



SOTER-LINE
OKTATÁSI KÖZPONT

Soter-Line Oktatási, Továbbképző és Szolgáltató Kft.

1073 Budapest, Erzsébet krt. 7. Tel: 887-2323; Fax: 887-2324, www.soterline.hu

Nyilvántartásba vételi szám: E-000201/2014

ZÁRÓDOLGOZAT

Gyűjtők Szentélye Web-áruház

Szoftverfejlesztő

OKJ száma: 54 213 05

Témavezető:

Dorogházi Donát

Készítette:

Rimóczy Kolos

2020

Tartalom

| | |
|---------------------------------------------------------------------------------------------------|-----------|
| 1. NYILATKOZAT A ZÁRÓ-DOLGOZAT KÉSZÍTÉSÉRE VONATKOZÓ SZABÁLYOK BETARTÁSÁRÓL ... | 3 |
| 2. BEVEZETŐ..... | 4 |
| 2.1. Téma:..... | 4 |
| 2.2. Specifikáció:..... | 4 |
| 2.3. Felhasznált technológiák:..... | 5 |
| 3. ADATSZERKEZETEK | 7 |
| 3.1. Adatbázisok | 7 |
| 3.2. A web-áruház adatbázis szerkezetének bemutatása egyed-reláció (ER) ábrázolással..... | 8 |
| 3.3. A web-áruház adatbázis szerkezetének bemutatása Crow's Foot ábrázolással..... | 9 |
| 3.4. Az adatbázis tábláinak attribútumai | 9 |
| 3.5. A webáruház adatbázisának tervezési szempontjai | 10 |
| 3.6. A táblák közötti kapcsolatok | 11 |
| 3.7. Az adattáblák és jellemzőik | 11 |
| 3.7.1. A 'felhasznalok' tábla fontosabb mezői..... | 11 |
| 3.7.2. A 'termeklap' tábla fontosabb mezői..... | 12 |
| 3.7.3. A 'vasarlas' tábla fontosabb mezői | 14 |
| 3.7.4. A 'kategoria' tábla fontosabb mezői | 15 |
| 3.7.5. A 'regnelkulfelhasznalok' (Regisztráció nélküli felhasználók) tábla fontosabb mezői: | 15 |
| 4. HASZNÁLATI MODELL ÉS ESET DIAGRAMOK | 17 |
| 4.1. Modell és elemzés | 17 |
| 4.2. Jogosultsági szintek..... | 17 |
| 4.3. Nem regisztrált vásárló jogai | 17 |
| 4.4. Regisztrált vásárló jogai..... | 18 |
| 4.5. Nem regisztrált és regisztrált felhasználók eset diagramjai | 18 |
| 4.6. Adminisztrátor jogai..... | 19 |
| 4.7. Az adminisztrátor eset diagramja: | 19 |
| 5. OSZTÁLYOK..... | 20 |
| 5.1. 'AdminNezet' osztály, UML diagram | 21 |
| 5.2. 'DbMuveletek' osztály, UML diagram | 22 |
| 5.3. 'Funkciok' osztály, UML diagram | 22 |
| 5.4. 'Weboldal' osztály, UML diagram..... | 23 |
| 5.5. 'FelhasznaloDbMuveletek' osztály, UML diagram | 24 |
| 6. IMPLEMENTÁLÁS | 25 |
| 6.1. Adatbázis csatlakozás, dbconnect.php | 25 |
| 6.2. AdminNezet osztály | 26 |
| 6.3. Dbmuveletek osztály | 27 |

| | | |
|-------|-------------------------------------------------------------------------------------------|----|
| 6.4. | <i>Funkciók osztály</i> | 30 |
| 6.5. | <i>Weboldal osztály</i> | 31 |
| 6.6. | <i>FelhasználóiDbMűveletek osztály</i> | 31 |
| 7. | FELHASZNÁLÓI FELÜLETEK | 32 |
| 7.1. | <i>Adminisztrátor felület</i> | 32 |
| 7.2. | <i>Felhasználói felület</i> | 35 |
| 8. | TESZTEZÉS | 37 |
| 8.1. | <i>Admin felület: Felhasználó Felvitel</i> | 37 |
| 8.2. | <i>Admin felület: Felhasználó Törlés/Módosítás</i> | 38 |
| 8.3. | <i>Admin felület: Termék Felvitel, Törlés/Módosítás</i> | 38 |
| 8.4. | <i>Admin felület: Kategória: Felvitel / Módosítás</i> | 39 |
| 8.5. | <i>Felhasználói felület: Regisztrál / Belépés</i> | 39 |
| 8.6. | <i>Felhasználó felület: Adatmódosítás</i> | 39 |
| 8.7. | <i>Felhasználó felület: Kosárba tétel / Kosárból törlés</i> | 40 |
| 8.8. | <i>Felhasználó felület: Megrendelés véglegesítése</i> | 40 |
| 8.9. | <i>Felhasználói felület: Keresés a keresőoldaltól, eladottak megjelenítése</i> | 40 |
| 8.10. | <i>Felhasználói felület: Bejelentkezés nélkül, keresés, eladottak megjelenítése</i> | 41 |
| 8.11. | <i>Felhasználói felület: Bejelentkezés nélkül, Admin felület elérése</i> | 41 |
| 8.12. | <i>Felhasználói felület: Bejelentkezés nélkül, Kosárba tétel / Kosárból kivétel</i> | 41 |
| 8.13. | <i>Felhasználói felület: Bejelentkezés nélküli megrendelés</i> | 41 |
| 9. | FELHASZNÁLÓI DOKUMENTÁCIÓ – A PROGRAM ISMERTETÉSE | 42 |
| 9.1. | <i>Felhasználói felület: Regisztrál / Adatmódosítás – Jelszóegyeztetés</i> | 42 |
| 9.2. | <i>Felhasználói felület: Keresés gomb</i> | 44 |
| 9.3. | <i>Felhasználói felület: Kezdőlap, Információk gomb, Kijelentkezés gomb</i> | 45 |
| 9.4. | <i>Felhasználói felület: Kosár gomb</i> | 45 |
| 9.5. | <i>Felhasználói felület: Képek megjelenítése</i> | 47 |
| 9.6. | <i>Admin felület</i> | 47 |
| 9.7. | <i>Admin felület: Felhasználó műveletek</i> | 48 |
| 9.8. | <i>Admin felület: Termék Műveletek</i> | 49 |
| 9.9. | <i>Admin felület Kategória Műveletek:</i> | 51 |
| 9.10. | <i>Admin felület Archiv műveletek</i> | 52 |
| 9.11. | <i>Admin felület: Eladások megtekintése</i> | 52 |
| 10. | BŐVÍTHETŐSÉG, JÖVŐBELI TERVEK | 53 |
| 11. | KÖSZÖNET | 54 |
| 12. | IRODALOMJEGYZÉK | 54 |

1. NYILATKOZAT A ZÁRÓ-DOLGOZAT KÉSZÍTÉSÉRE VONATKOZÓ SZABÁLYOK BETARTÁSÁRÓL

Alulírott Rimóczy Kolos, jelen nyilatkozat aláírásával kijelentem, hogy a **Gyűjtők Szentélye Web-áruház** című záró dolgozatot magam készítettem.

Mindezek alapján jelen záró-dolgozat önálló munkám, annak elkészítésekor betartottam a szerzői jogairól szóló 1999. LXXXVI törvény szabályait, valamint az intézmény által előírt, a dolgozat készítésére vonatkozó szabályait, különösen a hivatkozások és idézetek tekintetében.

Kijelentem, hogy az elektronikus és a papír alapú dokumentum mindenben megegyezik.

Fentiekén kívül kijelentem, hogy az önállóságra vonatkozóan, a dolgozat készítése közben konzulensemét és a feladatot kiadó oktatót nem tévesztettem meg.

Budapest, 2020.11.23

.....

Aláírás

2. BEVEZETŐ

2.1. Téma:

A web-áruházban, amelyet létrehozok, már-már egyedi termékek, gyűjtőknek szánt dolgok, relikviák lesznek találhatóak.

Ezen a területen ugyan számos más oldal kínál lehetőségeket értékesítésre, de az általam létrehozott áruházban csak ilyen termékekkel lesz érdemes foglalkozni.

Azért választottam ezt a témát, mert magam is ezzel foglalkozom és tudom, hogy még sokan mások is kereskednek ilyen jellegű tárgyakkal, viszont a piaci résztvevőknek kevés esetben van saját, jól működő, hatékony web-áruházuk.

A vásárlásra az oldal, mint közvetítő ad lehetőséget. A két üzletfélnek egymással kell majd az adásvétel folyamatát lebonyolítani. Az áruházban az eladásra kínált termékek ritkán lesznek teljesen egyformák, mint például egy ruházati web-shopban, ezért a web-áruház egyik fő célja, hogy a már eladott egyedi tárgyak (műtárgyak) adatai elérhetőek maradjanak egy archívumban is.

Ez azért hasznos a műtárgyak esetében, mert az olyan egyedi termékek, amelyek ritkán bukkannak fel a piacon, sok esetben nagy tapasztalattal rendelkező műtárgy becsüs által is nehezen felbecsülhetőek, és a róluk tárolt információk önmagukban is értékkel bírnak.

Ez a folyamatosan bővülő archív adatbázis lesz az egyik fő vonzereje a web-áruházamnak. Ez egy olyan dinamikusan növekvő szakmai információ-halmaz, ami miatt várhatóan egyre többen keresik fel az oldalt, adatokat fürkészve vagy akár csak az oldalon található termékek érdekessége miatt is.

2.2. Specifikáció:

Az áruházban a lehető legletisztultabb felületet alakítom ki, hogy ne vesszen el a vásárló a sok menü és választási lehetőség között. Az oldalon keretes szerkezetet hozok létre, fejléc, lábléc, tartalom elemekkel.

A letisztult és dizájnjos oldalképet a 'Bootstrap' keretrendszer segítségével is alakítom ki, ezzel teszem modernné és felhasználóbaráttá.

A weboldalon két felhasználói szint lesz: az adminisztrátor és a regisztrált vásárlók. Mindkét szintnek külön-külön megjelenítési felülete lesz.

A regisztrált vásárlói szintről csak a regisztráció, a saját adataik módosítása, és a vásárlás folyamata lesz elérhető.

A vásárlók a regisztráció után, vagy az úgynevezett "kosár" funkció használatával adhatják le megrendeléseiket. Utóbbi esetében nem szükséges a regisztráció. Ebben az esetben is szükséges lesz eltárolni az adatokat, hogy később visszakereshetők legyenek.

Az áruházban adminisztrátor joggal lehet majd eladásra kínált terméket létrehozni, azoknak tulajdonságokat adni, képeket feltölteni, árat meghatározni, és a felhasználók adatait szükségszerűen módosítani. A termékek egyedi volta miatt mindenképp bővíthetőek lesznek a termékkategóriák, melyet szintén erről a felhasználói

szintről lehet majd elérni. Ez az oldalkép egy egyszerűbb megjelenésű felület lesz, eltérően attól, amit a felhasználó/vevő lát.

Az eladandó termékekről minden lényegi információ adatbázisokban kerül tárolásra, 'MySQL' adatbázis kezelő rendszert használva. A különféle táblákon, rekordokon, mezőkön módosítási jogköre csak az adminisztrátor felhasználónak lesz. Ezt a felhasználók adattábla megfelelő mezőjében eltárolt adattal tároljuk majd.

Az eladott termékek adatai a termékek adattáblában lesznek, ezek a termékek a honlapról is megtekinthetők lesznek utólag, felhasználói szintről is, de csak amennyiben regisztrált felhasználó próbálja meg megnézni.

A termékeknél hangsúlyt kap a 'kulcsszavak' elmentése is, olyan fogalmaké, amelyek ezt a szakmát jellemzik, és ezeket a termékfeltöltésnél kötelezően kell majd megadni.

Az archív adatbázis folyamatos bővülése miatt az internetes keresők is előbbre helyezik majd az oldalt.

A felhasználók személyes adatait az adatvédelmi irányelvek betartásával, azokat figyelembe véve tárolom.

A technikai információkat, például a felhasználók jelszavait titkosítva helyezem el az adatbázisban.

A program kódolásánál elsődleges szempont az 'objektum-orientált' programozási alapelvek betartása és az úgynevezett 'clean', (tiszt) kód használata.

Az általam használt fejlesztői környezet a PHP lesz, az oldalak leírónyelve a HTML5, és a grafikus megjelenítésért a CSS3 leíró nyelvet veszem segítségül.

Későbbi terv lehet a bankkártyás fizetési mód, vagy más pénzforgalmi szolgáltatást nyújtó rendszerek (pl.: PayPal) beépítése az áruházba.

Felmerülhet még az is, hogy a már létrejött archív adatbázisban az érdeklődők számára később csak pénzügyi előfizetéssel lehessen az adott műtárgyak árait és tulajdonságait látni. Bővítési lehetőség még a hírlevél beépítése, hogy a felhasználók a promóciókról, akciókról email értesítést kaphassanak.

2.3. Felhasznált technológiák:

- **HTML:**

A HTML angol szavak kezdőbetűiből alkotott mozaikszó. Hyper Text Markup Language. Ez egy leíró nyelv melyet weboldalak készítéséhez használnak. A weboldalak alap szerkezetét lehet benne létrehozni. Ma a HTML5 a legelterjedtebb, én is ezt használtam.

- **CSS:**

A CSS (Cascading Style Sheets – Lépcsőzetes stíluslapok) egy stílusleíró nyelv, mely a HTML-ben alkotott weboldalak megjelenítését írja le. Itt szabható arculat a HTML oldalunknak. Elrendezések, dizájnok, színek, minden megjelenítéssel kapcsolatos tulajdonság e nyelv segítségével hozható létre.

- **MySQL:**

A MySQL egy relációs adatbázis kezelő szoftver (RDBMS : Relational database management system) mely az SQL lekérdező nyelvet használja. (Structured Query Language)

A rendszer lehetővé teszi az adatok tárolását, keresését, módosítását hatékony formában. Igen népszerű, ingyenes és a kitűzött feladat nem indokolta ennél összetettebb rendszer használatát.

A választásom azért esett erre, mert tanulmányaim során ezt ismertem meg.

- **PHP:**

A Hypertext Preprocessor (PHP) egy szerveroldali programozási nyelv. Segítségével dinamikus weboldalak lehet létrehozni. A PHP kód HTML oldalba ültethető, mely a kiszolgáló oldalán fut le és végezhet műveleteket az adatbázisokkal, fájlokat hozhat létre, képeket készíthet. Jelenleg a PHP 5-ös és 7-es verziója a leggyakrabban használatos.

- **Bootstrap:**

A 'Bootstrap' a legnépszerűbb ingyenes megjelenéssel kapcsolatos keretrendszer CSS és HTML alapokon. Használata könnyű, lehetővé teszi a reszponzív weboldalak készítését és a rácsszerkezetes használatot.

A 'Bootstrap' alpból mobilra megjelenő formában valósul meg. Több nagyobb böngészővel kompatibilis.

A keretrendszer fájljait a bootstrap.com oldalról lehet letölteni, vagy akár hivatkozni is lehet rá a kódból. Beépített 'JS' (JavaScript) komponenseket is tartalmazza. HTML5-öt használ alapként.

- **Xampp:**

A 'Xampp' egy sokoldalú web-szerver szoftvercsomag, mely ingyenes, nyílt forráskódú, és lehetővé teszi, hogy webes alkalmazásokat futtassunk rajta úgy, mintha a weben néznénk. Így könnyen készíthető és tesztelhető nem élesben egy weboldal.

3. ADATSZERKEZETEK

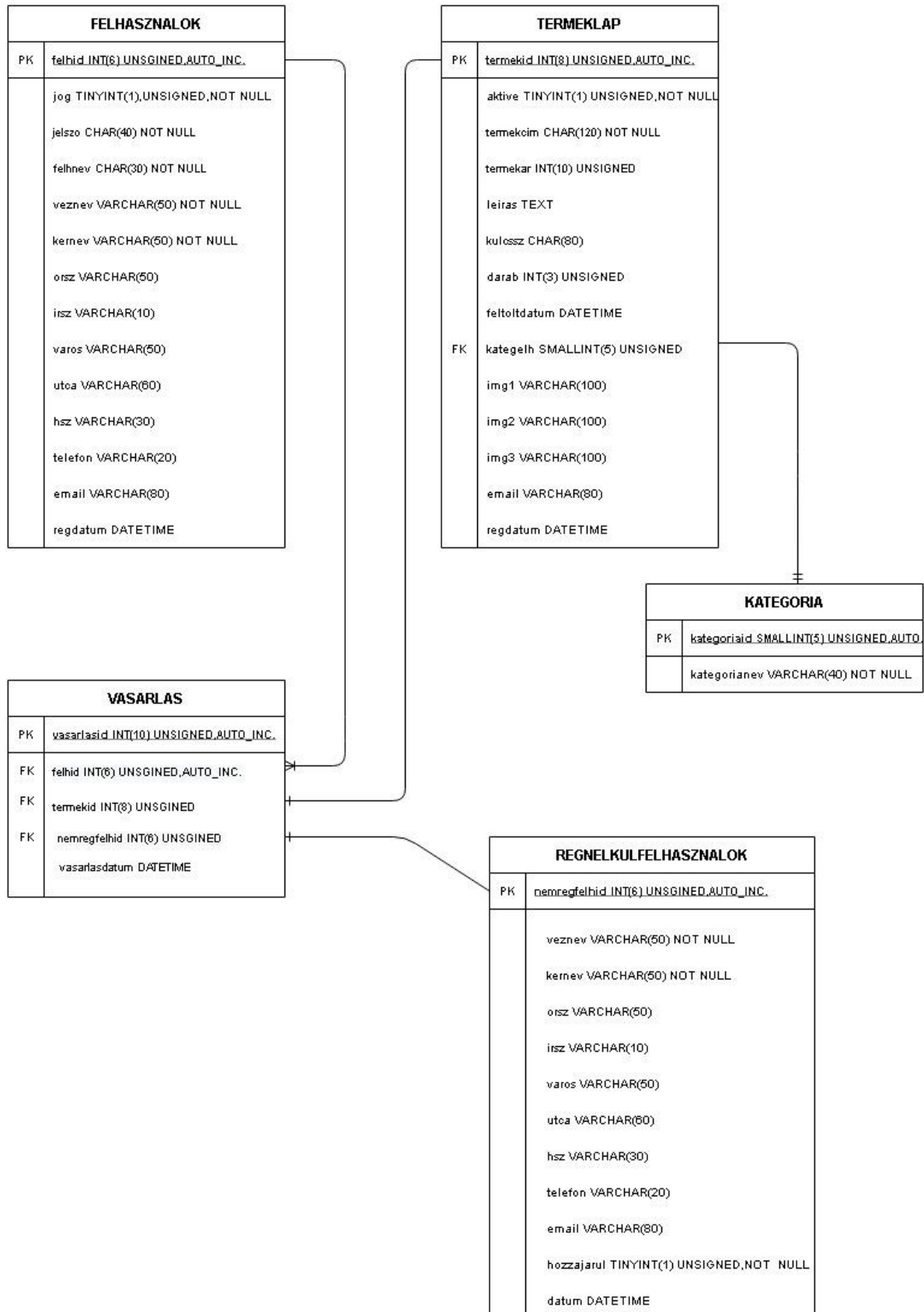
3.1. Adatbázisok

Az adatbázis megtervezésekor a valós világbeli objektumokat kell meghatározni. Az objektumokról és a kapcsolataikról tárolunk adatokat. Az adatbázisunk meghatározásához az első lépés az egyedek meghatározása volt, majd az egyedek attribútumait kellett megtervezni. Az egyedek, amelyekről információkat kell tárolni a web-áruház létrehozásakor:

- Felhasználók,
- Nem regisztrált felhasználók,
- Termékek,
- Kategóriák,
- Vásárlás

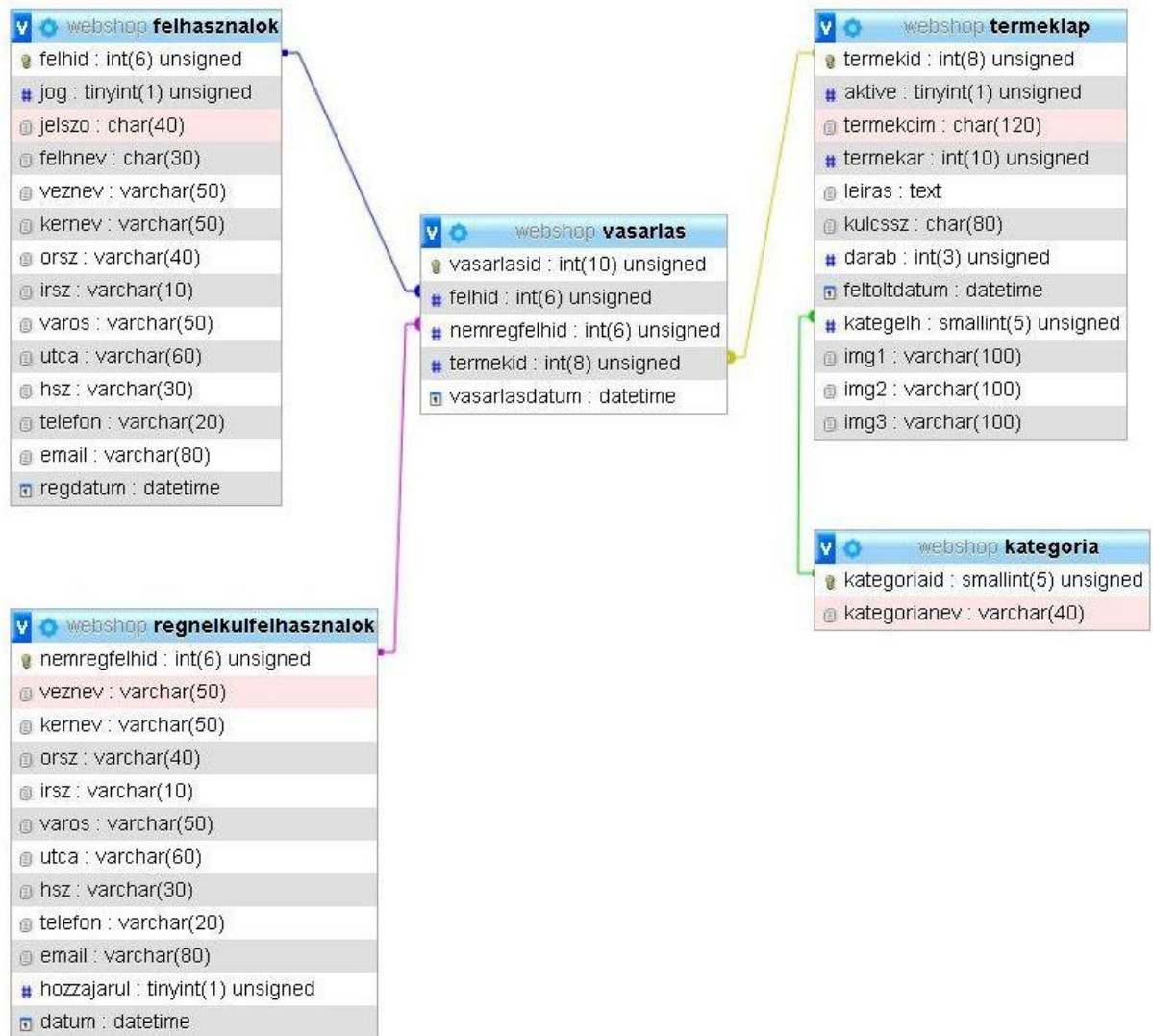
3.2. A web-áruház adatbázis szerkezetének bemutatása egyed-reláció (ER) ábrázolással

ER diagram.



3.3. A web-áruház adatbázis szerkezetének bemutatása Crow's Foot ábrázolással

Crow's Foot ábrázolással az adatbázis szerkezete ((MySQL tervező segítségével.)



3.4. Az adatbázis tábláinak attribútumai

A tárolandó attribútumokat az adatbázis tervezésekor a megvalósítás sikere érdekében szükséges pontosan meghatározni és az összetett attribútumokat egyszerűsíteni, összetevőire bontani. Ezt a normalizálás folyamatával lehet megvalósítani.

A normál formára hozás a tervezés egy fontos lépése, a projekt esetén a 3. normál formáig valósítottam ezt meg.

További figyelmet kíván az adatbázis megtervezésekor, hogy elkerüljük a redundanciát.

A redundancia jelentése: Felesleges adatismétlődés a táblákban, ami az adatbázisban törlési vagy módosítási anomáliákat okozhat.

Szükségtelen olyan adatok tárolása, amit már eltároltunk vagy kiszámíthatóak más adatokból.

3.5. A webáruház adatbázisának tervezési szempontjai

A web-áruház létrehozásakor az egyik legnehezebb feladat az adatbázis felépítése. A későbbi működés, gyorsaság, megbízhatóság szempontjából próbáltam ideálisan létrehozni az adattáblákat. A táblák mezőit a megfelelő változó típusokkal hoztam létre és megfelelő méretű hosszúságot engedélyeztem nekik, de nem mindig az adott adattípus legnagyobb fajtájával. Mert pl. az nem reális hogy a web-áruházamban többen fognak regisztrálni, mint 999999 személy.

A számokat tartalmazó mezőket minden esetben **UNSIGNED** módban hoztam létre, hisz a web-áruházban negatív adatok nem szerepelnek majd.

A mezők egy részét **NOT NULL** megkötéssel láttam el, szintén jelenthet egy kis sebesség növekedést, bár a kódolásnál nagyobb odafigyelést igényel.

A tervezés kezdeti szakaszában még terveztem egy adminisztrátor adattáblát is, de aztán a felhasználók tábla kialakításakor úgy terveztem meg a struktúrát, hogy ne kelljen ezért egy külön tábla, és a későbbi bővíthetőség sem sérül.

A **MySQL**-ben az adott táblákat **INNodb** módban kell létrehozni, hogy az idegen kulcsokkal tudjunk dolgozni.

Az összes táblát **INNodb** tároló motorral alakítom ki, elkerülendő a későbbi ebből adódó hibákat. Ezek mellett az **INNodb** előnye, hogy támogatja a tárolt eljárásokat is.

A tervezéskor az optimális gyorsaság szempontjából még fontos kérdés hogy a karaktereket tartalmazó adattípusokat változó karakterhosszúságú típusként (**VARCHAR**) vagy fix hosszúságú (**CHAR**) típusként hozzuk létre.

A **VARCHAR** típus csak az adott beleírt adatmennyiséget tartalmazza, ezért kevesebb a tárhely igénye. A **CHAR** típusnak fix a tárhely igénye, de gyorsabb.

Általában **VARCHAR** mezőket hoztam létre, de azoknak a karakterszámát próbáltam a gyakorlatból vett tapasztalatok alapján korlátozni, hogy valahol félúton legyen a gyorsaság és a tárhely igény között.

Egy kiterjedt web-shop esetén az adatbázis mérete is számít, hisz a tárhely szolgáltatóknál egyenesen arányosak az árak az igényelt tárhelyhez képest.

Bár az utóbbi években az erőforrások egyre könnyebben elérhetőek, de azért a tervezéskor próbáltam az 'arany' középutat választani, tekintettel arra, hogy az adatbázis szerkezete nem homogén.

Az adatbázis műveleteket igyekeztem korlátok közé szorítani, tehát adott esetben egy listázás esetén nem az adott tábla minden adatát érinti, vagy minden rekord minden mezőjét, csak a keresett adatát. Listázások esetén a kereséseket a lehető legszűkebbre alakítottam.

Még fontos kiemelni a táblák kialakításával kapcsolatban, hogy az összes idegen kulcs (**FOREIGN KEY**) mező, amely egy másik tábla elsődleges kulcsára mutat, pontosan ugyanolyan típusú mezőt kapott és pontosan ugyanakkora méretre korlátozva, hogy elkerülhetőek legyen az integritási problémák.

3.6. A táblák közötti kapcsolatok

Egy felhasználónak több vásárlása is lehet, tehát a felhasználók és vásárlások közötti kapcsolat egy 1:N-hez kapcsolódik. (felhasznalok -> vasarlas)

Viszont egy felhasználó is több terméket vásárolhat meg, tehát a 'felhasználók' és 'termékek' között egy 1:N kapcsolat valósul meg. (felhasznalok->termeklap)

A nem regisztrált felhasználók vásárlásai és a termékek között ugyanolyan relációk állnak fenn, mint az előzőleg leírt két bekezdésben.

3.7. Az adattáblák és jellemzőik

3.7.1.A 'felhasznalok' tábla fontosabb mezői

Elsőként a regisztrált felhasználók táblát terveztem meg. A tárolandó tulajdonságok között voltak összetett tulajdonságok is, melyeket normalizálni kellett.

Itt nagyobb figyelmet kapott a **FELHID** mező. A gyorsaság szempontjából fontos lehet, hogy ne egy **BIGINT** hosszúságú számon kelljen műveleteket végezni, így ez egy korlátozott **INT** típus lett, 6 számjegyre.

Elegendő, ha a web-áruházban „csak” 999.999 felhasználó regisztrálhat.

(Érdekességgé említem, hogy a www.vatera.hu oldalnak 2.6 millió regisztrált tagja volt 2013-ban, és a www.ebay.com oldalnak kb. 100 millió tagja volt ugyanekkor.)

A '**JOG**' mezőben jelzem az adott felhasználó jogait, a későbbi bővíthetőség szempontjából lehet az hasznos, hogy itt egy TINYINT adattípust használok 1 számjegyre korlátozva. 10 szintnél többre nem lesz szükség később sem jogosultságokból. Az adminisztrátor felhasználó 1-es számot kap a 'jog' mezőben. A felhasználók 3-as számot.

A '**FELHID**' mezőben egy egyedi azonosítót tárolunk, mely a felhasználó azonosítására fog szolgálni, 6 számjegyre korlátozva (UNSIGNED) előjel nélkül. (PRIMARY KEY) Elsődleges kulcs, (AUTO INCREMENT) , automatikus inkrementálással.

A '**JELSZO**' mezőben a felhasználó belépési kódja lesz tárolva, SHA1 kódolással mely 40 karaktert képes kódolni 8 bit-en, így a mező hosszát is ennyiben korlátoztam.

A '**FELHNEV**' mező a munkamenet kezelés, a felhasználó azonosításához, ellenőrzéséhez szükséges. Maximum 30 karakter lehet a hossza.

Az '**IRSZ**' az irányítószám tárolására szolgáló mező. Külföldi vásárló esetén nem ritka a 8-10 betű és számjegy kombináció sem. Ezért VARCHAR lett és nem INT mező típusa.

A '**VAROS**' mezőben elegendőnek gondoltam 50 karaktert mert a világ leghosszabb városának a neve 170 karakter, de a hazai és magyarul írt településnevek maximum 22-25 karakter körül vannak.

'HSZ' (VARCHAR)(30):

A házszámot tároló mező. A különféle házszámok, betűk miatt szükséges jóval több karakter mint 4-5, 30-ban maximáltam.

'REGDATUM' (DATETIME):

Az regisztráció pontos dátumát tartalmazó mező, melyet majd a **MySQL NOW()** függvényével töltök fel az aktuális dátummal.

FELHASZNALOK tábla:

| Megkötések | Típus | Mezőnév |
|-------------------------------------|-------------|----------|
| UNSIGNED,PRIMARY KEY,AUTO INCREMENT | INT(6) | FELHID |
| UNSIGNED,NOT NULL | TINYINT(1) | JOG |
| NOT NULL | CHAR(40) | JELSZO |
| NOT NULL | CHAR(30) | FELHNEV |
| NOT NULL | VARCHAR(50) | VEZNEV |
| NOT NULL | VARCHAR(50) | KERNEV |
| nincs | VARCHAR(40) | ORSZ |
| nincs | VARCHAR(10) | IRSZ |
| nincs | VARCHAR(50) | VAROS |
| nincs | VARCHAR(60) | UTCA |
| nincs | VARCHAR(30) | HSZ |
| nincs | VARCHAR(20) | TELEFON |
| nincs | VARCHAR(80) | EMAIL |
| nincs | DATETIME | REGDATUM |

3.7.2.A 'termeklap' tábla fontosabb mezői

A '**TERMEKID**' MEZŐ: A termékek egyedi azonosító száma, automatikus számlálóval (**AUTO INCREMENT**), mely elsődleges kulcs (Primary Key).

A '**TERMEKCIM**' a legfontosabb mező ebben a táblában. Ennek a típusa **CHAR**. Erre lesz a legtöbb lekérdezés a potenciális vásárlóktól és eladói szemszögből is nagyon fontos ez a mező. Itt kell megadni kellő figyelemmel az eladó terméket, jól körülírva azt. Azért is lett fix tárhely igényű **CHAR** típus a termékcím mező, mert ennek a mezőnek a hosszát mindig ajánlott végig (vagy ameddig lehet és szükséges) kitölteni ez a lekérdezések gyorsaságát is segíti.

A '**LEIRAS**' mező hosszabb szövegek tárolására készült, itt a **TEXT** típust használjuk, melyet erre találtak ki. Ez 65535 karaktert (kb. 300-700 szó) enged. Saját üzleti gyakorlatból vett tapasztalat alapján 400 szó elég a széleskörű leíráshoz, tájékoztatáshoz, a szállítási részletekkel is, ami kb. 3000-4000-5000 karakter.

A **'KULCSSZ'** mező a termékre jellemző kulcsszavakat tárolja. Itt is **CHAR** lett a mező típusa, de csak 30 karakter engedélyeztünk. Ezen a mezőn alapesetben már gyorsabb kereséseket lehet futtatni, mint a termékcímen. A későbbi bővíthetőség végett (is) került bele ez a mező a táblába, hogy majd a META adatokat dinamikussá lehessen tenni és ezekkel a szavakkal bővíteni.

A **'DARAB'** mező a termékek darabszámát tárolja. Bár egyedi termékeket árusítására hozzuk létre, de nyitva hagyjuk, hogy mégis lehessen több darabot árulni.

A **'FELTOLTESDATUM'** mező a feltöltés dátumát jelzi. A termékeknel fontos információ lehet, hogy mikor töltöttük fel őket eladásra, előfordulhat ugyanis, hogy már túl régóta nincs rájuk kereslet, ennek vizsgálatához hoztam létre ezt a mezőt. Feltöltés dátumát a **MySQL NOW()** függvényével fogjuk itt rögzíteni.

Az **'AKTIVE'** mezőben a termék aktivitását lehet vizsgálni. A gyakorlatban ez azt jelenti, hogy ha 1, akkor az adott termék megvásárolható, ha 0, akkor már nem. Ez esetben a **TERMEK LAP** tábla a már a specifikációban is megfogalmazott ARCHÍVUM szerepét is ellátja. Később ennek a mezőnek az 1-re állításával újra eladósorba lehet állítani a terméket.

A **'KATEGELH'** mező egy idegen kulcs a Kategória táblára, amely a **'KATEGORIAID'**-re hivatkozik.

Az **'IMG'** mezők:

Kiemelt fontosságú az egész munka és tervezés alatt, hogy az adott termékhez tartozó képeket hogyan tároljuk. Az utóbbi időben már nem szentírás hogy a képeket mindenképp fájlrendszerben tároljuk, és azok neveit hivatkozásként helyezzük el az adatbázisban. A tervezés során itt ütköztem a legnagyobb kihívásokba. Az adatok integritása szempontjából, és a biztonsági mentések egyszerűbb megvalósítása miatt célszerű lenne a képek az adatbázisban való tárolása.

A fájlrendszeren való tárolás a gyorsaság szempontjából lenne előnyös, de hátrány lehet, hogy a fájlok kezelése még több munkát igényel a tervezésnél és implementálásnál is. Továbbá biztonsági szempontból sem minden esetben a legjobb. Ha egy betörés történik a szerverre és törlik a képeket, akkor használhatatlanná válik az egész adatbázis. Természetesen erről is lehet biztonsági másolatot készíteni. Továbbá fájlfeltöltésnél / képfeltöltésnél is lehetne feltételezni, hogy támadás éri az adatokat, ha jogosulatlanul lépnek be, ennek a kivédésére is fel kell készülni.

Megoldás lehetne még, egy külön a képek tárolására létrehozott képek adattábla is, de ez is újabb és újabb programozási nehézségeket generálna. (Újabb tábla létrehozása, az azokkal való műveletek, stb.)

Vizsgáltam a fájlrendszerben való tárolásnak is vannak előnyei. A legfontosabb hogy sokkal gyorsabban elérhetőek, mint az adatbázisban tárolt képek, és ez a XXI. századi világban talán a legfontosabb, így végül is a fájlrendszerben való tárolás mellett döntöttem.

TERMEKLAP tábla:

| Megkötések | Típus | Mezőnév |
|-----------------------------------------------|--------------|----------------|
| UNSIGNED,PRIMARY KEY,AUTO INCREMENT | INT(8) | TERMEKID |
| UNSIGNED,NOT NULL | TINYINT(1) | AKTIVE |
| NOT NULL | CHAR(120) | TERMEKCIM |
| nincs | TEXT | LEIRAS |
| nincs | CHAR(30) | KULCSSZ |
| UNSIGNED | INT(3) | DARAB |
| nincs | DATETIME | FELTOLTDATUM |
| UNSIGNED,FOREIGN KEY (kategoria->kategoriaid) | SMALLINT(5) | KATEGELH |
| UNSIGNED | VARCHAR(100) | IMG1,IMG2,IMG3 |

3.7.3.A 'vasarlas' tábla fontosabb mezői

A következő tábla, amit megterveztem egy kapcsoló tábla, a vásárlás tábla. Ebből a táblából lesznek elérhetőek a vásárlások. A tábla a terméklap és a felhasználók táblát kapcsolja össze, egy-egy kapcsolatot testesít meg. A táblának négy sora van:

A '**VASARLASID**' a vásárlást azonosító, egyedi azonosító, elsődleges kulcs, automatikus inkrementálással, értéke egy tíz számjegyre korlátozott **INT** típus. (4.294.967.295 vásárlást lehet rögzíteni.)

A '**FELHID**' mező egy idegen kulcs, a regisztrált felhasználók táblájának '**FELHID**' mezőjére mutat.

A '**NEMREGFELHID**' mező szintén idegen kulcs, a nem regisztrált vásárlók adatait tárolja, annak is a '**NEMREGFELHID**' nevű mezőjére mutat.

A '**TERMEKID**' mező szintén idegen kulcs a terméklap tábla '**TERMEKID**' mezőjére mutat.

A tábla még tartalmazza a vásárlás dátumát '**VASARLASDATUM**' (**DATETIME**) típusként, melynél mindig az aktuális idő tárolódik a **MySQL NOW()** függvény segítségével.

VASARLAS tábla:

| Megkötések | Típus | Mezőnév |
|-----------------------------------------------------------|----------|---------------|
| UNSIGNED,PRIMARY KEY,AUTO INCREMENT | INT(10) | VASARLASID |
| UNSIGNED,FOREIGN KEY (felhasznalok->felhid) | INT(6) | FELHID |
| UNSIGNED,FOREIGN KEY (regnelkufelhasznalok->nemregfelhid) | INT(6) | NEMREGFELHID |
| UNSIGNED,FOREIGN KEY (termeklap->termekid) | INT(8) | TERMEKID |
| | DATETIME | VASARLASDATUM |

3.7.4.A 'kategoria' tábla fontosabb mezői

A kategória tábla szorosan kapcsolódik a terméklap táblához, hisz minden terméket el kell helyezni, egy a rá legjobban jellemző kategóriában.

A **'KATEGORIAID'** egy egyedi azonosító, ezzel tudjuk a kategóriák neveit megkeresni. **SMALLINT(5)** típusú mező,előjel nélkül,automatikus számlálással. Maximum 65535 kategóriát lehet létrehozni.

A **'KATEGORIANEV'** maga az elnevezése az adott kategóriának, **VARCHAR(40)** típusú lett a mező 40-es hossza korlátozva.

KATEGORIA tábla:

| Megkötések | Típus | Mezőnév |
|-------------------------------------|-------------|--------------|
| UNSIGNED,PRIMARY KEY,AUTO INCREMENT | SMALLINT(5) | KATEGORIAID |
| NOT NULL | VARCHAR(40) | KATEGORIANEV |

3.7.5. A 'regnelkulfelhasználók' (Regisztráció nélküli felhasználók) tábla fontosabb mezői:

Azoknak a felhasználóknak, akik regisztráció nélkül óhajtanak az áruházban vásárolni, itt tároljuk el az adatait. A tábla oszlopai hasonlóak a sima **'felhasznalok'** tábla oszlopaihoz.

Az eltérések:

Itt a vásárlót azonosító egyedi mező neve: **'nemregfelhid'**. Ezen felül van egy **'hozzajarul'** mező is, amiben arról tárolunk információkat, hogy az adott vásárló hozzájárul e vagy sem hogy számára marketing célú megkereséseket intézzük email formájában. (1 - igen , 0 - nem)

Emellett természetesen jelszó mező nincs benne, hisz egyszeri vásárlás esetén nem szükséges ezt megadni.

A **'NEMREGFELHID'** mező egy elsődleges kulcs, **INT(6)** előjel nélküli típus, automatikus inkrementálással, a nem regisztrált vásárlók azonosítására.

A **'VEZNEV','KERNEV', 'ORSZ','IRSZ', 'VAROS','UTCA','HSZ','TELEFON','EMAIL'** mezők ugyanolyanok, mint a **'felhasznalok'** tábla ugyanilyen nevű mezői.

A **'HOZZAJARUL'** mező egy **TINYINT** típus 1 számjegyre koncentrálva. A nem regisztrált felhasználóknak kell a vásárlás előtt nyilatkozniuk arról, hogy szeretnének e marketing jellegű megkereséseket. Ha a vásárló szeretne, akkor egy '1'-es kerül értékként a mezőbe, ha nem akkor egy '0'.

'DATUM' DATETIME

A nem regisztrált vásárló adatainak az adatbázisba kerülésének dátuma. (Tulajdonképp a vásárlás dátuma is ez lesz.)

REGNELKULFELHASZNALOK tábla:

| Megkötések | Típus | Mezőnév |
|-------------------------------------|-------------|--------------|
| UNSIGNED,PRIMARY KEY,AUTO INCREMENT | INT(6) | NEMREGFELHID |
| NOT NULL | VARCHAR(50) | VEZNEV |
| NOT NULL | VARCHAR(50) | KERNEV |
| nincs | VARCHAR(40) | ORSZ |
| nincs | VARCHAR(10) | IRSZ |
| nincs | VARCHAR(50) | VAROS |
| nincs | VARCHAR(60) | UTCA |
| nincs | VARCHAR(30) | HSZ |
| nincs | VARCHAR(20) | TELEFON |
| nincs | VARCHAR(80) | EMAIL |
| UNSIGNED,NOT NULL | TINYINT(1) | HOZZAJARUL |
| nincs | DATETIME | DATUM |

4. HASZNÁLATI MODELL ÉS ESET DIAGRAMOK

4.1. Modell és elemzés

Két fő igényt kell elemezni, a felhasználók információs igényeit és funkcionális követelményeket.

A modell lényege, hogy bemutatja, hogy mely felhasználók mire akarják használni a rendszert.

Funkcionális követelmények a rendszerrel szemben tulajdonképp a felhasználók által remélt/elvárt szolgáltatások.

A használati eset diagramok a modellezendő rendszert mutatják be.

A használati eset diagram a felhasználókból, használati esetekből, és a közöttük lévő kapcsolatból áll.

A felhasználó az, aki magát a rendszert használja, 'használati' eseteket kezdeményezhet, információt adhat, vagy adhat és fogadhat.

Áruházamban a használati eseteket a regisztrált felhasználókra, a nem regisztrált vásárlókra és az adminisztrátor felhasználókra kell vizsgálni.

4.2. Jogosultsági szintek

Az áruházamban két jogosultsági szint van, a regisztrált vásárló és az adminisztrátor. A '**felhasznalok**' táblában a '**jog**' mezőben van ennek az értéke rögzítve. **TINYINT** típust használtam 1 számjegyre korlátozva, ezáltal lehetőség van még 8 különféle szint beállítására a későbbiekben.

Az első adminisztrátort manuálisan kell létrehozni. (Regisztrálni majd az adott felhasználó '**jog**' mezőjében a jogosultságot átírni 1-re!)

A regisztrált felhasználók a regisztráció során kapják a jogosultságuk alapértéket (3). Ezt később nem lehet módosítani. Viszont az adminisztrátor hozhat létre a felületen keresztül is felhasználót, és az elején ennek az értéknek a megadás lehetséges.

A program kódja nem enged hozzáférést az adminisztrátor által elérhető funkciók használatához. Továbbá a kód semmilyen jogosulatlan hozzáférést nem enged a program fájljaihoz kliens oldalról. A felhasználók csak a jogosultsági szintjüknek megfelelő php fájlokat és kódrészeket érhetnek el.

A web-shop szerkezete is kétszintes. Van egy felhasználói felület és egy eltérő adminisztrátori felület is. Szándékosan alakítottam két külön felületet ki, a későbbi bővíthetőség miatt.

A nem regisztrált felhasználók külön nem kaptak jogosultságot, viszont a kódból lehetőségük sincs a vásárláson kívül semmire.

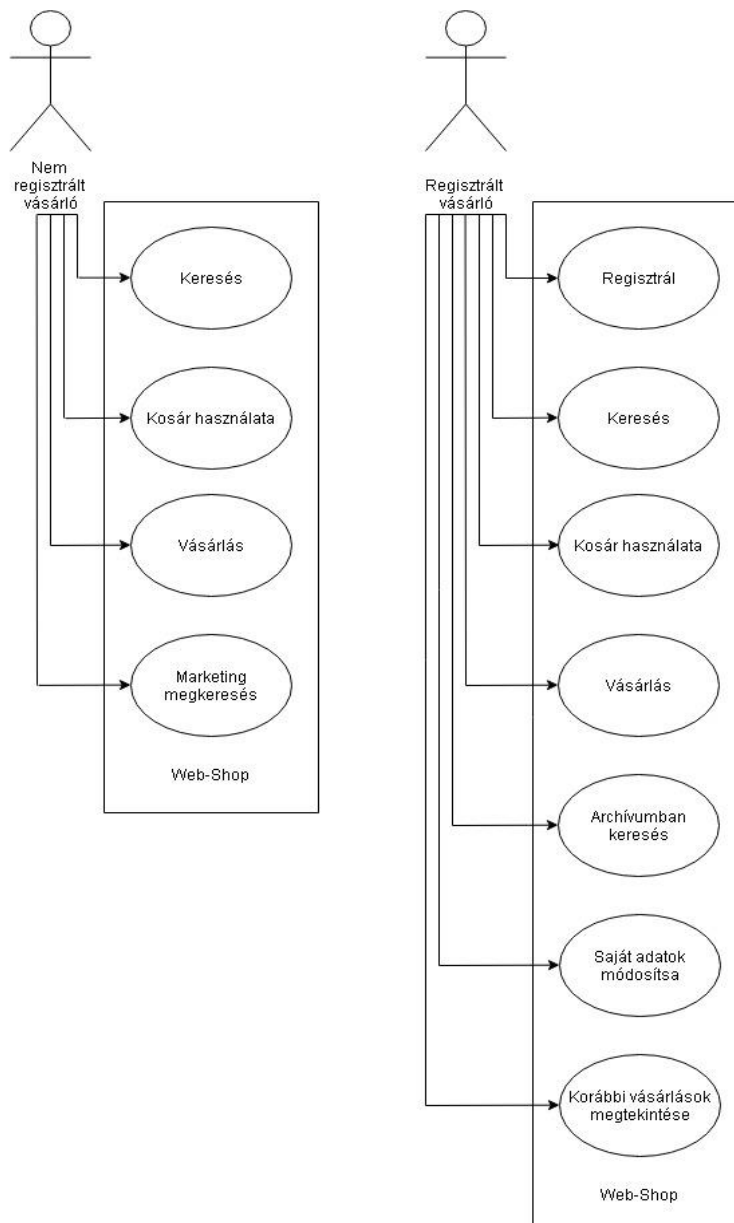
4.3. Nem regisztrált vásárló jogai

- Tud keresni a termékek között.
- Tud a kosarába tenni és kivenni termékeket.
- Tud vásárolni, a megrendelését véglegesíteni.
- Vásárlás közben felhatalmazást adhat arra, hogy a későbbiekben megkeressük marketing célú ajánlatokkal.

4.4. Regisztrált vásárló jogai

- Tud regisztrálni, létrehozni a saját profilját az adataival.
- Tud keresni a termékek között.
- Tud keresni az eladott termékek között is.
- Tud a kosarába tenni termékeket, kivenni onnan.
- Tud vásárolni, a megrendelését véglegesíteni.
- A regisztrációval hozzájárul, hogy megkeressük marketing célú megkeresésekkel. (A bemutatkozás fülön tájékoztatjuk erről!)
- Tudja a saját adatait módosítani.
- Meg tudja tekinteni a saját korábbi vásárlásait

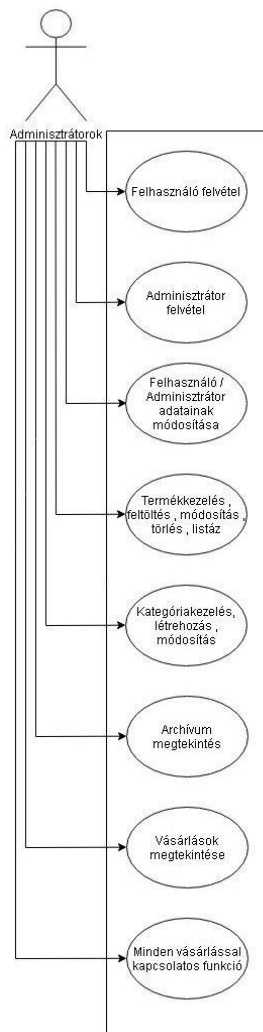
4.5. Nem regisztrált és regisztrált felhasználók eset diagramjai



4.6. Adminisztrátor jogai

- Tud felhasználót / vásárlót létrehozni.
- Tud adminisztrátor jogosultságú felhasználót is létrehozni.
- A már regisztrált felhasználók adatait is tudja módosítani, törölni őket.
- Az adminisztrátor szintű felhasználók adatait is tudja módosítani, de törölni nem tudja az adminisztrátorokat.
- (azt csak manuálisan lehetséges!)
- Terméket tud feltölteni, törölni, módosítani, listázni.
- Kategóriát tud feltölteni, módosítani. (Törölni az esetleges anomáliák miatt nem lehet!)
- Az archívumot is meg tudja tekinteni.
- Megtekintheti a regisztrált vásárlók vásárlásait.
- Megtekintheti a nem regisztrált vásárlók vásárlásait.
- Minden vásárlással kapcsolatos funkciót is elér a felhasználói felületről.

4.7. Az adminisztrátor eset diagramja:



5. OSZTÁLYOK

Egy alkalmazás felépítéséhez szükséges előre tervezni. A webes alkalmazásoknál is igaz ez. Nem lehet egy program megvalósulását a véletlenekre bízni. A strukturált programozási alapelv manapság már elavult, bár egyszerűbb rövid projektekben még lehetne használni. A manapság használatos programozási elv, az objektumorientált programozás. (A továbbiakban OOP.) Az OOP. egy összetett, sok lehetőséggel rendelkező módszer, mely lehetővé teszi a kódunk többszöri felhasználását, osztályok létrehozását. Ennek a módszernek talán ez a legnagyobb előnye. Ha így kódolunk és tervezünk, tudjuk példányosítani az osztályainkat, és létrehozhatunk belőle objektumokat.

Több osztályt hoztam létre a kódban, a két külön felület miatt ötöt.

Van egy külön felület az adminisztrátornak, és egy külön felület a felhasználói, vásárlói oldalról.

Az osztályokban a tagfüggvények egy része fogad paramétereket, egy része nem, szükség szerint. A továbbiakban a dokumentációban nem térek ki minden tagváltozóra, tagfüggvényre, csak a legfontosabbakra, tekintettel a megvalósult projekt méretére.

5.1. 'AdminNezet' osztály, UML diagram

Az adminisztrátor felhasználó működéséhez az **adminosztaly.php** fájlra van szükség (és az átirányító php fájlokra). Ebben három osztály van kódolva.

Az '**AdminNezet**' osztály és a függvényei főképp a megjelenítéssel kapcsolatos kód részeket tartalmazzák.

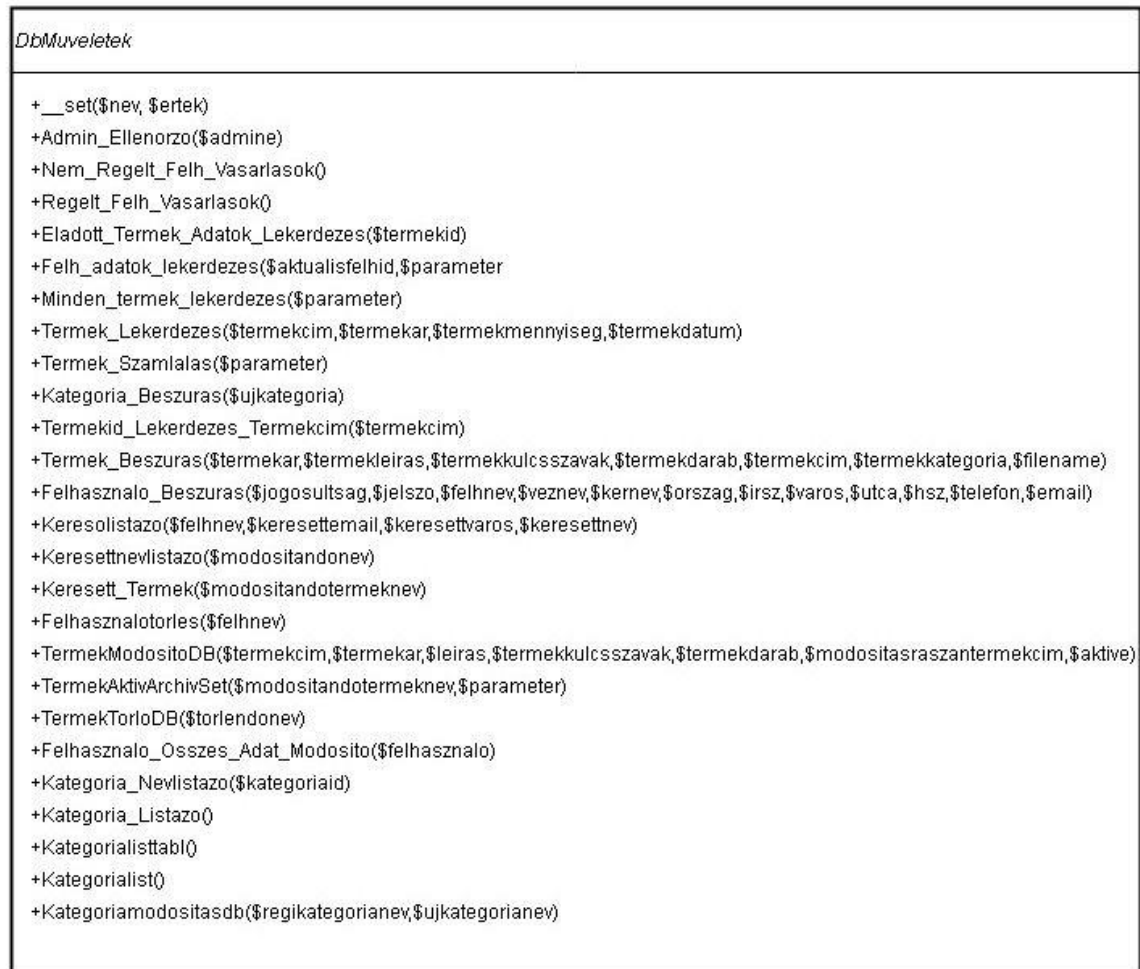
AdminNezet osztály diagram:



5.2. 'DbMuveletek' osztály, UML diagram

Az osztály az adatbázis műveletekkel foglalkozó osztály, de itt is vannak részben megjelenítéssel is foglalkozó tagfüggvények.

DbMuveletek osztály diagram:



5.3. 'Funkciok' osztály, UML diagram

A 'Funkciok' osztály olyan tagfüggvényeket tartalmaz, amelyek kapnak valamiféle paramétert, majd műveletet végeznek a kapott paraméteren és a visszatérési értékük a kívánt eredmény lesz. Nem az adatbázissal dolgoznak, és nem is megjelenítéssel kapcsolatosak.

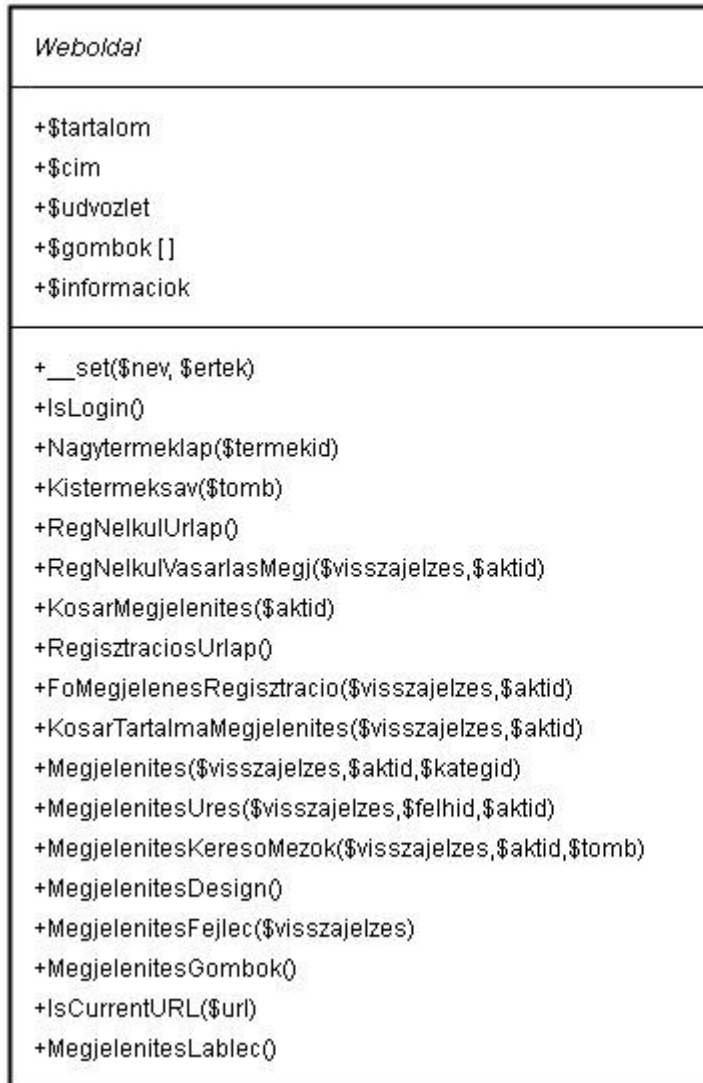
Funkciok osztály diagram:



5.4. 'Weboldal' osztály, UML diagram

A felhasználó által használható funkciók üzemeléséhez a **'weboldal.php'** fájlra van szükség (és az átirányító php fájlokra). Ebben két osztály van kódolva.

Weboldal osztály diagram:



5.5. 'FelhasznaloDbMuveletek' osztály, UML diagram

FelhasznaloDbMuveletek osztály diagram:



6. IMPLEMENTÁLÁS

Az implementálás gyakorlatilag a kódolás folyamata.

Igyekeztem a kódra vonatkozó tisztasági szabályokat betartani, és az OOP. elvei szerint kódolni.

Hozzá kell tennem, hogy a kódolás és a megvalósítás közben, főleg az elején, több olyan hibát ejtettem, amit később már nem volt idő kijavítani. Például bizonyos függvények adott esetben túl sok paramétert fogadnak, illetve a tagváltozók használatát is hatékonyabban lehetett volna alkalmazni. A tervezés során két nagyobb részre osztottam a projektet, szándékosan. Az egyik az adminisztrátor működésével kapcsolatos, a másik a felhasználó által használt dolgokkal kapcsolatos. A két rész kódolás részleteiben és felépítésében is eltér.

A kód teljes egészét PHP-ből hoztam létre, így minden HTML megjelenést a PHP biztosít. A későbbiekben a sebesség szempontjából érdemes lesz különválasztani a PHP és a HTML kódokat.

Az adminisztrátor felület, és annak a fájljai védelemmel vannak ellátva a külső jogosulatlan használattól. Az eljárások elején először vizsgáljuk, hogy van-e valaki bejelentkezve, vagy nincs. Ha nincs, akkor értesítjük a felhasználót, hogy be kell jelentkeznie a megtekintéshez. Viszont a következő vizsgálat azt vizsgálja, hogy a belépett felhasználónak van-e adminisztrátori jogosultsági szintje. Amennyiben nincs ez a jogosultsága, úgy hibát írunk ki a képernyőre, melyben jelezzük, hogy jogosulatlanul akar megtekinteni egy fájlt.

A kódolásnál további figyelmet kapott, hogy a különféle beviteli mezőket minden esetben a HTML5-től használatos REQUIRED attribútummal láttam el, és a megfelelő határoló értékekkel korlátoztam. Egy alapvető védelmet ez is nyújt.

A felhasználói felület implementálásával kapcsolatban már sikerült hatékonyabban dolgoznom, így az azt működtető osztályokban kevesebb függvényt írtam. Itt is megjelenítő, és adatbázissal dolgozó függvények vannak.

A kódolás során számos php fájlt is létrehoztam, melyek a működést és az átláthatóságot segítik. A `vasarlas.php` fájlban a vásárlással kapcsolatos kód van, a `kosar.php` fájlban a kosár műveletekkel kapcsolatosak. Később ezeket is érdemes osztályokban elhelyezni.

6.1. Adatbázis csatlakozás, `dbconnect.php`

Fontos még külön megemlíteni, hogy az adatbázishoz való csatlakozáshoz nem hoztam létre egy külön osztályt, vagy nem tettem bele pl. a **'Funkciok'** osztályba. Az ezzel kapcsolatos műveletek a **`dbconnect.php`** fájlban vannak implementálva egy függvényként, melyet egy php függvény (**`require_once`**) segítségével illeszték bele a kódba, illetve az osztályokat tartalmazó php fájlokba. Így könnyebb az adatbázishoz való csatlakozási adatokat módosítani, ha szükséges.

Kód:

```
„function dbconnect(&$con){  
$con;  
$servername = "localhost";  
$username = "root";
```

```

$password = "";
$dbname = "webshop";

$con = new mysqli($servername,$username,$password,$dbname);

if($con->connect_error>0)
    die("<h1>Hiba az adatbázis csatlakozáskor!</h1>");
else
    $con->set_charset("utf8"); }

```

6.2. AdminNezet osztály

Ebben az osztályban először az adminisztrátor fő oldal megjelenésével kapcsolatos függvényeket alkottam meg. Létrehoztam az oldalon a gombokat, az elrendezést, és az oldal grafikai elrendezését is, a karakterkódolást, a meta adatokat, melyeket a HTML HEAD elemében tároltam el. Az adminisztrátor felület fő oldalához az egyedi megjelenítést a **styleadmininfo.css** fájl tartalmazza.

Következő lépés volt az adminisztrátor felület al oldalainak funkcióinak lekódolása. Létre kellett hozni az összes funkcióhoz a megfelelő gombokat, és azokhoz a megfelelő átirányító fájlokat. Számos függvény gondoskodik arról, hogy felhasználót lehessen felvinni, törölni, módosítani, terméket lehessen feltölteni, törölni vagy, módosítani vagy listázni. Ezen felül a kategóriák felvitelét és módosítását is lekódoltam. Az archívum megtekintése, és az eladott termékek megtekintése opciók a kódolás végén került sorra. Az 'al' oldalak grafikai megjelenítéséhez a **styleadminal.css** fájl tartozik.

Az osztály egyik fontos alkotója EladottTermekMegjelenites(\$siker,\$parameter) függvény.

A létrehozásánál fontos szempont volt, hogy ne kelljen két külön függvényt írni a regisztrált felhasználók és a nem regisztrált felhasználók vásárlásainak megtekintéséhez.

Az eljárás elején az implementálás részben leírt kód védelem van. Amennyiben van jogosultsága a belépőnek a megtekintéshez, úgy a függvény megjeleníti a korábbi vásárlásokat, függően attól, hogy a \$parameter változóban milyen értéket kapott.

Amennyiben 1 értéket kapott, úgy a regisztrált vásárlásokat mutatja meg nekünk, amennyiben 0 értéket kapott, a nem regisztrált vásárlásokat jeleníti meg a függvény.

Mindkét paraméter esetén 1-1 függvényt hívunk meg az adatbázis kezeléssel foglalkozó osztályból, mely egy SQL lekérdezés eredményével tér vissza egy tömb formájában, melyet \$regeltvasarlasoktomb-nek neveztem el. Az eltérés annyi hogy 1 értéknél a regisztrált felhasználók táblából olvassa ki a felhasználó 'id'-jét, 0 értéknél a nem regisztrált felhasználók táblából olvassa ki a 'nemregfelhid'-t.

Miután megvannak a keresett értékek, egy for ciklussal megyünk végig a feltöltött tömbön. Mivel a nem regisztrált és a regisztrált felhasználók adatai külön táblában vannak tárolva, ezért a for cikluson belül még tovább vizsgáljuk az átadott \$parameter nevű változót. Ha nem regisztrált vásárló vásárlásait tekintjük meg, akkor a korábban említett \$regeletvasarlasoktomb tömbben csak akkor lesz érvényes felhasználót azonosító tömbem, ha a kiolvasást végző függvény tudja azonosítani a felhasználót. Tehát 1 és 0 érték vizsgálatnál

bevezettem egy változót, mely az *\$aktualisfelhid* nevet kapta, melybe mindig a megfelelő azonosító fog kerülni, hisz az 1 vagy 0 értéknél a tömb korábban feltöltött értéke a megfelelő táblából való adatot fogja a tömbelem értékévé tenni. Miután ezek a lekérdezések a 'vasarlas' táblából történnek, a cikluson belül még kimentem egy változóba az *\$aktualisvasarlasid*-t, az *\$aktualistermekid*-t is, majd egy újabb meghívott SQL lekérdezéssel lekértem az adott vásárló adatait, és az adott vásárláshoz tartozó felhasználó adatait is. Majd ezeket a megfelelő függvényekkel megjelenítjük.

A függvény bemutatása azért volt szándékomban, mert sok hasonló függvényt írtam a kódban, melyek kapcsolóként működnek, függően a kapott paramétertől.

6.3. Dbmuveletek osztály

Ennek az osztálynak a tervezése közben szembesültem azzal, hogy a tervezés egy nagyon összetett feladat, és már-már egy külön szakma épülhetne erre, mint ahogy épül is.

Az implementálás közben számos előre már megtervezett művelethez tudtam lekódolni a függvényeket, viszont néhány esetben a tervekben nem szereplő eljárást is le kellett kódolni.

Az osztály függvényei adatbázis műveleteket végeznek, és megjelenítéssel is foglalkoznak. Először a felhasználókkal kapcsolatos függvényeket alkottam meg, később a kategóriákkal kapcsolatosakat, majd a termékekkel kapcsolatosakat.

Az általam bemutatni kívánt egyik függvény, az Admin_Ellenorzo(\$admine) függvény. Az eljárás elején csatlakozunk az adatbázishoz, majd egy stringben eltároljuk az SQL lekérdezést, amit átadunk az adatbázis kapcsolatán hívott lekérdezés függvény paraméterének, mely eredményét a *\$result* változóban tároljuk el. Később definiálunk egy *\$jog nevű* változót és deklarálunk és *\$felhnev* változót. A következő lépés hogy az *\$result* változóban tárolt eredményt megvizsgáljuk, hogy adott e vissza az SQL lekérdezés bármennyi sort. Amennyiben nem, úgy jelezzük egy kiíratással a felhasználó felé a hibát. Viszont ha az érintett sorok száma nagyobb, mint nulla, akkor a `fetch_assoc()` php függvény segítségével kiolvassuk az adatbázisból a 'jog' és a 'felhnev' mezőket egy while ciklussal, és elmentjük őket a korábban ugyanilyen néven létrehozott változókban. Az eljárás következő része egy vizsgálat, melyben azt vizsgáljuk, hogy a *\$jog* nevű változó értéke '1' –e és hogy a korábban a függvény paramétereként átvett *\$admine* string, mely egy felhasználónevet tartalmaz, pontosan megegyeznek e. Erre a `php strcmp()` függvényét alkalmazzuk. Amennyiben ez a feltétel teljesül, akkor megtudtuk, hogy az adott felhasználó jogosultsági szintje '1', ami az adminisztrátori szint, és ha az adatbázisból kiolvasott felhasználónév és paraméteren kapott felhasználónév azonos, a függvény visszatér egy boolean típusú változó TRUE értékével. A függvény felhasználása előtt a belépett felhasználó azonosítóját kimentjük és azt adjuk át ennek a vizsgáló függvénynek. Ha valamilyen támadás érne korábban a `$_SESSION['felhasznalo_nev']` elemét, akkor mivel ennek a függvénynek adjuk át ezt a támadott 'adatot', csak akkor térhet vissza TRUE értékkel, ha egy adminisztrátor jogosultságú felhasználó felhasználónevét sikerül a 'támadónak' megadnia.

Az Admin_Ellenorzo(\$admine) függvény kódja:

```
function Admin_Ellenorzo($admine){  
    $con;
```

```

dbconnect($con);

$sql = "SELECT jog,felhnev FROM felhasznalok WHERE felhnev LIKE '$admine'";
$result = $con->query($sql);
$jog;
$felhnev = "";

if($result->num_rows>0)
{
    while($row = $result->fetch_assoc())
    {
        $jog = $row['jog'];
        $felhnev = $row['felhnev'];
    }
}
else
{
    echo "<h1>Nem adott a lekérdezés jó eredményt!</h1>";
}

if($jog == 1 && strcmp($felhnev,$admine) == 0)
{
    return true;
}
else
{
    echo "<h1>Nincs admin jogosultságod</h1>";
    exit();
}

$con->close();
}

```

A következő bemutatandó függvény a `TermekAktivArchivSet($modositandotermeknev,$parameter)`. Az eljárás célja hogy egy olyan terméket, ami még 'aktív' státuszban van és legalább 1 db elérhető belőle, átmódosítsa nem elérhetővé, és a darabszámát 0-ra csökkentse. Vagy adott esetben, ha nincs belőle elérhető, akkor az elérhetőséget aktívra állítja, és beállítson belőle 1 db-ot.

Ezt a kapott paramétertől függően teszi meg, majd átadja a vezérlést a megjelenítést végrehajtó függvénynek. A függvény elején csatlakozunk az adatbázishoz, majd a kapott paramétereket megtisztítjuk a felesleges karakterektől és az esetleges 'SQL Injection' támadások ellen.

Apró kitérő itt, hogy elméletben ide már úgy érkezik a program, hogy levizsgált adatot kap, ám több szakkönyv is leírta, hogy érdemes minden esetben biztonsági alapon levizsgálni a kódból érkező adatokat is, nehogy közben az támadást kapjon, és mégis kárt tehessenek az adatainkban a támadással.

Az eljárásban ezután rögzítünk a *\$parameter* változó értéke szerint egy-egy SQL lekérdezést, majd végrehajtjuk azt. Egy rövid ellenőrzésben vizsgáljuk hogy sikerrel zárult e az SQL művelet, majd visszaadjuk a megjelenítésnek a vezérlést és lezárjuk az adatbázissal való kapcsolódásunkat a **close()** függvénnyel.

A *TermekAktivArchivSet(\$modositandotermeknev,\$parameter)* kódja:

```
function TermekAktivArchivSet($modositandotermeknev,$parameter){  
    $con;  
    dbconnect($con);  
    $ellenorzes = new funkciok();  
    $ellenorzes->sql_tamadas($modositandotermeknev);  
    $ellenorzes->bemenet_ellenorzes($modositandotermeknev);  
    $ellenorzes->sql_tamadas($parameter);  
    $ellenorzes->bemenet_ellenorzes($parameter);  
  
    if($parameter == 0)  
        $sql = "UPDATE termeklap SET aktive = 1,darab = 1 WHERE termekcim LIKE  
        '$modositandotermeknev'";  
  
    if($parameter == 1)  
        $sql = "UPDATE termeklap SET aktive = 0,darab = 0 WHERE termekcim LIKE  
        '$modositandotermeknev'";  
  
    $result = $con->query($sql);  
    if($result)  
    {  
        if($parameter == 0)  
            $eredmeny = "Sikeresen áthelyeztük a terméket az aktív termékek  
            közé,és 1 db-ot tettünk elérhetővé!";  
  
        if($parameter == 1)
```

```

        $eredmeny = "Sikeresen áthelyeztük a terméket az archívumba,és a
        darabszámát nullára csökkentettük!";
    }
    else
    {
        $eredmeny = "Sikertelen áthelyezés!";
    }

    $am = new AdminNezet();
    $am->AdminAICsakTermekListazo($parameter,$eredmeny);
    $con->close();
}

```

6.4. Funkciók osztály

A `bemenet_ellenorzes($input)` függvény a nem kívánatos támadásoktól tisztítja meg a neki átadott paramétert. Levágja a szövegből a space karaktereket, a vezérlő karakterektől is megtisztítja, és a HTML vezérlő karakterektől is.

A `jelszo_vedelem($input)` eljárás az `sha1` php függvény segítségével kódolja a neki átadott jelszót.

Az `sql_tamadas($input)` az úgynevezett 'SQL Injection' elleni támadásoktól tisztítja meg a neki átadott paramétert.

A `fajl_feltolto()` metódus a php `$_FILES` szuperglobális változók segítségével tölt fel a szerverre fájlokat, melyben egy hibaellenőrzés is található. A fájlok feltöltéséhez szükséges két könyvtár létrehozása a szerveren. A két könyvtár neve : 'konyvtar' és 'ujkonyvtar'. Ezek nélkül nem tudnak feltölteni a fájlok. A feltöltött fájlok egyébként további almappákba kerülnek az 'ujkonyvtar' mappán belül, mely mappák nevei az adott termék feltöltött ID-jai lesznek. A függvény egyébként egy átirányító fájlból hívódik meg, a **termekbeszuras.php**-ból, és csak akkor tud működni ha POST-on kapja az általa várt azonosítót, melyet azonban a függvényben ellenőrzök külön, nem az átirányító fájlban. A függvény egy tömbbel tér vissza, melyekben a fájlok elérési útjai vannak tárolva. Ezt adjuk át a **termekbeszuras.php** fájlban a terméket az adatbázisba beszűrő függvénynek. Fontos még itt megemlíteni, hogy a fájlok törlésére nincs beépítve funkció a web-áruházban. Azért nincs, mert a termékek, mint archívum tulajdonképpen elérhetőek maradnak, bár megvásárolni azokat már nem lehet. A fájlok törlése esetén a képek nem jelennek meg.

6.5. Weboldal osztály

A weboldal.php **Weboldal** osztálya a felhasználói felületet megjelenítő függvényeit tartalmazza.

A tagváltozóknak lehet módosítani az információs oldalon megjelenő adatokat, az üdvözlő szöveget, és a gombsort is mely egy asszociatív tömb.

Ebben az osztályban külön függvényekben hoztam létre a termékeket megjelenítő függvényeket. Külön a fő oldalon megjelenítőt: kistermeksav(\$tomb) és külön a teljes terméklistát megjelenítő függvényt: a Nagytermeklap(\$termekid)-t. Később ezeket több helyen is felhasználom a kódban.

Külön függvényben kaptak helyet a regisztráció nélküli vásárláshoz szükséges űrlapok megjelenítéséért felelős kódok, és a regisztrációhoz szükséges űrlapokat megjelenítő függvények is. Ezen felül a dizájnért felelős kódok is itt vannak, és a kosárral kapcsolatosak is.

Az egyik fő függvény a Megjelenites(\$visszajelzes="", \$aktid="", \$kategid="").

A függvény tervezésekor fontos szempont volt, hogy több dolgot is képes legyen elvégezni. Ennek megfelelően az eljárás megjeleníti a fejléct, lábléct, a kosarat, a gombokat és a kapott paramétertől függően más-más oldalt. Ha a \$visszajelzes változó kap egy értéket, annak függvényében jeleníti meg az adott művelet végrehajtásával kapcsolatos visszajelzést a felhasználó felé. Ha alapértéket kap, úgy nem jelenít meg visszajelzést. Az \$aktid változó amennyiben a függvényből alapértéket kap, úgy a fő oldalt jeleníti meg, mely 20 terméket listáz ki, véletlenszerűen egy SQL lekérdezés és tömbrendezés után. Ha az alapértéktől eltérő értéket kap, az azt jelenti, hogy valaki egy teljes termék adatlapját óhajtotta megnézni. Ez esetben az ezt megjelenítő függvény töltődik be. Ha a \$kategid változó kap egy az alapértelmezettől eltérő értéket, akkor egy kategóriára kattintott a felhasználó, és a listázott termékek csak az adott termékkategóriából lesznek láthatóak.

6.6. FelhasznaloiDbMuveletek osztály

A weboldal.php fájlban található ez az adatbázissal dolgozó osztály. A függvények megvalósításánál igyekeztem olyan függvényeket létrehozni, amelyek az adatbázison hajtanak végre módosításokat és megjelenítéssel nem foglalkoznak. Függvények foglalkoznak a regisztrált vásárlásokkal és a nem regisztrált vásárlásokkal, a termékek aktivitását változtató függvényekkel, a regisztrált felhasználók és a nem regisztrált felhasználók adataival dolgoznak, és a termékekhez kapcsolódó kategóriákkal dolgozó kódok is ide kerültek.

Az egyik általam bemutatni kívánt függvény a Belepett felhasznalo modositasa() függvény, mely paraméteren a belépett felhasználó adatait kapja meg. Egy átirányító fájl hívja meg ezt a függvényt, ahol a POST-on érkezett adatokat vesszük át. Amennyiben ebben a fájlban (regisztral-modosit.php) egy bejelentkezett felhasználótól érkezünk, akkor az adatok módosítására készülünk. Ha pedig nincs bejelentkezve, akkor regisztrációra készülünk, de ezt nem ez a függvény végzi el. A függvény elején vizsgáljuk, hogy valóban a paraméteren kapott felhasználói név adatait kérdezzük-e le, ezt biztonsági okokból építettem be. A függvény, következő lépésben azt vizsgálja, hogy a két módosításra átadott jelszó egyforma-e. Ha nem akkor azt visszajelzésként jelezzük a felhasználó számára. Ez után vizsgálat alá vesszük, hogy a felhasználónév, amit módosításra átvettünk, nem-e foglalt már. Amennyiben foglalt, ezt is hibaként jelezzük a felhasználónak. Amennyiben nem foglalt, úgy végrehajtjuk az SQL lekérdezést, és ha ez is sikeres volt, akkor a \$_SESSION tömb [felhasznalo_nev] elemét egyenlővé tesszük a módosított névre, hogy a továbbiakban is működjenek a kódnak azon részei ahol a \$_SESSION tömbbel dolgozunk. A függvény végén átadjuk a vezérlést a megjelenítő függvényeknek.

7. FELHASZNÁLÓI FELÜLETEK

A felhasználó felület megtervezésekor egy egyedi megjelenésű oldal képét gondoltam el. Külön szerettem volna venni az adminisztrátor felületet és a felhasználó felületet. Amennyiben adminisztrátor lép be, úgy őt egyből az 'admin' felületre irányítjuk. Ha felhasználó lép be, akkor ő az alap felhasználói felületet látja. A formázásnál főképp egyedi megoldásokat alkalmaztam a CSS segítségével. A felhasználói oldal megjelenését is egyedinek terveztem. A vonalakra, elhatároló területekre világos színeket készítettem. A gombokat, és a különféle műveletek után a visszajelzéseket a BOOTSTRAP osztályai segítségével jelenítem meg. Ezeknek a visszajelzéseknek a helye a fejlécben került kialakításra. Minden végrehajtott akcióról visszajelzést kap a felhasználó.

7.1. Adminisztrátor felület

Az adminisztrátor felület kialakítását egy egyszerű megjelenésű oldallal valósítottam meg. Az oldal tetején látható a főcím. Alatta a bejelentkező sáv, amelyben a belépett adminisztrátor felhasználó nevét is kiírjuk, és üdvözljük őt. Itt is kialakításra került egy visszajelző sáv, mely az adminisztrátor felhasználókat értesíti szöveges üzenetben az általuk végrehajtott műveletekről.

A felületen haladva lefelé, eltérő gombsorok következnek. A különféle csoportokhoz tartozó műveletek gombjai egy sorban találhatóak.

Így fentről lefelé a következők műveletcsoportok vannak :

- Felhasználók,
- Termékek,
- Kategóriák,
- Archívum,
- Eladott termékek

Az ADMIN felület terve, egyszerűsítve:

Gyűjtők Boltja WebShop

Üdvözöllek : felhasználónév

Felhasználók (műveletek,gombok)

Felvitel

Törlés/Módosítás

Termék (műveletek,gombok)

Felvitel

Törlés/Módosítás

Összes termék listázás

Kategória (műveletek,gombok)

Felvitel

Módosítás

Archívum (műveletek,gombok)

Összes megtekintése

Eladott termékek (műveletek,gombok)

Eladások megtekintés

Nem reg.elt eladások

Navigáló gombok (műveletek,gombok)

Webshop Főoldal

Kilépés

Admin főoldal

Aktív termékek megjelenítésének terve:

Gyűjtők Boltja WebShop

Üdvözöllek : felhasználónév. Aktív termékek listázása:

VISSZAJELZŐSÁV

Termék cím:

Termék cím:

Termék cím:

Termék cím:

Termék cím:

Termék cím:

Termék cím:

Termék cím:

Termék archivba tétele

Képek:

Kép 1.

Kép 2

Kép 3

Navigáló gombok (műveletek,gombok amelyek a termékekhez kapcsolódnak)

Admin Főoldal

Termék felvétel

Termék törlése

Kilépés

Eladások megjelenítésének terve:

Gyűjtők Boltja WebShop

Üdvözöllek : felhasználónév. Eladások termékek és vevők adatai:

VISSZAJELZŐ SÁV

A vásárlás adatai:

Adott termék adatai:

Cím:

Ár:

Feltöltés dátuma

Kép I.

Kép II.

Kép III.

Adott vásárló adatai:

Felh. ID:

Felh. neve:

Felh. vezetéknéve:

Felh. keresztnéve

Cím:

Telefon:

Email:

Reg. dátuma:

Navigáló gombok

Webshop főoldal

Kilépés

Admin főoldal

7.2. Felhasználói felület

Amikor a böngészőből felkeresi a látogató az oldalt, erre a felületre jut. Itt kell belépni az oldalra, és innen jelenik meg az adminisztrátor felület, ha a belépett 'user' adminisztrátor. Amennyiben nem akkor jobb oldalon fenn az 'ADMIN OLDAL' gomb nem jelenik meg!

Szintén az oldal jobb felső oldalán van a 'kosár' gomb, melybe minden a 'kosárba' tett termék bekerül, és annak a számlálóján láthatjuk, hogy hány terméket tettünk eddig be. Jobb oldalon végig a kategóriák láthatóak, amikre kattintva az adott kategória termékeit listázzuk ki.

Amennyiben nincs bejelentkezve felhasználó, úgy a bejelentkezési mező látszódik a 'Kijelentkezés' gomb felett.

35

Ez alatt látható a gombsor. A 'kezdőlap' a 'fő' oldal szerepét tölti be. A 'regisztrál/adatmódosítás' gomb kettős funkciót tölt be: belépve az adatainkon tudunk módosítást végrehajtani, nem belépve pedig regisztrálni tudunk.

Az 'információk' fül alatt az áruház általános szabályai olvashatóak.

Ezek alatt kezdődnek a listázott termékek megjelenítése. Minden termék esetén a képre kattintva egy új ablakban nyílik meg a fotó ahol teljes méretben is meg lehet tekinteni. A 'kosárba' tesz gombra kattintva a termék a kosárba kerül, és visszajutunk a főoldalra. A 'teljes terméklap' gombra kattintva egy részletes adatlap jelenik meg a feltöltött termékekről. Itt a teljes leírás is látható.

A felhasználói felület megjelenítési terve:

Az admin oldal gomb irányít át az admin felületre, ha a belépett felhasználó admin jogosultsággal rendelkezik

A visszajelző sávban informáljuk a felhasználót a lezajlott műveletekről

The wireframe shows a web application layout for 'A Gyűjtők Szentélye'. At the top is a blue header with the site name and an 'Admin Oldal' button. Below the header is a light gray bar with a welcome message and a 'Felhasználó' label. A blue 'visszajelző sáv' (notification bar) is positioned below the welcome message. A navigation bar contains buttons for 'Kijelentkezés', 'Kosár(1)', 'Kezdőlap', 'Regisztrál/Adatmódosítás', 'Információk', and 'Keresés'. The main content area is divided into two columns. The left column displays two product cards, each with a title, price, availability status, keywords, and three image placeholders. The right column lists categories from 'Kategória I.' to 'Kategória IV.'. At the bottom is a light gray footer labeled 'Lábléc'. Two red arrows point to the 'Admin Oldal' button and the notification bar.

A Gyűjtők Szentélye Admin Oldal

Üdvözöllek a gyűjtő szentélyében kedves : **Felhasználó**

visszajelző sáv

Kijelentkezés Kosár(1)

Kezdőlap Regisztrál/Adatmódosítás Információk Keresés

Termélcím
Ár
Elérhető
Kulcsszavak: Kép I. Kép II. Kép III.
Terméklap Kosárba teszem!

Termélcím
Ár
Elérhető
Kulcsszavak: Kép I. Kép II. Kép III.
Terméklap Kosárba teszem!

Kategóriák:
Kategória I.
Kategória II.
Kategória III.
Kategória IV.

Lábléc

8. TESZTELÉS

A tesztelést két féle módon végeztem el az alkalmazásban. A fejlesztés során, folyamatosan manuálisan teszteltem minden bekerülő új opciót. Ezek tulajdonképpen komponens és integrációs tesztek voltak. Ezekről vártam, hogy elkerüljem a nagyobb hibákat, és azt hogy később ezek óriási hibahalmazzá nőjenek. A második módszer a fejlesztés befejezésével történt, amikor már a kód készen volt. Minden fontos funkciót leteszteltem, egyenként végigmenve rajtuk, és vizsgáltam, hogy a várt eredményt kaptam-e vissza, ez volt tulajdonképp a rendszerteszt.

A fejlesztési és tesztkörnyezet:

- Google Chrome böngésző: 87.0.4280.66 (Hivatalos verzió) (64 bites)
- Operációs rendszer: Windows 10 Pro (64 BIT)
- Processzor : Intel® Core™ i3-4130 CPU @ 3.40GHZ
- Memória: 8 GB.
- A felületek 1920*1080 pixelen vagy attól nagyobb kijelzőkön optimálisak.

8.1. Admin felület: Felhasználó Felvitel

Tesztelés leírása (1) : Próbá a 'tesztuser' felhasználó nevű user beszúrására.

Beviteli mezők 'teszt' értéket kapnak vagy a HTML validáció által elvárt adatokat és a jogosultság mezőben 3-at. (Mezők: string-ek, vagy számok, vagy '@'-al írt email cím.)

Várt eredmény: Az adatok bekerülnek az adatbázisba, lehetséges bejelentkezni felhasználóként.

Kapott eredmény: Az adatok helyesen bekerültek az adatbázisba, a bejelentkezés lehetséges.

Tesztelés leírása (2): Próbá a 'tesztuser' felhasználó nevű user ismételt beszúrására!

Beviteli mezők 'teszt' értéket kapnak vagy a HTML validáció által elvárt adatokat és a jogosultság mezőben 3-at.

Várt eredmény: Hibajelzés hogy van már ilyen nevű felhasználó.

Kapott eredmény: A hibát visszakapjuk, tehát azonos felhasználó nevű user-t nem lehet hozzáadni.

Tesztelés leírás (3): Próbá a 'tesztadminisztrator' felhasználó nevű user beszúrására.

Beviteli mezők 'tesztadminisztrator' értéket kapnak vagy a HTML validáció által elvárt adatokat és a jogosultság mezőben 1-et állítva.

Várt eredmény: Az adatok bekerülnek az adatbázisba, és lehetséges ADMIN bejelentkezés.

Kapott eredmény: Az adatok helyesen bekerültek az adatbázisba, majd kijelentkezés után lehetséges ADMIN-ként bejelentkezni.

Tesztelés leírás (4): Próbá az 'ujadminisztrator' felhasználó nevű user beszúrására.

Beviteli mezők 'ujadminisztrator' értéket kapnak vagy a HTML validáció által elvárt adatokat és a jogosultság mezőben 2,4,5,6,7,8,vagy 9-et állítva.

Várt eredmény: Hibaüzenet hogy ilyen jogosultsági szintű felhasználót nem lehet beszúrni.

Kapott eredmény: Az adatok bekerülnek az adatbázisba!

Hiba javítása: Az 'Adminosztály' felhasznalo_beszuras() függvényben kell, egy vizsgálat mely ellenőrzi, hogy a jogosultsági szint csak 1 vagy 3 lehessen. A vizsgálatot beépítettem.

A másodszori teszt eredmény : Hibát kapunk, hogy a beszúrás nem lehetséges.

8.2. Admin felület: Felhasználó Törlés/Módosítás

Teszteset leírás(1): A 'tesztadminisztrator' nevű felhasználó adatainak módosítása.

Beviteli mezők 'modositotttesztadminisztrator' értéket kapnak vagy a HTML validáció által elvárt adatokat.

Várt eredmény: Az adatok módosulva bekerülnek az adatbázisba.

Kapott eredmény: Az adatok bekerültek az adatbázisban!

Teszteset leírás(2): A korábban felvitt 'modositotttesztadminisztrator' felhasználót név alapján megkeressük, majd töröljük a törlés gombra kattintva.

Várt eredmény: Visszajelzés a sikeres törlésről.

Kapott eredmény: A visszajelzés sávbán értesülünk a sikeres törlésről. Az adatbázisból is eltűnt a felhasználó.

8.3. Admin felület: Termék Felvitel, Törlés/Módosítás

Teszteset leírás: A termék paramétereinek létrehozott beviteli mezőket kitöltjük HTML validáció alapján (3 jpg fájlt feltöltve!) érvényes adatokkal, és a termék feltöltése gombbal elérhetővé tesszük azt az áruházban.

Várt eredmény: Visszajelzés szöveges üzenetként hogy a termékbeszúrás sikeres volt.

Kapott eredmény: A visszajelzés megjelent, a termék feltöltődött.

Teszteset leírás(2): A tesztet I. szerint ugyanúgy valid adatokkal töltjük fel a beviteli mezőket, de csak 1 , majd 2 fényképet – kép fájlt töltünk fel a termékhez.

Várt eredmény: Visszajelzés érkezik szöveges üzenetként hogy sikeres volt a termékbeszúrás.

Kapott eredmény : Mindkét esetben a várt eredmény teljesült.

Teszteset leírás(3): A korábban felvitt (termékteszt1) nevű termék adatait módosítjuk a HTML validáció által lehetővé tett módon, majd végrehajtjuk.

Várt eredmény: A termék adatainak módosítása megtörténik az adatbázisban, amiről értesítjük a felhasználót.

Kapott eredmény: Az elvárt eredmény teljesül.

Teszteset leírás(4): A korábban felvitt teszt terméket törölni kívánjuk az adatbázisból.

Várt eredmény: A termék és adatai véglegesen törlődnek az adatbázisból!

Kapott eredmény: A termék és adatai (kivéve a szerveren a kép fájlok) törlődtek.

8.4. Admin felület: Kategória: Felvitel / Módosítás

Teszteteset leírás: A HTML validáció alapján érvényes nevet adunk meg kategórianévnek.

Majd ugyanazt a kategória nevet újra!

Várt eredmény: A bevitel után a kategória megjelenik az adatbázisban és a felületen is. Majd a másik bevitel esetén hibát kell kapnunk szöveges üzenetben, melyben jelezzük, hogy ugyanolyan név nem szerepelhet.

Kapott eredmény: A kategória létrejött. Második próbálkozás során a kategória nem került be újra az adatbázisba!

8.5. Felhasználói felület: Regisztrál / Belépés

Teszteteset leírás: Kísérletet teszünk egy felhasználó regisztrálására. A beviteli mezőkben csak a HTML validáció által megengedett adatokat vihetünk be.

Várt eredmény: A felhasználó adatai bekerülnek az adatbázisba. A felhasználói felületen pedig a regisztrálni kívánt user egyúttal bejelentkezve is marad.

Kapott eredmény: A várt eredmény teljesült.

Teszteteset leírás (2): Regisztráció során a felhasználó jelszó mezőjébe két eltérő jelszót adunk meg!

Várt eredmény: A regisztráció sikertelen, jelezzük a felhasználónak, hogy nem adhat meg eltérő jelszavakat!

Kapott eredmény: A regisztráció sikerül, és az első jelszót fogadja el a rendszer validnak.

Hiba javítása: A javítás során az adott függvényben a paraméteren érkező jelszavakat összehasonlítjuk, és vizsgálatot hajtunk rajtuk végre. Ha nem egyeznek a jelszavak, hibajelzést adunk.

Újratesztelés eredménye: A regisztrációt nem engedi véghezvinni a rendszer, hibajelzést ad.

Teszteteset leírás (3): A regisztráció során a felhasználónévnek egy már létező felhasználónevet adunk meg.

Várt eredmény: A regisztráció nem sikerül, mert azonos felhasználónevű user-ből nem lehet kettő.

Kapott eredmény: A várt eredmény.

8.6. Felhasználó felület: Adatmódosítás

Teszteteset leírás: A regisztrált felhasználó adatait átmódosítjuk a HTML validáció által engedélyezett adatkora.

Várt eredmény: A felhasználó adatai átmódosulnak, és a sikeres műveletről értesítjük a felhasználót.

Kapott eredmény: A várt eredmény.

Teszteteset leírás(2) :A regisztrált felhasználó adatai között a jelszó mezőket módosítjuk.

Várt eredmény: Két eltérő jelszó bevitele esetén jelezzük ezt hibaüzenetként. Azonosan megadott két jelszó módosítás esetén, értesítjük a felhasználót a jelszómódosítás sikerességéről.

Kapott eredmény: Azonos és eltérő jelszó bevitel esetén is sikeres adatmódosítási visszajelzést kapunk.

Viszont amennyiben kiléptünk, majd visszalépünk, a rendszer már nem enged belépni!

Hiba javítása: A már regisztrált felhasználó adatainak módosításakor az átirányító fájl módosító kód részében volt egy újabb jelszókódolás (sha1 függvény), mely a már hash-elte jelszót újra hash-elte! Ezért módosítás után nem lehetett belépni. Akármilyen adatmódosítást hajtottunk végre, a jelszó újra hash-elődött.

Újratesztelés: Az adatmódosítás után kiléptünk, majd engedett belépni.

8.7. Felhasználó felület: Kosárba tétel / Kosárból törlés

Tesztelés leírása: Regisztrált felhasználó a:

- Kosárba helyez/kivesz egy terméket.
- Kosárba helyez/kivesz több terméket.
- Kosárba helyez több azonos terméket.
- Kosár teljes tartalmát törli.

Várt eredmény: A kosár gombon a termékszámoló a megfelelő számú terméket tartalmazza minden opciónál. A kosár megjelenítésére kattintva minden beletett/kivett tétel megjelenjen/eltűnjön és a végösszeg számoló a megfelelő összeget mutassa. Teljes törlésnél minden törlődjön. Több azonos terméket hibaüzenettel nem enged a kosárba tenni.

Kapott eredmény: Minden opciónál az elvárt.

8.8. Felhasználó felület: Megrendelés véglegesítése

Tesztelés leírása: A kosárba tett terméket / termékeket a megrendelés véglegesítése gomb megnyomásával megvásárolni.

Várt eredmény: A felhasználót szöveges üzenetben értesítjük a felhasználó vásárlásának sikerességéről, kiírjuk a vásárlás teljes összegét számára.

Kapott eredmény: Az elvárt eredmények.

Tesztelés leírása(2): A kosárba tett terméket időközben megvásárolták. Ezt azzal szimuláljuk, hogy az adatbázis megfelelő mezőit úgy módosítjuk a megrendelés véglegesítése gomb lenyomása előtt, hogy a kosárba tett terméket már nem lehet elérni.

Várt eredmény: A felhasználót szöveges üzenet értesíti, hogy az általa megvásárolni kívánt termék már nem elérhető, de annak az adatait is kiírjuk.

Kapott eredmény: Az elvárt eredmények.

8.9. Felhasználói felület: Keresés a keresőoldaltól, eladottak megjelenítése

Tesztelés leírása: A keresés fülön a belépett felhasználó tud keresni az eladott termékek között az archívumban.

Várt eredmény: A korábban megvalósult vásárlás megjelenik a keresőmezőben az eladott termék keresésénél.

Kapott eredmény: Az elvártak megfelelően.

8.10. Felhasználói felület: Bejelentkezés nélkül, keresés, eladottak megjelenítése

Teszteteset leírása: Látogatóként kell vizsgálni, belépés nélkül, hogy megtekinthetők e az eladott termékek.

Várt eredmény: Hibaüzenetben értesítjük a felhasználót, hogy ehhez be kellene jelentkeznie.

Kapott eredmény: A várt eredmény.

8.11. Felhasználói felület: Bejelentkezés nélkül, Admin felület elérése

Teszteteset leírása: Megpróbáljuk az adminisztrátor felületet elérni a munkafájlok neveit beírva a böngészőbe.

Várt eredmény: Nem jelenik meg az adminisztrátor felület, vagy értesítjük, hogy nincs rá jogosultsága.

Az index.php és kategszures.php, kereses.php, kontakt.php, kosarmegjelenites.php , logout.php, megrendelesvegleg.php, regisztracio.php, teljestermeklap.php, vasarlas.php fájlok esetén nem várjuk ezt! Ott vagy üres oldalt kapunk vagy visszatöltődik az oldal az alap megjelenítést szolgáltatva.

Kapott eredmény: A nem felsorolt fájlok esetén szöveges üzenetet kapunk, hogy nincs jogosultság megtekinteni a kért fájlt.

Kapott hibás eredmény: A vasarlas.php fájlnál hibát kaptunk: Undefined variable több változóra.

Hibajavítás: A fájlban egy vizsgálat igaz ágának zárását rossz helyen tettük meg. Átjavítottuk a megfelelő helyre.

Újratesztelés: Ennél a fájlnál is visszatöltődik az oldal bármiféle eredmény nélkül, az elvártak megfelelően.

8.12. Felhasználói felület: Bejelentkezés nélkül, Kosárba tétel / Kosárból kivét

Teszteteset leírása: Bejelentkezés nélkül is kell tudni a kosárba tenni és törölni terméket. Az esetek, mint a 8.7-es pontban.

Várt eredmény: Mint a 8.7-es pontban.

Kapott eredmény: A várt eredmények

8.13. Felhasználói felület: Bejelentkezés nélküli megrendelés

Teszteteset leírása: A kosárba helyezett termékeket a megrendelés véglegesítés gombbal lehet megvásárolni, miután a nem regisztrált felhasználó adatait bekértük, majd értesítjük szöveges üzenetben a sikerességről.

Várt eredmény: A vásárlásról megjelenik a tájékoztató üzenet, és az adatbázis megfelelő tábláiba bekerülnek a kívánt adatok.

Várt eredmény: A teszteteset leírása szerint.

Kapott eredmény: A várt eredmények.

9. FELHASZNÁLÓI DOKUMENTÁCIÓ – A PROGRAM ISMERTETÉSE

Ebben a részben bemutatom a programot felhasználói oldalról és az adminisztrátor oldaláról. A tervezés során próbáltam minél egyértelműebben létrehozni mindent, hogy a felhasználó ne érezze magát labirintusban.

Próbálom az oldalról vett képekkel segíteni a tájékozódást az oldalon

9.1. Felhasználói felület: Regisztrál / Adatmódosítás – Jelszóegyeztetés

A főoldalra érkezve a látogatónak lehetősége van regisztrálni. A menüsorban látható 'regisztrál/adatmódosítás' gombra kattintva érhető el a regisztrációs felület. Itt 12 beviteli mezőt lehet látni, melyeket minden esetben érvényes adatokkal kell kitölteni. Amennyiben a mezőkre van kattintás úgy azok széle vastagabb kékkel látszik.

Elvárt, hogy minden mezőben minden adatot ki kell tölteni. A mezők felett látható hogy melyik mezőbe mennyi karaktert lehet beleírni.

Fontos hogy később az itt megadott *felhasználói névvel és jelszóval* lehet belépni.


Amennyiben a *két jelszó mezőbe* eltérő jelszavakat visznek be, úgy a regisztráció sikertelen lesz melyről hibaüzenetet is kapunk.

Az *email cím mezőben* is érvényes adatok kell bevinni, fontos hogy a '@' nélkül nem enged tovább a rendszer.

A *regisztrál/adatmódosítás gombra* kattintva az adatok bekerülnek az adatbázisba, és innentől be vagyunk jelentkezve. A hibaüzenet, illetve minden a műveletekkel kapcsolatos esemény tájékoztatását a gombsor feletti úgynevezett visszajelző sávban kapjuk meg szöveges értesítés formájában.

Ha már belépett felhasználóként kattintunk ide, akkor az adatainkat tudjuk módosítani. A beviteli mezőkben a korábban bevitt adatok jelennek meg. Ha bármit módosítunk és a lap alján lévő módosítás gombra kattintunk, akkor az átírt mező adatai kerülnek be az adatbázisba, és máris kész a módosítás. Módosítás esetén sem lehet eltérő jelszavakat megadni. Ezt a rendszer visszautasítja, és a korábbi belépő jelszó marad érvényben.

Név és jelszó mezők a bejelentkezéshez:



Név: Jelszó: Bejelentkezés KOSÁR(0)




[Kezdőlap](#) [Regisztrálás/Adatmódosítás](#) [Információk](#) [Keresés](#) [Kijelentkezés](#)

Commodore 64 Számítógép

15000 Ft.

Elérhető: 1 darab.


Kulcsszavak: Retrő,C64,Számítógép



Kategóriák:

- Bakelit lemez
- Dedikált könyvek
- Egyéb gyűjtői tárgyak
- Fénykép
- Festmények
- Filatélia
- Filmplakát

Beviteli mezők a regisztrációnál:



Név: Jelszó: Bejelentkezés KOSÁR(0)

[Kezdőlap](#) [Regisztrálás/Adatmódosítás](#) [Információk](#) [Keresés](#) [Kijelentkezés](#)

Kategóriák:

- Bakelit lemez
- Dedikált könyvek
- Egyéb gyűjtői tárgyak
- Fénykép
- Festmények
- Filatélia
- Filmplakát
- Különleges tárgyak
- Numizmatika
- Órák
- Papírpénz

Vezetéknév: (Max. 50 kar.)

Keresztnev: (Max. 50 kar.)

Felhasználónév: (Max. 30 kar.) Ezzel a névvel és a jelszóval lehetséges a bejelentkezés.

Ország: (Max. 40 kar.)

Jelszó: (Max. 40 kar.)

Jelszó újra:

Irányítószám: (Max 10 kar.)

Város: (Max 50 kar.)

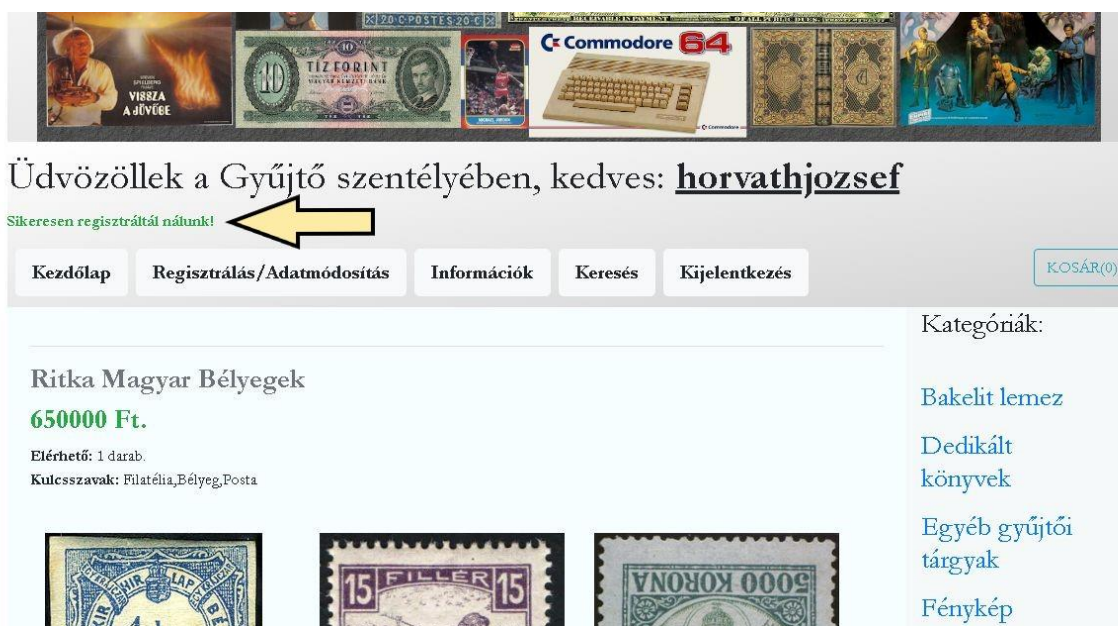
Utca/Község: (Max. 60 kar.)

Házszám/Emelet/Ajtó: (Max. 30 kar.)

Telefonszám: (Max 20 kar.)

Email cím: (Max 80 kar.)

Visszajelző sáv:



9.2. Felhasználói felület: Keresés gomb

A kereső mezőkben az éppen elérhető termékek között lehet keresni. Amennyiben regisztrált felhasználó lép be, úgy a nem elérhető (eladott) termékek között is keresni. Nem regisztrált felhasználó nem kereshet itt! Összesen négy beviteli mező van itt, és két választható elem. A listázott találatok a kereső mező alatt jelennek meg.

A termékcím kereső mezőben a keresés megtalál töredék szavakat, részleteket is. Így pl. ha a mezőbe a 'birodalom' szót írjuk, találatokat fogunk kapni a szó belső töredék részeire is. Illetve ha pl. 'irodalom' szót írunk be, akkor a 'birodalom' szót is megtalálja, de ha csak azt írjuk be, hogy, 'dal', akkor 'birodalom' szót is megtalálja, de a 'fogadalom' szót is. Tehát a beírt kifejezés előtt és után is lehet tetszőleges karakter.

A keresés ár alapján minimum illetve a maximum esetén is kiadja a vele egyenlő összegű találatot is. Intervallum keresés a kettő egyszerre való használatával nem lehetséges.

A kulcsszó mező keresés működési elve teljesen azonos a termékcímmel, de itt a terméknel megadott kulcsszavak között lehet keresni.

Kategória keresés alapján az összes az adott kategóriában található terméket megtalálhatjuk.

Az aktív és eladott termékek közötti választásról már volt szó. Működési elve összefügg a többi mezővel, ugyanis ezt szűrőként lehet alkalmazni a többi mezőre. Így bármelyik korábban bevitt mezőbe írt értékre kereshetünk, amennyiben 'eladott'-ra van állítva ez a kapcsoló, úgy a már eladott termékek között lehet keresni. Aktív esetén a még elérhető termékekre.

Keresősávok:

Nem írtál be semmit a keresőbe!

Kezdőlap Regisztrálás/Adatmódosítás Információk **Keresés** Kijelentkezés KOSÁR(0)

Részletes keresés a termékek között:

Termék cím alapján:

Keresés ár alapján: Min: 0 Max:

Keresés kulcsszó alapján:

Keresés kategória alapján:

Keresés az eladott / nem eladott termékek alapján:

Kategóriák:

- Bakelit lemez
- Dedikált könyvek
- Egyéb gyűjtői tárgyak
- Fénykép
- Festmények
- Filatélia
- Filmplakát
- Különleges tárgyak
- Numizmatika
- Órák
- Papírpénz
- Régi

9.3. Felhasználói felület: Kezdőlap, Információk gomb, Kijelentkezés gomb

A főoldalon még ez a két gomb látható. Az információ gombra kattintva a web-shopról található általános információkat találjuk meg. A kezdőlap gomb minden más oldalról a fő megjelenítési oldalra lép át.

Amennyiben a felhasználó be van lépve, a visszajelző sáv alatt és a gombsor felett egy külön helyen jelenik meg a kijelentkezés gomb. Minden esetben érdemes kilépni az alkalmazásból, ha befejeztük a munkát. Kilépés után is megmarad a kosár tartalma!

9.4. Felhasználói felület: Kosár gomb

A kosár gomb a látható felület mindig jobb felső oldalán látható. A benne lévő szám azt jelzi, hogy éppen hány termék van a kosarunkban. Alapesetben 0 van benne, és a vásárlás befejezése után is 0 lesz benne.

A 'kosárba tesz' gombra kattintva kerül bele a kosárba az adott termék, és minden egy darab kosárba tett termékkel 1-el nő a kosár melletti szám.

Fontos tudni, hogy minden termékből csak 1 db-ot lehet a kosárba tenni. Ha többször akarjuk beletenni akkor a visszajelző sávban értesítjük erről a felhasználót. Ennek az az oka, hogy a web-áruház felépítése egyedi termékekre van felépítve, olyan tárgyakra, amikből csak 1-1 darab van. Természetesen előfordulhat olyan is hogy több darab elérhető egy termékből, de ez esetben be kell fejezni a vásárlást, majd egy új

vásárlásban újra beletenni egy terméket a kosárba és ismét befejezni a vásárlást. Kosárba tétel esetén is visszajelzést kell kapnunk annak sikerességéről.

A Kosár gombra kattintva megnyílik a kosarunk, és láthatóvá válik, hogy mit tettünk eddig bele. Kilistázva látszik minden fontos információ, és minden terméknel két plusz gomb jelenik meg: A 'terméklap megtekintés' és a 'tétel törlése a kosárból' gomb. Ezeken felül alul állandó gombként látszik a 'megrendelés véglegesítése' és a 'kosár teljes ürítése gomb'.

A 'terméklap megtekintés' gomb átirányít egy újabb oldalra, ahol a termék összes adata megtalálható. Amennyiben hosszú leírás készült a termékleírása mezőben, akkor itt mindent megjelenít abból. Azért volt szükség erre, mert a termékek listázásánál széttördelné az oldalt a hosszú leírás.

A 'tétel törlése a kosárból' gomb kidobja a kosárból a valamiért már nem megvásárolni kívánt terméket, és újraszámítja a fizetendő végösszeget, és a kosár gomb mellett darabszámot is csökkenti.

A 'kosár teljes ürítése' gomb az összes tételt kidobja a kosárból, és minden számlálót (végösszeg és darabszám) nullára állít vissza.

Kosár gomb és kosár megjelenítése:

Üdvözöllek a Gyűjtő szentélyében, kedves: **horvathjozsef**

Kezdőlap Regisztrálás/Adatmódosítás Információk Keresés Kijelentkezés **KOSÁR(2)**

A kosarad tartalma:

Cím: Halálosztó Terminátor Filmes plakát, moziplakát
Termék ID: 328
Ár: 40000
Még elérhető darabok: 1
Kategória Neve: Reklámplakát

Tétel törlése a kosárból Terméklap megtekintés!

Cím: Nokia 3310-es Retrő Mobiltelefon
Termék ID: 348
Ár: 5000
Még elérhető darabok: 1
Kategória Neve: Reklámplakát

Tétel törlése a kosárból Terméklap megtekintés!

Fizetendő: 45000

Megrendelés véglegesítés! Kosár teljes ürítés!

Kategóriák:

- Bakelit lemez
- Dedikált könyvek
- Egyéb gyűjtői tárgyak
- Fénykép
- Festmények
- Filatélia
- Filmplakát
- Különleges tárgyak
- Numizmatika
- Órák
- Papírművészet

9.5. Felhasználói felület: Képek megjelenítése

A felhasználói felületen minden megjelenített képre kattintva egy külön ablakban megjelenik a kép nagyban is, melyre kétszer kattintva rá is tudunk nagyítani az adott képre.

9.6. Admin felület

Amennyiben adminisztrátorként lépünk be a felületre, egy teljesen más felületre jutunk be, mint ha nem adminisztrátorként lennénk be.

Alul a fő oldali gombsorban bal alul van egy kapcsoló gomb (Webshop Főoldal), mely a felhasználói oldalra irányít át.

Itt 5 fő csoport van, ahol eltérő gombok vannak. Minden gombra kattintva egy másik oldal tölt be, de alul egy másik gombsor is jelenik meg. Mindig az adott műveleti célcsoporttal kapcsolatos gombok jelennek meg, könnyítve a műveleteket.

A különféle beviteli mezőket nem lehet beírás nélkül használni. Ahhoz hogy tovább mehessünk, kell beírni információt. A beviteli mezők felett mindig közölve vannak annak a korlátai, hogy attól eltérő adatmennyiséget ne lehessen beírni.

A főcím alatt a belépett felhasználót üdvözlés látszik.

Minden művelet után értesítjük a visszajelző sávban az adminisztrátort arról, hogy az adott művelet sikeres volt e. Ez az értesítés a belépett felhasználó üdvözlő szövege alatt jelenik meg.

Adminisztrátor főoldal:



9.7. Admin felület: Felhasználó műveletek

Az adminisztrátor felületen a felhasználókkal kapcsolatban két fő művelet lehetséges. Felhasználó felvitele, és felhasználó Törlése/Módosítása.

A felvitel gombra kattintva egy űrlap jön be, ahol manuálisan is lehet felhasználót felvinni, aki aztán később be tud lépni, és tud vásárolni, illetve a saját adatait módosítani. Itt 12 beviteli mező jelenik meg. A korábban a regisztrációnál megismert megkötések itt is érvényesek, két eltérés van, méghozzá a 'jogosultság' mező és a jelszó mező.

Jogosultság mező:

Nagyon fontos tudni, hogy itt két értéket lehet adni. Az 1-es érték esetén adminisztrátor jogosultságot adunk. 3-as érték esetén egy sima felhasználót adunk az adatbázishoz, akinek nincs admin jogosultsága.

Alapértékként 3-as van beállítva. Minden ezektől eltérő szám megadása hibával jelez vissza.

Jelszó mező: A regisztrációval ellentétben itt nincs jelszó ellenőrzés, itt csak egy jelszót lehet megadni. Később a belépett felhasználó tudja ezt módosítani.

Felhasználó beviteli mezők:

Gyűjtők Boltja WebShop

Kedves: stamini! Új felhasználó felvitel menüpont.

Sikeres felhasználó beszúrás!

Jogosultság
1-es ADMIN jogok
3-as felhasználói jogok (KÖTELEZŐ)

Jelszó (Min. 5 - max. 40 karakter) Kötelező!

Vezetéknév (Min. 2 - max. 30 karakter) Kötelező!

Felhasználónév (Min. 2 - max. 50 karakter) Kötelező!

testuser

A következő gomb a felhasználókkal kapcsolatos műveletek. (Törlés/Módosítás)

Itt először egy keresőmező jelenik meg, hogy meg tudjunk találni egy adott felhasználót. Négy jellemzőre lehet itt keresni.

- Vezetéknév
- Város
- Email cím
- Felhasználó név

Miután megtaláltuk a keresett felhasználót, további két gomb jön fel: Felhasználó törlése, Az adatok módosítása! A törlésre kattintva végleg kitöröljük a keresett felhasználót a rendszerből.

Az adatok módosítása esetén megkapjuk az adatait, amiken módosítást lehet végrehajtani. Az adatok módosítása gombra kattintva a kívánt adatokra módosulnak az adatok. Itt **NEM** lehet a jogosultságot módosítani. Azt csak a felhasználó létrehozásakor lehet beállítani.

Amennyiben adminisztrátor jogosultságú felhasználót találunk, azokat nem lehet törölni biztonsági okokból. Ilyen szándék esetén forduljon a web-shop üzemeltetőjéhez.

Felhasználó kereső mezők:

The screenshot shows the 'Gyűjtők Boltja WebShop' interface. At the top, there's a header with the shop name. Below it, a message says 'Kedves: stamina Felhasználó törlés vagy módosítás menüpont:'. The main section contains four search filters, each with a text input field and a 'Keresés!' button: 'Keresés vezetéknév alapján', 'Keresés város alapján' (with 'pl. Budapest' entered), 'Keresés email cím alapján', and 'Keresés felhasználó név alapján'. Below these is a box for 'Felhasználónév: tesztelek' with a list of user details: Felh. ID: 261, Jogosultság: 3, Vezetéknév: Teszt, Keresznév: Elek, Lakhelye: Budapest, and Email címe: tesztelek@tesztelek.hu. At the bottom of this box are buttons for 'A felhasználó törlése' and 'Az adatok módosítását'. The footer contains navigation links: 'Admin Főoldal', 'Felhasználó módosítása/törlése', 'Felhasználó felvitele', and 'Kilépés', along with a timestamp: 'Ma 2020/11/22-e van. Az idő: 11:45:21pm'.

9.8. Admin felület: Termék Műveletek

Termék felvitele:

A termék felvitelénél öt mező van, egy elemválasztó, és egy fájlfeltöltő. A mezőknél a megkötések a mezőneveknél látszódnak. Ajánlott a termékcímet a lehető legbővebben kitölteni. A termék árát is meg lehet adni. Fontos hogy nincs megadva a pénznem, így akár más pénznemben is adhatjuk az árat. Ebben az esetben figyeljünk, hogy minden helyen következetesen azt a pénznemet használjuk, illetve az információknál közölni kell ezt a vevőkkel is. Alapesetben természetesen forintban számolunk, és az árak tartalmaznak áfát.

A termék leírása mezőben érdemes a termék jellemzőit bővebben megadni, illetve esetleg a vásárlással kapcsolatos általános információkat is lehet itt tárolni.

A fájlok kiválasztásánál 3 fájlt választhatunk ki. Ezeknek a fájloknak kép fájlformátumú fájloknak kell lenniük.

A fájlok méreteit 10 megabájtban korlátoztuk, ennél nagyobb fájl nem lehet feltölteni.

A kategória választóban a létrehozott kategóriákból lehet választani. A cél hogy a termékre minél jellemzőbb terméket találjunk.

A képek feltöltésével kapcsolatban még fontos tudni, hogy a tároló szerveren szükséges egy könyvtár és egy ujkönyvtár nevű mappa létrehozása. Az ujkönyvtár mappában kerülnek fel a termékekhez tartozó képek. Az almappák nevei egy számból állnak, ide kerülnek a hozzá tartozó kép fájlok. Sok termék esetén szükséges a fájlrendszer tulajdonságairól érdeklődni. A tárhely korlátjaira is szükséges figyelni a termékek feltöltésénél.

Termék törlése/módosítása:

Amennyiben egy adott termék adatait szeretnénk módosítani, úgy erre a gombra kell kattintani. Itt többféle kereső mezőben szükséges megkeresni az adott terméket. A listázott termékek feltöltés szerint növekvő sorrendben jelennek meg. További két gomb jelenik meg a listázott termékeknel.

'Az adatok módosítása' gombra kattintva lehetséges a termék adatainak módosítása. Figyelem, a termékhez tartozó képek és kategória módosítása nem lehetséges, így a felvitelnél körültekintően kell eljárni. Az adatok módosítás gombra kattintva a kívánt módosítások végrehajthatók.

'A termék törlése' gombra kattintva a terméket töröljük, végérvényesen az adatbázisból. A termékhez tartozó mappa és a képek fenn maradnak a szerveren.

Termékkereső/Módosító oldal:

Gyűjtők Boltja WebShop

Kedves: stamina! Termék módosítása oldal

Jelenleg : 36 termék elérhető.

Keresés termékcím alapján

Keresés termékár alapján

Keresés mennyiség alapján

Keresés feltöltés dátuma alapján (A keresett dátumtól régebbi dátum keresése!Formátum: '2020-07-13')

Termék címe:
'Magyar köztársasági Sportérmérem kitüntetés 1947'

- Termék ID : 311
- Aktív e : 1
- Termék ára : 120000
- Leírás : Magyar köztársasági Sportérmérem kitüntetés 1947
- Kulcsszavak : kitüntetés,érem,jelvény
- Darab : 1
- Felt. dátuma : 2020-09-07 21:30:40
- Numizmatika

Képek:



Admin főoldal Termék felvétele Termék törlése Kilépés

Ma 2020/11/23-e van.
Az idő: 12:13:10am

Az összes terméklistázás:

Erre a gombra kattintva az összes aktív terméket megjelenítjük. Itt egy fontos funkció érhető el: A 'termék archívba tétel' gombra kattintva a termék aktivitásának adatbázis mezőjét módosítjuk úgy, mintha a terméket eladtuk volna. Figyelem! Maga a termék ettől még nem lesz a vásárlások között, csupán az eladott termékeknel láthatjuk ezt. Fontos, hogy ha a termékből több darab is elérhető, akkor is 0 darabra állítja be ez a funkció a darabszámát!

9.9. Admin felület Kategória Műveletek:

A Kategóriák a termékekre jellemző, azokat legjobban meghatározó olyan kulcsszavak, amelyek gyűjtőfogalomként körülírják azokat.

Erre a gombra kattintva a Kategória Felvitel opciót illetve a Kategória Módosítást használhatjuk.

A 'Kategória felvitel' esetén ugyanolyan kategória nevet nem adhatunk meg, ezt segíti, hogy az oldal alján azonnal láthatjuk a már elérhető kategóriákat. A felvitelre vonatkozó megkötések láthatóak a mező felett. Érdeemes tömör, több rag nélküli szavakat használni.

Kategória beszúrás:

The screenshot shows the 'Gyűjtők Boltja WebShop' admin interface. At the top, there's a blue header with the shop name. Below it, a message says 'Kedves: stamina!Új kategória felvitele!'. A text input field is present with a placeholder: 'Add meg a kategória nevét! Az első betű nagy betű legyen, egyes számban írva! (Min. 2 - max. 40 hosszúság!)'. Below the input field is a green button labeled 'Kategória felvitel!'. At the bottom, there's a section titled 'Az elérhető kategóriák listája:' followed by a list of existing categories: 'Autóalkatrész', 'Bakelit lemez', and 'Dedikált könyvek'.

A 'Kategória Módosítása' gombra kattintva táblázatos formában listázzuk ki az elérhető kategóriákat, melyek alatt a kategória nevét, egy üres mezőt, és egy gombot találunk, melyre módosítás van írva.

Amennyiben az adott kategóriát módosítani akarjuk, beírjuk az új nevet, a módosításra kattintunk, és azonnal átmódosul a kategória neve. Fontos tudni, hogy ez után minden korábban az ebben a kategóriában levő termékhez tartozó kategória erre a névre változik. A korábbi elnevezés teljesen megszűnik. Ez lesz helyette.

Kategória Módosítás:

The screenshot shows the 'Gyűjtők Boltja WebShop' admin interface for modifying categories. It displays a table with three rows, each representing a category. Each row has columns for 'Eredeti kategória neve:', 'A kategória módosítása:', and 'Új név!'. The first row shows 'Reklám Plakát', the second row shows 'Szakkönyv', and the third row shows 'Számítógép egér'. Each row has a green button labeled 'A kategória módosítása:' next to the 'Új név!' field.

9.10. Admin felület Archív műveletek

Erre a gombra kattintva, azok a termékek jelennek meg, amiket már eladtunk, vagy korábban az archívumba helyeztünk. Itt egy opció van, a 'Termék Aktívra Tétel'. Erre a gombra kattintva a terméket újra az elérhetőek közé helyezzük át. Fontos tudni, hogy ilyenkor csak 1 db-ot teszünk elérhetővé így.

9.11. Admin felület: Eladások megtekintése

Az eladások műveleteknél két gomb látható. 'Eladások Megtekintése' és a 'Nem. reg Eladások'. A két gomb egy-egy listázás. A vásárlások adatait listázza ki részletesen és az adott terméket, annak adatait. Itt tudja az adminisztrátor megtekinteni a vásárlók adatait, a vételárakat, és minden az adásvétel lebonyolításához szükséges információt.

Gyűjtők Boltja WebShop

Kedves: **stamina!**
Eladott termékek és a vevők adatai:

Regisztrált felhasználók vásárlásai:

A vásárlás adatai:

Aktuális vásárlás azonosítója (ID-je): **384**. Aktuális termék ID: **320**. A terméket vásárló felhasználó ID-je: **71**.

Adott termék adatai :

A Termék címe: Retró Motorkerékpár.
A tétel ára: **780000**
A termékhez tartozó kulcsszavak: **Motorkerékpár,Veteránautó,Jármű**
A termék feltöltésének dátuma **2020-10-25 13:56:31**
Képek:



Adott vásárló adatai :

Felh. ID.: 71
Felh. neve: stamina
Felh. vezetéknév: Rimóczy
Felh. keresztnév: Kolos
Ország: ADMIN
Irányítószám: 1137
Város: ADMIN
Utca: ADMIN
Házszám: ADMIN
Telefon: ADMIN
Email: ADMIN@ADMIN.HU
Reg. Dátuma 2020-07-26 18:18:18

A vásárlás adatai:

Aktuális vásárlás azonosítója (ID-je): **385**. Aktuális termék ID: **344**. A terméket vásárló felhasználó ID-je: **71**.

Adott termék adatai :

A Termék címe: Szép fogadalom.
A tétel ára: **1000**
A termékhez tartozó kulcsszavak: **moziplakát,filmplakát,poszter**
A termék feltöltésének dátuma **2020-11-22 20:34:15**
Képek:



Adott vásárló adatai :

Felh. ID.: 71
Felh. neve: stamina
Felh. vezetéknév: Rimóczy
Felh. keresztnév: Kolos
Ország: ADMIN
Irányítószám: 1137
Város: ADMIN
Utca: ADMIN
Házszám: ADMIN
Telefon: ADMIN
Email: ADMIN@ADMIN.HU
Reg. Dátuma 2020-07-26 18:18:18

10. BŐVÍTHETŐSÉG, JÖVŐBELI TERVEK

A tervezés és a kódolás elején is látszott már hogy még funkciók és opciók sokaságát lehetne beépíteni, hogy valóban korszerű web-shopot tudjunk használni. Számos ötlet illetve terv nem valósult meg, a teljesség igénye nélkül néhány:

- Számítunk külföldi regisztrációra is, természetesen később a lapot angolul is elkészítjük.
- Későbbi kényelmi funkció lehetne, hogy az országot egy listából lehessen kiválasztani.
- Továbbá a projekt jövőbeni fejlesztési terve szerint - több eladó is lehetne.
- A több eladónál biztonsági kockázatot vetne fel, hogy az eladó által feltöltött fájlok ne legyenek kártékonyak legyenek, ezeket majd komolyan kell vizsgálni. A gyakorlatban már most is lehetséges több adminisztrátor, de ennek tesztelésére nem volt mód.
- A feltöltött kép fájlokon un. MIME típus ellenőrzést kellene végrehajtani.
- A későbbi tervek között szerepel a kategória fa létrehozása, a főkategóriákban alkategóriák szerepeltetése. Ennek
- jelenléte egyébként gyűjtői szemszögből nem feltétlen indokolt, de hasznos.
- Jövőben tervek a jelszó 'Sózás'-a , mely egy karaktersorozatot generál véletlenszerűen a jelszó mellé, és így mentik el ezt összefűzve és titkosítva, ezáltal is biztonságosabbá tenni a web-áruházat.
- További funkció lehet az admin felületről elérhető tárhely információk, a szabad terület jelzése, hogy a feltöltött képek tárhely igényét tudjuk.
- Tömeges hírlevél küldés lehetősége a felhasználóknak az admin felületről.
- Későbbi terv a kulcsszavak dinamikussá tétele és a meta tag-ekben való elhelyezése.
- Az adatbázis gyorsabb működése érdekében később az sql műveleteket tárolt eljárások módjára létrehozni. Szintén e célból, a az archív termékeknek később egy 'új' táblát lehet létrehozni, és a már nem aktív termékek átkerülnének ide.
- Később a már eladott termékek megsokasodnak, így célszerű lenne az archivált termékekre való szűrés sok módon. Erre létre lehetne hozni egy gombsort.
- Továbbá az eladott termékek megjelenítésénél beépítésre kerülhetne számos szűrés, mely az eladások átláthatóságát segítené.
- Mindenképp szükséges beépíteni az automatikus email küldést a vásárlások visszajelzéséről.

11. KÖSZÖNET

Szeretném megköszönni az oktatást és konzultációt tanáromnak, Dorogházi Donátnak. Fiatalos lendületével sokat segített abban, hogy kedvem legyen ehhez a szakmához.

Szeretném megköszönni páromnak, Papp Szilviának az önzetlenségét, hogy az utóbbi másfél évben elnézte nekem hogy a tanulás lett az első.

12. IRODALOMJEGYZÉK

Internetes forrásjegyzék:

<https://getbootstrap.com/docs/4.5/getting-started/introduction/>

<https://www.w3schools.com/>

<https://developer.mozilla.org/>

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

Nyomtatott könyvek:

Reiter István – C# programozás lépésről lépésre (2018)

Steven Suehring, Janet Valade – PHP, MySQL, JavaScript & HTML5 Tantusz könyvek 7 könyv 1-ben

Matt Zandstra Tanuljuk meg a PHP5 használatát 24 óra alatt (2005)

Bruce Lawson, Remy Sharp – Bemutatkozik a HTML5 (2013)

Luke Welling, Laura Thomson – PHP és MySQL webfejlesztőknek, hogyan építsünk webáruházat? (2010)

Kozmajer Viktor – PHP és MySQL az alapoktól. (2011)