

**UNIVERZITA J. SELYEHO – SELYE JÁNOS EGYETEM
EKONOMICKÁ FAKULTA – GAZDASÁGTUDOMÁNYI KAR**

**DATABÁZA NA PREDAJ A PRENÁJOM
NEHNUTELNOSTÍ
ELADÓ ÉS BÉRELHETŐ INGATLANOK ADATBÁZISA**

Bakalárska práca - Szakdolgozat

**Zoltán Héthy
2016**

**UNIVERZITA J. SELYEHO – SELYE JÁNOS EGYETEM
EKONOMICÁ FAKULTA – GAZDASÁGTUDOMÁNYI KAR**

**DATABÁZA NA PREDAJ A PRENÁJOM NEHNUTEĽNOSTÍ
ELADÓ ÉS BÉRELHETŐ INGATLANOK ADATBÁZISA**

Bakalárska práca – Szakdolgozat

Študijný program:	Aplikovaná informatika
Tanulmányi program:	Alkalmazott informatika
Študijný odbor:	aplikovaná informatika
Tanulmányi szak:	alkalmazott informatika
Vedúci záverečnej práce:	PaedDr. Ladislav Végh
Témavezető:	PaedDr. Ladislav Végh
Školiace pracovisko:	Katedra matematiky a informatiky
Tanszék megnevezése:	Matematika és Informatika Tanszék

**Zoltán Héthy
Komárno, 2016**

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Zoltán Héthy
Študijný program: Aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: 9.2.9. aplikovaná informatika
Typ záverečnej práce: Bakalárska práca
Jazyk práce: maďarský
Sekundárny jazyk: slovenský

Téma: Eladó és bérelhető ingatlanok adatbázisa

Anotácia: A szerző feladata egy weboldal létrehozása, amely eladó és bérelhető ingatlanokat tartalmaz. A weboldalt PHP, MySQL technológiák felhasználásával kell megvalósítani. Az oldal tartalmazzon adminisztrátori felületet, ahol új ingatlanok adatai és fényképei vihetők be, módosíthatók vagy törölhetők. A felhasználóknak legyen lehetősége az adatbázisban szereplő ingatlanok között keresni különböző szempontok szerint és az eredményt rendezni a hozzáadás dátuma vagy az ár alapján.

Vedúci: PaedDr. Ladislav Végh
Katedra: KMI – Katedra matematiky a informatiky
Vedúci katedry: RNDr. Zuzana Árki, PhD.

Dátum zadania: 14.10.2015

Dátum schválenia: 20.10.2015

RNDr. Zuzana Árki, PhD.
vedúci katedry

Köszönetnyilvánítás

Köszönetet szeretnék mondani PaedDr. Végh László tanár úrnak, amiért vállalta a munkám témavezetését, és javaslataival, ötleteivel segített a munkám elkészítésében.

Továbbá szeretném megköszönni Szénási Sándor, PhD. tanár úrnak, amiért segítséget nyújtott az adatbázis tervezésénél.

Végezetül szüleimnek és családomnak, akik mindvégig támogattak egyetemi tanulmányaim során.

Abstrakt

Cieľom bakalárskej práce je vytvorenie webovej stránky zameranej na predaj a prenájom nehnuteľností. Používatelia webovej stránky majú mať možnosť medzi nehnuteľnosťami nachádzajúcimi sa v databáze vyhľadávať na základe rôznych kritérií a výsledky vyhľadávania zoradiť podľa dátumu zverejnenia či ceny. Webová stránka musí obsahovať administratívne rozhranie, prostredníctvom ktorého je možné pridávať údaje a fotky nových nehnuteľností, ako aj upraviť a vymazať údaje existujúcich nehnuteľností. Práca pozostáva z piatich kapitol a obsahuje dvadsaťjeden ilustrácií. Prvá kapitola práce sa zaoberá cieľom práce a analyzuje webové stránky. Druhá kapitola predstaví programovacie jazyky a nástroje použité v rámci práce. Tretia kapitola je zameraná na návrh a realizáciu webovej stránky a databázy. Nasledujúca kapitola nám predstaví samotný systém stránky s nehnuteľnosťami. Záverečná kapitola opíše testovanie stránky a načrtne možnosti ďalšieho rozvoja stránky. Ako výsledok bakalárskej práce nám vznikol jednoduchý, dizajnový, ľahko ovládateľný systém, ktorý môže byť veľmi užitočný pre realitné kancelárie.

Kľúčové slová:

dynamická webová stránka, databáza, nehnuteľnosť, administratívne rozhranie

Absztrakt

A szakdolgozat célja egy weboldal létrehozása, amely az eladó és bérelhető ingatlanokkal foglalkozik. A weboldal felhasználóinak legyen lehetőségük az adatbázisban szereplő ingatlanok között keresni különböző szempontok szerint és a keresés eredményét rendezni a hozzáadás dátuma vagy az ár alapján. Továbbá az oldalnak tartalmazni kell egy adminisztrációs felületet, amelyen új ingatlanok adatai és fényképei vihetők be, vagy a meglévő ingatlanok módosíthatók vagy törölhetők. A munka öt fejezetből áll és huszonegy ábrát tartalmaz. Az első fejezet a szakdolgozat céljával és a weboldalak elemzésével foglalkozik. A második fejezet a munkánk során felhasznált programozási nyelveket és eszközöket ismerteti. A harmadik fejezet a webfelület és az adatbázis tervezésével és megvalósításával foglalkozik. A negyedik fejezet az ingatlanos rendszerünk bemutatását tartalmazza. A befejező rész a teszteléseket és a továbbfejlesztési lehetőségeket taglalja. A szakdolgozat eredménye egy egyszerű, dizájnos, könnyen kezelhető rendszer lett, amely jól hasznosítható az ingatlanirodák számára.

Kulcsszavak:

dinamikus weboldal, adatbázis, ingatlan, adminisztrációs felület

Abstract

The aim of this Bachelor's thesis is to create a web page that deals with the saleable and hireable real estate. The users of this page should have the possibility to search between the real estate found in the database by various criteria, and sort the results of the search by the date of addition or price. Furthermore the page must have an administrative interface that is capable of adding new real estate along with pictures and edit or delete the existing real estates. The thesis consists of five chapters and contains twenty-one illustrations. The first chapter deals with the aim of the thesis and the analysis of web pages. The second chapter describes the programming languages and tools used in the thesis. The third chapter deals with the realization of the interface and database. The fourth chapter characterizes our system of real estate. The final chapter outlines the testing and possible developments. The result of this thesis is a simple, good-looking, user-friendly system that can be used by offices dealing with real estate.

Keywords:

dynamic web page, database, real estate, administrative interface

Tartalomjegyzék

Bevezetés	11
1. A téma elemzése	12
1.1. A dolgozat célja.....	12
1.2. Az internet és a weblapok kapcsolata	12
1.2.1. Weblapok típusai	12
1.3. Statikus és dinamikus weboldalak	13
1.3.1. Statikus oldalak	13
1.3.2. Dinamikus oldalak.....	14
1.4. A jó weboldalak alapkövetelményei	15
2. Felhasznált eszközök ismertetése	16
2.1. HTML	16
2.1.1. HTML5	16
2.2. CSS.....	17
2.3. PHP	17
2.3.1. PHP5.....	18
2.4. MySQL.....	18
2.5. JavaScript.....	18
2.5.1. jQuery.....	19
2.6. EasyPHP.....	19
2.6.1. Apache	19
2.7. Notepad++.....	19
3. Webfelület és adatbázis	20
3.1. Feladat specifikáció.....	20
3.2. Képernyőképek, felülettervezés	20
3.2.1. Felhasználói és adminisztrátori felülettervek	21
3.2.2. A menürendszer.....	21
3.2.3. Webfelület megvalósítása.....	22
3.3. Adatbázis tervezése	23
3.3.1. Egyed-kapcsolat diagram	23
3.3.2. Adatbázis normalizálása	24
3.4. Adatbázis megvalósítása	25
3.4.1. Adatbázis szerkezete	26
3.4.2. Táblák bemutatása	26

4. Rendszer bemutatása.....	30
4.1. Felhasználói felület	30
4.1.1. Főoldal.....	30
4.1.2. Mysqli burkoló készítése	31
4.1.3. Eladó és kiadó ingatlanok oldal.....	33
4.1.3.1. Ingatlan ajánlatok listázása	34
4.1.3.2. Oldal lapozó	34
4.1.3.3. Munkamenet-kezelés	35
4.1.3.4. Keresés.....	35
4.1.3.5. Rendezés	36
4.1.4. Ingatlan ajánlatok részletei	37
4.1.5. Elérhetőségeink oldal	38
4.2. Adminisztrátori felület	38
4.2.1. Bejelentkezés	38
4.2.2. Adminisztrátori főoldal	39
4.2.3. Ingatlan felvétele	40
4.2.3.1. Képek feltöltése	42
4.2.4. Ingatlan módosítása	43
4.2.5. Ingatlan törlése	44
4.2.6. Kijelentkezés	45
5. Tesztelés és továbbfejlesztés.....	46
5.1. Tesztelés és üzembe helyezés	46
5.2. Továbbfejlesztési lehetőségek.....	47
5.2.1. Reszponzív webdesing	47
5.2.2. Részletes keresői oldal	48
5.2.3. Levelező-rendszer.....	48
5.2.4. Saját ingatlan feltöltése.....	48
5.2.5. Adminisztrátorok hozzáadása.....	49
5.2.6. SQL injection.....	49
Befejezés.....	50
Resumé.....	52
Felhasznált irodalom	56
Mellékletek	59

Ábrák és táblázatok jegyzéke

Ábrák jegyzéke

1. ábra: Statikus oldal előállítása	14
2. ábra: Dinamikus oldal előállítása.....	14
3. ábra: Felhasznált programnyelvek	16
4. ábra: Felhasználói felületterv	21
5. ábra: Adminisztrátori felületterv	21
6. ábra: Felhasználói menü	21
7. ábra: Adminisztrátori menü	21
8. ábra: Egyed-kapcsolat diagram.....	23
9. ábra: PhpMyAdmin kezelőfelület.....	25
10. ábra: Adatbázis szerkezete.....	26
11. ábra: Kiemelt ingatlanok.....	30
12. ábra: Eladó ingatlanok	33
13. ábra: Kereső	36
14. ábra: Rendező	36
15. ábra: Ingatlan részletei	37
16. ábra: Bejelentkezés	39
17. ábra: Adminisztrátori főoldal.....	40
18. ábra: Ingatlan felvétel	41
19. ábra: Ingatlan módosítás	44
20. ábra: Ingatlan törlés	45
21. ábra: Reszponzív design	48

Táblázatok jegyzéke

1. táblázat: Mysql burkoló	31
----------------------------------	----

Rövidítések és jelölések jegyzéke

CSS	Cascading Style Sheets
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
JPEG (JPG)	Joint Photographic Experts Group
LAMP	Linux, Apache, MySQL, and PHP
MD5	Message Digest 5
PHP	Hypertext Preprocessor
PNG	Portable Network Graphics
SQL	Structured Query Language
URL	Uniform Resource Locator
UTF-8	Unicode Transformation Format (8-Bit)
WWW	World Wide Web
WYSIWYG	What You See Is What You Get
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language

Bevezetés

Napjainkban az internethasználat egyre jobban az életünk szerves részét képezi. Segítségével szinte már bármit meg tudunk oldani. Használjuk a munkánkhoz, tanuláshoz, szórakozáshoz, információgyűjtéshez, kapcsolattartáshoz, illetve az ügyintézésben is óriási hasznát vesszük. Az internet lett az elődleges tájékozódási eszköz. Ezt az előnyt nagyon jól kihasználják a különböző cégek és vállalatok is. Minden vállalatnak, illetve cégnek már saját honlapja van. A honlapok a cégek elsődleges reklámozási formái. Segítségével az érdeklődő részletesebb információt kaphat a termékekről, szolgáltatásokról, és nem utolsósorban magáról a márkáról. Az ingatlan kereskedelmi piacon is óriási jelentősége van az információ megosztásnak. Egyre több ingatlaniroda gondolja úgy, hogy az emberek többsége igényli azt, hogy az interneten tudjanak keresgélni az eladásra szánt ingatlanok között. A honlapok használatával kiválthatóak részben a papíralapú hirdetési formák és megkönnyíthető, illetve felgyorsítható az információ továbbítása a célközönség felé.

Szakdolgozatunk témájának egy ingatlanokkal foglalkozó weboldal elkészítését választottuk. A választásunk azért esett erre a témára, mert úgy gondoltuk, hogy nagyon hasznos lehet számunkra, ha az elvárásoknak megfelelő weboldalakat tudunk készíteni. A szakdolgozatunk elkészítésével nagyon jól el tudtuk sajátítani az ehhez szükséges ismereteket, mint például a weboldal tervezése, programozása és implementálása, valamint a hozzá tartozó adatbázis tervezése, normalizálása és felhasználása.

A munka célja egy olyan weboldal létrehozása, amellyel megkönnyíthettük az ingatlanok után érdeklődő vagy információt gyűjtő felhasználók dolgát. A felhasználói felület egyszerű, ám hatékony kialakításával könnyedén tudnak keresgélni az eladó, illetve kiadó ingatlanok között. A weboldalt működtető adminisztrátorok életét is rendkívül megkönnyíti a könnyen áttekinthető és használható adminisztrációs felület, ahol aztán lehetőségük nyílik újabb ingatlanok felvételére vagy a meglévő ingatlanok módosítására, illetve eltávolítására.

A munkánk öt fejezetből tevődik össze. Az első fejezetbe ismertettük a dolgozat célját és részletesen foglalkoztunk a weblapok elemzésével. A következő fejezetben a munkánk során felhasznált programozási nyelveket és eszközöket ismertettük. Ezek után rátértünk a gyakorlati részre, ahol előbb a webfelületünk és az adatbázisunk tervezésével, majd megvalósításával foglalkoztunk. A negyedik fejezet az ingatlanos rendszerünk bemutatását tartalmazza, amelybe részletesen leírtuk rendszer működését. Az utolsó fejezetben pedig a teszteléseket és a weboldal továbbfejlesztési lehetőségeit ismertettük.

1. A téma elemzése

1.1. A dolgozat célja

A szakdolgozatunk célja egy olyan dinamikus weboldal létrehozása volt, amely eladó és bérelhető ingatlanokat tartalmaz. Az ehhez hasonló weboldalak dinamikus tartalom megjelenítését igénylik, ezért a weboldalt PHP és MySQL technológiák felhasználásával valósítottuk meg. A felhasználóknak lehetőségük nyílik az adatbázisban szereplő ingatlanok között keresni különböző szempontok szerint és a keresés eredményét rendezni a hozzáadás dátuma vagy az ár alapján. Továbbá tartalmaz egy adminisztrátori felületet is, ahol új ingatlanok adatai és képei vihetők fel, illetve a meglévő ingatlan ajánlatok módosíthatók vagy törölhetők.

1.2. Az internet és a weblapok kapcsolata

Az internet kifejezés egy nemzetközileg elterjedt szó, aminek a jelentése az angol eredetű internetwork szóból ered, amely magyarul „hálózatok hálózatát” jelenti. Vagyis az internet olyan számítógépes hálózatok hálózata, amely az internet protokoll segítségével felhasználók milliárdjait köti össze. Továbbá lehetővé teszi olyan elosztott rendszerek működését, mint a világháló (World Wide Web, WWW vagy röviden Web) (Muha, 2005).

A világháló az interneten működő, hiperlinkekkel és URL-ekkel összekapcsolt oldalak és fájlok egysége. Ezt a rendszert a webböngészők segítségével tudjuk elérni, amik aztán képesek megjeleníteni az egyes dokumentumokat, weblapokat. A weblap egy olyan dokumentum, amely megfelel a World Wide Web rendszer számára, és alkalmas arra, hogy különböző webböngészők megjelenítsék. Fő jellemzője a hiperszöveg, amely más weboldalakra navigálhat el minket, linkek segítségével (Szűts, 2012).

1.2.1. Weblapok típusai

A weblapok típusai különböző kialakításúak lehetnek a weblap jellegétől és funkciójától függően. Ezen eltérésekből adódik az egyes honlapok ára is.

Az oldal jellege és funkciója alapján az alábbi típusokat különböztetjük meg:

- **Internetes névjegy** – Egy egyszerű, egyoldalas statikus lap, ami nem tartalmaz mást, csak a cég tevékenységi körét és az elérhetőségeiket.
- **Többoldalas bemutatkozó lap** – Ez az oldaltípus rendkívül alkalmas magánszemélyek, vállalkozások, vendéglátóhelyek és különböző szervezetek bemutatására. Különálló

oldalakon történik a cégbemutató, a termékek, a szolgáltatások és az elérhetőségek ismertetése. Az ilyen típusú oldalak már tartalmazhatnak képgalériát, különböző űrlapelemeket és adminisztrációs felületet is.

- **Portál** – Ez egy speciális weboldal típus, amolyan információs csomópont az Interneten egy vagy több témában. Folyamatosan változó, bővülő tartalommal rendelkezik.
- **Webáruház** – Ez az internetes értékesítés eszköze. Ez nemcsak a termékeket mutatja be, hanem az online megrendelést és vásárlást is lehetővé teszi. Egy terméklistához kosár-rendszer és egyéb funkciók kapcsolódnak. Az üzemeltető aztán az adminisztrációs felületen keresztül frissítheti a termék adatait.

Az alkalmazott technológia alapján a következő típusokat különböztetjük meg:

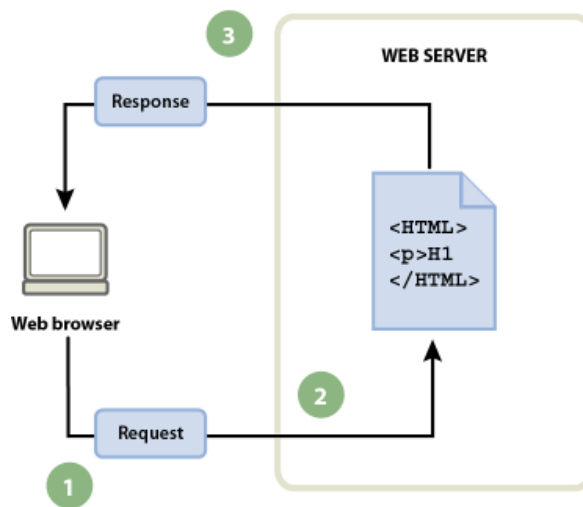
- **Statikus weblapok** – A statikus weblapok nélkülözik a tartalom megváltoztatásához szükséges programokat. Az ilyen oldalak olcsóbbak, rövidebb idő alatt elkészülnek, de módosításukhoz a fejlesztő közreműködése szükséges.
- **Flash oldalak** – Ez egy olyan webes technológia, amellyel animációkat, videoklipeket lehet létrehozni hanghatásokkal és vizuális effektusokkal. Interaktív kezelőfelülettel rendelkezik. Előnye, hogy hihetetlenül látványos. Viszont költséges és nem alkalmas gyakran módosításra szoruló tartalom megjelenítésére.
- **Dinamikus weblapok** – Gyakran változó adatok esetén használjuk. A weboldal különböző programozási nyelvek és adatbázis használatával készül. Tartalmaznia kell egy adminisztrációs felületet is, amely az adatok egyszerű és kényelmes megváltozathatására szolgál (K-Design.hu, 2016).

1.3. Statikus és dinamikus weboldalak

1.3.1. Statikus oldalak

Eleinte a weboldalak csak statikus tartalmú dokumentumokat jelenítettek meg. A statikus weboldalak teljesen változatlanok maradnak, ahogy elkészítődtek, abban az állapotban maradnak meg és minden lekérdezés esetén ugyanúgy jelennek meg. Ezeket HTML oldalaknak is nevezik. A kliens számítógépére az a HTML kód töltődik le, amit a weboldal készítője feltölt a szerverre. Alap építőelemei a táblázatok, címsorok, képek és hivatkozások. Ezen oldalak létrehozásához ma már a HTML5 és CSS3 szabványt használjuk. Továbbá léteznek olyan megoldások is, mint például a JavaScript és egyéb kliens oldali programnyelvek, amelyek a felhasználóba azt az érzést keltik, hogy egy dinamikus oldalt böngésznek (Kvaszingerné, 2013).

Statikus oldalak előállítása



1. A kliens kérést küld a webszervernek.
2. A kérés feldolgozása a webszerveren.
3. A webszerver válaszként vissza küldi a kért oldalt a böngészőnek.

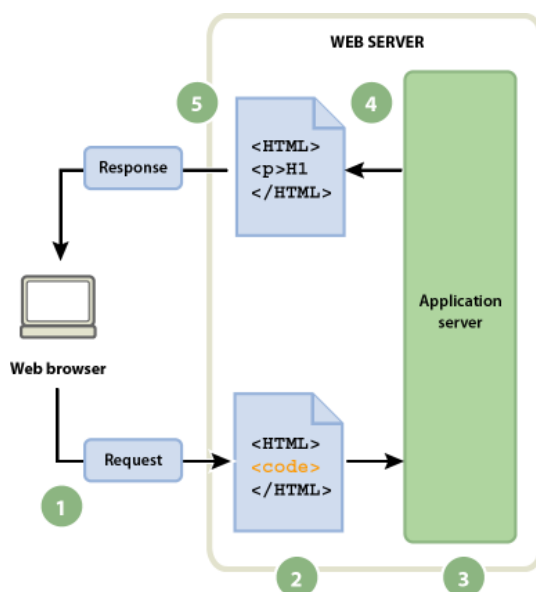
1. ábra: Statikus oldal előállítása

Forrás: http://help.adobe.com/en_US/dreamweaver/cs/using/images/ds_process_static.png

1.3.2. Dinamikus oldalak

A dinamikus weblapokat akkor használunk, ha változó tartalomra van szükségünk. Elkészítésük összetettebb, viszont a fejlesztőknek nem kell állandóan frissíteniük az oldal tartalmát. Ezek az oldalak interaktívak, képesek bemeneteket fogadni űrlapok segítségével. A tartalom nagy részét adatbázisokból szedik ki. Itt a kész HTML helyett programkódot töltünk fel a szerverre, amely a szerveren lévő futtatás után HTML kódokat generál ki. Ezek a programkódok általában PHP nyelven vannak írva (Kvaszingerné, 2013).

Dinamikus oldalak előállítása



1. A kliens kérést küld a webszervernek.
2. A webszerver fogadja a kérést, aztán továbbítja az alkalmazási szervernek.
3. Az alkalmazási szerver végrehajtja a módosításokat az oldalon.
4. Aztán visszaadja a már csak HTML kódból álló oldalt a webszervernek.
5. A webszerver válaszként továbbítja a kért oldalt a böngészőnek.

2. ábra: Dinamikus oldal előállítása

Forrás: http://help.adobe.com/en_US/dreamweaver/cs/using/images/ds_process_dynamic.png

1.4. A jó weboldalak alapkövetelményei

Egy jó weboldal legfőbb célja, hogy értékes információval bírjon, és igényes megjelenéssel rendelkezzen. Továbbá nagyon fontos, hogy jól használható legyen a weboldalunk. Ehhez a következő követelményeknek kell megvalósulniuk:

1. gyors és könnyű információszerzés
2. helyesen megfogalmazott szövegek az oldalon
3. olvashatóság, az oldal tartalmi áttekinthetősége
4. szemléletes médiaelemek
5. átlátható, logikus oldalrendezések
6. logikus weblap struktúra
7. egyértelmű navigáció
8. egyszerű felhasználóbarát felület

Továbbá biztosítanunk kell azt is, hogy a későbbiekben:

9. egyszerűen bővíthető és módosítható legyen
10. különböző böngészőkben és felbontáson is szépen jelenjen meg
11. különböző eszközökön is jól működjön

Ha ezeket a szempontokat betartjuk, akkor egy jól működő, látványos és gyakran látogatott weboldalt fogunk tudni tervezni (Kvaszingerné, 2013).

2. Felhasznált eszközök ismertetése

Ebben a fejezetben ismertetésre kerülnek a munka során felhasznált eszközök és programnyelvek. A weboldalunk létrehozása során HTML leíró nyelvet, CSS stílusleíró nyelvet, PHP szerveroldali szkriptnyelvet, JavaScript programozási nyelvet és MySQL adatbázist használtunk. A PHP kódok futtatásához az EasyPHP programcsomagot vettük igénybe, amely tartalmazta az Apache webszervert és MySQL adatbázist. A forráskód szerkesztése pedig NotePad++ szöveg- és forráskód szerkesztőben történt.



3. ábra: Felhasznált programnyelvek

Forrás: <http://www.jc-hicks.com/images/languages.png>

2.1. HTML

A HTML leírónyelvet (Hypertext Markup Language, magyarul Hiperszöveg leíró nyelv) használunk a weboldalak tartalmának és struktúrájának meghatározására. A HTML nem egy programozási nyelv, mert a három vezérlési szerkezet közül csak a szekvencia valósítható meg vele. Ez egy előírás, hogy hogyan írjuk meg a weblapokat helyesen, úgy hogy azt a számítógépek és a különböző berendezések képesek legyenek megjeleníteni (Kvaszingerné, 2013).

A hiperszöveg az Interneten található, formázatlan, egyszerű szöveg, amely a weblap tartalmát tárolja, továbbá programozási információkat, amelyek a weblap megjelenítéséhez és weblapok összekapcsolásához szükségesek. A hiperszöveget leírónyelv segítségével (markup language) formázzuk. A leírókód azt árulja el a szövegről, hogy milyen szerkezeti felépítést igényel és az elemek megjelenítése milyen legyen (DeBolt, 2005).

„A hiperszöveges jelölőnyelv írja le, hogy a szövegek, grafikák és más információkat tartalmazó fájlok hogyan épülnek fel és kapcsolódnak egymáshoz.” (Meloni, 2011, 2. o.)

2.1.1. HTML5

A HTML5 a HTML legújabb nemzedéke, a HTML 4.01 és az XHTML 1-et hivatott felváltani. A HTML5 olyan új szolgáltatásokat kínál, amelyek elengedhetetlenek a modern webalkalmazásokban (Pilgrim, 2011).

A HTML5 kifejlesztésének egyik célja, hogy a weboldalak működéséhez ne legyen szükség pluginok telepítésére. Továbbá számos új elem is került a nyelvbe, mint például az alkalmazások fejlesztése, a médiakezelés és az oldalelrendezést megvalósító elemek. Egyedüli hátránya, hogy vannak böngészők, amelyek egyes funkciókat még nem támogatnak (Kvaszingerné, 2013).

2.2. CSS

A CSS (Cascading Style Sheet – rangsorolt stíluslapok) egy programozási előírás. Ez úgynevezett stílus szabályok segítségével határozza meg a vizuális megjelenítést (DeBolt, 2005).

A nyelvezet független a HTML-től, de csak együttműködve használható. A használatához valamilyen módon össze kell kötni a HTML dokumentummal. Ez történhet beágyazás útján vagy csatolva. A csatolt stíluslapok nagy előnye, hogy hozzárendelhetjük több oldalhoz is, ezáltal ha változtatunk benne, hatással lesz az összes többi oldalra is.

A stíluslapok segítségével megváltoztathatók a HTML-en belüli elemek tulajdonságai. Például beállíthatjuk a színeket, szegélyeket, háttereket, térközöket és az elemek helyét az oldalon (Kvaszingerné, 2013).

„Röviden szólva, a stíluslapok kifinomultabb megjelenést tesznek lehetővé a Weben, ráadásul - elnézést a kihagyhatatlan szójátékért - mindezt stílusosan teszik.” (Meloni, 2011, 2. o.)

2.3. PHP

Eleinte személyes holnapok karbantartására használták, de képességei folyamatos bővülésével programozási nyelvvé nőtte ki magát. A PHP (Hypertext Preprocessor) egy általános szerver oldali leírónyelv, amely képes nagyméretű webes adatbázis-alapú alkalmazások működtetésére. Általában dinamikus és interaktív weboldalak létrehozására használják. Ez egy nyílt forráskódú szkriptnyelv, amelyet bárki elérhet és szerkeszthet (Zandstra, 2005).

Feladata, hogy a HTML kódba ágyazott kiszolgáló oldali parancsokat a szerver dolgozza fel és ne a böngésző értelmezze. Vagyis a weboldalak programja szervereken fut, a HTML szöveget viszont a kliens számítógép értelmezi majd. A PHP kódok végezhetnek adatbázis-lekérdezéseket, létrehozhatnak képeket, fájlokat olvashatnak és írhatnak, továbbá

feltételeket szabhatnak meg elemek megjelenésére vagy jogosultságaira. Gyakorlatilag felhasználási lehetőségeinek száma kimeríthetetlen (Programozásalapjai.hu, 2016).

Napjainkban az egyik legnépszerűbb webes platformra használt programozási nyelvnek számít. A PHP-nak temérdek erőssége van a többi programozási nyelvvel szemben. A legfontosabb közé tartozik a teljesítménye, különböző könyvtárak a webes feladatokhoz, az egyszerű használat és az alacsony költsége (Zandstra, 2005).

2.3.1. PHP5

A PHP5 a PHP legújabb generációja, ami számos új és hasznos szolgáltatással bővült. Ezáltal a programokat rövidebbé, elegánsabbá és rugalmasabbá teszik. Legfontosabb újításai közé tartozik az objektum orientált programozás és az XML támogatása, továbbá új MySQL kiterjesztés, beépített SQLite adatbázis-kezelő és átláthatóbb hibakezelés kivételekkel. A PHP5-nek számos verziója jelent meg, amelyek további újításokat tartalmaznak, ide sorolható a PHP5.1-től egészen az 5.6-os verzióig (Rimelek, 2011).

2.4. MySQL

Nagy mennyiségű adatok tárolására és előhívására SQL (Structured Query Language) adatbázis-kezelő rendszereket használhatunk. Az egyik legnépszerűbb adatbázis-kezelő szoftver a MySQL. Ez egy többfelhasználós, többszálú, SQL-alapú relációs adatbázis-kezelő szerver. Népszerűségét a nyílt forráskódú LAMP (Linux–Apache–MySQL–PHP) összeállításának köszönheti, amely olcsó és egyszerű megoldást kínál a dinamikus weboldalak szolgáltatására. Ez a típusú adatbázis egyre jobban elterjedt minden platformon és a Webes rendszerek nagy része is erre épül (Kolozár, 2016).

A MySQL használata nem túl bonyolult, mert az SQL nyelv az angol nyelvtanra épül, mivel ezt a nyelvet korábban nem csak a számítástechnikai szakemberek részére fejlesztették ki. Továbbá az SQL nyelv nem érzékeny a kis és nagybetűkre se (Sági, 2005).

2.5. JavaScript

A JavaScript programozási nyelv egy objektumalapú szkriptnyelv, amelyet weboldalakon előszeretettel használnak. A szerkezete közel áll a JAVA programozási nyelv szerkezetéhez, de működésük nem teljesen egyforma. A JavaScriptet arra használjuk, hogy a weboldal HTML kódját dinamikusan megváltoztathassuk némely események bekövetkezésekor. A JavaScript kódot a HTML-be illesztjük be, általában a <head> részbe.

A JavaScript kódot a böngésző értelmezi és futtatja, ezért nincs szükség webszerverre a használatához (Hörb, 2016a).

2.5.1. jQuery

A JavaScript-nek több keretrendszere van, amelyek előre megírt JavaScript függvényeket tartalmaznak. Ezek közül a legnépszerűbb a jQuery, amit arra használnak, hogy lerövidítsék a böngészők közti különbségeket kezelő hosszú kódokat egyetlen egy sorra. Használata megkönnyíti a HTML dokumentumok bejárását, manipulálását, animálását, stílusok módosítását és az események kezelését (Hörb, 2016b).

2.6. EasyPHP

A PHP kódok futtatásához telepíteni kell a megfelelő hozzávalókat, ami jelen esetben a PHP. Az EasyPHP egy ingyenes, nyílt forráskódú PHP fejlesztői környezet. Ez a programcsomag könnyen telepíthető és használható. A PHP-n kívül tartalmazza még az Apache webszervert és a MySQL adatbázist is (Harder, 2005).

2.6.1. Apache

Az Apache jelen pillanatban a legnépszerűbb webkiszolgáló az Interneten. Ez egy nyílt forráskódú, szabad felhasználású webkiszolgáló rendszer, ami nagy szerepet játszott a Világháló elterjedésében. Nagy előnye a hordozhatósága, vagyis képes működni a legtöbb operációs rendszeren. Továbbá a moduláris, bővíthető szerkezete, a nyílt forráskódja és hogy kompatibilis a HTTP protokollal (Lopez, 2007).

2.7. Notepad++

A Notepad++ teljesen ingyenes forráskód-szerkesztő program, amely számos programozási nyelvet támogat. Az alapját a Scintilla szövegszerkesztő komponens képezi, amit C++ programnyelven írtak. A Notepad++ legfontosabb képességei a kódkiemelés és kódblokkok egységbe zárása, a WYSIWYG szerkesztés, a felhasználó által állítható kódkiemelés, az automatikus kiegészítés és még számos jól használható funkció (Patai, 2009).

3. Webfelület és adatbázis

3.1. Feladat specifikáció

A rendszer tervezésénél az elsődleges szempont a témakiírásban szereplő fő célok betartása volt, vagyis egy olyan weboldal tervezése, amely az eladó és bérelhető ingatlanok árusításával foglalkozik. Továbbá, hogy a felhasználóknak legyen lehetőségük az ingatlanok közti keresésre és rendezésre, illetve a weboldal üzemeltetőinek legyen lehetőségük ingatlanok felvételére, módosítására és törlésére.

Ezeknek a követelményeknek eleget téve, egy dinamikus többoldalas bemutatkozó honlapot terveztünk. A weboldal nem tartalmaz regisztrációt, a honlap látogatói anélkül is igénybe vehetnek minden szolgáltatást. Módjukban áll a honlapra feltöltött ingatlanok között keresgélni különböző szempontok szerint és rendezni az ingatlanok listáját a saját igényeik alapján. Amennyiben, ha valaki komolyan érdeklődik egy ingatlan iránt, akkor fel kell vennie a kapcsolatot a weboldal üzemeltetőivel. A konzultáció történhet személyes, elektronikus, illetve telefonos úton is. A szükséges elérhetőségek megtalálhatók a weboldalon. Saját ingatlan feltöltésére egyelőre nincs lehetőségük a felhasználóknak, ehhez is az adminisztrátorok segítségét kell majd igényelniük.

Az adminisztrátori feladatok elvégzéséhez kialakításra kerül egy adminisztrátori felület is. Ezt azonban csak a weboldal üzemeltetői érhetik el, mivel a használatához adminisztrátori bejelentkezés szükséges. Az adminisztrátori felületen lehetőségük van az adminisztrátoroknak új ingatlanokat feltölteni, továbbá a meglévő ingatlanokat módosítani és törölni. Az ingatlan felvételénél és módosításánál képek feltöltése és módosítása is lehetséges. A weboldal üzemeltetőinek tehát minden olyan szükséges funkció elérhető lesz, amely egy ingatlanközvetítő iroda munkájához szükséges.

3.2. Képernyőképek, felülettervezés

A weboldal felületének megtervezésénél szem előtt tartottuk, hogy nem mindegyik felhasználó rendelkezik komoly számítástechnikai ismerettel, ezért egy könnyen kezelhető és egyszerűen átlátható oldalstruktúra került kialakításra. Fontos szempont volt még, hogy formailag egy igényes, jól kinéző webfelületet hozzunk létre. A stílusos kinézethez ötleteket az alábbi két ingatlanos weboldalból merítettünk:

<http://www.biztosingatlan.hu/>

<http://www.ingatlannet.hu/>

3.2.1. Felhasználói és adminisztrátori felülettervek

A weboldalunk szerkesztésének elkezdése előtt megterveztük a felhasználói és adminisztrátori felületünket.

A felhasználói oldal felületterve



4. ábra: Felhasználói felületterv

Forrás: saját szerkesztés

Az adminisztrációs oldal felületterve

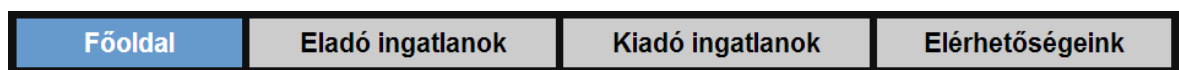


5. ábra: Adminisztrátori felületterv

Forrás: saját szerkesztés

3.2.2. A menürendszer

A felhasználói felület navigációs menüje (6. ábra) horizontálisan lett kialakítva a következő menüpontokkal: Főoldal, Eladó ingatlanok, Kiadó ingatlanok és Elérhetőségeink. Mindig az a menüpont lesz kiemelve, amelyik oldalon a felhasználó éppen tartózkodik.



6. ábra: Felhasználói menü

Forrás: saját szerkesztés

Az adminisztrátori felületen található navigációs menü (7. ábra) vertikálisan került kialakításra a következő menüpontokkal: Főoldal, Ingatlan felvétel, Ingatlan módosítás, Ingatlan törlés, Kijelentkezés. Ezeknél a menüpontoknál is megkülönböztethető az éppen használatban lévő oldal.



7. ábra: Adminisztrátori menü

Forrás: saját szerkesztés

3.2.3. Webfelület megvalósítása

A külső webfelület megvalósítása HTML leíró és CSS stílusleíró nyelvvel történt. A HTML-el létrehoztuk az oldal alapját, struktúráját. A HTML programozása formázó utasításokkal, úgynevezett tag-ek segítségével történt. Minden tag-nek van nyitó és záró része. A HTML legalapvetőbb tag-jei:

<HTML> : Azt jelzi a böngészőnek, hogy a közte lévő szöveget a HTML szabvány szerint kell értelmezni.

<HEAD> : Ez a HTML oldal fejrésze. Itt információkat helyezünk el, ide kerül például az oldal címe és karakterkódolása is.

<TITLE> : Ezzel adható meg a HTML oldal címe.

<BODY> : Ez a HTML oldalunk törzse, amit a böngésző megpróbál mindig megjeleníteni (Vári, 2007).

A HTML oldal elemeinek külső megjelenését CSS segítségével tudtuk formázni. A CSS-ben megírt stíluslapokat hozzá csatoltuk a HTML dokumentumokhoz. Egy külön fájlba íródtak meg a stíluslapok és a kívánt HTML oldalon csak hivatkozni kellett rájuk.

Részlet a felhasználói oldal CSS kódjából

```
*1 {  
    margin: 0;  
    padding: 0;  
    box-sizing: border-box;  
}  
body2 {  
    color: black;  
    background: #f0eee7 url('kepek/kekhatrer.png') repeat-x 0 -1px;  
    font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif;  
    font-size: 16px;  
}  
a3 {  
    text-decoration: none;  
    color: black;  
}
```

¹ Beállítja az összes tag-en a margót, a belső kitöltést és az alapértelmezett bokszt modellt.

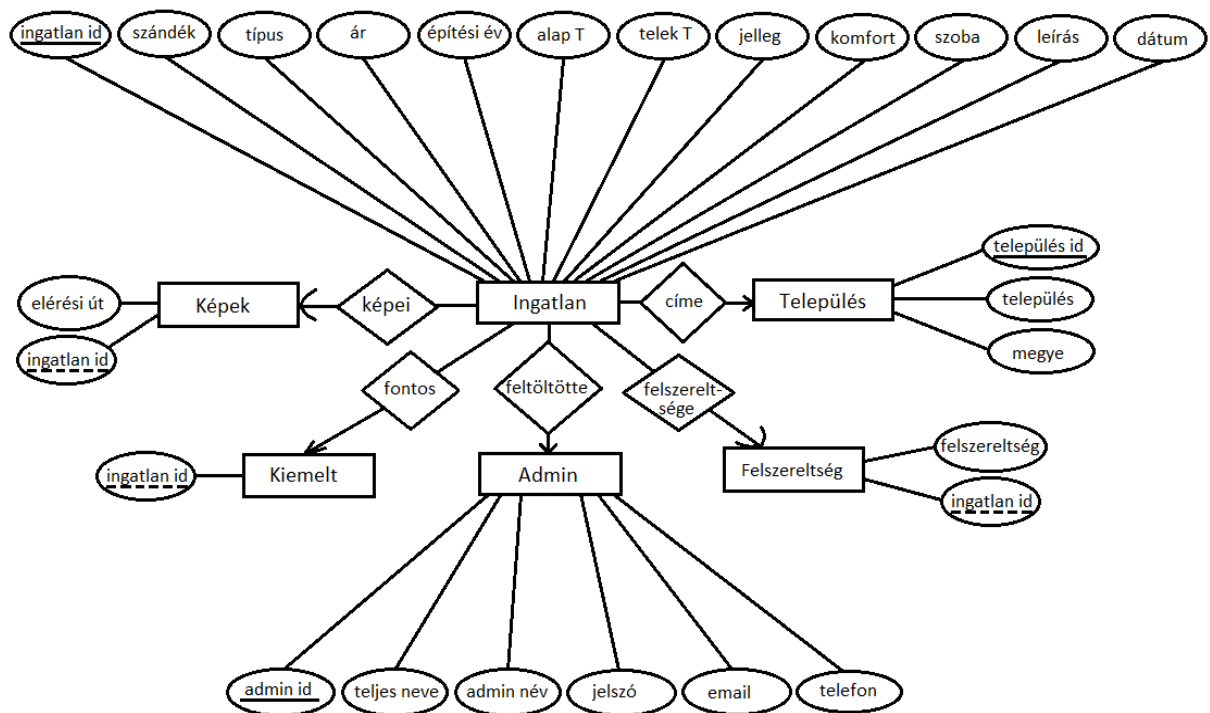
² Beállítja a body tag-en a szöveg színét, típusát és méretét, továbbá az oldal hátterét.

³ Beállítja az a tag-en a szöveg díszítettségét és a szöveg színét.

3.3. Adatbázis tervezése

Az adatbázis megtervezése során alakult ki az adatok tárolásához szükséges adatszerkezet. Első lépésben összegyűjtöttük, hogy milyen adatokat fogunk tárolni, majd ezen adatok alapján megterveztük a táblákat. A tervezés során a relációs adatmodellt használtuk. Ehhez előbb az egyed-kapcsolat diagramot (8. ábra) szerkesztettük meg, majd azt átvitük relációs adatmodell formájára. Az egyed-kapcsolat tervezésénél hat táblát használtunk. Az *Ingatlan* tábla volt a fő táblánk, ebből vezettük le a különböző relációs kapcsolatokat. A táblákat kulcsok segítségével kötöttük össze. Amelyik tábla nem tartalmazott kulcs mezőt, azokhoz külső kulcsokat rendeltünk.



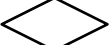
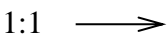
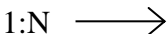
3.3.1. Egyed-kapcsolat diagram



8. ábra: Egyed-kapcsolat diagram

Forrás: saját szerkesztés

Az egyed-kapcsolat diagramban használt jelölések jelentései a következők:

- egyedhalmaz (tábla) 
- attribútumok (mezők) 
- kapcsolatok 
 - 1:1 
 - 1:N 

Az adatbázis egyed típusai:

- Adminisztrátorok (**ADMIN** egyedtípus)
- Ingatlanok (**INGATLAN** egyedtípus)
- Települések (**TELEPÜLÉS** egyedtípus)
- Felszereltségek (**FELSZERELTSÉG** egyedtípus)
- Ingatlanok képei (**KÉPEK** egyedtípus)
- Kiemelt ingatlanok (**KIEMLET** egyedtípus)

Az adatbázis attribútumai:

ADMIN (admin id, teljes neve, admin név, jelszó, email, telefon)

INGATLAN (ingatlan id, szándék, típus, ár, építkezési év, alap T, telek T, jelleg, komfort, szoba, leírás, dátum)

TELEPÜLÉS (település id, település, megye)

FELSZERELTSÉG (ingatlan id, felszereltség)

KÉPEK (ingatlan id, elérési út)

KIEMLET (ingatlan id)

3.3.2. Adatbázis normalizálása

A relációs adatbázis tervezésénél többlépcsős normalizálási módszereket végeztünk. Ennek a célja, hogy az adatbázis táblái biztonságosak legyenek és a lehető legkevesebb redundanciát (felesleges adatot) tartalmazzák. Ezeknek a módszereknek köszönhetően az adatbázis különböző „normál formákba” került (Szinyeigimibp.hu, 2011).

Első normál forma (1NF)

A tábla akkor van 1NF-ban, ha minden sor minden mezőjének értéke elemi.

Második normál forma (2NF)

A tábla akkor van 2NF-ban, ha 1NF-ban van, és minden másodlagos attribútum teljesen függ valamelyik kulcstól.

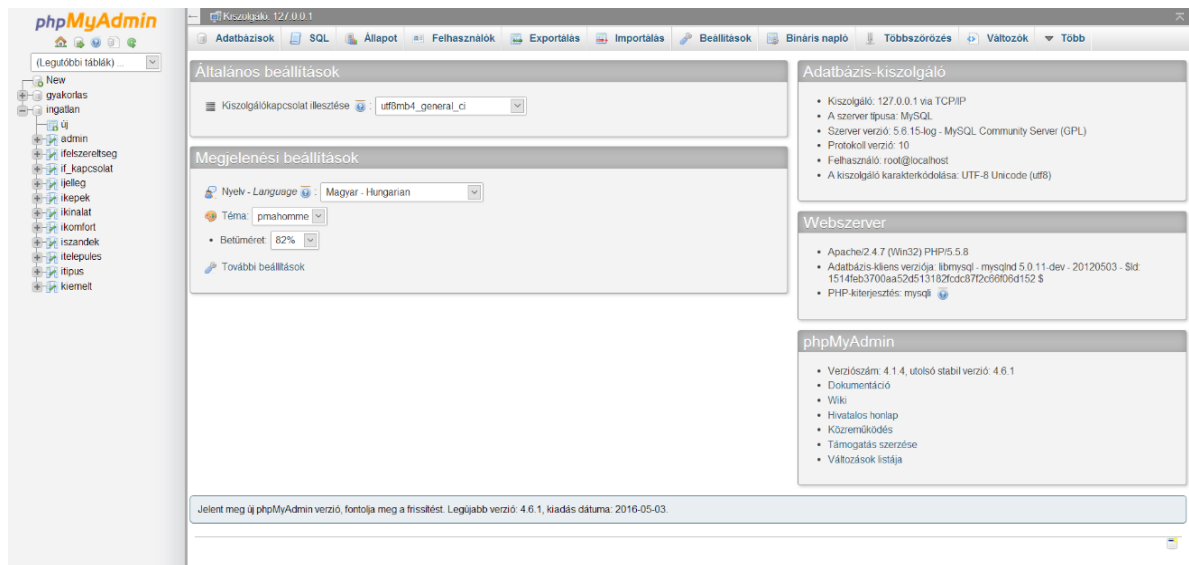
Harmadik normál forma (3NF)

A tábla akkor van 3NF-ban, ha 2NF-ban van, és nincs benne kulcstól való tranzitív függőség.

A normalizálási módszereknek köszönhetően az adatbázisunk 3NF-ba került. Így egy hatékony és jól áttekinthető adatbázis szerkezetet kaptunk, amelyek későbbi bővítése rugalmasan elvégezhető.

3.4. Adatbázis megvalósítása

Az adatbázis megvalósításához a phpMyAdmin nevű adatbázis-kezelő eszközt (9. ábra) használtuk. Ez egy nyílt forráskódú, PHP-ban megírt eszköz, ami a MySQL vezérlését teszi lehetővé az interneten. A grafikusan kialakított felületnek köszönhetően a kezelése nagyon egyszerű és gyorsan lehet benne dolgozni.



9. ábra: PhpMyAdmin kezelőfelület

Forrás: saját szerkesztés

Új adatbázis létrehozása az Adatbázisok menüpont alatt volt lehetséges. Mindössze az adatbázis nevét és az illesztés típusát (alapértelmezett karakterkódolást) kellett megadni. Az adatbázis nevének az *ingatlan*-t választottuk, az alapértelmezett kódolásnak pedig az *utf8_hungarian_ci*-t, mivel ez gyakorlatilag minden nyelv karaktereit tartalmazza. A későbbiekben aztán oda kellett figyelni arra is, hogy a szövegszerkesztőben a php fájlok írásához, a HTML fejlécében a `<meta>` tag-ben és az adatbázishoz való kapcsolódáskor is UTF-8-as kódolást használjunk.

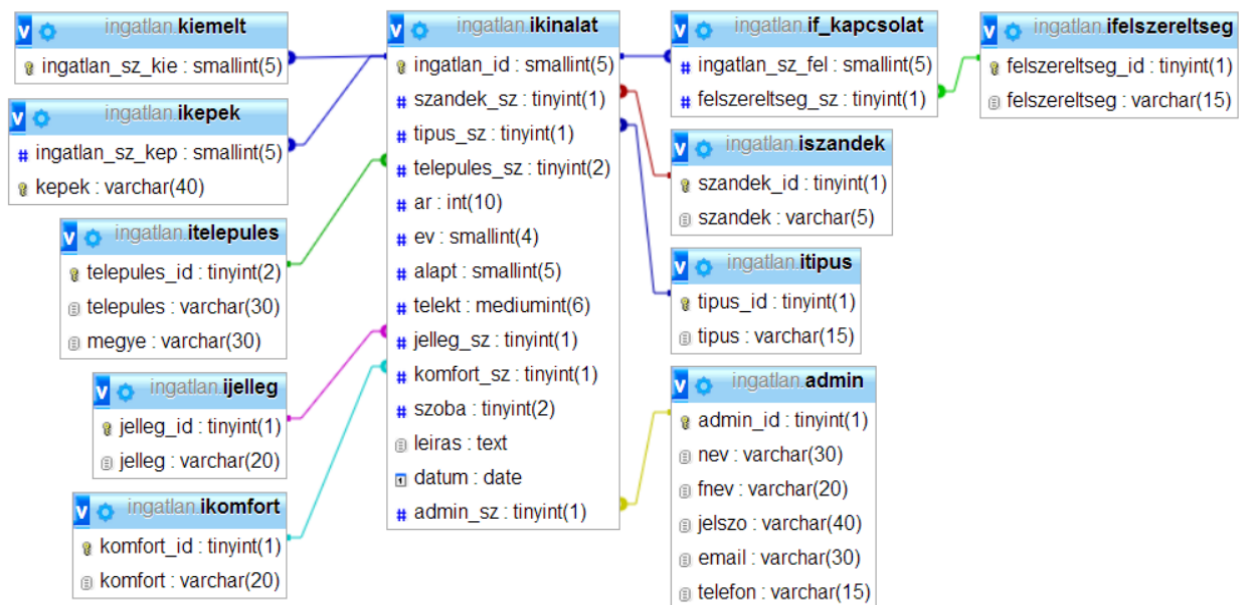
Új táblák létrehozása az adatbázisunkra kattintva volt lehetséges. Itt meg kellett adnunk a tábla nevét és az oszlopok számát, majd ezután, ha továbbléptünk, akkor megjelent egy űrlap, ahol az oszlopok neveit, típusait, néhol az adatok hosszát és végezetül a kulcs oszlopoknál az *auto_increment* tulajdonságot adtuk meg, ami az egyedi azonosítást tette lehetővé. A munkánkhoz összesen hét adattípust használtunk. Az egész számok tárolására *tinyint*, *smallint*, *mediumint*, *int* típusokat, a karakterláncok és hosszabb szöveg tárolására *varchar*, *text* és a dátum típus tárolására pedig *date*-t használtunk. Mindegyik táblában egyedi kulcsok alkalmazása tette lehetővé a különböző rekordok azonosítását.

A táblák összekapcsolása külső kulcsok használatával volt lehetséges. Az ilyen külső kulcsokat a Szerkezet menüpont alatt, a Reláció nézetre kattintva tudtuk beállítani.

Az üres táblák adatokkal való feltöltése a Beszúrás menüpont alatt történt. Itt mindössze annyi a dolgunk volt, hogy ki töltsük a tábla létrehozásakor megadott oszlop mezőket.

3.4.1. Adatbázis szerkezete

Az adatbázisunk végleges szerkezete a következőképpen néz ki.



10. ábra: Adatbázis szerkezete

Forrás: saját szerkesztés

3.4.2. Táblák bemutatása

1. Ikinalat tábla

Az *ikinalat* nevű táblában tárolódnak a feltöltött ingatlanok adatai. Ez a legfontosabb táblánk, ennek szinte mindegyik táblával össze kell lennie kötve. Egy ingatlant sok adat jellemez, ezért sok mezőt tartalmaz. Az *ikinalat* tábla a következő mezőket tartalmazza:

„*ingatlan_id*” – Ez a feltöltött ingatlan azonosítóját tartalmazza, amely a tábla elsődleges kulcsa is egyben. Az *ikepek* és a *kiemelt* táblánál is ezt használjuk azonosításra.

„*szandezek_sz*” – A feltöltött ingatlan szándékát határozza meg. Ez egy külső kulcs, amely összeköti az *ikinalat* táblát az *izsandezek* táblával.

„*tipus_sz*” – A feltöltött ingatlan típusát határozza meg. Külső kulcs, amely összeköti az *ikinalat* táblát az *itipus* táblával.

„**telepules_sz**” – A feltöltött ingatlan helyét határozza meg. Külső kulcs, amely összeköti az *ikinalat* táblát az *itelepules* táblával.

„**ar**” – A feltöltött ingatlan értékét tartalmazza. Az eladó ingatlanoknál az ingatlan értéke Ft-ban, a kiadó ingatlanoknál az ingatlan értéke egy hónapra Ft/hó-ban.

„**ev**” – A feltöltött ingatlan építkezési évét tartalmazza. A nem épület típusoknál ez az érték nulla.

„**alapt**” – A feltöltött ingatlan alapterületét tartalmazza.

„**telekt**” – A feltöltött ingatlan telekterületét tartalmazza.

„**jelleg_sz**” – A feltöltött ingatlan jellegét határozza meg. Külső kulcs, amely összeköti az *ikinalat* táblát az *ijelleg* táblával.

„**komfort_sz**” – A feltöltött ingatlan komfort fokozatát határozza meg. Külső kulcs, amely összeköti az *ikinalat* táblát az *ikomfort* táblával.

„**szoba**” – A feltöltött ingatlan szobáinak számát tartalmazza. A nem épület típusoknál ez az érték nulla.

„**leiras**” – A feltöltött ingatlan részletes leírását tartalmazza.

„**datum**” – Az ingatlan feltöltési dátumát tartalmazza „ÉÉÉÉ-HH-NN” formában.

„**admin_sz**” – Az ingatlant feltöltő adminisztrátor azonosítóját tartalmazza. Külső kulcs, amely összeköti az *ikinalat* táblát az *admin* táblával.

2. Iszandek tábla

Az *iszandek* táblában tárolódnak a szándék lehetőségek, vagyis, hogy egy ingatlan eladó vagy kiadó lesz. A normalizálások során hoztuk létre ezt a táblát. Ennek a táblának két mezője van.

„**szandek_id**” – A tábla kulcsa, ami a szándék típus azonosítására szolgál.

„**szandek**” – Ez tartalmazza az ingatlan szándék típusait.

3. Itelepules tábla

Az *itelepules* táblában tárolódnak a feltöltött ingatlanok települései és megyéi. Kizárólag különböző megyék esetén tartalmazhat egyforma nevű településeket. Ennek a táblának három mezője van.

„**telepules_id**” – A tábla kulcsa, ami a települések azonosítására szolgál.

„**telepules**” – Ez tartalmazza az ingatlanok településeit.

„**megye**” – Ez tartalmazza a településekhez tartozó megyéket.

4. Itípus tábla

Az *itípus* táblában tárolódnak az ingatlan típusok. A következő ingatlan típusokat különböztetjük meg: családi ház, lakás, üzleti, telek, mezőgazdasági. A normalizálások során hoztuk létre ezt a táblát. Ennek a táblának két mezője van.

„*tipus_id*” – A tábla kulcsa, ami a típus fajták azonosítására szolgál.

„*tipus*” – Ez tartalmazza az ingatlan típusokat.

5. Ijelleg tábla

Az *ijelleg* táblában tárolódnak az ingatlan jelleg fajták. A következő jelleg fajtákat különböztetjük meg: téглаépület, faház, beépítetlen terület, erdő, szántó, külterületi szántó, külterület. A normalizálások során jött létre ez a táblát. Ennek a táblának két mezője van.

„*jelleg_id*” – A tábla kulcsa, ami a jelleg fajták azonosítására szolgál.

„*jelleg*” – Ez tartalmazza az ingatlan jelleg fajtákat.

6. Ikomfort tábla

Az *ikomfort* táblában tárolódnak az ingatlan komfort fokozatok. A következő komfort fokozatokat különböztetjük meg: összkomfort, komfortos, félkomfort, komfort nélküli, szükséglakás, közművesített, közművek nélküli. A normalizálások során jött létre ez a táblát. Ennek a táblának két mezője van.

„*komfort_id*” – A tábla kulcsa, ami a komfort fokozatok azonosítására szolgál.

„*komfort*” – Ez tartalmazza az ingatlan komfort fokozatokat.

7. Ifelszereltség tábla

Az *ifelszereltség* táblában tárolódnak az ingatlan felszereltség típusok. Egy ingatlan összesen a következő felszereltségekkel rendelkezhet: víz, gáz, villany, központi fűtés, pince, csatorna, garázs. Ennek a táblának két mezője van.

„*felszereltség_id*” – A tábla kulcsa, ami a felszereltségek azonosítására szolgál.

„*felszereltség*” – Ez tartalmazza a felszereltség típusokat.

8. If_kapcsolat tábla

Az *if_kapcsolat* táblában két külső kulcs tárolódik, amelyek az *ikinalat* és az *ifelszereltség* táblákat kötik össze. A normalizálások során hoztuk létre ezt a táblát. Ennek a táblának két mezője van.

„*ingatlan_sz_fel*” – Külső kulcs, amely az ingatlanok azonosítására szolgál.

„*felszereltség_sz*” – Külső kulcs, amely a felszereltségek meghatározására szolgál.

9. Ikepek

Az *ikepek* táblában tárolódnak a feltöltött ingatlanok képei. Egy ingatlanhoz maximálisan nyolc kép tartozhat. Ennek a táblának két mezője van.

„*ingatlan_sz_kep*” – Külső kulcs, amely az ingatlanok azonosítására szolgál.

„*kepek*” – Ez tartalmazza az ingatlanokhoz tartozó képek elérési útjait.

10. Kiemelt

A *kiemelt* táblában tárolódnak a kiemelt ingatlan ajánlatok. Ennek a táblának mindössze egy mezője van.

„*ingatlan_sz_kie*” – Külső kulcs, amely a kiemelt ingatlanok azonosítására szolgál.

11. Admin

Az *admin* táblában tárolódnak az oldal adminisztrátorainak az adatai. Ennek a táblának hat mezője van.

„*admin_id*” – Az admin tábla elsődleges kulcsa, ami az adminisztrátorok azonosítóit tartalmazza.

„*nev*” – Az adminisztrátorok teljes nevét (vezetéknév, keresztnév) tárolja.

„*fnev*” – Az adminisztrátorok felhasználó nevét tárolja, ami az adminisztrátori felület belépéséhez szükséges.

„*jelszo*” – Az adminisztrátorok jelszavát tárolja, ami az adminisztrátori felület belépéséhez szükséges. Az adminisztrátorok jelszavai md5-ös kódolást kaptak, ami a védelem szempontjából volt fontos. Ez egy 128 bites, egyirányú kódolási algoritmus.

„*email*” – Az adminisztrátorok email címét tárolja.

„*telefon*” – Az adminisztrátorok telefon számát tárolja.

4. Rendszer bemutatása

Ebben a fejezetben bemutatásra kerül az ingatlanos rendszerünk és az oldal dinamikussá alakításához használt programozási módszerek. A programozás elsősorban PHP nyelv segítségével történt, ami a szerver oldali programozást tette lehetővé számunkra. A PHP segítségével tudtuk a különböző tartalmú ingatlanokat kiíratni, a keresést, a rendezést és a bejelentkezést megvalósítani, továbbá az űrlapok továbbítását, a képméretezést és az adatbázis kezelést is lehetővé tette számunkra. Szerencsénkre a PHP rengeteg beépített függvényt tartalmaz, ami nagyban megkönnyítette a weboldalunk programozását.

4.1. Felhasználói felület

4.1.1. Főoldal

A weboldalunk megnyitásakor először az *index.php* fájl hívódik meg, amely a weboldalunk főoldalát tartalmazza. A főoldal a felhasználók számára tartalmaz egy *kiemelt ajánlataink* részt és a *legfrissebb ingatlanok* listáját.

A kiemelt ajánlatoknál (11. ábra) mindig három ingatlan jelenik meg, amiket a *kiemelt* táblából véletlenszerűen hív elő a php programunk. Ehhez először egy két dimenziós asszociatív tömbbe lekérdezzük a kiemelt ingatlanokat, az adataikkal együtt, majd az *array_rand()* beépített függvény segítségével véletlenszerűen kiválasztunk három ingatlant ebből a tömbből.

Kiemelt ajánlataink

 Kiadó ház, Eger 520 000 Ft/hó Szobák: 2 Alapterület: 90 m² Telekterület: 990 m² RÉSZLETEK...	 Eladó üzlet, Kaposmérő 9 000 000 Ft Szobák: 0 Alapterület: 100 m² Telekterület: 400 m² RÉSZLETEK...	 Eladó lakás, Fertőd 14 800 000 Ft Szobák: 1 Alapterület: 55 m² Telekterület: 0 m² RÉSZLETEK...
--	--	--

11. ábra: Kiemelt ingatlanok

Forrás: saját szerkesztés

A legfrissebb ingatlanok részénél az adatbázisba utoljára bevitt négy ingatlan jelenítődik meg. Ezt egy SQL lekérdezéssel oldottuk meg. A lekérdezés során mindössze négy darab ingatlan adatait kértük le dátum szerint csökkenő sorrendbe.

A legfrissebb ingatlanok lekérdezéséhez használt SQL utasítás

```
SELECT ingatlan_id, szandek, tipus, telepules, ar, alapt, telekt, szoba, leiras, datum, kepek
FROM ikinalat, iszandek, itipus, itelepules, ikepek
WHERE szandek_sz = szandek_id AND tipus_sz = tipus_id AND telepules_sz = telepules_id
AND ingatlan_sz_kep = ingatlan_id AND kepek LIKE '%1.jpg'
ORDER BY datum DESC LIMIT 4
```

Az SQL lekérdezésünknel a SELECT utasítás egy eredménytáblát ad vissza, amely tartalmazza az ingatlanjaink azonosítóját, szándékát, típusát, települését, árát, alapterületét, telekterületét, szobák számát, részletes leírását, a feltöltés dátumát és egy képet az ingatlanról. A FORM után a lekérdezéshez használt táblák neveit adjuk meg. A WHERE kulcsszó után a különböző feltételeket, amelyek segítségével összekötjük a tábláinkat és meghatározzuk, hogy miket kérjen le az adatbázisból. Az ORDER BY segítségével a lekérdezés sorrendjét határozhatjuk meg, a LIMIT korlátozással pedig azt, hogy legfeljebb hány sora legyen a lekérdezésünknek.

4.1.2. Mysqli burkoló készítése

A leggyakrabban előforduló adatbázis-kezelő függvényekhez készítettünk egy *mysqli_burkolo.php* nevű fájlt, ami az adatbázis-kezelésnél előforduló feladatokhoz nyújt olyan speciális függvényeket, melyekkel egy sorba lerövidíthetjük az adatbázis-kezelő eljárásokat. Több függvényt is tartalmaz, amelyeket egy közös "DB_" előtaggal láttunk el.

1. táblázat: Mysqli burkoló

függvény	argumentumok	visszatérési érték	művelet
DB_connect()	5 db string	resource	Kapcsolódás egy adatbázishoz
DB_close()	resource	bool	Kapcsolat lezárása
DB_query()	string	resource	SQL utasítás lefuttatása
DB_getrows()	string	array	SQL lekérdezés eredménytáblája
DB_getrow()	string, integer	array	SQL lekérdezés eredménytáblájának egy sora

DB_getcolumns()	string	array	SQL lekérdezés eredménytáblája oszlopok szerint felsorolva
DB_getcolumn()	2 db string	array	SQL lekérdezés eredménytáblájának egy oszlopa
DB_getfield()	2 db string, integer	string	SQL lekérdezés eredménytáblájának egy cellája
DB_fetch_assoc()	resource	array	Eredménytábla következő sora
DB_num_rows()	resource	integer	Eredménytábla sorainak száma
DB_affected_rows()	-	integer	Legutóbbi művelet során érintett sorok száma
DB_insert_id()	-	integer	Legutóbbi INSERT művelet során beszúrt auto-increment értéke
DB_next_id()	string	integer	A tábla következő auto-increment értéke
DB_error()	bool	string	Hibakezelő függvény

Forrás: <http://php.webprog.biz/phptanfolyam/adatbaziskezeles/4-sql-fuggvenyek>

Az adatbázis használata esetén az *include()* függvény segítségével minden alkalommal beépítettük *mysqli_burkolo.php* fájlt, így fel tudtuk használni az összes benne szereplő függvényt. A munkánk során az adatbázis kapcsolódására a *DB_connect()*, a kapcsolat lezárására a *DB_close()* és a hibakezelésre a *DB_error()* függvényt használtuk. Az eredménytáblák lekérdezésére a *DB_getrows()*, egy sor lekérdezésére a *DB_getrow()*, egy oszlop lekérdezésére pedig a *DB_getcolumn()* függvényt használtuk. További SQL utasítások, mint például a beszúrás, módosítás, törlés (INSERT, UPDATE, DELETE) futtatásához a *DB_query()* függvényt használtuk. A beszúrás művelet után pedig, ha meg akartuk határozni a tábla auto-increment értékét, akkor ezt a *DB_insert_id()* függvénnyel tehetjük meg.

Az adatbázishoz való csatlakozás localhoston keresztül

```
DB_connect("localhost", "root", "bcmunka", "ingatlan", "UTF8");
```

Az adatbázishoz való csatlakozást egy külön *setting.php* nevű fájlban írtuk meg, és mindegyik php fájlnál, ahol adatbázissal dolgoztunk *include()* segítségével meghívtuk. Ezt abból a célból írtuk külön fájlba, hogy szerver vagy adatbázis változása esetén mindössze egy helyen kelljen átírnunk az adatbázis kapcsolódásához szükséges adatokat. Az ilyen esetekben változik az első három paraméter, a domain név, amin az adatbázis fut, az adatbázis felhasználóneve és jelszava.


4.1.3. Eladó és kiadó ingatlanok oldal

Az *eladó és kiadó ingatlanok* oldala (12. ábra) szerkezetileg teljesen egyforma, egyedül abban különbözik a két oldal egymástól, hogy az egyik az eladó ingatlanok, a másik pedig a bérelhető ingatlanok számára lett kialakítva. Ezek az *elado.php* és *kiado.php* fájlba lettek megírva. Ezeken az oldalakon keresztül kerülnek megjelenítésre az eladó vagy a kiadó ingatlan ajánlatok. Minden kis ingatlan ajánlat tartalmazza az ingatlan szándékát, települését, árát, típusát, szobák számát, alapterületét, telekterületét, feltöltési dátumát és egy fotót az ingatlanról. Amennyiben rákattintunk egy ingatlan ajánlatra, az tovább dob minket az *ingatlan.php* oldalra, ahol további adatok és képek jelennek meg az ingatlanról. Az *eladó és kiadó ingatlanok* oldalon egyszerre hat darab ingatlan megjeleníthető meg, a többi külön oldalakon érhetjük el. Az ingatlan lista több oldalba való törését egy lapozó segítségével oldottuk meg. Továbbá ezeken az oldalakon van lehetőségük a felhasználóknak keresni az adatbázisban szereplő ingatlanok között és rendezni az ingatlanokat ár vagy dátum alapján.


Eladó ingatlanok

(24 db találat)

Rendezés: legfrissebb elől ▼ Rendezés



Eladó ház, Érd
19 900 000 Ft
Típus: Családi ház Szobák: 4
Alapterület: 160 m² Telekterület: 3230 m²
Feltöltve: 2016.05.14 [Részletek >>](#)



Eladó lakás, Fertőd
14 800 000 Ft
Típus: Lakás Szobák: 1
Alapterület: 55 m²
Feltöltve: 2016.05.07 [Részletek >>](#)

1 2 3 4

Ingatlan kereső

Település:

Típus:

Szobák:

Terület: m²

Ár: - millió Ft

12. ábra: Eladó ingatlanok

Forrás: saját szerkesztés

4.1.3.1. Ingatlan ajánlatok listázása

Az ingatlanok listázásához először lekérdeztük az ingatlanok adatait és egy képét az adatbázisból, majd egy for-ciklus segítségével ezeket kiíratuk a megfelelő helyeken.

Az eladó ingatlanok lekérdezéséhez használt SQL utasítás

```
SELECT ingatlan_id, szandek, tipus, telepules, ar, alapt, telegt, szoba, datum, kepek  
FROM ikinalat, iszandek, itipus, itelepules, ikepek  
WHERE szandek_sz = szandek_id AND tipus_sz = tipus_id AND telepules_sz = telepules_id  
AND ingatlan_sz_kep = ingatlan_id AND kepek LIKE '%1.jpg' AND szandek = 'eladó'
```

4.1.3.2. Oldal lapozó

A lapozó segítségével tudjuk az ingatlanjainkat külön oldalakra szétosztani, ezáltal áttekinthetőbbé téve a weboldalunkat. A lapozó a 12. ábra alján látható.

A lapozó működését a következőképpen valósítottuk meg:

- Megszámoltuk a feltételeknek eleget tevő ingatlanokat az adatbázisban.
- Meghatároztuk, hogy mennyi ingatlant akarunk egyszerre megjeleníteni egy oldalon.
- Ebből a két értékből kiszámoltuk, hogy hány oldalra lesz szükségünk összesen az ingatlanok megjelenítéséhez.
- Megvizsgáltuk egy feltétellel, hogy van-e átadva oldalszám a kiválasztott oldal lekéréséhez. Amennyiben nem volt, akkor alapértelmezett oldalszámnak az egyest vettük, és az első oldalt jelenítettük meg.
- Az oldalszámok átadása GET metódussal történt, a következőképpen:

```
print '<a href="elado.php?sess=1&osz='.$_i.'" title="oldal'.$_i.'"> '.$_i.' </a>';
```

Az <a> tag href attribútumában adhatjuk meg, hogy melyik linkre ugorjon a weboldalunk. Az URL ? utáni részében pedig a változók listáját név=érték formában & jelekkel elválasztva adhatjuk meg. Itt láthatóan két változó került átadásra, az egyik az *osz* változó, amely a megnyitandó oldal oldalszámát tartalmazta, a másik pedig a *sess* változó, ez a session létezését továbbította. Ezek a változók az átadás után a *\$_GET* asszociatív tömbbe kerültek és az oldalszámra a *\$_GET["osz"]* formában hivatkozhattunk.

- Az átadott oldalszámnak megfelelő ingatlanok kiírása egy for-ciklus segítségével történt. A for-ciklus mindegyik oldalnál a megjelenítendő ingatlanok számaig ment, kivéve az utolsót, mert ott már csak a maradékot jelenítette meg.

4.1.3.3. Munkamenet-kezelés

A keresés és rendezés megvalósításához elengedhetetlen lépés volt a munkamenet használata. A munkamenet (session) az a folyamat, amely során a felhasználó a weboldalunk aloldalai között böngészik és különböző kérések által egy összefüggő folyamatot végezhet. Ezt használjuk például, hogy a bejelentkezett felhasználó bejelentkezett állapotban maradjon. A mi esetünkben arra használtuk, hogy a keresés vagy a rendezés használata után megjegyezve maradjanak a keresési feltételek is és a rendezési beállítás is. Továbbá az oldalak lapozásánál is hasznát vettük, hogy megmaradjon az aktuális keresés vagy rendezés szerint kigenerált ingatlan lista.

Egy munkamenet elindításához a `session_start()` függvényt használjuk. Ezt a függvényt minden aloldal elején meg kell hívnunk, ahol használni szeretnénk a munkamenetet. A függvény meghívása után használhatjuk `$_SESSION` asszociatív tömböt, ami egy szuperglobális változó. Vagyis amit belerakunk ebbe a tömbbe, azok elérhetőek lesznek azokon az aloldalakon is, ahol meg lett hívva a `session_start()` függvény. A munkamenet addig lesz használható, amíg meg nem szüntetjük. A teljes munkamenetet a böngésző bezárásával vagy a `session_destroy()` függvénnyel lehet megszüntetni. Viszont, ha nem szeretnénk megszüntetni a teljes munkamenetet, csak változókat törölni belőle, akkor használhatjuk az `unset()` függvényt is (Hörb, 2016d).

4.1.3.4. Keresés

Az *eladó és kiadó ingatlanok* oldalon található jobboldalt elhelyezve egy kereső panel (13. ábra). Ezzel a weboldal látogatóinak lehetőségük van az oldalon elhelyezett hirdetések közül csak azokat az ajánlatokat listázni, amelyek megfelelnek számukra. Ehhez egy űrlapot használtunk, ami hat beviteli mezőt tartalmazott (három szövegbeviteli mezőt és három legördülő menüt) és egy submit gombot az űrlap küldéséhez. A keresésnél megadhatjuk a keresendő ingatlan helyét, típusát, szobáinak számát, alapterületét és hogy milyen összeg között legyen az ingatlan ára. Túl sok adat megadása esetén előfordulhat, hogy nem talál számunkra megfelelő ajánlatot a kereső. Az *eladó ingatlanok* és a *kiadó ingatlanok* oldal keresője szinte teljesen megegyezik, annyi különbséggel, hogy az egyik az eladó ingatlanok között keres és az ingatlan ára millió Ft-ban adható meg, a másíknál pedig az ingatlan árát ezer Ft-ban kell megadni és a bérelhető ingatlanok között keres.

13. ábra: Kereső

Forrás: saját szerkesztés

Kereséskor, ha kitöltöttük vagy kiválasztottuk valamelyik űrlapelemet és elküldtük a *Keresés* gombbal, akkor POST metódus segítségével átadtuk az adatokat az oldalunknak. Miután átadtuk az adatokat, a *\$_POST* asszociatív tömbből érhattük el őket. Különböző feltételek vizsgálata után ezeket elmentettük a munkamenetünkbe, vagyis hozzáadtuk a *\$_SESSION* tömbhöz. Így ezek egészen addig megjegyezve maradtak, amíg nem törölődtek a munkamenetből. A keresés után az ingatlanok lekérdezését pedig úgy oldottuk meg, hogy az alapkérdezéshez a *WHERE* feltételbe hozzáfűztünk olyan feltételeket, amelyeket a keresővel küldtünk át az oldalnak. A keresőbe beírt mezők és a kiválasztott elemek addig megjegyezve maradtak, amíg a munkamenetből nem törölődtek az azokhoz elmentett változók. Ezek akkor törölődtek a munkamenetből, ha a weboldalunk navigációs menüjét használva egy másik oldalra léptünk át.

4.1.3.5. Rendezés

A keresés után, vagy akár előtt is módunkban áll változtatni a kilistázandó ingatlan ajánlatok sorrendjén. Ezt egy rendező (14. ábra) segítségével tudjuk kivitelezni. Ehhez egy űrlapot használtunk, ami csupán egy legördülő menüt és egy submit gombot tartalmaz. A rendezés négy féleképpen történhet: a dátum és az ár alapján csökkenő vagy növekvő sorrendbe. Az alapértelmezett rendezés szerint a legfrissebb ingatlan lesz elől. Ez is hasonló elven működött, mint a keresés. Itt is POST metódussal adjuk át az adatokat, és ezeket is el kellett tárolnunk session-el. A különbség csupán az, hogy az alapkérdezéshez az *ORDER BY* utasítással fűzzük hozzá az átadott adatot.

14. ábra: Rendező

Forrás: saját szerkesztés

4.1.4. Ingatlan ajánlatok részletei

Amikor egy felhasználó rákattint az általa kiválasztott ingatlan ajánlatra, akkor egy másik oldalon, előugrik az ingatlanról egy részletes leírás, a hozzátartozó összes adattal és képpel együtt. Az adatok rendezetten, táblázat formájában jelenítődnek meg. Továbbá az oldal tartalmaz egy képgalériát is, amellyel a felhasználóknak lehetőségük van az ingatlan képeit alaposabban megtekinteni.

Az oldal megvalósítása az *ingatlan.php* nevű fájlban történt. Meghívásakor először GET metódus segítségével áadtunk egy *id* nevű változót, amely a kiválasztott ingatlan azonosítóját tartalmazza. Ez segítségével lekérdeztük az ingatlan adatait és képeit, majd a megfelelő helyeken kiírtuk ezeket. A képgaléria megvalósításához a Slimbox2 nevű webes alkalmazást használtuk fel, ami JavaScript és jQuery programnyelven lett megírva. A képgalériához tartozó programfájlokat a HTML dokumentumunk fejrésében hívtuk meg.

Eladó ház, Fertőszentmiklós

Fertőszentmiklós városában, Soprontól 20, az ausztriai határátkelőtől 15 percre, nyugodt utcában található ez az 5 szobás, plusz különálló apartmannal rendelkező családi ház. A lakóház kétszintes, 1998-ban épült téglából, betonfödémekkel, külső hőszigeteléssel, gáz központi fűtéssel. A földszinten 2 szoba, konyha-étkező, fürdő (sarokkáddal, zuhanyzóval) külön WC, kamra, kazánház, előszoba helyiség került kialakításra. Az emeleten 3 szoba, konyha, fürdőszoba (szerelvényezés megvan de nincs burkolva) helyiség található. Az ingatlanhoz kb. 30 nm-es külön bejáratú téglából épült lakrész tartozik, 1 szobával, konyhával, fürdővel. Fűtése a főépületről és fatüzelésű kályhával is megoldott. Garázs és még egy autó tárolására alkalmas pajta is van az udvaron. A kert bekerített, konyhakertként hasznosított. Az ingatlan jó állapotú, több generáció együttélésére vagy akár kiadásra is alkalmas. Az ár alkuképes!



Település:	Fertőszentmiklós
Megye:	Győr-Moson-Sopron
Típus:	Családi ház
Ár:	26 000 000 Ft
Épitkezési év:	1999
Alapterület:	200 m²
Telekterület:	920 m²
Jelleg:	Téglaépület
Komfort:	Összkomfort
Szobák:	3
Víz:	Van
Gáz:	Van
Villany:	Van
Központi fűtés:	Van
Pince:	Van
Csatorna:	Van
Garázs:	Van
Feltöltés dátuma:	2015.09.30
Feltöltötte:	Hagyma István

15. ábra: Ingatlan részletei

Forrás: saját szerkesztés

4.1.5. Elérhetőségeink oldal

Amennyiben valamelyik ingatlan felkelti egy látogató érdeklődését, vagy saját ingatlanját szeretné reklámozni az oldalon, akkor ahhoz a weboldal üzemeltetőivel kell felvennie a kapcsolatot. A kapcsolat felvétel történhet személyes, elektronikus, illetve telefonos úton is. Az ehhez szükséges adatokat az *Elérhetőségeink oldalon* találja meg. Ez az oldal tartalmazza a weboldalt üzemeltető adminisztrátorok elérhetőségeit táblázat formájában, ahol a nevük, telefonszámuk és az email címük van megadva. Továbbá az ingatlan iroda pontos címe is elérhető, amely biztos megtalálásához egy Google Térképet is beépítettünk az oldalra.

Az oldal megvalósítása az *elerhetoseg.php* fájlban történt. Ennél az oldalnál nagyon egyszerű dolgunk volt, mindössze az adminisztrátorok adatait kellett lekérnünk az adatbázisból és kiíratnunk az oldalon. A Google Térképhez pedig egy javascript kódot kellett beszúrunk az ingatlan iroda pontos koordinátaival és címével megadva.

4.2. Adminisztrátori felület

4.2.1. Bejelentkezés

Az adminisztrátori felületre csak a jogosultsággal rendelkező személyek, az úgynevezett adminisztrátorok léphetnek be. Ezen feltétel biztosításához létrehoztunk az adminisztrátori felülethez egy bejelentkezési lapot (16. ábra). Itt minden alkalommal, ha be szeretnénk lépni, meg kell adni egy felhasználónév-jelszó párost, és csak akkor enged tovább, ha az adatbázisban szereplő valamelyik adminisztrátor bejelentkezési adatait adtuk meg, különben a „Nem megfelelő felhasználónév vagy jelszó!” hibaüzenettel tér vissza.

Mivel az adminisztrátori felülethez írt fájlok egy külön *admin* nevű mappába kerültek és a bejelentkezési lap az *index.php* fájlal lett megvalósítva, ezért a bejelentkezési lapot az URL-ben megadva egy *admin* kulcsszóval tudjuk előhívni.

A bejelentkezéshez egy űrlapot használtunk, ami két beviteli mezőt és egy submit nyomógombot tartalmaz. Ha beírtuk a nevet és jelszavat, akkor a POST metódus segítségével továbbítottuk az adatokat a *login.php* nevű fájlba. Ebbe a fájlba először lekértük az adatbázisban szereplő adminisztrátorok adatait, majd leellenőriztük, hogy valamelyik adminisztrátor felhasználóneve és jelszava megegyezik-e a POST metódus által elküldött adatokkal. A POST által küldött jelszót még az *md5()* függvénnyel át kellett alakítanunk, mielőtt az ellenőrzésekbe belekezdhattunk. Erre azért volt szükség, mert az adatbázisban szereplő jelszavak is md5-ös kódolással lettek ellátva a védelem céljából.

Amennyiben a vizsgálatok után megegyeztek az adatbázisból kinyert adatok a POST-olt adatokkal, akkor az adminisztrátor adatait elmenthettük egy munkamenetbe és a *header()* függvény segítségével továbbléptünk az adminisztrátori felület főoldalára. Ha viszont rossz felhasználónevet vagy jelszót adtunk meg, akkor visszadobott minket a bejelentkezési oldalra és kiírta a hibaüzenetet.

Az admin adatok munkamenetbe mentésének egyik célja, hogy az adminisztrátori oldalon lévő munkálatok során ismert legyen a bejelentkezett adminisztrátor. A másik célja pedig egy védelem, hogy különböző ügyeskedések által se tudjon belépni az egyszerű felhasználó az adminisztrátori felületre. Például ha az URL-be beírná a „fooldal.php” karakterláncot, akkor megkerülhetné a bejelentkezési lapot és használhatná az adminisztrátori felületünket. Ezért mindegyik PHP fájlunk elejére írtunk egy feltételt, ha a munkamenetbe nincs adminisztrátor, akkor dobjon vissza minket a bejelentkezési lapra.

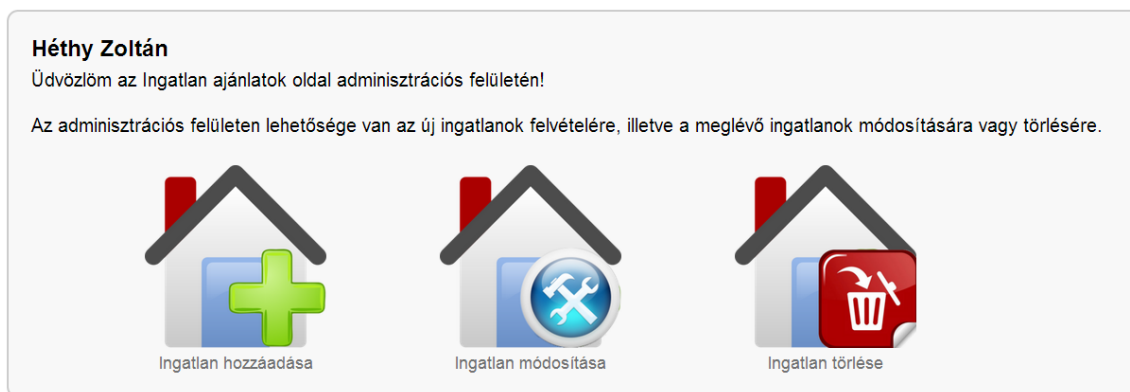


16. ábra: Bejelentkezés

Forrás: saját szerkesztés

4.2.2. Adminisztrátori főoldal

Bejelentkezés után ez az oldal jelenik meg az adminisztrátoroknak. Az adminisztrátori felület főoldala (17. ábra) mindössze annyit tud, hogy név szerint üdvözli a bejelentkezett adminisztrátort, és az adminisztrátorok látványos ikonok segítségével érhetik el a különböző adminisztrátori feladatokat. Bár ezek a feladatok az adminisztrátori felület navigációs menüjével is elérhetőek. Ez az oldal a *fooldal.php* nevű fájlban lett megírva.



17. ábra: Adminisztrátori főoldal

Forrás: saját szerkesztés

4.2.3. Ingatlan felvétele

A weboldal üzemeltetőinek, vagyis az adminisztrátoroknak három fontos feladatuk van az oldal karbantartása érdekében. Egy adminisztrátornak tudnia kell új ingatlan ajánlatokat feltöltenie, vagy a meglévő ingatlan ajánlatokat módosítani vagy törölnie.

Az ingatlanok felvétele (18. ábra) esetén egy darab ingatlan feltöltésére van lehetőségünk, amelynél meg kell adnunk az ingatlan összes adatát és legalább egy képet mellékelnünk kell mellé. Legfeljebb nyolc képet tölthetünk fel. Továbbá a képek csak JPEG, JPG és PNG formátumú fájlok lehetnek és a maximális fájl méretük se lehet több 2MB-nál. Képfeltöltés közben azonnal láthatjuk, hogy milyen képek lettek kiválasztva a feltöltéshez. Az ingatlanok adatainak a feltöltésénél meg kell adni az ingatlan szándékát, típusát, jellegét, szobáinak számát, komfort fokozatát, építkezési évét, alapterületét, telekterületét, felszereltségeit, milyen településen és milyen megyében található, egy részletes leírást, az árát és a feltöltés dátumát.

A rendszer tartalmaz egy hibakezelőt is, ami azért felelős, hogy ha hibásan vagy éppen nincs kitöltve az összes adatmező és nincs megadva legalább egy a feltételeknek eleget tevő kép, akkor nem engedi feltölteni az ingatlant és visszadob az ingatlan feltöltése űrlaphoz. Továbbá tartalmaz űrlapvisszatöltést is, ami az adminisztrátor által már egyszer kitöltött mezők értékeit visszatölti. A hibás mezők mellé pedig egy hibát jelző szimbólumot rak, aminek a jelentése az űrlap alján magyarázódik aztán el.

Az ingatlanok felvételéhez egy nagyobb méretű űrlapot használtunk, ami hét darab szövegbeviteli mezőt, három legördülő menüt, egy darab kételemű rádiógombot, egy jelölőnégyzetet, még egy hételemű jelölőnégyzetet, egy hosszabb szövegmezőt és egy dátum mezőt tartalmazott. Továbbá nyolc darab fájlfeltöltő mezőt a képek számára.

A szövegbeviteli mezők esetében, ahol egész számok megadására volt szükség, ott egy „num_formatting” nevű javascript programrészt használtunk, hogy számokon kívül ne fogadjon el más típusú bemenetet. A feltöltés gombbal küldhettük tovább ezeknek a beviteli mezőknek az értékeit. Az adatok továbbküldéséhez POST metódust használtunk. A képek továbbküldése is POST metódussal történt, de ahhoz, hogy képeket tudjunk küldeni, a *form*-nak meg kellett adni egy *enctype="multipart/form-data"* paramétert. Az ingatlan adatai az *insert.php* fájlra küldődtek tovább, ahol az adatok a *\$_POST* tömbbe, a képek pedig a *\$_FILES* tömbbe kerültek. Ezek aztán különböző ellenőrzéseken estek át, hogy minden adat korrekt legyen az ingatlan feltöltéséhez. A helyesen kitöltött adatmezőket mentettük munkamenetbe is, arra az esetre, ha a feltöltés nem valósulna meg, vissza tudjuk írni az adatokat az ingatlan felvételénél lévő űrlap mezőkbe. Ha viszont az átküldött űrlap összes mezője helyesen volt kitöltve, akkor elindíthattuk az adatok feltöltését az adatbázisba. Először az *itelepules* táblába töltöttük fel a megadott települést, ha még nem tartalmazott a táblánk egy ugyanolyan települést. Majd az *ikinalat* táblába, az *if_kapcsolat* táblába, az *ikepek* táblába, és ha kiemelt ingatlanról volt szó, akkor a *kiemelt* táblába is beszúrtuk a feltöltendő ingatlan adatait. A beszúráshoz az *INSERT INTO SQL* utasítást használtuk, amibe megadtuk, hogy a tábla melyik mezőjébe milyen értéket szúrjon be. Ezen értékek adatbázisba való beszúrása után már csak a képfeltöltés volt hátra.

Ingatlan felvétel

Szándék <input checked="" type="radio"/> eladó <input type="radio"/> kiadó <input type="checkbox"/> kiemelt		Típus <input type="text" value="Ház"/>
Település <input type="text"/>	Megye <input type="text"/>	
Jelleg <input type="text" value="Téglaház"/>	Szoba <input type="text"/>	
Komfort <input type="text" value="Összkomfort"/>	Építkezési év <input type="text"/>	
Alapterület <input type="text"/> m ²	Telekterület <input type="text"/> m ²	
Felszereltség <input type="checkbox"/> víz <input type="checkbox"/> gáz <input type="checkbox"/> villany <input type="checkbox"/> központi fűtés <input type="checkbox"/> pince <input type="checkbox"/> csatorna <input type="checkbox"/> garázs		
Leírás <input type="text"/>		
Ár <input type="text"/> Ft	Dátum <input type="text" value="----. --. --."/>	

Az ingatlan képei

Legalább egy kép feltöltése kötelező!
Támogatott fájlformátumok: JPEG JPG PNG
Maximális fájlméret: 2MB

1. kép

Nincs kiválasztott fájl

2. kép

Nincs kiválasztott fájl

3. kép

Nincs kiválasztott fájl

4. kép

Nincs kiválasztott fájl

5. kép

Nincs kiválasztott fájl

6. kép

Nincs kiválasztott fájl

7. kép

Nincs kiválasztott fájl

8. kép

Nincs kiválasztott fájl

NO IMAGE AVAILABLE

NO IMAGE AVAILABLE

NO IMAGE AVAILABLE

NO IMAGE AVAILABLE

18. ábra: Ingatlan felvétel

Forrás: saját szerkesztés

4.2.3.1. Képek feltöltése

A POST metódussal átküldött fájlok a `$_FILES` asszociatív tömbbe kerültek. Ebben a tömbben egy fájlhoz összesen öt adat tartozik, amik a következők:

- *name*: a fájl eredeti neve
- *type*: a fájl típusa
- *tmp_name*: a fájl jelenlegi elérési útja a szerveren
- *error*: egy hibakód (ha nulla, akkor nincs hiba)
- *size*: a fájl mérete (Hörb, 2016e)

Az átküldött képek vizsgálatánál ezek közül az adatok közül szükséges volt az *error* és *type* értékeinek ellenőrzésére. Ha valamelyik képnél az *error* értéke nem 0 volt, vagy a *type* értéke nem *image/jpeg* vagy *image/png*, akkor az a kép nem felelt meg a feltételeknek és nem került feltöltésre. Amelyik kép viszont eleget tett a feltöltési követelményeknek, azt átméreteztük és a *képek* mappán belül, az ingatlan szándékától és típusától függően egy mappába lett átmásolva. A képkezelési műveletekhez a PHP kiterjesztésű GD programkönyvtár függvényeit használtuk.

Kódrészlet az ingatlan feltöltő fájl képkezeléséből

```
//GD képkezelés
// fájl elérési útja
$filename = $kep["tmp_name"];
// eredeti kép méretei
$kepmeret = getimagesize( $kep["tmp_name"]);
$kep_w = $kepmeret[0];
$kep_h = $kepmeret[1];
// új kép méretei
if ($kep_w >= $kep_h) {
    $w = 700;
    $h = 464;
}
else {
    $w = 338;
    $h = 510;
}
// egy üres kép létrehozása
$newimage = imagecreatetruecolor($w, $h);
// eredeti kép beolvasása
switch ($ext) {
    case "jpg":
    case "jpeg": $oldimage = imagecreatefromjpeg($filename);
break;
    case "png": $oldimage = imagecreatefrompng($filename); break;
}
```

```
// eredeti kép rámásolása újra, átméretezve
imagecopyresampled($newimage, $oldimage, 0, 0, 0, 0, $w, $h, $kep_w,
$kep_h);
// kép mentése
imagejpeg($newimage, $target.$kov_id."_".$i.".jpg", 100);
```

A *getimagesize()* függvénnyel lekértük az eredeti kép méreteit és megvizsgáltuk, hogy álló vagy fekvő képről van-e szó. Fekvő kép esetén a képet 700x464 méretűre, álló kép esetén pedig 338x510 méretűre változtattuk. Ehhez az *imagecreatetruecolor()* függvénnyel egy új, üres képet hoztunk létre, amire rámásoltuk az eredeti képet. Az eredeti képet az *imagecreatefromjpeg()* és az *imagecreatefrompng()* függvényekkel tudtuk be olvasni. Az átméretezett másolást pedig az *imagecopyresampled()* függvénnyel valósítottuk meg. Ezután már csak ki kellett mentenünk az új átméretezett képünket a megfelelő mappába, ehhez pedig az *imagejpeg()* függvényt használtuk.

4.2.4. Ingatlan módosítása

A feltöltött ingatlanok adatainak módosítására is lehetőségünk van az adminisztrációs felületen. Mivel előfordulhat, hogy az adatbázisban található ingatlanok adatai közben megváltoztak, vagy hibásan lettek feltöltve. Az ingatlan módosítása esetén, a weboldalon kilistázásra kerülnek az adatbázisban szereplő ingatlanok (19. ábra). Az oldalon egyszerre csak korlátozott számú ingatlan listázódhat ki, ez jelen esetünkben tizennégy darabot jelent. Ha ennél több ingatlanunk van az adatbázisban, akkor egy lapozó segítségével több oldalba törjük az ingatlan lista megjelenítését. A listában szereplő ingatlanok azonosítására nem csak az ingatlan azonosítója szolgál, hanem megtalálható még az ingatlan szándéka, típusa, települése és a feltöltés ideje is. Továbbá minden kilistázott ingatlan mellett van egy *Módosítás* felirat, amivel módosíthatjuk a kiválasztott ingatlant.

A *Módosítás* feliratra ha rákattintottunk, akkor a *modositas.php* oldalról átdobott minket a *modositas2.php* oldalra, ahol egy hasonló űrlap jelent meg, mint az ingatlan feltöltése esetében. Az űrlap már kitöltve jelent meg a kiválasztott ingatlan adataival és képeivel. Ehhez a POST módszerrel átküldött ingatlan id-je alapján lekértük az ingatlan adatait, amiket aztán egyesével beírtunk az űrlap adatmezőibe. Az azonosítón kívül az ingatlan összes adatát módosíthattuk. A feltöltött képek módosítására is lehetőségünk volt. Helyére új képeket tölthettünk fel, vagy törölhettük is azokat. A törléshez nyolc darab jelölőnégyzetet használtunk. Ezek közül az első soha sem lehetett bepipálni, mivel egy képet kötelezően tartalmaznia kellett az ingatlannak, így nem törölhettük az első képet.

Miután megadtuk a változtatni kívánt adatainkat, a módosítás gombbal elküldhettük ezeket az *update.php* fájlunk. Itt az ellenőrzések után *UPDATE*, *DELETE* és *INSERT INTO* SQL utasítások segítségével módosíthattuk az adatbázisunk tábláit. Az ingatlan képeinek módosítása összetettebb feladat volt, mivel a képeket vagy töröltük, vagy módosítottuk, vagy nem csináltunk velük semmit. Ehhez egy összetett feltételvizsgálatot alakítottunk ki, amelyben SQL parancsokat és fájlkezelő utasításokat is egyaránt végrehajtottunk. Képek áthelyezéséhez és felülírásához a *rename()* függvényt, a képek törléséhez az *unlink()* függvényt használtuk. Az új képek hozzáadása pedig itt is a GD képkezelő függvények segítségével történt.

Ingatlan módosítás					
Ingatlan ID	Szándék	Típus	Település	Dátum	
1	Eladó	Családi ház	Komárom	2016.04.02	Módosítás
2	Eladó	Családi ház	Fertőszentmiklós	2014.12.22	Módosítás
3	Eladó	Családi ház	Petőháza	2015.10.21	Módosítás
4	Kiadó	Családi ház	Kazincbarcika	2016.02.25	Módosítás
5	Eladó	Családi ház	Edelény	2015.07.15	Módosítás
6	Eladó	Családi ház	Fertőszentmiklós	2016.04.10	Módosítás
7	Kiadó	Családi ház	Győrújfalú	2016.01.28	Módosítás
8	Eladó	Családi ház	Fertőszentmiklós	2015.09.30	Módosítás
9	Kiadó	Családi ház	Bánréve	2015.09.05	Módosítás
10	Eladó	Családi ház	Hajdúszoboszló	2014.08.20	Módosítás
11	Kiadó	Családi ház	Eger	2015.04.07	Módosítás
12	Eladó	Családi ház	Komárom	2015.05.27	Módosítás
13	Eladó	Családi ház	Gödöllő	2015.06.01	Módosítás
14	Eladó	Családi ház	Fertőszentmiklós	2015.04.13	Módosítás

1 2 3

19. ábra: Ingatlan módosítás

Forrás: saját szerkesztés

4.2.5. Ingatlan törlése

Ha már nincs szükségünk egy ingatlanra, akkor az adminisztrátornak lehetősége van törölnie azt az adatbázisból. Az ingatlan törlése nem túl bonyolult művelet. A törlés során egy ingatlan lista közül választhatjuk ki a felesleges ingatlanokat. Ha rákattintunk a *Törlés* feliratra, akkor a rendszer átugrik *torles.php* oldalról a *torles2.php* oldalra, ahol egy megerősítést kér az ingatlan törlésének végrehajtásához (20. ábra). Itt a rendszer megkérdezi tőlünk, hogy valóban törölni szeretnénk-e a kiválasztott ingatlant. Igen esetén törli az ingatlant, a nem esetén viszont visszadob a törlés első oldalára. Több ingatlan egyidejű törlése nem engedélyezett a rendszerben, ezzel is elkerülve a véletlen ingatlan törléseket.

A törléshez a *delete.php*-t használtuk feldolgozó fájlunk, amivel töröltük az adatbázisban szereplő ingatlanok adatait. Az adatbázisból való adatok eltávolítását *DELETE* SQL utasítással hajtottuk végre. A képek törlésénél pedig az *unlink()* függvényt használtuk.

Ingatlan törlés

Biztosan törölni szeretné a következő ingatlant?

Ingatlan ID	Szándék	Típus	Település	Dátum
2	Eladó	Családi ház	Fertőszentmiklós	2014.12.22

☒ igen ☐ nem

Törlés

20. ábra: Ingatlan törlés

Forrás: saját szerkesztés

4.2.6. Kijelentkezés

Az adminisztrátornak a munkáját befejezve ki kell jelentkeznie az adminisztrátori felületről, hogy egyéb felhasználó ne használhassa jogtalanul az ő nevében az adminisztrációs felületet. Amennyiben erről mégis megfélekedezne, a böngésző bezárása is elegendő, így is törlődik a munkamenetből bejelentkezett adminisztrátor.

A kijelentkezésre a navigációs menü alján lévő *Kijelentkezés* menüponttal van lehetőségünk. Ez a link a *logout.php* fájlt hívja meg, amivel törlődik a munkamenetből bejelentkezett adminisztrátor.

5. Tesztelés és továbbfejlesztés

5.1. Tesztelés és üzembe helyezés

A munkánk során a tesztelés folyamatosan történt a weboldal kódolása közben. Minden egyes funkció elkészítése során azonnal kipróbáltuk a weboldalunkat, hogy meggyőződjünk a megfelelő működéséről. Így a végső tesztelés során már alig találoztunk hibákkal. A gyors tesztelések érdekében egy otthoni gépre kialakított futtatókörnyezetet kellett létrehoznunk. A futtatókörnyezet kialakításának lépései a következők voltak:

1. Először egy webkiszolgálót kellett telepítenünk. Esetünkben az egyik legnépszerűbb webkiszolgálót, az Apache szervert választottuk. Népszerűségét az ingyenességnek és a megbízható gyors működésének köszönheti. Ennek a futtatása, a telepítés után bármikor felfüggeszthető, illetve elindítható volt.
2. A következő lépés a PHP szkriptnyelv konfigurálása és telepítés volt.
3. Továbbá telepítenünk és konfigurálnunk kellett az adatok tárolásához egy adatbázisrendszert is, ami esetünkben a MySQL adatbázisrendszer volt.

Az imént felsorolt lépéseket nagyon egyszerűen megvalósíthattuk az EasyPHP programcsomag telepítésével. Ennek a telepítése után megtalálható volt a gépünkön az Apache webszerver, a PHP szkriptnyelv és a MySQL adatbázisrendszer. Így ezeket a komponenseket nem kellett egyesével feltelepítenünk. A telepítések után, ha elindítottuk az EasyPHP-t, akkor a böngészőnkben localhoston keresztül már könnyedén futtathattuk a weboldalunkat. A teszteléseket négy féle böngészőn is végrehajtottuk. Ezek név szerint a Chrome, a Firefox, az Edge és a Maxthon böngészők voltak. A kisebb hibákat leszámítva a Firefoxon, mindegyiken szépen működött a weboldal. Továbbá több képernyőfelbontáson is remekül működött az oldalunk és a különböző oldal méretezésekre is jól reagált.

Azonban az otthoni gépen felállított futtatható környezet nem mindig nyújt pontos és megfelelő tesztelési lehetőségeket. A hibák javítására és a kezdetleges tesztelésekhez tökéletesen megfelelt, azonban a végső munkánk megvalósításához egy éles webkiszolgáló segítségével is le kellett tesztelnünk. A rendszerünk átköltöztetéséhez a következő lépésekre volt szükségünk:

1. Először meg kellett győződnünk arról, hogy az általunk választott webkiszolgáló támogatja a PHP és a MySQL nyelveket.

2. Azután a következő feladatunk volt a PHP fájlok feltöltése a webszerverre. Itt nagyon fontos volt, hogy a PHP fájlok ugyanúgy helyezkedjenek el egymáshoz viszonyítva, ahogy a saját gépünkön.
3. Továbbá a MySQL-ben található adatbázisunkat és tábláinkat kiexportáltuk és feltöltöttük a webkiszolgáló adatbázisrendszerébe.
4. Utolsó lépésként már csak annyi dolgunk volt, hogy a *setting.php* fájlunkban átírjuk az adatbázishoz való csatlakozási parancsot, hogy ne a saját gépünkön használt adatbázisrendszerre csatlakozzon.

A weboldalunk megosztására egy ingyenes, könnyen kezelhető és jól működő webkiszolgálót választottunk, amelyet az alábbi címen érhattünk el:

<http://cpanel.serversfree.com/>

A feltöltés után pedig a weboldalunk az alábbi címen vált elérhetővé:

<http://ingatlanajanlatok.bugs3.com/>

5.2. Továbbfejlesztési lehetőségek

Az ingatlanos rendszerünkben szépen működik minden, amit a munkánk elején a célunknak kitűztünk. A weboldalunk kinézete rendkívül dizájnosra sikeredett, a legtöbb böngészőn szépen működik és több képernyőfelbontáson is jól használható, sőt az oldal méretezést is elég jól tűri. Azonban még így is van lehetőségünk az oldal továbbfejlesztésére, hogy még felhasználóbarát környezetet biztosítsunk a weboldal látogatóinak és üzemeltetőinek, illetve a rendszer alkalmasságát és védelmét is tovább fokozzuk. Ezek közé a fejlesztések közé sorolnám egy reszponzív design kialakítását, részletes keresői oldal kialakítását, levelező-rendszer beépítését, saját ingatlan feltöltését a regisztrált felhasználók számára, új adminisztrátorok hozzáadását, és a védelem szempontjából egy SQL befecskendezések elleni védelmet.

5.2.1. Reszponzív webdesing

Mivel az informatika rohamosan fejlődik, ezért manapság már rengeteg eszköz van, amely webes böngészésre alkalmas. Ezért érdemes olyan weboldalt készíteni, amely minden eszköz kijelzőjén jól áttekinthető módon jelenik meg. Erre a megoldás nyújt a reszponzív webdesign (21. ábra) kialakítása. A lényege, hogy a különböző méretű és felbontású kijelzőkhöz eltérő stílus szabályokat alkalmazunk, de ugyanazt a tartalmat jelenítjük

meg minden eszközön (Primahonlap.hu, 2016). Megvalósításával a weboldalunk bármely eszközön szépen megjeleníthető lenne, ami rendkívül kifizetődő volna a későbbiekben.



21. ábra: Reszponzív design

Forrás: <http://primahonlap.hu/mi-is-az-a-reszponziv-responsive-webdesign>

5.2.2. Részletes keresői oldal

A felhasználói felületen létrehozásra kerülne egy külön oldal, amely az ingatlanok keresésénél az adatok részletes bevitelére adna lehetőséget. Keresésben meg lehetne adni az ingatlanokat jellemző összes adatot. Ezáltal a felhasználó rengeteg ingatlan esetén is nagymértékben leszűkíthetné a keresés eredményeit a saját igényeinek megfelelően.

5.2.3. Levelező-rendszer

Amennyiben egy ingatlan megtetszene a felhasználónak, és elektronikus úton szeretné felvenni a kapcsolatot a weboldal üzemeltetőivel, akkor nem kellene külső oldalak levelező-rendszerét használni, hanem a weboldalról is küldhetnének email-eket. Minden adminisztrátornak saját ingatlanos email címe volna. Az adminisztrátorok a nekik címzett üzeneteket az adminisztrátori felületen az üzenetek menüpont alatt tekinthetnék meg.

5.2.4. Saját ingatlan feltöltése

A regisztrált felhasználóknak lehetőségük volna saját ingatlan feltöltésére, ezzel megkönnyítve a saját és az adminisztrátorok dolgát is. A regisztrációnak és az ingatlan feltöltésnek is külön oldalt hoznánk létre a felhasználói felületen. A regisztrációs oldal bárki

számára elérhető volna, de a feltöltő oldalra már csak a bejelentkezett felhasználók léphetnének be.

5.2.5. Adminisztrátorok hozzáadása

Abban az esetben, ha új munkatárs érkezne az ingatlan irodához, akkor az adminisztrátoroknak lehetőségük volna az új munkatársnak is adminisztrátori jogokat adni. Ehhez az adminisztrációs felületen hoznánk létre egy külön oldalt, amin az új munkatárs adatait lehetne feltölteni. A felhasználónevet és a jelszót is saját maga választhatná meg, amivel a későbbiekben be tudna jelentkezni az adminisztrációs felületre.

5.2.6. SQL injection

A weboldal biztonságának megerősítéséhez egy SQL befecskendezések elleni védelmet alkalmaznánk. Ezt azoknál a mezőknél használnánk, ahol tetszőleges adatokat lehet bevinni, mivel a támadó ezen mezők használatával könnyedén manipulálhatja az oldal működését. Ezzel a védelemmel meggátolnánk, hogy a támadó nemkívánatos SQL-lekérdezéseket módosító parancsokat futtasson a rendszerben. Továbbá lehetőséget nyújtana a veszélyes karakterek ellenőrzésére is.

Befejezés

A munkánk elkészítésével sikeresen létrehoztunk az ingatlanok közvetítésére alkalmas weboldalt, amit aztán a későbbiekben akár hasznosítani is lehet kisebb ingatlan irodáknál. A weboldal alapkövetelményei közé tartozott, hogy tartalmazzon eladó és kiadó ingatlanokat, az ingatlanok között lehessen keresni és rendezni, és egy adminisztrációs felület létrehozása is, amelyen keresztül az adminisztrációs feladatokat hajthattuk végre. Ezeket a követelményeket mind sikerült megvalósítanunk és az elképzeléseinknek megfelelő munkát hoztunk létre.

A szakdolgozat írása előtt már rendelkezünk a szükséges alapokkal, mivel az egyetemi tanulmányaink során foglalkoztunk a webes rendszerek fejlesztésével és az adatbázis-kezeléssel is. Ezek az ismeretek a munka folyamán nem mindig bizonyultak elégségesnek, ezért folytonos önfejlesztést is igényelt a munka elkészítése. Ehhez különböző szakirodalmak és online források vizsgálatára volt szükségünk. Ezáltal programozási ismeretünket megfelelő képen kibővítettük a gyakorlati rész megvalósításához.

A munkánk legelején létrehoztuk HTML és CSS nyelvek segítségével a weboldal felületét, amely nagyon dizájnosra sikeredett. Sok problémát nem okozott ennek a megvalósítása. Egyedüli problémát az jelentette, hogy nagyon lassan haladtunk vele, mivel sok időt szántunk a stílusos kinézet kialakítására. Ezután az adatbázis tervezése és megvalósítása következett. Az adatbázis tervezésénél kisebb elakadásokba ütköztünk az adatbázis optimalizálásánál, amit tanáraink személyes konzultálásával orvosoltunk. Az adatbázis létrehozása nem vett nagyon sok időt el, de az ingatlanok adataival való feltöltése rengeteg időt vett igénybe. Az adatbázis megvalósítása után következett a munka legnehezebb része, az oldal dinamikussá alakítása. Ebben a részben a legfontosabb feladatokat sikerült megvalósítanunk, de akadtak különböző kisebb dolgok, amiket a JavaScript nyelv hiányosságainknak köszönhetően nem tudtunk megvalósítani. Nem sikerült a keresőnk beviteli mezőinek egy olyan tulajdonság beprogramozása, amely a beírt karakterek alapján kilistázza az adatbázisban szereplő értékeket. Próbálkoztunk a megvalósításával, de a felmerülő hibák miatt, inkább kihagyásra került ez a dolog. Továbbá az ingatlan feltöltésénél és módosításánál nem tudtuk megoldani az űrlap visszatöltésénél a képek visszatöltését a bementi fájl mezőkbe, ezért ezeket a képeket manuálisan kell visszatöltögetnünk. Problémánk volt még a keresés és a rendezés egyidejű használatával is, de ezt egy munkamenet-kezeléssel megoldottuk. Sok gondunk volt még az ingatlanok módosításának megvalósításával is, de különböző teszteléseknek köszönhetően végül ezt is

sikerült a megfelelőképpen kivitelezni. A többi rész elkészítésével nem volt különösebben sok problémánk, leszámítva azt, hogy mindegyik feladat megvalósítása nagyon időigényes volt. A fejlesztések után a weboldal üzembe helyezése is gond nélkül ment. Ehhez egy ingyenes webkiszolgálót használtunk. Az oldalunkat a tesztelések során kipróbáltuk különböző böngészőkön és képernyőfelbontásokon is.

A szakdolgozatunk befejeztével úgy gondoljuk, hogy elegendő tudást és tapasztalatot szereztünk a dinamikus weboldalak létrehozásához, amit a későbbi életünk során remekül hasznosíthatunk.

Resumé

Internet je v súčasnosti neoddeliteľnou súčasťou nášho života. Pomocou neho sme schopní prostredníctvom webových stránok získať alebo sprostredkovať rôzne informácie. Vďaka tomu sa internet stal prvotným informačným nástrojom súčasnosti. Možnosti internetu sú výhodné aj pre realitné kancelárie, ktoré ho naplno využívajú na inzerovanie online formou.

Cieľom našej bakalárskej práce bolo vytvorenie webovej stránky zameranej na inzerciu predávaných a prenajímaných nehnuteľností. Na webovej stránke majú návštevníci možnosť prezerať si nehnuteľnosti nachádzajúce sa v databáze. Taktiež ich môžu vyhľadávať, filtrovať a kategorizovať podľa určitých kritérií. Internetová stránka obsahuje aj administratívne rozhranie, prostredníctvom ktorého môžu prevádzkovatelia stránky pridávať, upravovať a vymazávať ponuky nehnuteľností. Tento systém bol vytvorený pomocou technológie PHP a MySQL.

V úvode práce sme sa oboznámili so základnými charakteristikami webových stránok a prezentovali sme najdôležitejšie informácie potrebné k ich vytvoreniu. Stručne sme spomenuli spojitosti internetu a webových stránok, ako aj ich typy. Obširnejšie sme sa zamerali na statické a dynamické webové stránky. V závere kapitoly sme čitateľovi vymenovali, aké sú základné požiadavky dobre navrhutej webovej stránky.

V nasledujúcej kapitole sme prezentovali nástroje, resp. programovacie jazyky, ktoré sme použili na vytvorenie webovej stránky. Definovali sme značkový jazyk HTML, ktorý je používaný na vytváranie webových stránok a spomenuli sme jeho najnovšiu verziu HTML5. Taktiež sme sa zamerali na štýlový jazyk CSS, ktorý je zodpovedný za výzor stránky, niekoľko informácií sme poskytli aj o programovacom jazyku PHP, a v rámci neho aj o jeho verzii PHP5. V záujme dosiahnutia dynamickosti webovej stránky sme ako základ použili programovací jazyk PHP a databázový systém MySQL. Spomenuli sme programovací jazyk JavaScript zodpovedný za programovanie používateľského rozhrania, ako aj najčastejšie používaný framework jQuery. Následne sme veľmi stručne opísali použité nástroje, medzi ktoré patrili Notepad++ na editovanie zdrojového kódu a balík programov EasyPHP, ktorý obsahoval okrem iného PHP, databázu a webový server Apache.

V praktickej časti práce sme sa najprv zamerali na návrh, resp. vytvorenie webového rozhrania a databázy. Stránku sme sa snažili navrhnuť tak, aby zodpovedala hlavným cieľom práce. Najprv sme sa zaoberali návrhom rozhrania, v rámci čoho sme vytvorili základné rozhranie pre administrátora a používateľa. K týmto rozhraniám sme navrhli rôzne

navigačné menu, k jednému vertikálne, k druhému horizontálne. Následne sme pomocou HTML a CSS zrealizovali navrhnuté rozhrania webovej stránky.

Po vytvorení vzhľadu stránky sme sa v ďalšom kroku zamerali na databázu. Najprv sme zhromaždili údaje, ktoré mali v databáze figurovať, potom sme na základe nich navrhli entitno-relačný diagram databázy a pomocou neho sme vytvorili relačnú databázu. V závere vytvárania databázy sme použili rôzne metódy normalizácie databáz, čím sme ju dostali do 3NF, to znamená, že štruktúra databázy bude efektívna a prehľadná. Po ukončení navrhovania databázy sme opísali samotný postup jej realizácie. Databázu sme vytvorili pomocou nástroja na vytváranie databáz phpMyAdmin. Tento nástroj je vytvorený pomocou PHP jazyka a umožňuje ovládanie MySQL cez internet prostredníctvom grafického zohrania. Následne sme vytvorili databázu, tabuľky databázy a nahrali sme údaje. Konečnú štruktúru databázy sme zobrazili pomocou obrázka a prezentovali sme aj tabuľky databázy spolu s poľami tabuliek. Na uskladnenie údajov sme celkovo použili 11 tabuliek.

V ďalšej kapitole sme predstavili náš systém predaja nehnuteľností. V osobitných podkapitolách sme opísali jednotlivé časti našej webovej stránky, charakterizovali sme ich úlohy a stručne načrtli postup realizácie.

Charakteristiku sme začali úvodnou stránkou a podstránkami používateľského rozhrania. Na úvodnej stránke našej webovej stránky sa nachádza jedna časť pre zvýraznené ponuky a časť pre zobrazenie najnovších ponúk nehnuteľností. Na vytvorenie ponúk nehnuteľností bolo nutné, aby sme z databázy vedeli stiahnuť údaje nehnuteľností, na tento účel sme použili databázové funkcie MySQL používané v PHP v kombinácii s SQL príkazmi. V záujme jednoduchšieho používania sme vytvorili „obal,“ pomocou ktorého sme najčastejšie postupy spravovania databáz skrátili na jeden riadok. Tieto funkcie sme predstavili pomocou tabuľky. Používateľské rozhranie obsahovalo osobitné stránky pre predávané a prenajímané nehnuteľnosti. Návštevníci stránky si ponuky nehnuteľností mohli prezerať na týchto stránkach. Vedeli vyhľadávať ponuky podľa svojich predstáv, mali možnosť zoradiť ich podľa dátumu alebo ceny. V mnohých prípadoch nebolo možné všetky nehnuteľnosti zobraziť na jednej stránke, zobrazenie ďalších záznamov sme vyriešili pomocou listovania medzi stránkami. Tieto funkcie sme osobitne definovali a podrobne opísali postup ich fungovania. Počas opisu funkcií sme sa oboznámili aj s reláciami, ktorých používanie bolo nevyhnutné v záujme správneho fungovania vyhľadávania, kategorizovania či listovania stránok. Úlohou relácie bolo, aby si pri listovaní medzi nehnuteľnosťami zapamätala kritériá vyhľadávania a kategorizovania, a umožnila ich používanie súčasne. Používateľ mal možnosť zobraziť si aj podrobné údaje a obrázky zvolenej nehnuteľnosti,

vd'aka čomu získal detailnejšie informácie o danej nehnuteľnosti. Na tejto stránke boli údaje zobrazené vo forme tabuľky a obsahovala aj podrobný opis konkrétnej nehnuteľnosti. Na vhodné zobrazenie obrázkov sme použili galériu obrázkov. Ak mal používateľ záujem o nejakú nehnuteľnosť alebo chcel využiť služby ponúkané prostredníctvom webovej stránky, musel sa skontaktovať s administrátormi stránky. Z tohto dôvodu sme vytvorili stránku s kontaktmi, ktorá obsahovala kontaktné údaje spolupracovníkov.

Zamerali sme sa aj na detailné predstavenie administračného rozhrania. Táto stránka bola dostupná len pre administrátorov, preto bolo nutné vytvoriť aj prihlasovaciu stránku. Pri prihlasovaní nás stránka presmerovala na hlavnú stránku administratívneho rozhrania iba po zadaní správneho používateľského mena a hesla. Z hlavnej stránky boli dostupné rôzne úlohy administrátora, napr. pridávanie, upravovanie a mazanie nehnuteľností. Jednotlivé úlohy boli dostupné zo samostatných stránok. Na pridávanie nehnuteľností sme vytvorili stránku, ktorá obsahovala rozsiahlejší formulár na nahratie údajov nehnuteľností. Popri nahrávaní údajov sme mali možnosť aj na nahrávanie obrázkov, ktoré bolo podmienené rôznymi kritériami. Nahratie aspoň jedného obrázka bolo povinné. Formulár obsahoval aj funkciu chybového hlásenia a funkciu opätovného načítania údajov. Taktiež sme spomenuli aj knižnicu s názvom GD, ktorá slúžila na nahrávanie obrázkov a obsahovala rôzne funkcie na manipuláciu s nimi. Tu sme vložili aj zdrojový kód, ktorý obsahoval príkazy na manipuláciu s obrázkami. Modifikácia údajov nehnuteľností fungovala na podobnom princípe ako pridávanie nehnuteľností. V tomto prípade bolo potrebné si zo zoznamu najprv vybrať konkrétnu nehnuteľnosť a následne sa údaje a obrázky nehnuteľnosti zobrazili v modifikačnom formulári. Ak sme tieto údaje upravili, tak sa modifikované údaje nehnuteľnosti zmenili aj v databáze. Mali sme možnosť aj na úpravu obrázkov alebo sme ich mohli vymazať, ak boli nepotrebné. V prípade chyby nás systém vrátil späť na formulár a zobrazil nám zadané údaje. Pri vymazávaní nehnuteľností sme si tiež museli vybrať konkrétnu nehnuteľnosť zo zoznamu. Následne nás systém vyzval na potvrdenie nášho úmyslu a daná nehnuteľnosť bola vymazaná len po tomto potvrdení.

V poslednej kapitole sme opísali testovanie nášho systému a jeho uvedenie do prevádzky. Tiež sme vysvetlili, aké boli požiadavky na realizáciu tohto testovania. Testovanie sme vykonávali postupne na našom domácom počítači s vytvoreným behovým prostredím. Záverečné testovanie bolo vykonané pomocou webového serveru dostupného na internete.

V tejto kapitole sme ešte spomenuli možnosti na ďalší rozvoj, ktoré by náš, už aj teraz dobre fungujúci systém, urobili ešte užitočnejším a praktickejším. Medzi tieto

rozvojové možnosti možno zaradiť responzívny dizajn, e-mailový systém, možnosť nahrávania vlastných nehnuteľností, pridanie nových administrátorov či zvýšenie ochrany a zabezpečenia.

Felhasznált irodalom

1. Könyvek és monográfiák

DEBOLT, V. *HTML és CSS Webszerkesztés stílusosan*. Debrecen : Kiskapu Kft., 2005. 480 o. ISBN 963 9301 96 5.

LOPEZ, D. – BLANCO J. *Apache zsebkönyv*. Budapest : Kiskapu Kft., 2007. o. 232. ISBN 9789639637320.

MELONI, C. J. – MORRISON, M. *Tanuljuk meg a HTML5 és CSS használatát 24 óra alatt*. Budapest : Kiskapu Kft., 2011. o. 446. ISBN 978 963 9637 78 8.

PILGRIM, M. *HTML 5 - Az új szabvány - Ugorjunk fejest a webfejlesztés jövőjébe!* Budapest : Kiskapu Kft., 2011. o. 278. ISBN 978 963 9637 77 1.

SÁGI, G. *Webes adatbázis-kezelés MySQL és PHP használatával*. Budapest : BBS-INFO Kft., 2005. o. 159. ISBN 9639425044

ZANDSTRA, M. *Tanuljuk meg a PHP5 használatát 24 óra alatt*. Budapest : Kiskapu Kft., 2005. o. 578. ISBN 9639301868.

2. Elektronikus dokumentumok és monográfiák

HARDER. *EasyPHP – otthoni webszerver*. [online]. 2005. [megtekintés dátuma: 2016.05.15.] Elérhető az interneten: <<http://www.tutorial.hu/easyphp-otthoni-webszerver/>>

HÖRB. *Beléptető rendszer*. [online]. 2016d. [megtekintés dátuma: 2016.05.24.] Elérhető az interneten: <<http://php.webprog.biz/phptanfolyam/urlapok/10-belepteto-rendszer>>

HÖRB. *Fájlfeltöltés*. [online]. 2016e. [megtekintés dátuma: 2016.05.26.] Elérhető az interneten: <<http://php.webprog.biz/phptanfolyam/urlapok/7-fajlfeltoltes>>

HÖRB. *JavaScript alapok*. [online]. 2016a. [megtekintés dátuma: 2016.05.14.] Elérhető az interneten: <<http://php.webprog.biz/phptanfolyam/urlapok/4-javascript-alapok>>

HÖRB. *jQuery gyorstalpaló*. [online]. 2016b. [megtekintés dátuma: 2016.05.14.] Elérhető az interneten: <<http://php.webprog.biz/phptanfolyam/jquery>>

HÖRB. *SQL függvények*. [online]. 2016c. [megtekintés dátuma: 2016.05.22.] Elérhető az interneten: <<http://php.webprog.biz/phptanfolyam/adatbaziskezeles/4-sql-fuggvenyek>>

K-DESIGN.HU. *Weblapok típusai*. [online]. [megtekintés dátuma: 2016.05.11.] 2016. Elérhető az interneten: <<http://www.k-design.hu/index2.php?pg=weblaptipusok>>

KOLOZÁR, G. *MySQL Server*. [online]. 2016. [megtekintés dátuma: 2016.05.13.] Elérhető az interneten: <<http://www.gadmin.hu/szolgaltatas/mysql/mysql.html>>

KVASZINGERNÉ, P. Cs. – NAGY, D. *Weblapfejlesztés*. [online]. Eger : Eszterházy Károly Főiskola, 2013. [megtekintés dátuma: 2016.05.12.] Elérhető az interneten: <http://people.ektf.hu/kvcsilla/sites/all/malaj_weblapfejlesztes_konyv_kvcsilla_nagydenes_2014.pdf>

MUHA, L. *Az informatikai biztonság kézikönyve*. [online]. Budapest : Verlag Dashöfer Szakkiadó, 2005. [megtekintés dátuma: 2016.05.10.] Elérhető az interneten: <https://books.google.hu/books?id=6HSIIJyOrcC&printsec=frontcover&hl=hu&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false>.

PATAI, L. *Notepad++ egy ingyenes forráskód-szerkesztő*. [online]. 2009. [megtekintés dátuma: 2016.05.15.] Elérhető az interneten: <<http://blog.wphu.org/2009/05/22/notepad-egy-ingyenes-forraskod-szerkeszto/>>

PRIMAHONLAP.HU. *Reszponzív webdesign*. [online]. 2016. [megtekintés dátuma: 2016.05.28.] Elérhető az interneten: <<http://primahonlap.hu/mi-is-az-a-reszponziv-responsive-webdesign>>

PROGRAMOZÁSALAPJAI.HU. *Programozás alapjai*. [online]. 2016. [megtekintés dátuma: 2016.05.12.] Elérhető az interneten: <<http://programozasalapjai.hu/programozas/php/a-php-programozasi-nyelv-bemutatas>>

RIMELEK. *PHP jelentősebb változásai napjainkig - Kezdetektől a trait-ekig*. [online]. 2011. [megtekintés dátuma: 2016.05.12.] Elérhető az interneten:
<<http://rimelek.hu/php-jelentosebb-valtozasai-napjainkig-kezdetektol-a-trait-ekig>>

SZINYEIGIMIBP.HU. *Adatbázis-kezelés*. [online]. 2011. [megtekintés dátuma: 2016.05.19.] Elérhető az interneten:
<<http://szinyeigimibp.hu/moodle/mod/page/view.php?id=316>>

VÁRI Cs. *HTML programozás az alapoktól*. [online]. 2007. [megtekintés dátuma: 2016.05.18.] Elérhető az interneten:
<<http://szatyika.hu/files/Vari-Csaba---HTML-programozas-az-alapoktol.pdf>>

3. Cikk elektronikus folyóiratokban és egyéb cikkek

SZÜTS, Z. *Az internetes kommunikáció története és elmélete*. [online]. Médiakutató, 2012/tavaszi [megtekintés dátuma: 2016.05.10.] Elérhető az interneten:
<http://www.mediakutato.hu/cikk/2012_01_tavaszi/01_internetes_kommunikacio_tortenete>. ISSN 1586 8389.

Mellékletek

A CD melléklet tartalma:

- Szakdolgozat elektronikus formátumban (.pdf)
- Forráskódok és képállományok
- Adatbázis szerkezete és tartalma (.sql)

Az ingatlan weboldal elérhetősége:

<http://ingatlanajanlatok.bugs3.com/>

Adminisztrációs felület esetén:

<http://ingatlanajanlatok.bugs3.com/admin/>

Az adminisztrációs felület belépési adatai:

- Felhasználónév: hethy7
- Jelszó: abc123456