI.4.1.	A rendszer biztonságáról	38
VI.4.2.	A rendszer megbízhatóságáról.....	39
VI.4.3.	A rendszer követelmények pontokba szedve.....	39
VI.5.	A közvetítés menete	41
VI.5.1.	A hirdető.....	41
VI.5.2.	A vásárlók.....	41
VI.5.3.	Az adminisztrátorok	41
VI.6.	A felhasználói csoportok bemutatása	42
VI.6.1.	A látogatók	42
VI.6.2.	Az adminisztrátorok	42
VI.7.	A felhasználói felület	43
VI.7.1.	Keresések.....	43
VI.7.2.	Az Ingatlanok kilistázása.....	45
VI.8.	Az admin felület.....	46
VI.8.1.	Bejelentkezés	46
VI.8.2.	Kínálatok listája.....	46
VI.8.3.	Új ingatlan hozzáadása	47
VI.8.4.	Az ingatlanhoz tartozó képek feltöltése.....	48
VI.8.5.	Ingatlan módosítása	48
VI.8.6.	Ingatlan törlése	49
VI.8.7.	Kijelentkezés	49
VI.8.8.	Hirdető hozzáadása.....	49
VI.8.9.	Hirdető törlése	50
VI.8.10.	Admin hozzáadása.....	51
VI.9.	Az oldal beépített funkcionalitása	51
VI.9.1.	Galéria	51
VI.9.2.	Datepicker.....	51
VI.9.3.	SQL injection.....	52
VI.9.4.	Mező ellenőrzések	52
VI.10.	A rendszer validálás	52
VI.11.	A rendszer üzembe helyezése.....	53
VI.12.	A rendszer üzembe helyezése éles környezetben	54
VII.	BEFEJEZÉS	55
VIII.	IRODALOMJEGYZÉK.....	56

I. BEVEZETÉS

Az internet térnyerésének és a számítógép-használat terjedésének köszönhetően, egyre több lehetőség adódik és nyílik meg a vállalatok és cégek számára, hogy sikeresebbé és jövedelmezőbbé tegyék az általuk végzett munkákat és feladatokat. A világháló sok olyan kiaknázásra váró részt rejt, melyek felhasználásával még közelebb hozható annak lehetősége, hogy sikerüljön számukra versenyképessé válni, a gyorsan változó és kiélezett cégek közt jelentkező versengésben.

Ezen célok eléréséhez a web segítségével működtethető alkalmazások elkészítésére volt szükség. Ezen alkalmazások olyan tulajdonságokkal vannak felruházva, melyek csábítóak lehetnek mindenki számára, aki akár egy kicsit is érdekelt ebben a témakörben.

Napjainkban már nem csak a vállalatoknál, hanem az otthoni felhasználók körében is elterjedt ez a technológiai felhasználás. Az otthoni felhasználóknak nem feltétlen az üzletkötés, vagy a haszonszerzés a céljuk.

Már léteznek a technológiának olyan felhasználásmódjai is, melyek segítségével dokumentumokat tudunk tárolni saját felhasználásra, vagy megosztani azokat az általunk megjelölt emberekkel. Ennek segítségével az otthoni felhasználók is bárhol és bármikor szerkeszteni tudják a dokumentumaikat, esetleg azokat csoportos együttműködéssel is.

Ezek alapján is jól látható milyen széles körben lehet felhasználni és alkalmazni ezeket a programokat. Ennek következtében az újabb és újabb alkalmazások elkészítésének száma, a mai napig egyre csak növekvőben van.

I.1. Előszó

Mielőtt belekezdenék a szakdolgozati munkám prezentálásába, köszönetet szeretnék mondani ezen részben, a szakirodalomnál felsorolt művek szerzőinek, mert segítséget kaptam és sok hasznos ötletet meríthettem munkáikból, melyekből még jobban és átfogóbban megismerhettem a PHP programozási nyelvet, az adatbázisrendszereket és az ehhez kapcsolódó egyéb témaköröket.

Külön köszönetet szeretnék mondani Dr. Kuki Attilának, amiért vállalta a munkám témavezetését, és javaslataival, ötleteivel segített még tökéletesebbre formálni azt.

Itt szeretném megköszönni, Lukácsi András tanár úrnak a sok segítséget és a kitartó biztatást, melyeket munkám megalkotása és tökéletesítése során kaptam tőle. Valamint, hogy elvállalta a felkérést a konzulensi feladatra a szakdolgozatomhoz.

Továbbá szüleimnek és családomnak, akik mindvégig bíztattak egyetemi tanulmányaim során, és ezt a mai napig hasonlóan teszik.

I.2. Betekintés

Szakedolgozatomban szeretnék betekintést nyújtani az informatikai világ, talán napjaink során az egyik legjobban fejlődő és legnagyobb hódítást élvező részébe. A webes oldalak által alkotott világba, amelyről az előbbiekben már szót ejtettem.

Ezen terület biztosan sokak számára ismertté vált és sokak találkoztak már az általa elért és nyújtott eredményekkel. A világháló és az egyszerű kezelhetőséget kínáló programok segítségével, már bárki könnyedén megformálhatja és megalkothatja saját weblapját az elképzeléseihez mérten. Ennek következtében bárki publikálhatja és feltöltheti az internetes tárhelyekre a saját vagy megbízással készített honlapját.

Ezen oldalak felhasználhatóságát és működését tekintve meghatározó alapjuk a statikusság, vagy ezzel szemben a rivális dinamikusság.

I.3. Statikus és dinamikus honlapok

I.3.1. A statikus oldalak

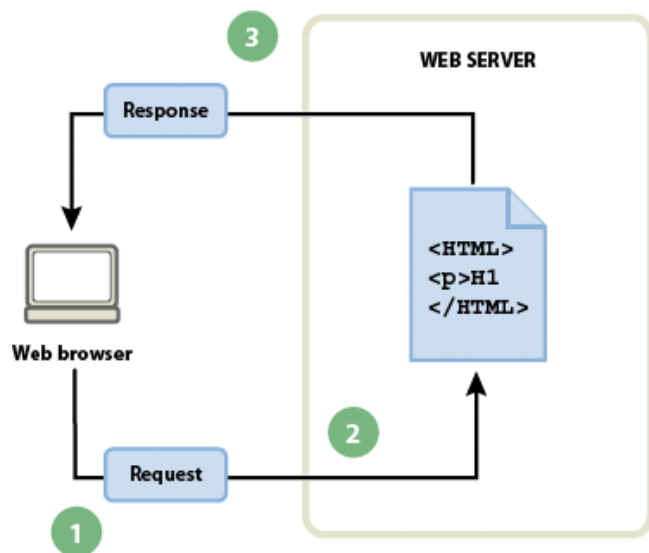
Az internet használat elterjedésének kezdetén a weboldalak szinte mindig statikus dokumentumokat jelenítettek meg. A statikus honlapok legjellemzőbb tulajdonsága, hogy csak az információ közlésére és bizonyos kapcsolattartási funkciók ellátására alkalmasak, az azt böngésző felhasználókkal. Ha egy statikus oldalt elkészítünk, feltesszük egy webszerverre és azt egy böngésző segítségével meg tudjuk nézni. Ezek a weboldalak mind azonos HTML leíró nyelven készültek. Alap építőelemei a táblázatok, címsorok, képek és hivatkozások, melyekkel más oldalakra juthatunk el. Nem áll módunkban adatbázis műveleteket futtatni, és az oldalon ebből fakadóan dinamikus elemeket elhelyezni. Ezen oldalak felépítéséhez napjainkban már az XHTML és a CSS segítségét is felhasználják. A statikus honlapoknak van azonban egy fejlettebb alkalmazási lehetősége is. Itt java scriptekkel és egyéb trükköket felhasználva próbálják azt az érzést kelteni, hogy egy dinamikus oldalt böngészünk.

I.3.2. A statikus weblapok korlátai

Mint láthatjuk az effajta dokumentumok elkészítése nem túl bonyolult feladat, de szükségünk lehet kellő időtartamra és megfelelő szaktudásra a megalkotásukhoz. Ezen oldalakkal a probléma az, hogy csak akkor változtatják meg a tartalmukat, ha a fejlesztő módosításokat hajt végre rajtuk, majd a változtatásokat újra feltölti a webkiszolgálóra. Ha új lapokkal szeretnénk bővíteni az oldalunkat, esetlegesen a tartalmát gyakran szeretnénk frissíteni, akkor időigényes feladatba ütközünk.

I.3.3. A statikus weblapok előállítása (informálisan)

Ha beírunk a böngészőbe egy címet, mely segítségével szeretnénk a címen található oldalt megnézni, akkor egy HTTP oldalkérési kérelmet nyújtunk be, a webkiszolgáló számára. A kiszolgáló ezt a kérést feldolgozva, a kért oldal HTML-kódját egy válasz üzenet formájában adja át a böngészőnek, és mi így tekinthetjük meg a kért oldal tartalmát.



1. Az oldal kérésre kérés benyújtása.
2. A kérés feldolgozása a webserveren.
3. A kért oldal átadása a böngészőnek válaszüzenet formájában.

1. ábra (A statikus weblapok generálása)¹

¹ Az 1. ábra forrása : http://help.adobe.com/en_US/dreamweaver/cs/using/images/ds_process_static.png

I.3.4. A dinamikus oldalak

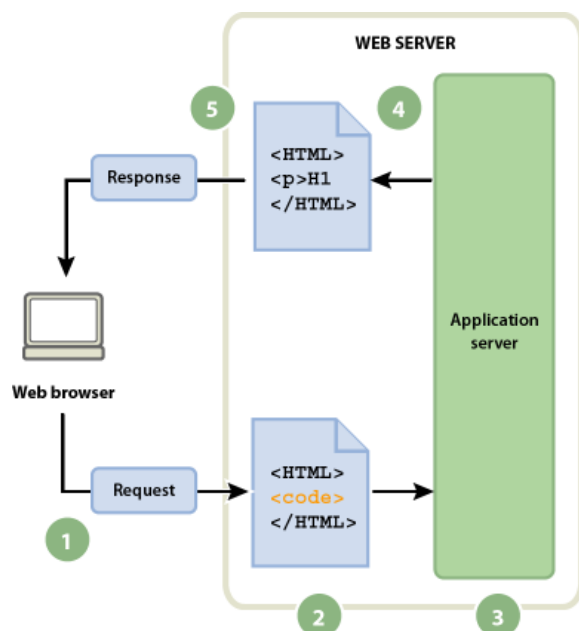
A statikus oldalakkal szemben a dinamikus oldalak lehetővé teszik számunkra, hogy a látogatást követő pillanatban generálódjon az oldal tartalma, az aktuális adatbázis és tartalom alapján. A statikusnál jól ismert egyirányú kommunikáció itt megszűnik, lehetőségünk nyílik kapcsolattartásra látogatóinkkal, és egyéb adatbázis műveleteket felhasználó parancsok beépítésére a weboldalunkba.

Segítségükkel személyre szabott oldaltartalom is megjeleníthető. Az esetleges fejlesztőnek itt nem kell folyamatosan frissítenie az oldal tartalmát, hanem a programozó beavatkozása nélkül is aktualizálható, frissíthető a felület tartalma.

Ezeknek a honlapoknak az elkészítése összetettebb feladatmegoldást igényel.

A dinamikus, PHP-vel elkészített oldalak lehetővé teszik, hogy a webkiszolgáló mielőtt a kért adatokat átküldené a böngésző számára, azokon változtatásokat hajtson végre.

A kiszolgálón egy php-parancsfájl fut, amely az igényekhez mérten hajt végre módosításokat a HTML-kódon. Ekkor a szerver hozzáférhet az adatbázis tartalmához és a konfigurációs fájlokhoz is, esetlegesen azonosíthatja ily módon a felhasználót is. Tehát egy PHP szkript futtatás eredménye, mindig HTML-lé alakul át. A böngésző ebből semmilyen változást nem érzékel, mert ugyan úgy egy HTML-kódú oldalt jelenít majd meg.



1. Oldal kérését a böngésző benyújtja.
2. A webszerver fogadja a kérést és átadja az alkalmazási szervernek.
3. Az alkalmazási szerver végrehajtja a kért változtatásokat az oldalon.
4. Az alkalmazási szerver átadja a tisztán HTML kódból álló oldalt a webszervernek.
5. A kért oldalt válaszüzenet formájában megkapja a böngésző.

2. ábra (A dinamikus weblapok generálása)²

² A 2. ábra forrása : http://help.adobe.com/en_US/dreamweaver/cs/using/images/ds_process_dynamic.png

I.3.5. Összefoglalás

Ezek után megfogalmazódhat bennünk a kérdés, hogy miért is választják oly sokan a statikus oldalakat? Ha a megnyerő tulajdonságuk oly kevés, a dinamikus oldalakkal szemben.

Mint minden szinten, az oldal elkészítésének az egyik legbefolyásolóbb tényezői közt az ára szerepel. Míg egy *statikus honlapot* bárki elkészíthet, aki rendelkezik egy minimális tudással és egy HTML-szerkesztő programmal, így ennek következtében az elkészítés költsége is nagyon alacsony az effajta oldalaknál. Ennek akkor jelentkezik a hátulütője, amikor azt szeretnénk, hogy látogatottabbá, sikeresebbé váljon az általunk készített oldal, esetlegesen túl gyakran kell frissítéseket és átalakításokat elvégeznünk alkotásunkon.

I.4. Az adatbázisokról röviden

Az egyes kutatások is azt bizonyítják, hogy már a korai világban találhatóak voltak olyan gyűjtemények, amelyek bizonyos tulajdonságokat felhasználva lettek összeállítva. Akkoriban is, valamilyen rendszer alapján szervezték sorrendbe a kívánt adatokat. Ezen gyűjtemények közé sorolhatóak, példa említés céljából a ma is használatos, hivatalok és a könyvtárak is. A könyvtárakban sok olyan könyvet találhatunk, melyeknek rengeteg tulajdonságuk van, melyek segítségével csoportosíthatóvá tehetőek, és bizonyos feltételek alapján rendezettségbe állíthatóak. A hivatalokban is sok olyan irat található, melyek megfelelő rendezési elv felhasználása nélkül, nehézkesen átlátható és böngészhető állapotba kerülnének. Ezen papír alapú gyűjtemények mindegyikénél a felmerülő probléma, hogy ha esetlegesen nagy mennyiségű, terjedelmes adathalmazt fognak alkotni, akkor feldolgozásuk és a karbantartásuk is nehéz feladattá válik számunkra.

A számítástechnika fejlődésének köszönhetően a számítógépen tárolható és segítségével működtethető adatbázis-rendszerek egyre kiemelkedőbb szerepekhez jutnak. Használatukkal, a nagy adathalmazokkal rendelkező rendszerek adatai is könnyebben átláthatóak lesznek, és könnyű kereshetőséget és felhasználhatóságot biztosítanak számunkra. Napjainkban a relációs adatbázis-rendszerek kerültek hatalomra, számos olyan megnyerő tulajdonságuknak köszönhetően, melyek nélkül ma már el sem tudnánk képzelni az adatbázis tervezést.

I.4.1. Az adatbázisok tervezésének lépései

Az adatbázisok megtervezése egy több lépésből álló folyamat, amely először a logikai adatbázis megtervezésével kezdődik, majd ezt követi a fizikai szinten megjelenő tényleges adatbázis elkészítése.

Az első lépések közt szerepel az adatbázisban létrehozandó rendszer és a felmerülő igények elemzése, majd a később benne szereplő adatok csoportjának meghatározása. Következő lépés a rendszertervezés, melyben vizsgáljuk a relációk közti kapcsolatokat. Ennek eredményeképpen említhetjük meg a rendszer specifikációját, más néven logikai modelljét. Utolsó lépésként következik a fizikai szinten való leképezése az általunk megalkotott logikai adatbázisnak, a hardver és szoftverek segítségével.

A logikai adatbázis megtervezésének a célja a redundanciák megszüntetése a táblákon belül. Redundanciáról akkor beszélhetünk, ha egy adatot többször is tárolunk az adatbázisban. Ezek a többször előforduló adatok azt eredményezhetik, hogy az adatbázisunk inkonzisztens állapotba kerülhet, amiben a frissítés és karbantartás is komplikálttá válik. A tervezés során különféle normalizációs eljárásokat kell alkalmazni, melyek a függőségi kapcsolatokon alapszanak a relációs modell és a modellezési fogalmak között. A relációs-adatbázisok esetén, már a logikai modell megalkotása közben elnyerhetik végleges formájukat az alkotó relációk.

II. TÉMAVÁLASZTÁS

II.1. A megalkotott webalkalmazás

Munkám során bemutatom, hogyan és miként épül fel egy olyan dinamikus tartalommal rendelkező és adatbázis-rendszeren alapuló oldal, mely tartalma akár minden pillanatban változhat. Ezen tényezőt kihasználva fog életre kelni egy webalkalmazás, amely nem csak egyirányú kommunikációt tesz lehetővé, és biztosít az oldalt látogató személyekkel, hanem azok személyes döntéseik és választásaik alapján tudnak majd az oldalon belül böngészni, és érdeklődést benyújtani.

Az általam megalkotott webalkalmazás keretein belül életre keltek egy konkrét ingatlan nyilvántartó rendszert. Ez a rendszer megfelel minden olyan elvárásnak, amelyet egy ingatlan közvetítő alkalmazás szolgáltat, és nyújthat az érdeklődők számára. Üzemeltetését egy ingatlanközvetítő iroda végzi. Az alkalmazás kiszolgálja mindazokat az igényeket, melyeket az üzemeltetők felállítottak. A felhasználói oldalról elérhető funkciók mind térítésmentesek, így elérésükhöz és használatukhoz sem regisztráció, sem anyagiak nem szükségesek.

II.2. A témaválasztásról

Azért is esett a választásom egy ingatlanközvetítő alkalmazás megvalósítása mellett, mert elég nagy kihívást láttam az elkészítésében, és úgy érzem ezen keresztül kellőképpen tudom szemléltetni a kiválasztott webes alkalmazásokról szóló szakdolgozati témát. Mindenképp szem előtt tartottam a választásomnál, egy olyan téma megalkotását, melynek valóban célja van és valóban felhasználható munkavégzésre akár most akár a későbbiekben. Esetleg bővítésekkel tovább fejleszteni, és szélesebb körű munkavégzésre kialakítani az elkészített alkalmazást. Egy ilyen nyilvántartó rendszert sok felhasználó vesz igénybe, így célszerű a felhasználói felület elérését a web segítségével megoldani. A dolgozatomban bemutatom ennek az alkalmazásnak a teljes fejlesztési fázisát, az első kezdeti lépésektől folyamatosan felépítve a végkifejletig, amikor is üzembe helyezzük az alkotásunkat.

II.3. Kezdőlépések

Feladatom megkezdése előtt szakirodalom gyűjtésébe és tudásom bővítésébe kezdtem, hogy ne csak általánosságban, hanem a szakmaiságon keresztül tudjam szemléltetni a választott témámat. Felkerestem egy ingatlanok közvetítésével foglalkozó céget, hogy ebből kifolyólag még pontosabb, és eredményesebb munkát lehessen elérni. Sokszor az alkalmazást megalkotó fejlesztő nem rendelkezik kellő, és olykor elegendő szaktudással a megalkotásra váró témával kapcsolatban. Ezen hiányosságokból eredő hibák elkerülése végett is ajánlatos egy a szakmában otthonosan mozgó, esetleg ezzel foglalkozó céget felkeresni.

A megszerzett információkat és a saját ötleteimet is felhasználtam a kivitelezéshez, emellett a vállalat által felállított követelményeket, kritériumokat és elvárásokat is. A cég segítségével kerültek kialakításra azok az elvárások amelyek a későbbi tényleges használatkor fontos tényezőket alakíthatnak ki. A munka során így nem a fejlesztő dönt a saját elvárásaihoz mérten, hanem teljes mértékben, szorosan a megbízó igényeihez alkalmazkodik. Az elkészített ingatlanközvetítő rendszer ennek a valós életben is működő vállalkozásnak a megrendeléseként készült el.

III. SZOFTVEREK

III.1. Az alkalmazott szoftverek

Ahhoz, hogy különböző platformokon webes programokat szerkesszünk és fejlesszünk, nem elegendő csupán feltelepíteni az adott nyelvhez tartozó kezelőprogramot. Ha csak ezt tennénk meg, akkor az általunk megírt szkriptek, lehet, hogy elméletben jól működnek, azonban annak lehetőségére, hogy azt böngészőben lefuttassuk, és eredményeit megfigyeljük nincs lehetőségünk. A böngésző, mint már korábban említettem, egy web kiszolgálóval történő kommunikációban kapott eredményeket jelenít meg. Tehát annak eléréséhez, hogy egy webes fejlesztői környezetet alakítsunk ki a saját gépünkön, egy webkiszolgálóra is szükségünk lesz. Az adatok tárolását is valahogyan meg kell oldanunk. Ehhez egy adatbázis rendszerre is szükségünk lesz. A következőkben szeretném bemutatni pár mondatban a munkámhoz választott és használt programokat, de nem célom az adott programok működésének és függvényeinek teljes ismertetése.

MySQL

A rendszer egy kellőképpen normalizált MySQL adatbázison alapszik, mely adatbázis tartalmaz minden olyan adatot melyek az ingatlanokhoz és az ahhoz kapcsolódó részekhez köthetőek. Azért emellett a szoftver mellett döntöttem, mert mind gyorsaságban és mind tudásban megfelelő segítséget nyújtott számomra, nem említve ingyenes felhasználhatóságát.

PHP

Az alkalmazásban a PHP programozási nyelvet felhasználva lekérdezéseket hajtunk végre egy adatbázisban, így mind a felhasználói mind az adminisztrátori felület ezen nyelvet felhasználva nyeri a szükséges adatait, melyek a működéséhez elengedhetetlenek. Ezen nyelv a MySQL adatbázis kezelővel tökéletest-párost alkot. Kiválasztásakor ismét szem előtt tartottam ingyenes és kiváló felhasználhatóságát.

Apache

Otthoni tesztelésre, konfigurálásra az APACHE webservert alkalmaztam. Az Apache a legelterjedtebb webkiszolgálók közé sorolható. Elterjedését talán annak is köszönheti, hogy ingyenesen beszerezhető és több operációs rendszeren is elérhető.

XAMPP

A XAMPP egy olyan fejlesztői környezet, mely több operációs rendszeren is elérhető, és segítségével pár könnyű telepítési és konfigurálási lépést elvégezve, a saját gépünkön is ki tudunk építeni egy alkalmazásfejlesztésre és tesztelésre alkalmas rendszert.

Ingyenesen elérhető és megtalálható benne a MySQL, PHP és Apache is, amely komponenseket én is választottam a munkámhoz. Használata mellett szól azon érv is, hogy telepítésével elérhetővé válik számunkra a phpMyAdmin nevű alkalmazás, mely segítségével grafikusan tudjuk az adatbázisunkat működtetni, és azon különböző műveleteket végrehajtani.

Dreamweaver

A Dreamweaver a professzionális weblapszerkesztő programok közé sorolható. Segítségével a tervezésnél lehetőségünk van vizuálisan és kódszerkesztő nézetben is szerkeszteni a honlapunkat. Ezen nézetek közt gondtalanul navigálhatunk munkánk során.

Ha bármely felületen változtatásokat hajtunk végre, akkor azok a változtatások a másik felületen is azonnal megnyilvánulnak. Támogatja a szerver oldali szkripteket és kiemelten kezeli az XML-t, HTML-t és a CSS-t is. Felhasználásával oldottam meg az oldalon található menük és egyéb grafikus elemek elhelyezését, valamint az oldal arculatának megtervezését.

Notepad++

A PHP programkód megírásához felhasználtam a Notepad++ forráskód szerkesztő programot, mely ingyenesen beszerezhető és nagy tudással rendelkezik. A program segítségével kiemelhetőek a kódok és az egyes kódblokkok egységbe rendezhetőek.

A programkód szerkesztésekor is nagy segítséget nyújtott számomra ez az alkalmazás, mert az összetartozó nyitó- és zárójeleket kiemeli színezéssel, így az esetlegesen le nem zárt utasítások már a futtatás előtt is észrevehetőek.

CSS:

A CSS ismertetését a képernyőtervekkel foglalkozó, következő fejezetben mutatom be.

IV. A TERVEZÉS ELSŐ LÉPÉSEI

A tervezés első lépései között szerepel a megrendelővel való egyeztetés és az igények felmérése. A webes felület úgy kerül kialakításra, hogy mind a felhasználói mind az adminisztrációs felület a közvetítő munkát szolgálja és ezáltal könnyebben és rövidebb idő alatt lehetséges a felhasználói igényekhez mértén megfelelő ingatlanra tálni.

A személyes, ill. telefonos konzultálásra csak azokban az esetekben van szükség, ha a vásárló komolyan gondolja célját. A felhasználói felület nem tartalmaz semmi olyan hivatkozást mellyel az ingatlanközvetítők munkája feleslegessé válna, így ha valaki érdeklődést mutat egy ingatlan iránt, mindenféleképpen fel kell venni a kapcsolatot a közvetítő irodával. Erre is lehetőséget biztosít az alkalmazás.

IV.1. Képernyőképek, felülettervezés

Az oldal megtervezésénél és a kivitelezésnél, szem előtt kell tartani a könnyű kezelhetőséget, és az átláthatóan kialakított oldalstruktúrát. A felhasználói célközönségről tudjuk, hogy nem rendelkezik komoly számítástechnikai ismeretekkel. Így ezt felhasználva, az alap számítógép kezelői tudásra építve kell megalkotni az oldalon található felületeket. Fontos tulajdonságaik közt kell szerepelniük a formailag igényes, és szép elemek elhelyezésének.

IV.1.1. A CSS-ről röviden

A CSS rövidítés az angol *Cascading Style Sheets* szóból ered. A CSS egy stílusleíró nyelv, amely a HTML típusú strukturált dokumentumok megjelenítését írja le. Segítségével különböző stíluslapokat hozhatunk létre, majd azokat beépíthetjük a HTML alapú honlapunkba. Segítségével befolyásolni tudjuk az oldal formai elemeinek megjelenését, és a HTML tagekben szereplő adatok elhelyezkedését is.

A saját alkalmazásom elkészítésénél azért is alkalmaztam ezt a technológiát, mert segítségével a forráskódok olvashatóbbak lettek. Az ismétlődő elemeket át lehet csoportosítani egy külön fájlba, és a kívánt oldalon csak hivatkozni kell rájuk. Így tegyük fel, ha egy elem elhelyezkedését meg szeretnénk változtatni az összes oldalon, csak egy fájlban belül kell átírni a kódrészletet. Sok lapból álló oldalak esetén ennek nagyon jelentős a szerepe.

Részlet a felhasználói oldal CSS kódjából

```
body,td,th3 {  
    font-family: Arial, Helvetica, sans-serif;  
    color: #069; }
```

```
body4 {  
    background-image: url(images/bckgrnd.jpg);  
    background-repeat: repeat-x; }
```

```
.oszlop15 { background-color: #EBEDEB; }
```

```
#logo2 {  
    position:absolute;  
    left:15px;  
    top:15px;  
    width:209px;  
    height:68px;  
    z-index:1; }
```

```
#menu_horizontal6 {  
    position:absolute;  
    left:15px;  
    top:87px;  
    width:522px;  
    height:151px;  
    z-index:9; }
```

³ Az oldal betűtípusára és betűszínére vonatkozó parancs.

⁴ Az oldal háttérének és az ismétlődés mértékének megadása.

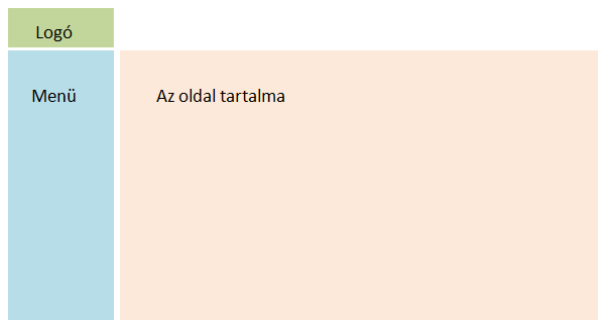
⁵ Az oldalon található táblázatokra vonatkozó formázási beállítás.

⁶ A menü elhelyezésére vonatkozó utasítás.

A kezdetleges oldaltervek

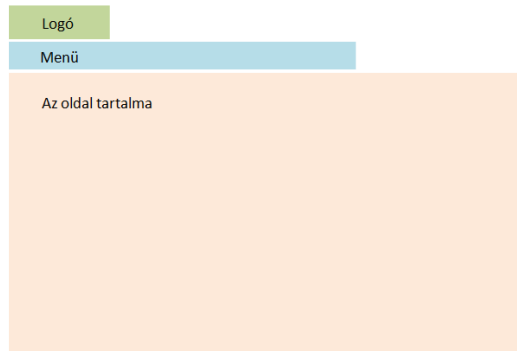
Az oldal kinézetének tényleges megalkotása előtt elkészült felülettervek a megrendelő igényeihez igazodva lettek kialakítva.

A felhasználói oldal felületterve



3. ábra (A felhasználói oldal)

Az admin oldal felületterve



4. ábra (Az adminisztrátori oldal)

IV.1.2. Követelmények a felületekkel szemben

1. A menürendszer

Felhasználói menü

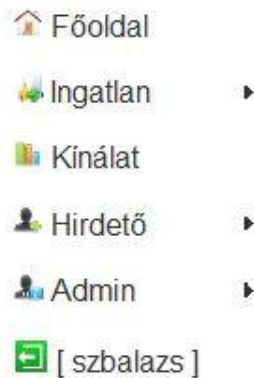


5. ábra (A felhasználói menü)

A felhasználói felületen megtalálható navigációs menü horizontális helyzetben került elhelyezésre a megrendelő kéréséhez igazodva.

- Főoldal
- Kínálat
- Keresés
 - *Egyszerű keresés*
 - *Részletes keresés*
 - *Azonosító szám segítségével történő keresés*
- Kapcsolat

Admin menü



6. ábra (Az admin menü)

Az adminisztrátori felületen megtalálható navigációs menü vertikális helyzetben került elhelyezésre a megrendelő kéréséhez igazodva.

- Főoldal
- Ingatlan
 - *Új ingatlan hozzáadása*
 - *Ingatlan módosítása*
 - *Ingatlan törlése*
- Kínálat
- Hirdető
 - *Új hirdető hozzáadása*
 - *Hirdető eltávolítása*
- Admin
 - *Új admin hozzáadása*
- Kijelentkezés

IV.1.3. Az oldal fő funkciói

1.Hirdető hozzáadása:

- A hirdető teljes neve – beviteli mező
- A hirdető településének irányítószáma – beviteli mező
- A hirdető települése – beviteli mező
- A hirdető utcája – beviteli mező
- A hirdető házszáma – beviteli mező
- A hirdető e-mail címe – beviteli mező
- A hirdető mobiltelefon száma – beviteli mező

2.Admin hozzáadása:

- Az adminisztrátor teljes neve – beviteli mező
- Felhasználónév a bejelentkezéshez – beviteli mező
- A későbbi bejelentkezéshez szükséges jelszó – beviteli mező
- Az admin email címe – beviteli mező
- az admin telefonszáma – beviteli mező

3.Ingatlan hozzáadása:

- Admin – legördülő lista
- Hirdető – legördülő lista
- Azonosító – beviteli mező
- Kínálat típusa (eladó / kiadó) – legördülő lista
- Típus * – legördülő lista
- Település – legördülő lista
- Utca – beviteli mező
- Házsám – beviteli mező
- Helyrajzi szám – beviteli mező
- Kor – beviteli mező
- Anyag – beviteli mező
- Alapterület – beviteli mező
- Telek terület – beviteli mező
- Szobaszám - legördülő lista
- A lakás emelete – legördülő lista
- Közös költség – beviteli mező
- Komfort (fél komfort / összkomfort) – radio button
- Fűtés típusa () – legördülő lista
- Külön vízóra – checkbox
- Csatorna – checkbox
- Víz – checkbox
- Gáz – checkbox
- Fűtés – checkbox
- Villany – checkbox
- Telefon – checkbox
- Kábel tv – checkbox
- Internet – checkbox
- Riasztó – checkbox
- Garázs – checkbox
- Kocsi beálló – checkbox
- Pince – checkbox
- Kamra – checkbox
- Lift – checkbox
- Erkély – checkbox
- Galéria – checkbox
- Irányár (Ft) – beviteli mező
- Feladás dátuma (dátum) – date picker
- Eladva (igen / nem) –
- Végösszeg (Ft) – beviteli mező
- Eladás dátuma (dátum) – date picker
- Egyéb – szöveg mező

*Ingatlan típusok:

Az ingatlan rendszerben 12 különböző kínálat típust különböztetünk meg.

- Lakás, Családi ház
- Ikerház
- Sorház
- Iroda
- Üzlethelyiség
- Telephely
- Műhely
- Raktár
- Nyaraló
- Építési telek
- Fejlesztési terület

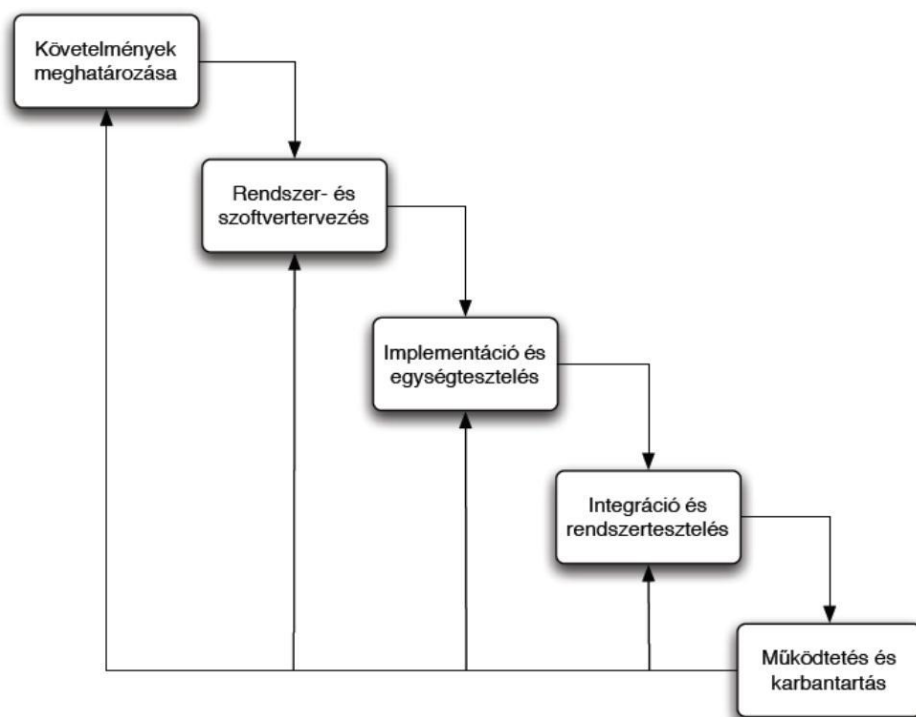
V. A SZOFTVERFEJLESZTÉS

A szoftvertervezés megkezdése komoly előkészületekkel és az ehhez kellő tudás elsajátításával jár. Ezáltal megfelelően precíz gondolkodást igénylő feladat. Munkám megkezdése előtt kiválasztottam egy a számomra megfelelő szoftverfejlesztési modellt, amely modell a munkámra legjobban jellemző, ezt felhasználom és segítségül hívom a későbbiekben.

Több ilyen életciklus modell is létezik, amelyet a fejlesztési és tervezési munkálatok során fel lehet használni. Mielőtt az általam kiválasztott modellt ismertetném, szeretnék kiemelni és bemutatni párat e modellek közül, jellemző tulajdonságaikon keresztül. Nem megfelelően az előnyök és hátrányok szemléltetéséről sem.

V.1. Szoftverfejlesztési modellek⁷

V.1.1. Vizesés modell



7. ábra(Vizesés modell)

⁷ A szoftverfejlesztési modelleknél bemutatott ábrák forrása: Ian Sommerville: *Software Engineering*, 2004, Gyula Simon 2005 (magyar verzió) 4.fejezet

1. A követelmények meghatározása
2. A rendszer és a szoftver megtervezése
3. Az implementációk tesztelése
4. Az integráció és a rendszertesztelés elvégzése
5. Működtetés és karbantartás

Talán az egyik legrégebbi életciklus modellek közt található meg a vízesés modell. A szoftverfejlesztés során felmerülő tevékenységek sorát, egymás után következő és egymástól jól elválasztható sorrendben írja elő. A modell ezeken a lépéseken lépcsőzetes sorrendben halad végig.

A vízesés modellnél a fejlesztőnek a tervezés korai szakaszában kell elköteleznie magát, sok nehezen változtatható részegységből áll, így nehezen lehet alkalmazkodni az ügyfél elvárásaihoz. Egy munkafázist be kell fejeznünk, mielőtt a következőre lépünk.

Nehézkesen lehet alkalmazkodni az ügyfél követelményeihez, ha azok idő közben megváltoznak. Azonban ha előre jól ismert minden követelmény, és csak nagyon kis változtatások fognak végbemenni a megalkotás során, akkor jól alkalmazható ez a modell. A gyakorlatban ez a tényező elég ritkán fordul elő. Ma már ezt a modellt meglehetősen elavultnak tartják.

V.1.2. Evolúciós fejlesztési modell

A modell megalkotásának alapelve, hogy kezdetben létrehozunk egy specifikációt a megrendelő igényeit szem előtt tartva. Ez először kipróbálásra és véleményezésre kerül a megrendelő részéről. Ezt a tesztelést befejezván, finomításokkal tökéletesítéseket hajtunk végre a kezdeti rendszeren, melynek következtében elő fog állni a végleges rendszer. A modell megközelítése kétfajta módon lehetséges.

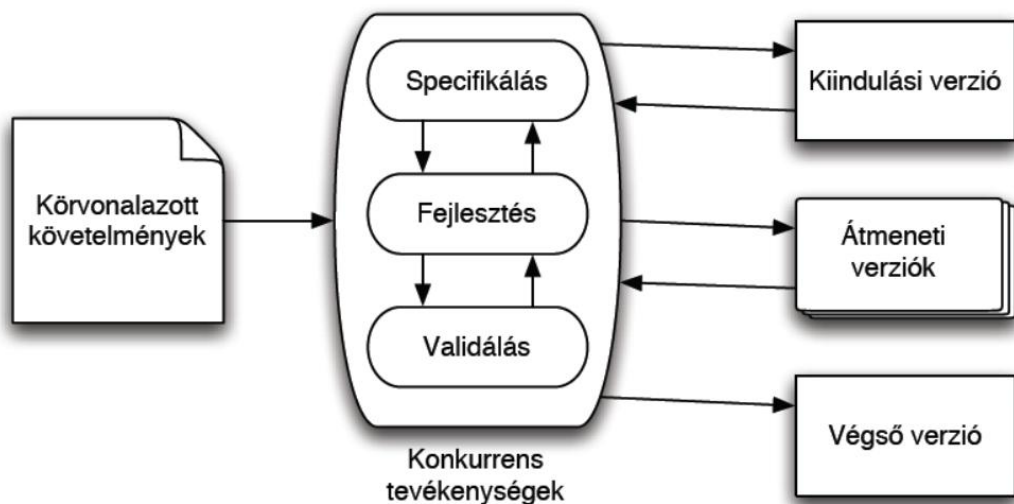
1. Kísérletező fejlesztés

Az egyik megközelítési módja az evolúciós modellnek a *kísérletező fejlesztés*. Első lépésként itt megemlíthetjük a követelmények listájának feltárását a megrendelőnkkel együttműködve, majd a következő lépésben megtörténik a kész szoftver megalkotása. Könnyen bővíthető új funkciókkal a későbbiekben, esetleges újabb igények felmerülése esetén.

2. Eldobható prototípus

A modell másik megközelítése az *eldobható prototípus*. Célja hogy tisztázzuk a homályos követelményeket. A megbízó elképzeléseinek megértésével kezdődik, és ennek segítségével alakul ki a későbbiekben a kész rendszer. Ennél a modellenél is adódhatnak problémák. A fejlesztés sokszor átláthatatlanná válik a folyamatos módosításoknak köszönhetően, ezáltal rosszul strukturált lesz a rendszerünk. A későbbiekben ebből, és a megfelelő dokumentálás hiányából adódóan, karbantartási nehézségekbe ütközhetünk.

Mint minden más modellenél az evolúciós fejlesztési modellenél is kialakulhatnak problémák a tervezés és kivitelezés során. A folyamatos változtatásoknak köszönhetően, a rendszer szerkezete javarészt átláthatatlanná, és rosszul strukturálttá válik. A gyorsaságot szem előtt tartva más olyan tényezők, mint a pontos dokumentálás, a rendszer megbízhatósága és biztonságossága háttérbe szorulnak. Így a specifikációk hiányából adódóan a későbbi karbantartás megdrágul és megnehezül. Azonban, ha egy rövid élettartamú kisebb rendszert szeretnénk készíteni, esetleg egy nagyobb rendszer kisebb részeként lesz felhasználva az alkotásunk, akkor megfelelő lehet számunkra ez a szoftverfejlesztési modell is.

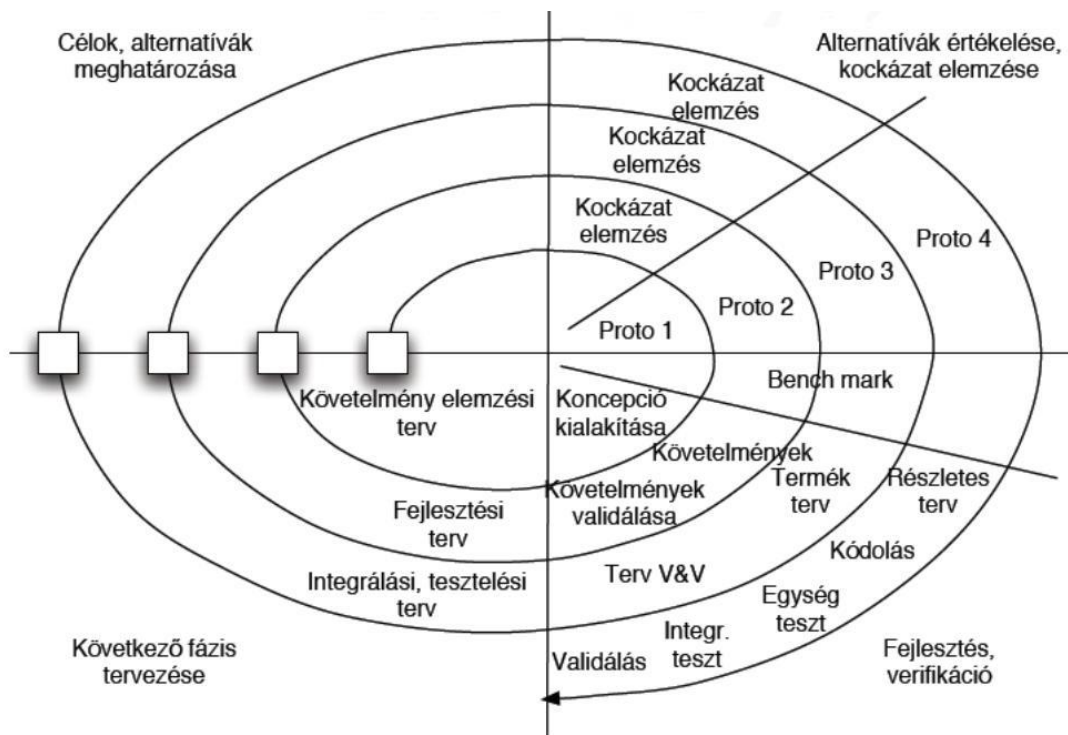


8. ábra(Evolúciós fejlesztési modell)

V.1.3. Spirális modell

A spirális szoftverfejlesztési modell azt az alapötletet használja ki, hogy a szoftverfolyamatra nem úgy tekint, hogy fázisokból és esetlegesen a közöttük fennálló tevékenységek sorozatából áll, hanem egy spirál reprezentációjaként. Amely spirálban megtalálható körök egy-egy fázisnak felelnek meg. Nem találhatóak benne fix hurkok, ezeket az igényekhez mérten mi alakítjuk ki a későbbiekben.

1. Meghatározzuk a célokat és az alternatívákat.
2. Kockázatok megbecslése és azok csökkentése.
3. A következő szintű termékfejlesztés és termék validálás.
4. A következő fázisok megtervezése.



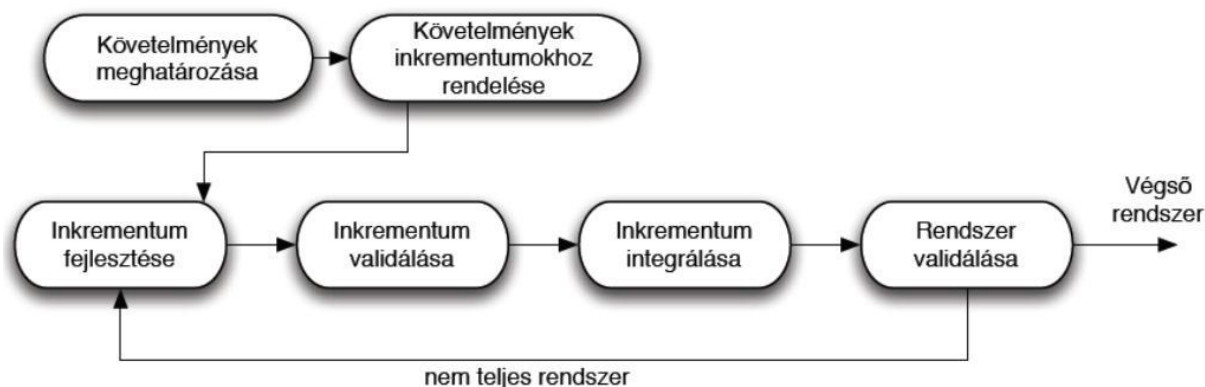
9. ábra (Spirális modell)

V.1.4. Inkrementális fejlesztési modell

Az igényelt funkcionalitás biztosítása inkremensekben történik. A korai inkremensekben a magas prioritással rendelkező szolgáltatásokat kell implementálni. A rendszert fejlesztés közben is lehet használni, így nem kell megvárni a teljes rendszer elkészültét. A korábban megalkotott inkremensek, mint prototípusok a későbbiekben újra felhasználhatóak.

Nagyobb a valószínűsége ennek következtében a tökéletes rendszer megalkotásának. Az elrontás valószínűsége nagyon alacsony. A legfontosabb komponenseknek több tesztelési lehetőséget biztosít.

1. Meghatározzuk vázlatosan a követelményeket (számít a fontossági sorrend közöttük)
2. Inkremenseket rendelünk a követelményekhez
3. A rendszer architektúra megtervezése
4. Fejlesztés
5. Validálás
6. Inkremens integrálás
7. Rendszer validáció



10. ábra(Inkrementális fejlesztési modell)

V.2. Szoftverfejlesztési modell választása

A szakdolgozatomban megtalálható ingatlanközvetítő-rendszer megtervezése előtt, az előzőekben bemutatott szoftverfejlesztési modellek előnyeit, és hátrányait is mind figyelembe vettem. Ezen információk segítségével az inkrementális szoftverfejlesztési modell felhasználása mellett döntöttem. Azért is esett erre a modellre a választásom, mert a felsorolt alkotó tulajdonságai nagyon jellemzőek voltak rám. Egy példán keresztül szeretnék betekintést nyújtani, miért is ezt a modellt választottam.

Az ingatlanhoz tartozó képek tárolásának kérdésénél több fajta ötlet is felmerült bennem. Az első gondolatom az volt, hogy az adatbázisban létrehozok egy képek táblát és abban fogom tárolni a később feltöltéskor bevitt képeket. Azonban, sok olyan felesleges tulajdonságot tároltunk volna így módon, amire a későbbiekben nem lesz szükségünk.

A lepraktikusabb és a legkényelmesebb megoldás alkalmazása mellett döntöttem. A képek feltöltésekor, a rendszerben (egy adott képfeltöltéseket tároló mappában) létrehozunk egy új mappát, az ingatlan azonosítójának nevével. Az azonosítóról tudjuk, hogy egyedi, így ez által tudunk hivatkozni az ingatlan képeire, és ezzel elkerülhetjük az esetleges kavargásokat, hogy minden ingatlan képeit rendezetten külön mappákban tároljuk. A képek feltöltésekor létrehozásra kerül egy tmb nevű almappa is, amely a megjelenítéseknél szükséges bélyegképeket tárolja.

Ebből a példából is jól látható, hogy a rendszert vázlatosan, pontokba szedve terveztem meg, ahogyan az a modell leírásában is megtalálható. Minden olyan hibalehetőséget szem előtt tartottam, amely a későbbiekben a rendszer működésének hanyatlásához vezetne. Fejlesztés közben teszteltem a rendszer bizonyos részeit és javításokat hajtottam végre az egyes inkrementeken, azonban az előbbiekhöz hasonló kérdések a munkám során többször is előfordultak és felmerültek.

VI. AZ ADATBÁZIS TERVEZÉS

VI.1. Az ER modell

Az ER modell a valóság egy modellezni kívánt darabját mutatja be. Az ER modell szolgál a később elkészítésre kerülő adatbázis alapjául. Az ER modellt alkalmazva lehet minden - az adatbázissal kapcsolatban felmerült - kérdést tisztázni. A modell megalkotásához nem szükséges bármiféle törvényt betartanunk, mert az elkészítéséhez a saját elképzeléseink szolgálnak alapul. A későbbi adatbázis megalkotása során, a körülmények megváltozása esetén a valós modell is változik.

VI.1.1. Az egyedtípusokról

Gyenge egyedtípusok:

Azokat az egyedtípusokat, amelyek nem rendelkeznek saját kulcs attribútumokkal, *gyenge egyedtípusok*nak nevezünk. A *gyenge egyedtípusok*nak részleges kulcsuk van, amely azon attribútumok halmaza, amelyek egyértelműen azonosítják azokat a gyenge egyedeket, amelyek ugyanazon tulajdonos egyedhez kapcsolódnak.

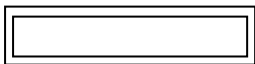
A gyenge egyedtípusok jelölése



Erős egyedtípusok:

A hagyományos egyedtípusokat, amelyeknek van kulcs attribútumuk, *erős egyedtípusok*nak nevezünk.

Az erős egyedtípusok jelölése



Feltételezzük, hogy:

- Egy hirdetőhöz több ingatlan is tartozhat
- Egy adminhoz több ingatlan is tartozhat
- Egy-egy településhez több ingatlan is tartozhat.
- Egy-egy kínálat típus több ingatlanra is jellemző lehet
- Egy-egy fűtés típus több ingatlanra is jellemző lehet

Az adatbázis egyedtípusai:

- Adminisztrátorok (**ADMIN** egyedtípus)
- Ingatlanok (**KÍNÁLAT** egyedtípus)
- Hirdetők (**HIRDETŐ** egyedtípus)
- Települések (**TELEPÜLÉSEK** egyedtípus)
- Kínálat típusok (**TÍPUS** egyedtípus)
- Fűtés típusok (**FŰTÉS** egyedtípus)

Az adatbázis attribútumai:

ADMIN (admin_id, a_nev, a_jelszo, a_email, a_telefon, a_felhnev)

KÍNÁLAT (ingatlan_id, hirdeto_id, admin_id, tipus_id, futes_id, telepules_id, azonosito, aktiv, feladas_datum, iranyar, eladva, eladva_ar, eladva_datum, irsz, utca, hazszam, hrsz, kor, anyag, lakas_emelet, komfort, alapterulet, szobaszam, kozos_koltseg, telek_terulet, lift, galeria, pince, kocsibeallo, csatorna, kamra, garazs, víz, gaz, telefon, internet, futes, riaszto, villany, kulon_vizora, kabletv, erkely, kiemelt, kinal_tip, egyeb)

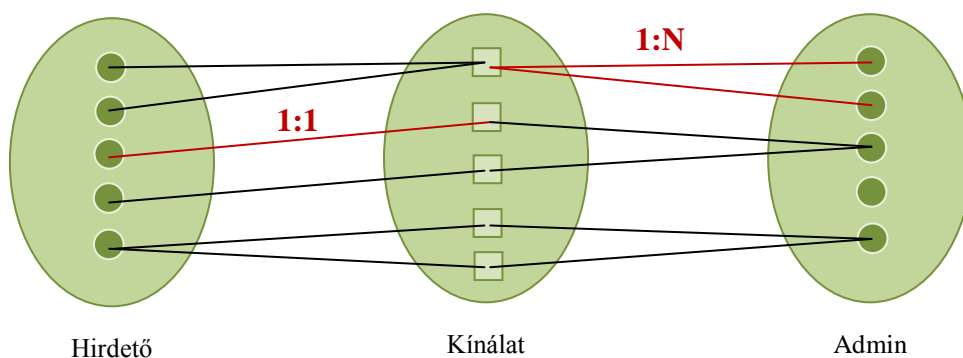
HIRDETŐ (hirdeto_id, h_nev, h_telepules, h_utca, h_hazszam, h_irsz, h_email)

TELEPÜLÉSEK (telepules_id, telepulesek)

TÍPUS (tipus_id, tipusok)

FŰTÉS (futes_id, futes_tip)

VI.1.2. Az egyedtípusok közti kapcsolatok szemléltetése

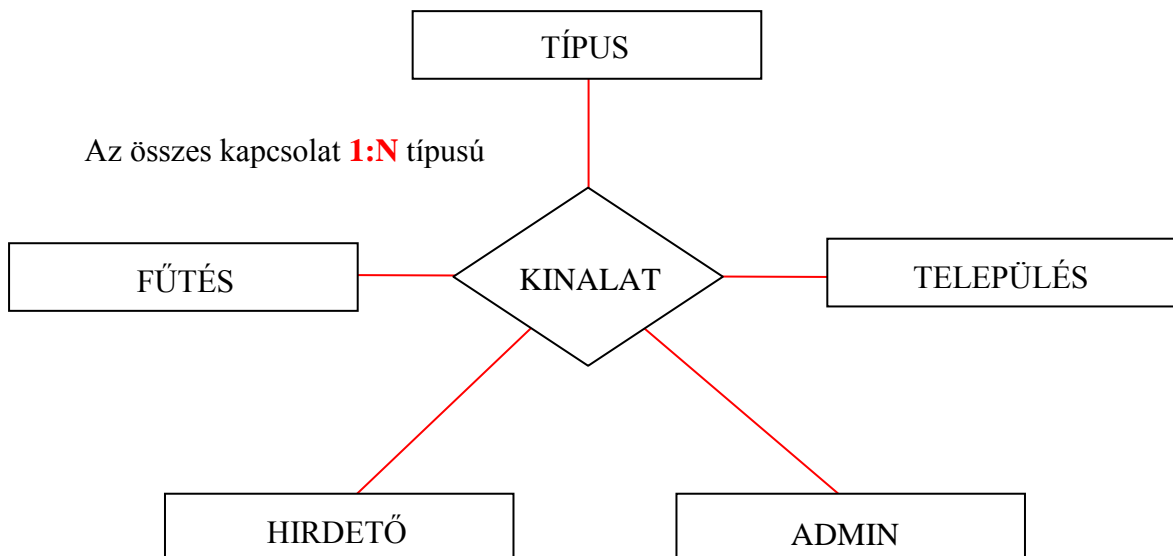


11. ábra (Az egyedtípusok közötti kapcsolat szemléltetése)

Az egyedtípusokon belül (az ábrán a kis körök, és kis négyzetek szemléltetik) találhatóak a valós világ elemei. Az egyedtípusok között *teljes részvétel*⁸ található. Hiszen minden ingatlanhoz tartoznia kell egy hirdetőnek és egy adminisztrátornak.

Az ER modellen belül többfajta kapcsolattípus is létezik. Az 1:1⁹, az 1:N¹⁰, az N:M kapcsolattípusok. A dolgozatomban megalkotott adatbázisra az 1:1 (egy az egyhez) és az 1:N (egy a többhöz) típusú kapcsolattípusok a legjellemzőbbek.

VI.1.3. A ER modell megalkotása



⁸ *Teljes részvételnél egy kapcsolatban lévő egyedtípus minden egyede részt vesz a kapcsolatban. A teljes részvételt egyenes vonallal, míg a parciális részvételt szaggatott vonallal szokás jelölni.*

⁹ *Egy-egy kapcsolatban az egyik egyedhalmaz minden egyes elemének, csak egy elem felel meg a másik halmazból. Ez a tulajdonság visszafelé is értendő.*

¹⁰ *Egy-sok kapcsolatban az egyik halmaz elemeinek, több elem is megfelelhet a másik halmazból.*

VI.2. A normalizálásról

Normalizálásnak nevezhető az a folyamat melynek során az adatbázisban található adatokat, ésszerű sorrendbe és csoportosításba rendezzük bizonyos szabályok betartásával. Ezeknek a szabályok az adatbázisunk fenntartását hivatottak megkönnyíteni. A normalizálás céljának tekinthetjük az adatbázis optimalizálását, tehát ha eleget tesz a kimondott feltételeknek, akkor az adatbázisban jelentkező *redundancia*¹¹ elkerülhető és a táblák megfelelő kapcsolatban fognak állni egymással. Célunk a normalizálással, hogy az ingatlan adatbázisba bekerülő adatokat, olyan relációkra osszuk fel, melyekben a redundancia a lehető legkisebb legyen. Így későbbi bővítésük rugalmasan elvégezhetővé válik. A normalizációs lépéseket végigkövetve, egy hatékony és jól áttekinthető adatmodellt fogunk kapni.

VI.2.1. A normálformák ismertetése

A normalizáláskor felhasznált szabályokat **normálformáknak** nevezzük, ezek az adatbázis belső szerkezetére nagyon jellemző tulajdonságok. Ha az általunk készített adatbázis eleget tesz bizonyos feltételeknek, akkor azt mondhatjuk, hogy valamelyik normálformában van.

Ha az első csoport által felállított szabályoknak felel meg az adatbázis, akkor azt mondjuk, hogy *első normálformában* van. Ha az első két szabálynak megfelel az adatbázisunk, akkor *második normálformában* van. Céлом a tervezési munkám során, a megalkotott adatbázis második normálformára hozása. További normalizálási szintek is léteznek, azonban a legtöbb esetben a harmadik normálforma a szükséges legmagasabb szint. A normalizálás eléréséhez új táblák létrehozásai szükségesek az adatbázisban.

A valós életben jelentkező sok esetben nem tudunk azonban tökéletesen megfelelni a felállított szabályoknak.

¹¹ *Redundancia: Redundanciáról akkor beszélünk, ha valamely adatokat többszörösen tároljuk az adatbázisban. A redundanciának köszönhetően az adatbázis inkonzisztens állapotba kerülhet.*

1. Első normálforma

Egy R relációséma első normálformában (1NF-ben) van, ha R minden attribútuma teljesen funkcionálisan függ ¹² az elsődleges kulcstól.¹³

- A tábla minden olyan adathalmazához, melyek kapcsolódnak egymáshoz új táblát kell létrehozni.
- A tábla minden olyan adathalmazát, melyek kapcsolódnak egymáshoz elsődleges kulcs azonosítsa.
- Egy táblán belül annak elkerülése, hogy több mező hasonló adatot tartalmazzon.

Ha egy reláció 0.NF-ben van, több módon is megtehetjük azt, hogy 1.NF-re hozzuk. Az egyik módja, hogy több attribútum értékkel rendelkező sorokat annyi sorra bontjuk szét, amennyi a benne lévő attribútum értékek száma. Ebből következik, hogy a 0.NF-ben található elemekből, kiválasztjuk az egymáshoz kapcsolódókat és külön táblába tesszük őket. Így a következő táblák kerülnek létrehozásra: Kínálat, Hirdető, Admin, Fűtés, Típus és Települések.

2. Második normálforma

Egy R relációséma második normálformában (2NF-ben) van, ha R minden másodlagos (leíró) attribútuma teljesen funkcionálisan függ R elsődleges kulcsától.¹⁴

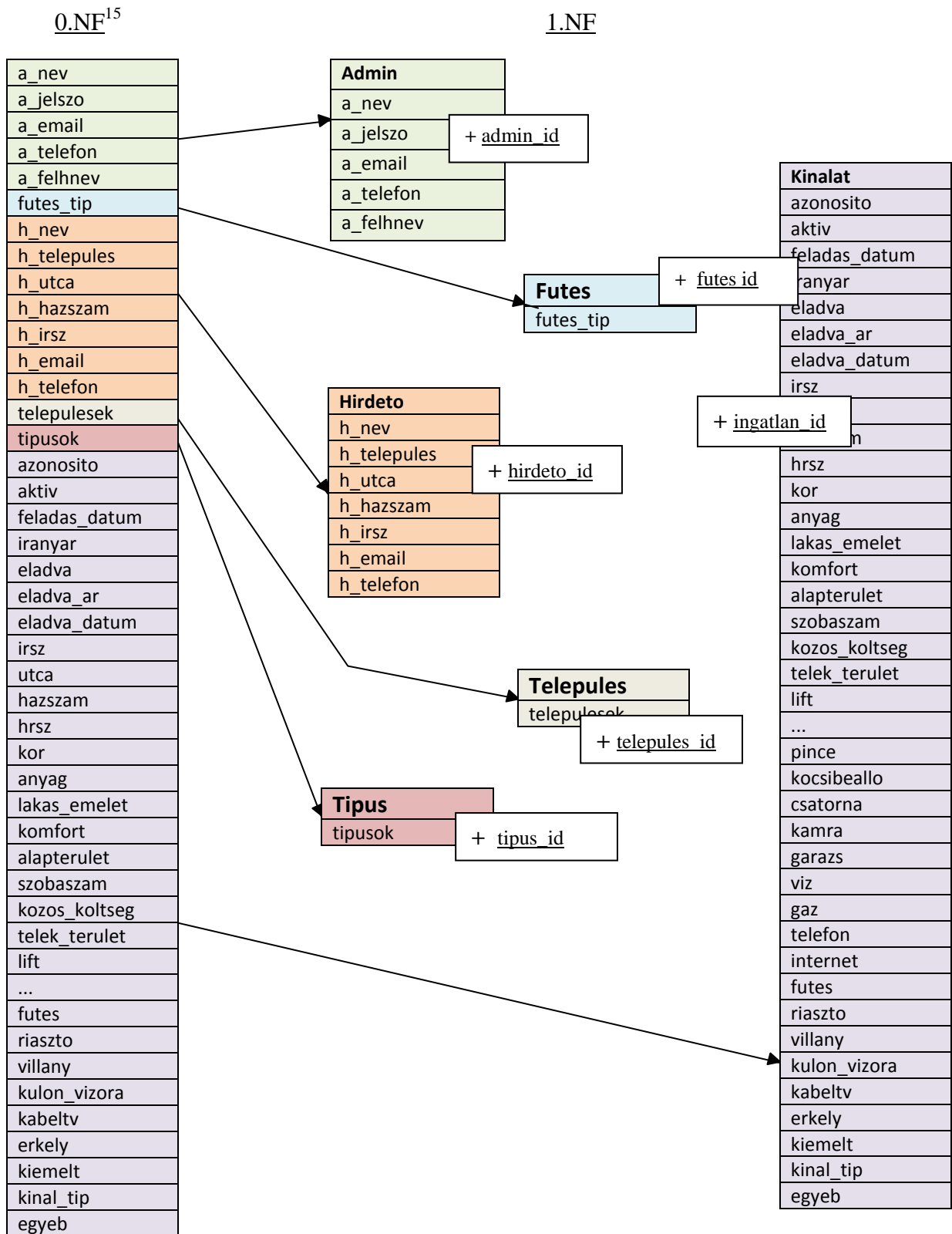
A 2.NF-re hozást a létrehozott táblák további táblákra való szétbontásával hajtjuk végre.

- Külön táblákat kell létrehozni a több rekordra is jellemző értékekhez.
- A létrehozott táblákat külső kulccsal kell összekapcsolni.
- A tábla rekordjai csakis a tábla elsődleges kulcsától függhetnek.

¹² Funkcionális függés: Egy egyeden belül az a tulajdonság funkcionálisan meghatározza B tulajdonságot, ha minden esetben azonos A tulajdonság-érték mellett ugyanaz a B tulajdonság-érték szerepel.

¹³ Kulcs: A kulcs attribútumainak értékei egyértelműen meghatározzák az egyed minden értékét.

¹⁴ A normálforma szabályok forrásai Kósa Márk – Adatbázisrendszerek jegyzet 2009, Szilágyi József – Adatbázis-kezelési alapok 2003.

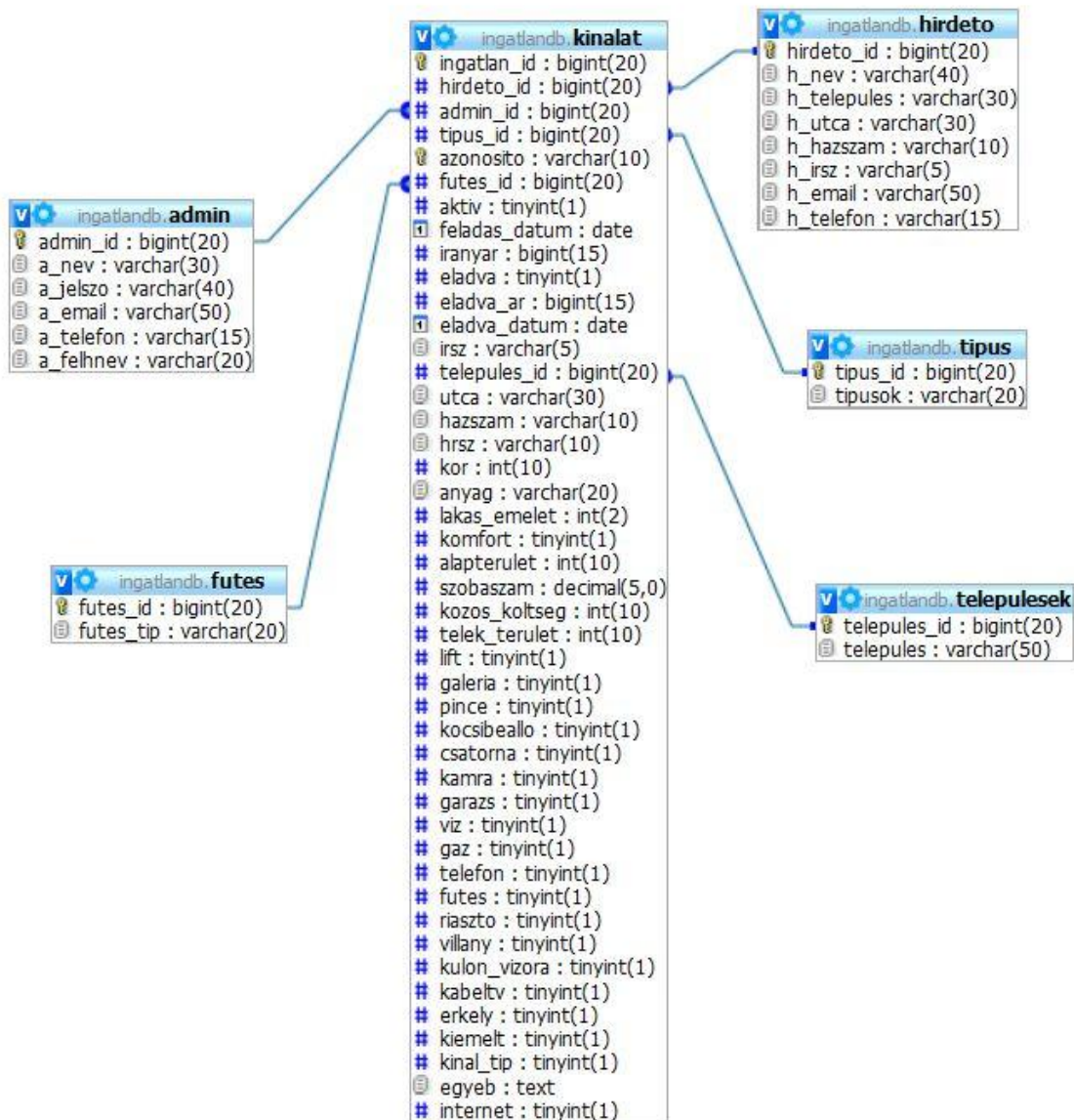


12.ábra (Normalizálás 0.NF, 1.NF)

¹⁵ Az ábrán található „...” a rövidítés célját szolgálja, hogy az ábra egy oldalon belül elférjen. A „...” alatt található még 10 'tinyint (1)' mező. Azonban ezek az ábra megértéséhez nem szükséges információk.

A 0.NF-ben található táblából, kiválasztjuk az egymáshoz kapcsolódó adathalmazát és külön táblába tesszük őket. A tábla minden adathalmazát, melyek kapcsolódnak egymáshoz elsődleges kulcs azonosítja. A Kínálat, Hirdető, Admin, Fűtés, Típus és Települések táblákat ellátjuk elsődleges kulcsokkal. Az ábrán így jól megfigyelhető a táblák közötti kapcsolat.

2.NF



12. ábra (Normalizálás 2.NF)

Adaptálás MySQL-be

A normalizációs lépéseket elvégezve, egy 2.NF-ben lévő adatbázist kapunk eredményül. A következő lépés az adatok bevitele és a hozzájuk tartozó szükséges értékek beállítása a fizikai adatbázisban, tehát esetünkben a MySQL-ben. A könnyebb adatbevétel érdekében alkalmaztam a phpMyAdmin nevű, php nyelven írt webes alkalmazást, amely segítségével könnyebben tudjuk kezelni a MySQL adatbázisunkat.

VI.3. Az ingatlan adatbázis

Ebben a részben szeretnék betekintést nyújtani az ingatlanközvetítő rendszerhez készített ingatlandb.sql adatbázisba, bemutatva az ott megtalálható táblákat és a mezőket. Az adatbázisban hat különböző tábla található meg, ezek tartalmazzák a rendszer működéséhez szükséges mezőket.

Az adatbázisban megtalálható táblák a következők:

- admin: az adminisztrátorok adatainak tárolása
- hirdeto: a hirdetők adatainak tárolása
- futes: a fűtés típusok tárolása
- telepules: a települések listájának tárolása
- tipus: az ingatlan típusok tárolása
- kinalat: az ingatlanokhoz kapcsolódó adatok tárolása

1. Admin tábla:

Mező	Típus	Illesztés	Extra
<u>admin_id</u>	bigint(20)		auto_increment
a_nev	varchar(30)	utf8_hungarian_ci	
a_jelszo	varchar(40)		
a_email	varchar(50)		
a_telefon	varchar(15)		
a_felhnev	varchar(20)		

Az admin táblában kerülnek tárolásra mindazon adatok, melyeket szükség szerint kell tárolni az adminisztrátorokról. Rendelkezik admin_id mezővel, amely a könnyebb beazonosíthatóságot és hivatkozhatóságot szolgálja. Ez a mező egy plusz auto_increment tulajdonsággal rendelkezik, tehát új admin hozzáadása esetén a táblába, értéke egyel mindig nő. Az ékezetes betűk felmerülésénél az a_nev táblában az utf8-as kódolás lett beállítva, az esetleges karakter elrontás elkerülése végett.

2. Fűtés tábla:

Mező	Típus	Illesztés	Extra
<u>futes_id</u>	bigint(20)		auto_increment
futes_tip	varchar(20)	utf8_hungarian_ci	

A futes táblában találhatóak meg azok a fűtés típusok, melyek közül választani lehet majd ahhoz mérten, hogy melyik található meg a hirdetésben szereplő ingatlanban. Rendelkezik futes_id mezővel, amely auto_increment típusú. Illetve utf8 kódolás található meg a fűtés típusok tárolására szolgáló futes_tip mezőben.

3. Hirdető tábla:

Mező	Típus	Illesztés	Extra
<u>hirdeto_id</u>	bigint(20)		auto_increment
h_nev	varchar(40)	utf8_hungarian_ci	
h_telepules	varchar(30)	utf8_hungarian_ci	
h_utca	varchar(30)	utf8_hungarian_ci	
h_hazszam	varchar(10)		
h_irsz	varchar(5)		
h_email	varchar(50)		
h_telefon	varchar(15)		

A hirdető tábla tartalmazza mindazon alapvető adatokat, melyek szükségesek az ingatlan közvetítő rendszerben való hirdetés feladásához. Tartalmaz hirdeto_id mezőt a könnyebb hivatkozhatóság érdekében, amely auto_increment típusú. A h_nev, h_telepules és h_utca mezőkhöz utf-8 kódolást állítottunk be, mert-e mezőkben ékezetes betűk is szerepelhetnek.

4. Települések tábla:

Mező	Típus	Illesztés	Extra
<u>telepules_id</u>	bigint(20)		auto_increment
telepulesek	varchar(50)	utf8_hungarian_ci	

A megfelelő gyorsaság elérése érdekében, létrehozásra került egy települések tábla, amely az összes magyarországi települést tartalmazza. Illesztésnél utf8as kódolás és auto_increment típus lett beállítva a telepules_id mezőnél..

5. Típus tábla:

Mező	Típus	Illesztés	Extra
<u>tipus_id</u>	bigint(20)		auto_increment
tipusok	varchar(20)	utf8_hungarian_ci	

A rendszerben megtalálható ingatlanok típusait tartalmazza. A tipus_id auto_increment értékű, új típus hozzáadása esetén értéke egyel nő. A típusokat tartalmazó mezőben utf8 kódolást állítottunk be.

6. Kínálat tábla:

Mező	Típus	Illesztés	Extra
<u>ingatlan_id</u>	bigint(20)		auto_increment
hirdeto_id	bigint(20)		
admin_id	bigint(20)		
tipus_id	bigint(20)		
futes_id	bigint(20)		
telepules_idűd	bigint(20)		
azonosito	varchar(10)	utf8_hungarian_ci	
aktiv	tinyint(1)		
feladas_datum	date		
iranyar	bigint(15)		
eladva	tinyint(1)		
eladva_ar	bigint(15)		
eladva_datum	tinyint(1)		
irsz	varchar(5)		
utca	varchar(30)	utf8_hungarian_ci	
hazszam	varchar(10)		

hrsz	varchar(10)		
kor	int(10)		
anyag	varchar(20)	utf8_hungarian_ci	
lakas_emelet	int(2)		
komfort	tinyint(1)		
alapterulet	int(10)		
szobaszam	decimal(5,0)		
kozos_koltseg	int(10)		
telek_terulet	int(10)		
lift	tinyint(1)		
galeria	tinyint(1)		
pince	tinyint(1)		
kocsibeallo	tinyint(1)		
csatorna	tinyint(1)		
kamra	tinyint(1)		
garazs	tinyint(1)		
viz	tinyint(1)		
gaz	tinyint(1)		
telefon	tinyint(1)		
internet	tinyint(1)		
futes	tinyint(1)		
riaszto	tinyint(1)		
villany	tinyint(1)		
kulon_vizora	tinyint(1)		
kabeltv	tinyint(1)		
erkely	tinyint(1)		
kiemelt	tinyint(1)		
kinal_tip	tinyint(1)		
egyeb	text	utf8_hungarian_ci	

A kínálat táblában találhatóak meg mindazon adatok, melyek az ingatlanokhoz tartozó adatokat tartalmazzák. Az ingatlan_id elsődleges kulcs értékkel rendelkezik. Azonosítja az ingatlanokat, auto_increment értékű, így új ingatlan hozzáadása esetén értéke mindig egyel nő. Több id érték is tárolásra került ebben a táblában (hirdeto_id, admin_id, tipus_id, futes_id, telepules_id), amelyek segítségével másik táblákra hivatkozunk. Ezzel mind a gyorsaságot, mind pedig az átláthatóságot és rendezettséget szolgálva. Tartalmazza, hogy melyik ingatlan melyik hirdetőhöz és adminisztrátorhoz tartozik.

VI.4. A rendszerrel szemben támasztott követelmények

Az első legfontosabb teendő a felhasználói követelményrendszer megfogalmazása. Mire és hogyan fogják használni az üzemeltetők és a látogatók a kialakított alkalmazást. Az ingatlan közvetítő alkalmazás célkitűzése, hogy a hagyományos értelemben vett irodában történő ingatlanközvetítés munkavégzését megkönnyítse, azonban azt működése közben semmiben se gátolja meg.

A meghatározásra került kritériumok közt szerepet kap, hogy az internet használatának bevonásával a cég még nagyobb ügyfélkört alakítson ki, így még sikeresebbé és jövedelmezőbbé tegye vállalkozását. Az rendszer megalkotásának - talán egyik legfontosabb - célja, hogy megkönnyítse, és megfelelő mértékben átvállalja a tárolás közben felmerülő nehézségeket és azokat az adminisztrációs folyamatokat melynek során a hirdetők és vásárlók adatait az irodában, úgymond elkönnyvelik. Célunk tiszta átláthatóságot és megkönnyített munkavégzést biztosítani.

VI.4.1. A rendszer biztonságáról

A rendszerben szükség van megerősített biztonság beépítésére, avégett, hogy az adatbázisban található adatokat illetéktelenek ne láthassák és ezekkel a személyes adatokkal ne lehessen jogtalanul visszaélni. A felhasználói felületre ez nem terjed ki a rendszergazdai felülettel szemben. Az oldalt böngészők számára, semmilyen módszerrel nem lehetséges a beviteli oldalak bármelyikének elérése is. A beviteli mezők számát minimálisra kell csökkenteni, és ezzel elkerülhetővé kell tenni az esetlegesen az adatbázist módosító parancssorral elvégezhető csalások futtatását. A kevés beviteli mező segítségével, így még a lehetőség sem biztosított az effajta csalások elvégzésére.

Minden adminisztrátori oldal kódjában elhelyezésre kerül, egy olyan parancssor mellyel az előzményekben található hivatkozásokra kattintva sem érhető el az adott oldal tartalma, csak a bejelentkezést követően. Az adatbázisban, a bejelentkezéshez szükséges jelszó tárolását titkosított formában kell elhelyezni, a direkt módon történő tárolással szemben.

VI.4.2. A rendszer megbízhatóságáról

Az alkalmazást úgy kell elkészíteni, hogy az esetlegesen más operációs rendszert, vagy más típusú web böngészőt használó ügyfelek is meg tudják nyitni és jeleníteni a saját számítógépükön. Nem szükséges az egész böngésző kínálatot lefedő kompatibilitás biztosítása. Azonban, a legtöbbet, leggyakrabban használt, divatos és elterjedt böngészőknek, kezelniük kell tudni a megalkotott oldalt. Biztosítani kell még ezen felül, az oldal megjelenítésének optimalizálását a különböző monitor típusokra és képernyő felbontásokra.

VI.4.3. A rendszer követelmények pontokba szedve

A rendszerkövetelmények a megbízó segítségével lettek kialakítva, ezek mutatták meg a kellő irányt az oldal felépítéséhez és az igényelt funkciók megalkotásához. A későbbi adatbekérések mintájául a jól megszokott, papír alapú adatbekérő nyomtatványt használtam fel.

- Az oldal megalkotásakor biztosítani kell, egy felhasználói és egy (csak is a cég munkatársai számára elérhető) adminisztrátori felületet. Az adminisztrátori felület eléréséhez egy biztonsági oldalt kell megalkotni, ahol a bejelentkezésre váró munkatársak megadhatják az azonosításhoz szükséges adataikat.
- Lehetőség szerint a munkamenet megszakítását is ki kell alakítani az adminisztrátori felületen, ezáltal lehetővé kell tenni a kijelentkezést.
- A felhasználói felületet mindenki számára elérhetővé és böngészhetővé kell tenni.
- Regisztrációra nincs szükség a keresések és az ingatlan lista adatainak megtekintéséhez. Sok érdeklődő a regisztráció láttán, inkább új oldalt keres, ahol szabadon böngészhet, minthogy a regisztrációs folyamattal bajlódjon.
- Az oldalon ki kell alakítani egy kereső ablakot, amiben tetszőleges adatokat lehessen megadni, ezáltal biztosítva a még pontosabb találat elérését.
- A keresési eredményeket formázott állapotban, az ingatlanra legjellemzőbb adatokat megjelenítve kell egy listában kiírni.

- Új hirdetőket regisztrálására a rendszerben, biztosítani kell egy regisztrációs űrlapot.
- Az adminisztrátor jogosultsággal rendelkező felhasználóknak minden ingatlan közvetítői feladatot el kell látniuk, amely a munkájukhoz kapcsolódik.
- Lehetőséget kell nyújtani, új ingatlanok bevitelére az adatbázisba, amelyeken később, esetleges adat módosításokat lehet végrehajtani.
- Képek feltöltését is lehetővé kell tenni az ingatlanokhoz.
- A felhasználói jogosultsággal rendelkezők, az oldalon, szabadon böngészhetnek, és esetlegesen lekérdezéseket futtassanak, különböző keresési feltételeket megadva.
- A cég munkatársainak kezelniük kell a hirdetőket listáját. Azokból esetlegesen partnereket kell tudniuk törölni.
- A keresési eredményeket, és a teljes ingatlan listát lapozhatóvá kell tenni, illetve minden olyan oldalon biztosítani kell a lapozhatóságot, ahol több adatot szeretnénk kilistáztatni, és azt esztétikai szempontból több oldalon szeretnénk megvalósítani. Szem előtt tartva a könnyű és tiszta átláthatóságot.
- Az ingatlanokat böngésző ügyfelek számára biztosítani kell egy olyan lehetőséget, hogy egy megfelelően tűnő ajánlat esetén, bővebb adatokat tudjanak előhívni.
- Galéria elhelyezése, melyben az ingatlanhoz kapcsolódó képeket lehet böngészni, ennek segítségével még közelebb kerülhessen az érdeklődő az adott ingatlanhoz.
- Minden ingatlan iránt érdeklődőnek biztosítani kell lehetőséget a kapcsolatfelvételre, a közvetítő cég munkatársainak valamelyikével.
- Lehetőséget kell biztosítani arra az adminisztrátorok számára, hogy az esetleges adat eltávolításokat, el tudják végezni az adatbázisban.
- A jogosultsággal rendelkezőknek, lehessen hirdetőt és ingatlant törölni a rendszerből.

VI.5. A közvetítés menete

VI.5.1. A hirdető

Első lépésként az oldalon hirdetni kívánó ügyfelek, felkeresik az oldalt üzemeltető munkatársak valamelyikét, ahol jelzik szándékukat azzal kapcsolatban, hogy a rendszerben megtalálható ingatlantípusok valamelyikével rendelkeznek, és bérbe szeretnék adni, vagy el szeretnék árúsítani azt. A formai szerződés megkötésével és a kívánt ingatlan adatainak átszolgáltatásával a közvetítő cég megkezdheti a már jól ismert munkáját. A formai követelményeket teljesítve, az ingatlan regisztrációját követően az ingatlan megjelenik az oldalon kínált ingatlanok listájában, kereshetővé válik és esetlegesen az árlistában megjelölt ár befizetésével a 'kiemelt' ingatlanok listájába is bekerülhet.

VI.5.2. A vásárlók

Az oldal eközben már böngészhető állapotba kerül a felhasználók számára. A lehetséges vevők a kínálat listát tallózva, és az esetleges keresési feltételeket lefuttatva válogathatnak a találatok közt. A számukra megfelelőnek tűnő ingatlant kiválasztva leadhatják az érdeklődésüket az ingatlan iránt. A megjelölt ingatlan azonosítóra hivatkozva a bővebb információkat szerezhetnek azokról az ingatlanokról, melyek elnyerték a tetszésüket.

VI.5.3. Az adminisztrátorok

Az alkalmazásban, csakis az adminisztrátori jogosultsággal rendelkező munkatársak láthatják az ingatlant meghirdető ügyfelek listáját és az adatbázisban szereplő ingatlanokat.

Ezen adatokon, a munkavégzést is a cég munkatársai végzik el. Ezáltal kerül biztosításra, hogy az alkalmazás használatával nem lehet az ingatlan közvetítő irodát kivonni az üzlet lebonyolításának folyamatából. A lehetséges vevőknek legelőször, az ingatlan közvetítő irodától egy időpontot kell kérniük, (erre lehetőség van telefonon és e-mailben is), a telefonos konzultációban az ingatlan azonosító számára kell hivatkozniuk. Majd ezt követően a személyes konzultáció után lehetőség nyílik egy szakemberrel a kiválasztott ingatlan meglátogatására, és annak felmérésére.

VI.6. A felhasználói csoportok bemutatása

VI.6.1. A látogatók

A felhasználóknak lehetősége van az oldalt szabadon látogatni, az ingatlan kínálat listából adatokat lehívni, különböző keresési feltételeket megadva böngészni a találatok közt, esetleg azokat szűkíteni, a még pontosabb eredmény megtalálása érdekében. Lehetőség nyílik az oldallal kapcsolatos észrevételekkel felkeresni az üzemeltető céget, illetve ha érdeklődést tanúsítanak egy ingatlan iránt. A munkatársak mindig készséggel állnak rendelkezésre.

VI.6.2. Az adminisztrátorok

Az adminisztrátorok hatásköre bővebb, mint az egyszerű látogatóké, azonban a bejelentkezéshez egy felhasználónévvel és egy jelszóval kell azonosítania magát. Ha sikeres bejelentkezést hajt, végre akkor szabadon végezheti a munkáját a webes felületen, amely a gyakorlatban már jól megszokott.

Az adminisztrátor által elvégezhető feladatok:

- Új ingatlan bevitele az adatbázisba.
- Az adatbázisban szereplő ingatlanok adatszolgáltatásának végrehajtása.
- Ingatlan törlése a rendszerből.
- Új adminisztrátor hozzáadása a rendszerhez.
- Új hirdető bevitele a rendszerbe.
- Hirdető törlése a rendszerből.

Hirdető törlésével kapcsolatban egy olyan megszorítás lett beépítve az alkalmazásba, amely meggátolja annak lehetőségét, hogy hirdetőt lehessen törölni a rendszerből, ha tartozik hozzá meghirdetett ingatlan. A törlést megelőzően egy megerősítő oldalra irányítjuk át az admin, és ott felkínálunk neki egy olyan lehetőséget, hogy a hirdetőt és a hirdetőhöz kapcsolódóan az összes hirdetését is mind eltávolíthatjuk.

VI.7. A felhasználói felület

VI.7.1. Keresések

Három különböző keresési módra van lehetőségük a rendszerben a látogatóknak.

1. Egyszerű keresés:



13. ábra (Egyszerű keresés)

- Eladó-kiadó – legördülő lista
- Típus – A rendelkezésre álló ingatlan típusok
- Település – A rendszerben megtalálható települések listája

2. Részletes keresés:



14. ábra (Részletes keresés)

- Eladó-kiadó – legördülő lista
- Típus – A rendelkezésre álló ingatlan típusok
- Település – A rendszerben megtalálható települések listája
- Ár – Az ingatlan meghirdetett ára, forintban értve
- Szobák száma – Az ingatlanban található helyiségek száma, darabszámmal megadva
- Ingatlan területe – Az ingatlan alapterülete, m²-ben megadva

Kódrészlet a részletes keresés PHP fájlból

```
$ar=$_POST["ar"];
$terulet=$_POST["terulet"];
$parancs="SELECT kinalat.azonosito, kinalat.ingatlan_id, kinalat.kinal_tip,
kinalat.tipus_id, kinalat.iranyar,kinalat.telepules_id, kinalat.alapterulet,
kinalat.szobaszam, kinalat.eladva_ar, telepulesek.telepules, tipus.tipusok
FROM kinalat
INNER JOIN telepulesek ON telepulesek.telepules_id=kinalat.telepules_id
INNER JOIN tipus ON tipus.tipus_id=kinalat.tipus_id
WHERE kinal_tip='".$_$_POST['kinal_tip']."' AND kinalat.tipus_id='".$_$_POST['tipus']."'
AND kinalat.telepules_id='".$_$_POST['telepules']."'
";
// Ha az ár a 'mindegy' opción van.
if ($ar<999)
{
// A lehetségesen választható árak.
if($ar==1) { $parancs.=' AND kinalat.iranyar<10000000';}
if($ar==10) { $parancs.=' AND kinalat.iranyar BETWEEN 10000000 AND 20000000';}
if($ar==20) { $parancs.=' AND kinalat.iranyar BETWEEN 20000000 AND 30000000';}
if($ar==30) { $parancs.=' AND kinalat.iranyar BETWEEN 30000000 AND 40000000';}
if($ar==40) { $parancs.=' AND kinalat.iranyar BETWEEN 40000000 AND 50000000';}
if($ar==50) { $parancs.=' AND kinalat.iranyar BETWEEN 50000000 AND 100000000';}
if($ar==100) { $parancs.=' AND kinalat.iranyar>100000000';}
}
// Ha a terület a 'mindegy' opción van.
if ($terulet<999)
{
// A lehetségesen választható területek.
if($terulet==1) { $parancs.=' AND kinalat.alapterulet<30';}
if($terulet==30) { $parancs.=' AND kinalat.alapterulet BETWEEN 30 AND 50';}
if($terulet==50) { $parancs.=' AND kinalat.alapterulet BETWEEN 50 AND 70';}
if($terulet==70) { $parancs.=' AND kinalat.alapterulet BETWEEN 70 AND 100';}
if($terulet==100) { $parancs.=' AND kinalat.alapterulet>100';}
}
// Ha a szobaszám a 'mindegy' opción van.
if ($_POST["szobaszam"]<>999)
{
// Ha a kiválasztott szobaszám 6 vagy annál több.
if ($_POST["szobaszam"]==6)
{
$parancs.=' AND szobaszam>='.$_POST["szobaszam"];
}
// Más esetekben a kiválasztott szobaszámot fűzze hozzá a lekéréshez.
else
{
$parancs.=' AND szobaszam='.$_POST["szobaszam"];
}
}
}
$_SESSION['parancs']=$parancs;
}
```

15. ábra (A részletes keresés PHP kódjának egy részlete)

3. Azonosító alapján történő keresés:



16. ábra (Keresés azonosító alapján)

- Azonosító száma – az ingatlan 5 jegyű azonosító száma.

A harmadik keresési változat, az ingatlanhoz tartozó azonosító szám kiválasztása a legördülő listából.

A keresés eredményeképpen kilistázódnak a kiválasztott kritériumoknak megfelelő ingatlanok. A listában megtalálhatóak az ingatlan főbb adatai és egy az ingatlanhoz kapcsolódó kép. Minden ingatlan mellett található egy részletek gomb, mely megnyomásával bővebb adatok hívhatóak elő a kiválasztott ingatlanról. A részletes adatok mellett egy galéria is szerepel, mely tartalmazza a rendszerben található összes képet az ingatlanról. Több találat esetén, a lista alján elhelyezett lapozó hivatkozással tetszőlegesen lapozható.

VI.7.2. Az Ingatlanok kilistázása

Az ingatlan kilistázás a felhasználói és az adminisztrátori felületen annyi különbséggel rendelkezik, hogy míg a hirdetéseket böngésző ügyfél a kiválasztott ingatlannál az 'ingatlan részletei' gombot találja (melynek segítségével bővebb információkhoz juthat), az admin felületen megtalálható 'módosít' gomb helyett. Az admin felületen található kínálat listában azonban lehetőség nyílik a kiválasztott ingatlan adatainak módosítására az oldalt használó munkatársaknak.

VI.8. Az admin felület

VI.8.1. Bejelentkezés

A munkafelületre, csak a jogosultsággal rendelkező munkatársak jelentkezhetnek be. Minden egyes illetéktelen behatolót a rendszer hibaüzenettel figyelmeztet, hogy azonosítás szükséges a további lapok megtekintéséhez. Felhasználónév és jelszó hiányában nem lehet hozzáférni az adminisztrátori felületeken található alkalmazások egyikéhez sem. A sikeres bejelentkezést követően, a bal oldalt megtalálható menüpontok segítségével, különböző feladatokat és módosításokat lehet elvégezni az adatbázisban található ingatlanokkal.



17. ábra (Belépés az admin felületre)

VI.8.2. Kínálatok listája

A felhasználói felület mintájára, az admin felületen is elhelyezésre került egy táblázat, melyben, az adatbázisban szereplő összes ingatlan megtalálható. Ebben a listában is, minden ingatlanhoz hozzáfűzésre került egy kicsinyített kép, amely az ingatlant ábrázolja. Azonban amelyik ingatlanhoz nem található kép, itt is a jól ismert módon egy előre megadott kép kerül az ingatlan sorának elejére. Minden sorban megtalálható a 'szerkeszt' gomb, mely segítségével az előugró ablakban lehetőség nyílik az adott ingatlan adatainak módosítására. A kínálat lista alján elhelyezésre került egy lapozó, mely segítségével navigálhatunk az oldalak közt.

VI.8.3. Új ingatlan hozzáadása

Új ingatlan bevitelére a rendszerbe, a bal oldalt megtalálható menüpontok segítségével, azok közül az *ingatlan hozzáadása* sort kiválasztva van lehetőségünk. Az új ablakban megnyíló regisztrációs űrlap minden mezőjét kitöltve, a rendszer ellenőrzi a mezők helyességét, és a 'következő' gombra kattintva a rendszer elmenti az ingatlant az adatbázisba. Ha a mentés vagy a mezőellenőrzés során, bármi féle hiba merülne fel, akkor a rendszer a lap tetején, ezt hibaüzenettel jelzi számunkra. Ha minden mezőt pontosan kitöltöttünk, akkor megnyílik egy következő beviteli oldal. A megjelenő újabb oldalon van lehetőség az ingatlanhoz tartozó képek feltöltésére. A rendszer akkor is elmenti a hirdetést, ha nem csatolunk hozzá egyetlen képet sem. Azonban ekkor az ingatlan listában történő megjelenítésnél, egy előre - a rendszernek - megadott képet fogunk látni. Azonban az ingatlan részleteihez tartozó galériában egy kép sem fog megjelenni azokhoz az ingatlanokhoz, melyekhez nem kerültek feltöltésre képek.

Admin Szabó Balázs	Hirdető Dobó Péter	Azonosító száma* dobp	Kínálat típusa Eladó	Típus Családi ház
Irányítószám 2600	Település Vác	Utca Badacsony	Házszám 23	Helyrajzi szám 12371/A/23
Kor 19	Anyag Tégla	Alapterület 210	Telek terület 120	Szobaszám 5
[A lakás emelete] -	[Közös költség] 	<input checked="" type="checkbox"/> Külön vízóra	<input checked="" type="checkbox"/> Csatorna	Komfort <input type="radio"/> Félkomfort <input checked="" type="radio"/> Összkomfort
Fűtés típusa gáz	<input checked="" type="checkbox"/> Víz <input checked="" type="checkbox"/> Gáz <input checked="" type="checkbox"/> Fűtés <input checked="" type="checkbox"/> Villany	<input checked="" type="checkbox"/> Telefon <input checked="" type="checkbox"/> Kábel tv <input checked="" type="checkbox"/> Internet <input checked="" type="checkbox"/> Riasztó	<input checked="" type="checkbox"/> Garázs <input checked="" type="checkbox"/> Kocsibeálló <input checked="" type="checkbox"/> Pince <input checked="" type="checkbox"/> Kamra	<input type="checkbox"/> Lift <input checked="" type="checkbox"/> Erkély <input type="checkbox"/> Galéria
Képek: Képek feltöltésére a következő oldalon lesz lehetősége.		Egyéb: Az ingatlanban van infrasauna. A nappaliban padlófűtési található.		
<input checked="" type="checkbox"/> Aktív? <input checked="" type="checkbox"/> Kiemelt?	Irányár 32.000.000 Feladás dátuma 2011-01-14	Eladva? <input type="radio"/> Igen <input checked="" type="radio"/> Nem	Végösszeg Az eladás dátuma	Tovább >> Mégse

18. ábra (Ingatlan hozzáadása)

VI.8.4. Az ingatlanhoz tartozó képek feltöltése

A képek feltöltésére az ingatlan hozzáadási ablakban a második lépésben lehetséges. Tetszőleges számú kép tölthető fel az ingatlanhoz. A képfeltöltések ingatlanonként külön mappába történnek. A mappa neve az azonosító alapján kerül létrehozásra. Az ingatlanhoz tartozó képek mappában található egy tmb nevű mappa, amely a thumbnaileket (bélyegképeket) tartalmazza, ezt az alkalmazás hozza létre.

Az ingatlanok kilistázása esetén ezek a csökkentett méretű képek jelennek meg az ingatlanok mellett. Feltöltéskor a képekről másolat készül, és azokból lesznek létrehozva a bélyegképek. A képek neve változatlan marad a feltöltés után. A kép megjelenítésénél, a feltöltött képre hivatkozni az ingatlan azonosítójával lehet. A rendszer a következő képtípusokat fogadja el:

- JPEG / JPG
- GIF
- PNG

Ha az imént felsorolt képtípusoktól különböző típusú fájlt szeretnénk feltölteni, a rendszer hibaüzenettel tér vissza. Hibás fájlfeltöltésre nincs lehetőség, a beépített fájl típus ellenőrzésnek köszönhetően.

VI.8.5. Ingatlan módosítása

A regisztrált ingatlanok adatainak módosítására is lehetőséget kínál az alkalmazás. Ha az adatbázisban található ingatlan adatai esetlegesen megváltoznának, vagy hibásan szereplnének az adatbázisban. Ezt a módosítást úgy hajthatjuk végre, hogy ismételten a bal oldalt megtalálható menüben az ingatlan módosítása fülre kattintva, megnyitjuk az ingatlanokat kilistázó űrlapot. Itt az összes, a rendszerben megtalálható ingatlan listázásra kerül.

Minden ingatlan mellett található egy ' szerkeszt ' gomb melyre rákattintva megnyílik a már jól ismert hozzáadási panel. Azonban annyi különbséggel, hogy tartalmaz minden, korábban bevitt az ingatlanhoz tartozó adatot. A módosítási ablak tartalmaz megszorításokat, melyek kimondják, hogy az ingatlan azonosítója nem megváltoztatható tényező. Ez csak egy olvasásra felkínált beviteli mezőben kerül kiíratásra.

VI.8.6. Ingatlan törlése

Ingatlan törlésére, a menüben megtalálható ingatlan törlése fülre kattintva van lehetőség. Egy megnyíló új ablakban kilistázódik a törölhető ingatlanok listája. Minden ingatlan mellett megtalálható a törlés gomb melyre kattintva egy új, törlést megerősítő oldalra jutunk. Itt a rendszer megkérdezi, hogy valóban törölni szeretnénk-e a kiválasztott ingatlant? Az igen opciót kiválasztva az ingatlan törlésre kerül a rendszerből. Ha a nem gombra kattintunk, hibaüzenetet kapunk a rendszertől a sikertelen eltávolításról. Több ingatlan törlési szándéka esetén is csak egyesével áll módunkban eltávolítani az ingatlanokat, az esetleges véletlen ingatlan törlés elkerülése érdekében.

VI.8.7. Kijelentkezés

Munkáját befejezve, az oldat használó személynek távozás előtt, mindenféleképpen ki kell jelentkeznie. Kijelentkezésre a felhasználónévre kattintva van lehetősége a menü utolsó sorában. Ezt követően, újra a bejelentkezési doboz jelenik meg. Ha ezt a munkatárs elfelejtené valamilyen okból kifolyólag, akkor a böngésző bezárása is elegendő a kijelentkezéshez, így a munkamenet megszűnik, és a rendszer kilépett önmagától. Így újabb használat, esetén ismételten be kell írni a felhasználónevet és a jelszót. Ennek hiányában, a rendszer elutasítja a belépési kérelmünket.

VI.8.8. Hirdető hozzáadása

Új szerződéskötés esetén az adminisztrátor feladata a hirdető adatait feltölteni a rendszerbe. Erre lehetőséget biztosít az alkalmazás a menüben elhelyezett ' hirdető hozzáadása ' gomb megnyomásával. Egy új ablakban felugró regisztrációs lapon a kért mezőket kitöltve, az új hirdető hozzáadódik a rendszerhez. Ha a kijelölt mezők valamelyikét elfelejtjük kitölteni, akkor hibaüzenetet kapunk.

Ezt korrigálnunk kell, mert máskülönben nem adódik hozzá az adatbázishoz a kért hirdető. A sikeres hozzáadást követően, az ingatlan hozzáadása lapon elhelyezett hirdetők mezőben, megtalálható és kiválasztható lesz az imént bevitt ügyfél.

VI.8.9. Hirdető törlése

Hirdető törlését kizárólag adminisztrátor végezheti el. A menüben kiválasztva a hirdető törlését egy új oldalon megjelenik az adatbázisban található hirdetők listája. Két fajta módon lehetséges az eltávolítás. Az egyik módja, hogy ha a hirdető nevén nincs egyetlen ingatlan sem a hirdetések között, akkor egyszerűen kitöröljük a rendszerből a törlés gombra kattintva. Egy megerősítő oldalon megbizonyosodunk arról, hogy biztosan azt a hirdetőt választottuk-e ki a listából, akit szerettünk volna. Ha minden az elvárásainkhoz híven zajlott le, akkor a megerősítő oldalon az igen gombra kattintva töröljük a hirdetőt. Ha a nem gombra kattintunk, hibaüzenet jelzi számunkra, hogy nem sikerült az eltávolítás. A másik opció ha a hirdető névéhez egy illetve több hirdetés is társul, akkor nem lehet törölni addig a rendszerből a hirdetőt, amíg a nevén lévő ingatlanokat ki nem töröltük a rendszerből. Ezeket a lehetőségeket egy megerősítő oldalon kínáljuk fel.



19. ábra (Hirdető törlése)

VI.8.10. Admin hozzáadása

Abban az esetben, ha az irodához új adminisztrátor kollega érkezik, lehetőség van a már felhasználónévvel és jelszóval rendelkező adminisztrátorok részére, hogy adatait felvigyék a rendszerbe, és ez által ő is szabadon használhassa a webes felületen kialakított alkalmazást. (A későbbiek során, a rendszer újabb fejlesztésénél tervbe van véve egy supervisor felület kialakítása, az új adminisztrátor hozzáadására.) A regisztrációs űrlapon vannak olyan mezők, melyeket kötelező kitölteni. Ennek elmulasztásával a rendszer hibaüzenetet küld, és az elmulasztott adatok beírásának pótlásáig nem lehetséges a regisztráció. Hozzáadást követően az admin mezőkben az ingatlan hozzáadása fülön elérhetővé válik az új adminisztrátorok kiválasztása is.

VI.9. Az oldal beépített funkcionalitása

VI.9.1. Galéria

Az ingatlan részleteinél, szükséges galéria elhelyezése, amelynek tartalmaznia kell az összes képet, amelyet az ingatlanhoz feltöltöttek. Egy képre kattintva, azt nagyobb méretben kell megjeleníteni, egy felugró ablakban. A felugró ablakban lapozást is lehetővé kell tenni, így bármelyik képre kattintva körbe tudjuk járni a képek mappa teljes tartalmát.

VI.9.2. Datepicker¹⁶



20. ábra

Az ingatlan hozzáadása, ill. módosítása lapon azokhoz a mezőkhöz, amelyeknél dátum megadása szükséges, ott egy kis naptár alkalmazást kell elhelyezni, amely segítségével könnyebben és pontosabban meg lehet adni a választott dátumot.

¹⁶ A datepicker forrása: <http://plugins.jquery.com/project/datepicker>

A beviteli mezőre kattintva kell előugrania az alkalmazásnak, és ha a dátumot kiválasztottuk, akkor vissza kell rejtett módba lépnie a felugró naptárnak. Ez után a mezőnek a kért dátumot, ÉÉ-HH-NN formában megadva kell tartalmaznia.

Az alkalmazás használatához az oldalon engedélyezni kell a java scripteket. Ha nem tartunk igényt az alkalmazásra akkor szabad kezűleg is be lehet vinni a szükséges dátumot. Azonban használatával elkerülhető a helytelenül bevitt dátumok mentése.

VI.9.3. SQL injection

```
$a_nev = mysqli_real_escape_string($csatlakozas, trim($_POST['a_nev']));17
```

Azoknál a mezőknél, ahol tetszőleges adatokat lehet bevinni a rendszerbe, be kell építeni egy támadás védelmet azzal szemben, hogy nemkívánatos SQL-lekérdezés módosító parancsot lehessen futtatni. A megoldásként alkalmazott módszer az SQL befecskendezések elleni védelem beépítése mely meggátolja, hogy az űrlapadatok felhasználásával, módosítható legyen egy lekérdezés működése. Ezen függvény lehetőséget biztosít a veszélyes karakterek ellenőrzésére is.

VI.9.4. Mező ellenőrzések

Az esetleges akarva vagy akaratlanul kimaradt mező kitöltések elkerülése végett a rendszerbe egy validátor került, amely figyel a mezőbe beírt adatok helyességét, illetve, hogy nem lett-e a kért mező üresen hagyva. Az effajta ellenőrzéseknél figyelniük kell, hogy a kért mezőben, pontosan és érvényesen adtuk-e meg telefonszámunkat illetve e-mail címünket.

Hiba! Kérjük töltsön ki minden mezőt!

VI.10. A rendszer validálás

A korábban bemutatott részek elkészítése után, szükség van tesztelések elvégzésére a megalkotott programrészekben. Fel kell mérni, hogy azok helyesen működnek és az általunk beállított funkciókat pontosan végzik-e.

¹⁷ Az admin hozzáadása PHP fájlból egy kódrészlet, amely szemlélteti az SQL befecskendezést.

VI.11. A rendszer üzembe helyezése

1. Első lépésként egy webkiszolgálót kell telepítenünk. Jelen esetben a nyílt forráskódú Apache webservert. Ennek működése, telepítés után tetszőlegesen felfüggeszthető, illetve újra elindítható.
2. Második lépésként telepítésre és konfigurálásra kerül a PHP szkriptnyelv .
3. Harmadik lépésben feltelepítjük a MySQL adatbázisrendszert és elvégezzük a szükséges konfigurációkat.
4. Az ingatlandb.sql fájl létrehozása, az ingatlan adatbázis minden egyes elemének a tárolására.
5. A PHP és HTML részek létrehozása és konfigurálása.

Az imént felsorolt lépéseket, könnyebben és rövidebb idő alatt is el lehet végezni. Erre kínál lehetőséget az általam is alkalmazott alkalmazás, a XAMPP. Mint ahogyan már korábban bemutattam, feltelepítését követően egy fejlesztői környezet alakul ki az általunk használt rendszeren. Segítségével a telepítés után megtalálható lesz az Apache webservert, a PHP szkriptnyelv és a MySQL adatbázisrendszer. Így ezeket a komponenseket nem kell külön-külön egyesével telepítenünk.

VI.12. A rendszer üzembe helyezése éles környezetben

A saját gépünkön felállított futtató környezet nem mindig nyújt pontos és kielégítő tesztelési lehetőségeket. A hibák javítására és a kezdetleges tesztelésekhez tökéletesen megfelelő, azonban a későbbi munkánk elvégzéséhez korántsem elegendő. Mielőtt a PHP és MySQL páros alkotta rendszerünket egy éles webkiszolgálóra költöztetnénk át, gondosan meg kell terveznünk, hogyan is hajtjuk mindezt végre .

1. Meg kell bizonyosodnunk afelől, hogy az általunk választott webkiszolgáló támogatja a PHP MySQL párost és a számunkra megfelelő verziószámú változat van-e telepítve. Ha esetlegesen más verzió számú változatok szerepelnek a kiszolgálón, lehet hogy át kell alakítani az általunk megírt kódokat. Ha mindezekkel végeztünk, akkor áttérhetünk a következő lépésekre, amelyek már könnyen elvégezhetőek.
2. A PHP fájlok feltöltése a webszerverre. Fontos arra figyelni, hogy a fájlok egymáshoz viszonyított elhelyezkedése azt a szerkezetet őrizze meg ahogyan az a saját gépünkön is megtalálható.
3. A MySQL-ben található adatbázis és a benne lévő táblák áthelyezése.
4. A PHP fájlokban található csatlakozási parancsok átalakítása, hogy az a jelenlegi kiszolgálóra, és ne a saját gépünkön található, általunk felállított rendszerre hivatkozzanak.

VII. BEFEJEZÉS

Remélem, hogy sikerült a választásomra esett témát kellőképpen körüljárni, bemutatni, és egy jól felhasználható rendszert készítenem. Tudom, hogy tökéletes, végkifejlett programot nem lehet készíteni, a programozást és a finomítást csak abbahagyni lehet.

Úgy hiszem, hogy az általam felállított követelményekhez alkalmazkodva, sikerült megoldanom és megvalósítanom elképzeléseimet. Annyit hozzá szeretnék fűzni azonban, hogy a megalkotás közben sok olyan ötlet és elképzelés keletkezett bennem, melyeket a későbbiekben meg szeretnék valósítani, ezzel a rendszer továbbfejlesztését és funkcióinak bővítését szolgálva. Mint már említettem szakdolgozatom elején is, célom egy mindenképp használható és fejleszthető rendszer elkészítése volt. Melyet a későbbiekben teljesen, vagy akár részegységeit felhasználva be lehessen építeni egy éles környezetben működő rendszerbe.

VIII. IRODALOMJEGYZÉK

Ian Sommerville: Software Engineering, 2004, Gyula Simon 2005 (magyar verzió)

Csetényi Arthur: Informatikai rendszerek fejlesztése, Computerbooks, 2001.

Sági Gábor: Webes adatbázis-kezelés PHP és MySQL használatával, (BBS-INFO) 2005

Kósa Márk: Adatbázisrendszerek jegyzet, 2009

Szilágyi József: Adatbázis-kezelési alapok, 2003

Matt Zandstra: Tanuljuk meg a PHP5 használatát 24 óra alatt, 2005

<http://dev.mysql.com/doc>

<http://php.net/manual/en/>

<http://www.php.net/docs.php>

<http://hu.wikipedia.org/wiki/CSS>

Plágium - Nyilatkozat

Szakdolgozat készítésére vonatkozó szabályok betartásáról nyilatkozat

Alulírott (Neptun kód: ... JV8OPY...) jelen nyilatkozat aláírásával kijelentem, hogy a

.....Webes.alkalmazásfejlesztés...-..Ingatlanközvetítő.rendszer.....

című szakdolgozat/diplomamunka

(a továbbiakban: dolgozat) önálló munkám, a dolgozat készítése során betartottam a szerzői jogról szóló 1999. évi LXXVI. tv. szabályait, valamint az egyetem által előírt, a dolgozat készítésére vonatkozó szabályokat, különösen a hivatkozások és idézések tekintetében.

Kijelentem továbbá, hogy a dolgozat készítése során az önálló munka kitétel tekintetében a konzulenszt, illetve a feladatot kiadó oktatót nem tévesztettem meg.

Jelen nyilatkozat aláírásával tudomásul veszem, hogy amennyiben bizonyítható, hogy a dolgozatot nem magam készítettem vagy a dolgozattal kapcsolatban szerzői jogsértés ténye merül fel, a Debreceni Egyetem megtagadja a dolgozat befogadását és ellenem fegyelmi eljárást indíthat.

A dolgozat befogadásának megtagadása és a fegyelmi eljárás indítása nem érinti a szerzői jogsértés miatti egyéb (polgári jogi, szabálysértési jogi, büntetőjogi) jogkövetkezményeket.

hallgató

Debrecen,