

Debreceni Egyetem Informatika Kar

DINAMIKUS WEBLAPOK KÉSZÍTÉSÉNEK LEHETŐSÉGEI

Témavezető:
Dr. Rutkovszky Edéné
Egyetemi tanársegéd

Készítette:
Pető Zsolt
Programtervező
Matematikus

Debrecen
2008

Bevezetés.....	3
Apache webservert	4
MySQL.....	5
PHP	6
Követelmények és tervek	7
1. A termék felhasználási területe	7
2. A termék áttekintése	7
3. A rendszer funkciói	7
4. Általános megszorítások.....	9
5. A felhasználói felület felépítése	9
6. Az egyes funkciók és oldalak működése bővebben leírva	10
Az adatmodell.....	21
ER-Modell	21
Relációs modell.....	22
A táblák és mezők felsorolása, valamint típusai.....	23
Fejlesztői dokumentáció.....	25
Gépi és nyelvi igények	25
Algoritmusok szintjeinek leírása	25
Az oldal fejlesztésének lehetőségei.....	56
Összefoglalás	57
Irodalomjegyzék	58

Bevezetés

Ahogy a számítógépek egyre elterjedtebbé váltak, az internet használók tábora is megnövekedett, főleg ahogy az árak is csökkentek. Manapság szinte mindenki életébe beférkőzött az internet. Kezdetben szigorúan munkára használták, majd később kezdték szórakozásra is igénybe venni, később az internetes kereskedelem és az internetes reklámozás is egyre elterjedtebbé vált. A napjainkban egyre több internetes szolgáltatás érhető el. Az emberek felismerték az internet előnyeit.

Azért választottam ezt a témát, hogy tájékozottabb legyek a weboldalak programozása terén. Ez egy olyan terület, amin mindenképpen szerettem volna új dolgokat tanulni, már csak azért is, mert van esély rá, hogy az egyetemet elvégezve olyan céghez kerülök dolgozni mely dinamikus weboldal készítéssel foglalkozik, ezen kívül érdekel a webes programozás, és izgalmasnak találom a területet.

A szakdolgozatom céljával egy dvd filmeket értékesítő tetszetős külsejű webáruház létrehozását tűztem ki. Ezzel jól demonstrálni lehet a dinamikus weboldalak mibenlétét. Szerettem volna egy viszonylag jól használható rendszert, ami bár nem állítható be azonnal tényleges értékesítésre, például csak bruttó árakkal dolgozik az oldal, de némi ráfordítás utána teljes értékű webáruház alakítható ki belőle (pl. megfelelő ÁFA kezelés, bővebb kimutatások, elérhetőségek pontos beállítása).

Célom volt, hogy csak bizonyos látogatók tudjanak rendelni az áruházban, ezért szükségesnek találtam a regisztrációt. Így a regisztrált az oldal teljes funkcionalitását élvezhetik szemben a nem regisztrált felhasználókkal. Ilyen funkciók például a felhasználói kosár, amibe szabadon tárolhat termékeket, majd megrendelheti a kosár tartalmát, vagy az egyes filmekhez való hozzászólás, kritika írása. Valamint szükségét éreztem egy adminisztrátori felület létrehozásának, melyben az adminisztrátor törölhet, felvehet, módosíthat termékeket, ill. felhasználókat, rendeléseket törölhet, listázhat. Az oldal elkészítésénél főleg a használhatóságot helyeztem előtérbe a nagyon díszes felülettel szemben.

Egy webes áruház létrehozásához nem elég ismerni a statikus html nyelvet, hanem valamilyen webes programozási nyelvet kell használni. A PHP-t választottam, mert ez eléggé népszerű, ingyenes és használható szkript alapú nyelv. Ráadásul már csak azért is könnyebb elsajátítani, mert szintaxisban, nagyban hasonlít a C nyelvre. PHP egy szerver oldali nyelv, ami azt jelenti, hogy a kód a szerveren fut le és a webszerver csak a generált html kimenetet küldi el az oldalt lekérő böngészőnek.

Webszervernek a több platformos, ingyenes és robosztus Apache-ot választottam. Az Apache http szerver szintén nagyon népszerű az webprogramozók körében.

Révén, hogy webáruházról van szó, termékeket kell tárolni, azaz szükség van az adatbázis kezelésre. Választásom a MySQL adatbázis kezelőre esett. Ez egy szintén több platformon elérhető és ingyenes rendszer, valamint sok helyen olvastam róla, hogy az Apache, PHP és MySQL hármas jól használható együtt és könnyen összehangolható. A PHP 5.0 verzió előtt a MySQL adatbázis kezelőre volt legfőképpen kihegyezve, mint ha ez lett volna az alapértelmezett adatbázis kezelője. Így kézenfekvőnek tűnt a választás a MySQL mellett.

Az áruház tetszetős felhasználó felületéhez szerettem volna CSS-t (Cascading Style Sheets) használni, de erre már nem lett volna időm.

Apache webservert

[6]

Az Apache HTTP Server (röviden Apache) egy nyílt forráskódú webkiszolgáló alkalmazás, szabad szoftver, mely kulcsfontosságú szerepet játszott a World Wide Web elterjedésében. A projekt célja egy olyan webservert program létrehozása, karbantartása, és fejlesztése, amely megfelel a gyorsan változó Internet követelményeinek, biztonságos, üzleti, vállalati felhasználásra is megfelelő és szabadon használható. Az Apache a régi NCSA httpd szerverre épül, az Apache Szoftvert Licenc feltételei alatt terjesztik.

Az Apache egy robusztus, erőteljes és rugalmas webservert, amely kompatibilis a HTTP/1.1 (RFC2616) protokollal. Az Apache projekt koordinálását az Apache Software Foundation végzi. Néhány vezető és több száz fejlesztő van e projekt mögött.

Az Apache volt az első használható alternatíva a Netscape Communications Corporation webserverttel szemben (melynek mai neve Sun Java System Web Server). A későbbiekben továbbfejlődött és más unixalapú webservertekkel is felvette a versenyt funkcionalitás és teljesítmény tekintetében.

Többek között a következő operációs rendszerekhez készítették el az Apache-ot: Unix, FreeBSD, Linux, Solaris, Novell NetWare, Mac OS X és Microsoft Windows.

Szolgáltatások és tulajdonságok

Az Apache sok szabványt támogat, melyeknek nagy része fordított modulok formájában áll rendelkezésre a mag kiegészítéseként. Ezek a modulok sok területet lefednek a kiszolgálóoldali programnyelvtámogatástól kezdve a hitelesítési sémáig. Az ismertebb, támogatott programnyelv modulok a `mod_perl`, a `mod_python`, a Tcl és a PHP. A népszerűbb hitelesítési modulok a `mod_access`, `mod_auth` és a `mod_digest`. További modulokban megvalósított tulajdonságokra példa még az SSL és TLS támogatást nyújtó `mod_ssl`, a proxy modul, egy hasznos URL átíró a `mod_rewrite` modulban, testreszabható loggolás a `mod_log_config` modulban és szűrési támogatás a `mod_include` és `mod_ext_filter` modulok segítségével.

A `mod_gzip` modul tömöríteni tud, segítségével csökkenthető a weboldalak mérete. Az Apache logok elemezhetőek böngészővel az AWStats/W3Perl vagy a Visitors programok segítségével.

A virtual hosting segítségével egyetlen Apache installáció ki tud szolgálni több különböző webhelyet is. Például egyetlen gép egyetlen Apache szerverrel képes kiszolgálni a `www.valami.com`, a `teszt42.trallala.hu` és a `www.burgonya.org` címekre érkező HTTP kéréseket.

Az Apache hibaüzenetei konfigurálhatóak. Számos grafikus felhasználói felület (röviden GUI) támogatja a szerver könnyebb és intuitívabb konfigurálását.

Felhasználás

Statikus és dinamikus weboldalak közzétételére egyaránt használják. Sok webalkalmazást az Apache által nyújtott környezethez és szolgáltatásokhoz terveznek. Az Apache alkotja a webservert komponensét a népszerű LAMP alkalmazáscsomagban, melynek további komponensei a MySQL adatbázisszerver és a PHP/Perl/Python programozási nyelvek mellett.

Az Apache webservert különböző kereskedelmi szoftvercsomagok (például az Oracle Database és az IBM WebSphere) részeként is terjesztik. A Mac OS X is magába foglalja az

Apache-ot beépített webserverként. A Novell NetWare 6.5 alapértelmezett webservere is az Apache.

Az Apache nem csak weboldalak, hanem egyéb tartalom publikálására is használható, például tetszőleges file-ok megosztására is. Ha egy felhasználó telepíti az Apache-ot a személyi számítógépén, akkor tetszőleges file-okat meg tud azon keresztül osztani, ha bemásolja őket az Apache document root könyvtárba.

A webalkalmazásokat készítő szoftverfejlesztők is gyakran használnak a személyi számítógépükön telepített Apache webservert a fejlesztett kód tesztelésére.

Számos nagy honlap Apache webserveren lakik. A Google keresőmotor felülete egy módosított Apache-on fut, melyet a Google Web Server (röviden GWS) névre kereszteltek. A Wikimedia projektek, beleértve a Wikipédiát is Apache szervereken futnak

MySQL

[7]

A MySQL egy többfelhasználós, többszálú, SQL-alapú relációs adatbázis-kezelő szerver.

A szoftver fejlesztője a svéd MySQL AB cég, amely kettős licenccel teszi elérhetővé a MySQL-t; választható módon vagy a GPL, vagy egy kereskedelmi licenc érvényes a felhasználásra. 2008 januárjában a Sun felvásárolta 800 millió dollárért a céget.

Az MySQL az egyik legelterjedtebb adatbázis-kezelő, aminek egyik oka lehet, hogy a teljesen nyílt forráskódú LAMP (Linux–Apache–MySQL–PHP) összeállítás részeként költséghatékony és egyszerűen beállítható megoldást ad dinamikus webhelyek szolgáltatására.

Elérhetősége programnyelvekből:

Egyedi illesztőfelületekkel az adatbázis-kezelő elérhető C, C++ , C#, Delphi, Eiffel, Smalltalk, Java, Lisp, Perl, PHP, Python, Ruby és Tcl programozási nyelvvel. Egy MyODBC nevű ODBC interfész további, ODBC-t kezelő nyelvek számára is hozzáférhetővé teszi az adatbázis-kezelőt. A MySQL számára az ANSI C a natív nyelv.

Adminisztrációja:

A MySQL adatbázisok adminisztrációjára a mellékelt parancssori eszközöket (mysql és mysqladmin) használhatjuk. A MySQL honlapjáról grafikus felületű adminisztráló eszközök is letölthetők: MySQL Administrator és MySQL Query Browser.

Széles körben elterjedt és népszerű alternatíva a PHP nyelven írt, nyitott forráskódú phpMyAdmin. A phpMyBackupPro (amelyet szintén PHP-ban írtak) adatbázisok (akár időzített, ismétlődő) mentésére szolgál eszközzel.

A webáruház elkészítéséhez én a MySQL GUI Tools 5.0-át használtam.

Platformok

A MySQL különböző platformokon futtatható: AIX, BSDi, FreeBSD, HP-UX, Linux, Mac OS X, NetBSD, 0Netware, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP és a Windows frissebb verziói. Már MacOSX 10.4-re is elérhető.

Főbb jellemzők

- Az ANSI SQL 99 bő részalmazának és bővítéseinek széleskörű támogatása

- Keresztplatformos támogatás
- Az adatbázis-kezelőtől független tárolómotorok (MyISAM a gyors olvasáshoz, vagy a tranzakciókat és hivatkozási integritást kezelő InnoDB)
- Mentési pontokat is kezelő tranzakciók
- SSL támogatás
- lekérdezések gyorstárazása
- al-lekérdezések (beágyazott SELECT-ek)
- replikáció (one-master)
- teljesszövegű indexelés és keresés a MyISAM motor használatával
- beágyazott adatbázis-könyvtár
- teljes Unicode-támogatás
- ACID megfelelés az InnoDB motorral
- klaszterezés a MySQL Cluster használatával
- A MySQL 5.0 legfontosabb újítása a tárolt eljárások, felhasználói függvények és a triggerek megjelenése volt.

PHP

[8]

A PHP (PHP: Hypertext Preprocessor) nyílt forráskódú, számítógépes szkriptnyelv, legfőbb felhasználási területe a dinamikus weboldalak készítése. Emiatt a PHP-t jórészt szerver-oldalon használják, bár létezik parancssori interfésze is, illetve önálló, grafikus felületű alkalmazások is létrehozhatóak vele.

A nyelvet eredetileg Rasmus Lerdorf alkotta meg 1994-ben, de a ma létező egyetlen (és hivatalos specifikáció híján de facto szabvánnyá vált) PHP implementációt már a PHP Group tartja karban és fejleszti. A PHP a saját licensze alatt kerül kiadásra, a Free Software Foundation így szabad szoftverként tartja számon.

A PHP a legtöbb webszerverre, operációs rendszerre és platformra ingyenesen telepíthető. Manapság több mint 20 millió weboldal és egymillió szerver futtat PHP-t, bár a nyelvet használó oldalak száma 2005 augusztusától kezdve folyamatosan csökken. A PHP emellett az Apache webszerver egyik legnépszerűbb beépülő modulja.

Én a PHP 5.2.4-et használtam. De ennél már van újabb verzió is.

A PHP működése, leegyszerűsítve

A PHP oldalak elkészítésénél a HTML-t gyakorlatilag csak mint formázást használják, ugyanis ezen lapok teljes funkcionalitása a PHP-re épül. Amikor egy PHP-ben megírt oldalt akarunk elérni, a kiszolgáló először feldolgozza a PHP utasításokat, és csak a kész (HTML) kimenetet küldi el a böngészőnek, így a programkód nem is látható kliens oldalról. Ehhez egy ún. interpretert (értelmezőt) használ, amely általában egy külső modulja a webszervernek.

A PHP nyelv lényegében nagymértékű kiegészítése a HTML-nek, ugyanis rengeteg olyan feladat végezhető el vele, amelyre az ügyféloldali szkriptek nem képesek (vagy ha igen, korlátozottan). Ilyen például a bejelentkezés, az adatbáziskezelés, filekezelés, kódolás, adategyeztetés, kapcsolatok létrehozása, e-mail küldése, adatfeldolgozás, dinamikus listakészítés stb. Minden olyan esetben, ahol nagyszámú ismétlődő feladatsort kell végrehajtani (például képek listázása és linkelése, listakészítés stb.), ott ez a programnyelv nagyszerű segítség.

A PHP programok futhatnak közönséges (parancssori) programként is, nem HTML oldalba építve. Ezt azonban ritkán használják.

A PHP nyelv egészen egyszerű és könnyen megtanulható.

Főbb jellemzők:

- Platform független
- Könnyen telepíthető az Apache webserververhez
- ZEND2 névre hallgat a PHP magja
- Nem létezik benne változó deklaráció
- Nem túl típusos nyelv, a típusegyeztetés nem szigorú, automatikus konvertálások zajlanak a háttérben
- Sok kiterjesztés van hozzá (pl.: mysql, gd2 képmanipulálás,..stb.)
- Sokféle adatbázishoz lehet vele csatlakozni, de az alapértelmezett adatbázis a MySQL volt jó ideig

Követelmények és tervek

1. A termék felhasználási területe

1.1.1. A webáruházban műsoros dvd-eket lehet rendelni és a rendelt termékek, eljutnak majd az ügyfélhez.

2. A termék áttekintése

2.1. A dvd webáruház egy nagyon sok szempontból hasznos dolog, az egyik legfontosabbak közül ilyen a kényelem, az embernek nem kell leballagnia egy dvd üzletbe, hanem akár a karosszékből is rendelhet magának filmeket vagy más ilyen tulajdonsága, hogy akár helyben az adott filmhez esetleg megtekinthet egy előzetest a filmből, de ezeken kívül nagyon sok tényező van.

3. A rendszer funkciói

3.1. Aki az oldalt meglátogatja keresni, szűrni, nézelődni... stb. tud, de vásárolni csak regisztrált felhasználók vásárolhatnak. Azon funkciókat, amiket a nem regisztrált felhasználó nem jogosult használni többnyire nem látja.

3.2. A látogatónak van lehetősége az egyes filmekhez előzetesek megtekintésére, letöltésére

3.3. Minden egyes dvd-hez az adminisztrátor megadhat képet, kategóriát, megjelenés időpontját, rövid leírást és hosszú leírást

3.4. A látogató először a rövid leírással találkozik, amikor rákattint az adott dvd-re akkor egy részletesebb lap jelenik meg a dvd-ről ami, a részletesebb leírást tartalmazza itt található meg a már említett előzetes a filmből, a regisztrált felhasználók által írt kritikákat és az általuk beállított értékelést is, amely felhasználónként nullától tízig terjedhet.

3.5. A termékek kategóriákba vannak besorolva, mely kategóriákba besorolást is az adminisztrátor végez el

3.6. A regisztrált felhasználók a számukra megtetszett filmeket (akár több darabot ugyanabból a filmből) virtuális bevásárló kosárba helyezhetik el, melynek tartalmát

bármikor megtekinthetik és belőle az egyes tételeket törölhetik. A felhasználó csak a kosár megtekintése esetén tud tovább haladni a megrendelés folyamatán.

3.7. Miután a felhasználó a kosárnál úgy dönt, hogy megrendeli a benne levő dolgokat, még utoljára megjelenik a felhasználó kézbesítési címe. Majd a program közli, hogy a megrendelést felvette. A felhasználó pár napon belül megkapja a rendelt termékeit.

3.8. A felhasználó az eddigi rendeléseit megjelenítheti a megfelelő menüpontról. Itt láthatja az egyes tételeket, a megrendelés dátumát és, hogy a rendelése még folyamatban van, vagy lezárult, tehát megkapta az adott rendelés dvd-it.

3.9. Egy film megjelenhet alapáron, vagy akciósan az adminisztrátor által megadott százalékkal engedve az árból, ha a termék akciós, akkor a program jelzi.

3.10. A kezdőoldalon a legújabb filmek közül megjelenik véletlenszerűen pár darab az oldalon, valamint az akciós termékekből adott darab, az hogy mi számít legújabbnak azt az időintervallumot az adminisztrátor állítja be, valamint a kirakott képek darabszámát is

3.11. A kirakott képek szélessége előre meghatározott tehát a program húzza adott méretűre, és ehhez arányosan állítja be a magasságot is

3.12. Mivel a filmeket a regisztrált felhasználók értékelhetik is melyről volt már szó így minden oldalon láthatóvá válik egy toplista ezen értékelések alapján, mely alaphelyzetben az 5 legjobb filmet tartalmazza, melyek közül bármelyikre rákattintva az adott film részletes lapjánál találja magát a látogató. Ezt a lista csak bizonyos időszak filmjeire vonatkozik ezt az intervallumot szintén az adminisztrátor állítja be.

3.13. Az oldalakon található egy másik toplista is, mely alaphelyzetben az 5 legkelendőbb dvd-t jeleníti meg.

3.14. Szűrés illetve a filmek megtekintése az adott menüpontra kattintással érhető el alaphelyzetben nincsenek szűrési feltételek megadva szóval minden film látszik, ha változtatunk valamit, akkor azt mindenképpen egy gomb lenyomásával kell jeleznünk, tehát nem frissül az eredmény automatikusan.

3.15. Minden felhasználó maga állítja be, hogy egy oldalon a szűrés eredményei közül hány látszódjon ez alapértelmezésként 10.

3.16. Az eredmények megjelenítése a cím, kategória, megjelenés dátuma hármából lehet kiválasztani alaphelyzetben a cím szerinti kiírás látható, ha mást választunk a másodlagos rendezés a cím szerinti lesz. Adminisztrátori nézetre is igazak az itt elmondottak.

3.17. A szűrés a cím, kategória és megjelenés dátuma szerint történik itt intervallum adható meg, mely kezdetben 1900-2100 közti időben készült filmeket engedi megjeleníteni. A cím szerinti szűrésnél megadható, hogy a keresett szöveg a címek elejére illeszkedjen-e vagy bárhol megtalálható legyen a szövegben. Adminisztrátori nézetre is igazak az itt elmondottak.

3.18. A regisztrált felhasználók regisztráláskor jelszót és felhasználói nevet, e-mail címet és kézbesítési címet adnak meg. A felhasználó jelszót változtathat a személyes adatai menüpontjánál.

3.19. Az adminisztrátor is felhasználó, csak speciális felhasználó, felhasználói neve: admin

3.20. Az adminisztrátor végzi fentebb már említett dolgokon kívül az esetleges felhasználók törlését ez esetben a felhasználó összes rendelése és a kosara is törölve lesz. A felhasználó megrendeléseit szintén törölheti, valamint beállíthatja a rendelés állapotát aszerint, hogy mi történt az adott rendeléssel ez lehet „folyamatban” vagy „lezárt”. és ezzel együtt automatikusan a hozzájuk tartozó tételek, kosaruk megtekintését, törlését. Ez

a következő módon valósul meg az adminisztrátor lekéri a felhasználókat az adott menüponttal mely csak adminisztrátor nézetben látható. Majd itt kiválaszthatja, hogy törölje-e a felhasználót vagy csak megtekintse az adatait. Ha megtekinti, akkor megjelenik a felhasználó összes adata, szállítási címe, felhasználó neve. Itt kiválaszthatja, hogy meg akarja tekinteni a felhasználó kosarát, vagy törli a tartalmát. Valamint megtekintheti a felhasználó rendeléseit. Itt kap listát a felhasználó rendeléséről az egyes tételeket tartalmazva. Itt választhat, hogy vagy törli azokat vagy csak az egyes tételeket, vagy beállítja a rendelés állapotát, melyről már volt szó. Ha megtekinti, akkor

3.21. Az adminisztrátor végzi a termékek felhelyezését az oldalra és levételét. Nála a termékek listájánál lesz megadható, hogy új terméket vegyen fel, létező terméket töröljön, vagy módosítson. Felvétel esetén beállítja a termék árát, akciót hozzá, részletes és rövid leírást, kategóriába sorolást a dvd címét, megjelenési dátumát. Valamint módosíthatja is a címet, ez nem lesz hatással toplistára, tehát a termék neve, ha változik itt, a toplistákban is fog.

3.22. Az adminisztrátor az egyes felhasználókhoz és a rendelésekhez megjegyzéseket fűzhet, a felhasználói megjegyzéseket csak az adminisztrátor láthatja, míg a rendelések megjegyzéseit a felhasználó is láthatja de, nem módosíthatja.

4. Általános megszorítások

4.1. A webáruház készletezése olyan, hogy egy adott termékből bármennyit lehet rendelni bármikor, tehát nem fogy ki.

4.2. Két fajta látogató léphet az oldalra, aki regisztrált felhasználó és a nem regisztrált, a regisztrált felhasználó közé tartozik a speciális regisztrált személy az oldal adminisztrátora

4.3. Aki az oldalt meglátogatja keresni, szűrni, nézelődni... stb. tud, de vásárolni csak regisztrált felhasználók vásárolhatnak. Azon funkciókat, amiket a nem regisztrált felhasználó nem jogosult használni többnyire nem látja

4.4. Az oldalon az árak Bruttó árat jelentenek mindenhol, a kiírás és a beolvasás esetén is.

5. A felhasználói felület felépítése

5.1. A főoldal 3 részből áll:

a) Felső rész a fejlécnek

Itt jelenik meg a cég logója, és valós áruház esetén itt jelenhetnének majd meg az áruház elérhetőségei, valamint ha a felhasználó bejelentkezett, akkor a felhasználó neve, vagy „admin” az adminisztrátor esetén, vagy „regisztrálatlan” nem belépett felhasználók esetén.

b) Baloldali rész

A menüt tartalmazza valamint az akciós és legnépszerűbb dvd filmeket. Ezek az adminisztrátor minden egyes bejelentkezésével frissülnek. Minden itteni film egy linket tartalmaz, melyre kattintva az adott film részletes adatait láthatjuk.

c) Középső munkaterület

Itt fog megjelenni minden más tartalmi rész

5.1.2. Az egyszerű felhasználó által látott menüpontok:

a) Főoldal: rákattintva a fő oldalra jut

- b) Filmek: a teljes film kínálat, cím majd megjelenési dátum szerint rendezve
 - c) Keresés: filmet kereshet magának a felhasználó
 - d) Belépés/regisztráció: itt léphet be vagy regisztrálhat
- 5.1.3. A bejelentkezett felhasználó által látott menüpontok:
- a) Főoldal: rákattintva a fő oldalra jut
 - b) Filmek: a teljes film kínálat, cím majd megjelenési dátum szerint rendezve
 - c) Keresés: filmet kereshet magának a felhasználó
 - d) Profil: jelszót, a keresésnél az egy oldalra megjelenő találatok számát tudja módosítani
 - e) Rendelések: az eddig leadott rendeléseit és a rendelésekkel kapcsolatos adatokat láthatja (pl. állapot)
 - f) Kosár: a kosarát nézheti meg ebben törölhet termék(keket), vagy innen azonnal a rendelés gombbal megrendelheti a kosár tartalmát
 - g) Levél: levél küldése az adminisztrátornak, ha valami gondja vagy kérdése támadna
 - h) Kilépés: kiléphet
- 5.1.4. A bejelentkezett adminisztrátor által látott menüpontok:
- a) Főoldal: rákattintva a fő oldalra jut
 - b) Filmek: a teljes film kínálat, cím majd megjelenési dátum szerint rendezve
 - c) Keresés: filmet kereshet magának a felhasználó
 - d) Profil: jelszót, a keresésnél az egy oldalra megjelenő találatok számát tudja módosítani
 - e) Rendelések: az eddigi rendelésekkel kapcsolatos adatokat láthatja (pl. állapot)
 - f) Új dvd: új dvd-t vehet fel a webáruházba
 - g) Felhasználók
 - h) Kilépés: kiléphet
- 5.1.5. A munkaterület minden oldalán lesz egy link, ami az azt megelőző oldalra mutat majd így a felhasználónak elég erre kattintania, ha vissza akar jutni

6. Az egyes funkciók és oldalak működése bővebben leírva

6.1. Bejelentkezés folyamata

6.1.1. Meg kell adni a felhasználó nevet és a jelszó

6.1.2. Az oldal session-öket használ a bejelentkezés nyilvántartásához, a session-öket adatbázisba menti. Amikor valaki a bejelentkezési oldalra lép és megadja az adatait, akkor több lehetőség van:

- a) a felhasználó még most lép be először
- b) már be volt lépve de lejárt a session ideje ezért egy alkalommal egy oldal visszairányította a bejelentkezéshez
- c) úgy akar valaki belépni, hogy közben más vagy ugyanaz a személy már belépett egy másik böngésző ablakkal vagy egy másik böngésző fülön vagy ugyanazzal az ablakkal

6.1.3. Az a) eset

Ha érvényes felhasználó nevet és jelszót adott meg akkor belép az oldalra, megnyílik neki egy munkamenet, ami tartalmazza a felhasználó nevét és jelszavát

6.1.4. A b) eset

Szintén ugyanaz az eset van, mint 6.1.3. esetén, de így mindene törlődik, ami a munkamenetében tárolódott a kosarának a tartalma is.

6.1.5. A c) eset

A rendszer ellenőrzi, hogy aki belép az be van-e már jelentkezve, ha igen akkor a már belépett felhasználót lépteti ki, azaz törli a session-jét és új session-t nyit a most belépett felhasználónak. Ez több szempontból is jó megoldás mivel egyrészt biztonságos másrészt a felhasználó be tud lépni, ha előzőleg nem lépett ki hanem csak bezárta a böngészőjét, akkor be tud lépni újra.

6.2. Regisztráció folyamata

6.2.1. Felhasználó által megadandó adatok: nev(kötelező), város, irányítószám, utca, házszám, telefonszám(kötelező), e-mail cím(kötelező), felhasználó név(kötelező), jelszó(kötelező), az adatok szövegdozokba kerülnek, majd a kitöltést befejezve a Regisztráció gombra kattintva az adatokat feldolgozza egy php fájl a fenti adatokon kívül rögzítésre kerül még a regisztráció dátuma is. A következő esetekben visszakapjuk a regisztráció űrlapját, hibaüzenettel:

- a kötelező mezők nincsenek kitöltve
- a két jelszó mező értéke nem egyezik
- már létező e-mail cím vagy felhasználó név

6.2.2. Az ügyfél adatai a címe, alapértelmezett értékek lesznek, vásárláskor ezt fogja felajánlani a rendszer, amit megváltoztathat majd.

6.2.3. Ha a regisztráció sikeres akkor az oldal üzen, hogy egy e-mailt fog küldeni a megadott címre, amely egy 2 linket tartalmaz, a felhasználónak az elsővel fogja tudni megerősíteni a regisztrációját.

6.2.4. Ha a megerősítés megtörtént akkor azt az oldal tudatja a felhasználóval, majd átirányítja a belépés ablakra egy linkkel.

6.2.5. Ha a 2. linkre kattint, akkor törlődik az adatbázisból

6.3. Elfelejtett jelszó

6.3.1. Az oldal kéri a felhasználó nevet és az e-mail címet

6.3.2. Ha az adatbázisban ehhez a felhasználóhoz az az e-mail társul akkor elküldi az

6.3.3. új jelszót a megadott címre, egyébként közli a hibát a felhasználóval

6.4. Főoldali filmválasztás

6.4.1. Ha valaki megnyitja a webáruház főoldalát akkor ez a lap jön be, 3 részre van osztva: fejléc (fent), menü választó rész (bal oldalt) és munkaterület (középső rész)

6.4.2. Középen megjelennek a legújabb filmek és az akciós filmek egy csoportja.

6.4.3. Regisztrálatlan felhasználó vagy az adminisztrátor itt csak böngészhet, regisztrált felhasználó a kosárba is tehet. Megadhat darabszámot az adott termékhez, aztán ha rákattint a Kosárba feliratú gombra, akkor a termék a kosárba kerül, adott darabszámmal, a felhasználó pedig átkerül a kosár weblapra, hogy lássa a kosara tartalmát.. Az adminisztrátor nézetében itt is törölhet terméket, nála nem jelenik meg a kosár gombja és a darabszám szerkesztő mezője.

6.4.4. Ha a filmhez rendelt képre(ami egy nagy kérdőjel ha nincs a filmhez kép hozzárendelve) vagy a film címére kattint valaki akkor megjelennek egy filmről a részletes adatok egy új lapon a középső munkaterületen amik a következők: a fő oldalon látható adatok és hozzá még a filmről egy részletesebb leírás ami hosszú is lehet valamint a film kritikáit az adatlap alján ami tartalmazza a szerző becenevét, a dátumot és a kritikát, egymás alatt. Bejelentkezett felhasználó szintén látja a DVD kosárba rakásához szükséges vezérlőket, valamint láthatja a film által kapott pontokat

is ill., pontoszthatja a filmet egy rádió gombsorral (1-5 pont) amit beküldhet, ha a „szerintem” gombra kattint, vagy írhat kritikát a filmről, amit a „küldés” feliratú gombbal küldhet be. Az adminisztrátor is szavazhat, vagy hozzászólhat a filmhez, valamint nála szintén megjelenik a dvd törlésének vezérlő gombja, de nem jelennek meg a kosár vezérlők. Ezen a lapon egy „<<vissza” feliratú linkkel kerülhet ki a felhasználó a főoldalbeli munkaterületre, ill. ahonnan jött.



A főoldal

6.5. Keresés funkció

6.5.1. A baloldali menüben minden felhasználónál elérhető funkció

6.5.2. A keresés szintén a munkaterületi részben zajlik.

6.5.3. Megjelenik egy kereső/szűrő űrlap, mellyel a filmek címében, a leírásaiban kereshető a megadott szöveg, kis/nagybetű érzéketlen módon. Ezen kívül kereshet a felhasználó arra, ami adott zárt intervallumban van, a film megjelenési dátumára egy adott intervallumban, ha valamelyik intervallum érték nincs kitöltve, akkor értelemszerűen csak a másikat veszi figyelembe. Kereshetünk még kategória szerint is, amit egy kombinált listából választhatunk ki. A keresési/szűrési feltételek szűkítők, azaz ÉS-el vannak összekapcsolva. A keresés a keresés feliratú gombbal indítható el.

6.5.4. Alapértelmezésként minden kitöltendő mező üres kivéve a kategória, ami a Minden értékre van állítva. Ha így zajlik a keresés akkor minden filmet kihoz találatként.

6.5.5. A filmek itt is úgy jelennek meg, mint a fő oldalon, 3 film kerül egy sorba (ha van ennyi találat) középre rendezve.

6.5.6. Nem regisztrált felhasználók egy oldalon max. 10 filmet láthatnak, a regisztrált felhasználó megváltoztathatja ennek a mértékét a profiljában

6.5.7. A keresés indítása után a munkaterület jobb felső sarkában megjelenik egy lapozási lehetőség

6.5.8. A keresés eredményeként megjelent DVD-k megjelenítési formátuma megegyezik a főoldalon levővel.

6.5.8.1. A lapok fel vannak sorolva, legfeljebb 10 darab lap vesszővel elválasztva, ezt bezárja, ha lehet két oldalról egy előre (>>) és egy vissza (<<) mutató link és ezt az egészet bezárja egy első és egy utolsó feliratú link.

6.5.8.2. Először az első lapon állunk. Ha több lap van, mint 10 akkor a 10. lapra állván a következő lapot választva egy újabb 10-es csoport jelenik meg. Ha több lap van, mint 10 akkor azt, hogy merre lehet tovább lépni az aktuális 10 lapos csoportból három pont (...) jelzi a megadott irányba, ezekre nem lehet kattintani, csak jelzik, hogy előre, vagy vissza van még 10-es csoport.

6.5.8.3. Az első ill. utolsó link, az első ill. utolsó lapra navigál. Az előre vagy vissza linkeket egy lappal előre vagy vissza visznek minket, ha lehetséges. Az aktuális lap, amin állunk félkövérrel van kiírva.

6.6. Felhasználói profil funkció

6.6.1. A menüben a Profil menüponttal érhető el

6.6.2. Az űrlapon megtalálhatók a felhasználó által a regisztráció során megadott adatok, ezek vannak alapértelmezett értéként a szerkesztő dobozokban. Az adatok a Név, Város, Irányítószám, Utca, Házszám, Telefonszám, E-mail cím, Felhasználó név (ezt nem lehet megváltoztatni), Új jelszó mező, Találati eredmények darabszáma egy oldalon, a mostani jelszó mező, valamint a Rögzít feliratú gomb. Ezekből a Név, az eredmények darabszáma és a jelszó mezőt mindig kötelező kitölteni, ezek meg is vannak csillagozva, ha nem teszi a felhasználó, akkor írjon hibaüzenetet az oldal.

6.6.3. A jelszót azért kéri, mert ez egy biztonsági intézkedés, hogy ténylegesen az a felhasználó változtat, akinek ez a felhasználói fiókja.

6.6.4. Ha a felhasználó nem akar jelszót változtatni, akkor üresen kell hagynia az új jelszó mezőt, ezt az oldal írja is ki

6.6.5. Miután a felhasználó a végzett a változtatásokkal a rögzít gombbal mentheti a módosításokat. Az oldal ezután egy levelet küld a változtatás sikerességéről a megadott levélcímre, ha ez lehetséges, különben jelzi, hogy nem tudta elküldeni a levelet. A levélben mindenképpen szerepel a felhasználó név és a jelszó, ha a felhasználó változtatott

6.6.6. A sikeres változtatás esetén az oldal közli ezt és egy Vissza feliratú linkkel segíti a felhasználót a kezdő oldalra

6.7. A felhasználó rendelései

6.7.1. A menüben a Rendelések menüponttal érhető el a bejelentkezett felhasználók részére.

6.7.2. Az ablak középső kerete ekkor két részre bontódik fel: a felvett de még nem kézbesített rendelések és a kézbesített rendelések. Előbb tehát azok a rendelések jelennek meg az adott felhasználóhoz, amiket megrendelt de az adminisztrátor nem állította át a rendelés állapotát kikézbesítettre, azaz még nem szállították le fizikailag a felhasználóhoz. A második csoportban a már leszállított rendelések találhatók. Majd

végül egy Vissza feliratú link az oldal alján, ami a főoldalra viszi vissza a felhasználót, ha rákattint.

6.7.3. Minden rendeléshez a következő adatok lesznek feltüntetve: a rendelés dátuma és ideje, a rendelés összértéke, valamint a rendelés tételei. A tételeknél megjelenik egy sorszám, a dvd címe, a rendelt darabszám, az egység ár az adott filmhez, és a részlet összeg, ami az egység ár megszorozva a mennyiséggel. A rendelések meg vannak számozva.

6.7.4. A rendelések közt kezdetben nem lesz semmi. Később pedig esetleg kiürítheti az adminisztrátor, ha pl. egy adott rendelést visszamond a felhasználó, amit levélben tehet meg.

6.8. A kosár funkció

6.8.1. A menüben a Kosár menüponttal érhető el csak a bejelentkezett felhasználók részére.

6.8.2. Az egyes DVD-eket a kosáron keresztül rendelheti meg. A kosár teljes tartalmát megrendeli, ha rendelni akar, ez fog egy rendeléshez tartozni.

6.8.3. Minden bejelentkezett felhasználó rendelkezik kosárral. A kosárba bele tehet filmeket és törölhet belőle filmeket.

6.8.4. Minden olyan dvd-nél, amelyet meg szeretne vásárolni meg kell határoznia a vásárolt darabszámot is. Így a vásárolt dvd és a darabszám belőle alkotnak egy tételt a kosárban.

6.8.5. Az ablakban látható a kosár egyes tételei, ahol minden tételnél fel vannak tüntetve a következő adatok:

- sorszám
- dvd címe
- az ár: ha akciós a termék, akkor az akciós ár található itt meg
- összeg: az ár szorozva a kosárba tett darabszámmal
- egy jelölő négyzet: kiválasztható vele, hogy mely tételeket szeretnénk

törölni a kosárból

6.8.6. Látható még a teljes kosár össz. ára, egy Vissza feliratú gomb, amivel eljuthat a felhasználó a keresés oldalra, valamint található még ezen felül három gomb:

- mindent kijelöl: az összes tételhez bejelöli a jelölő négyzetet
- kijelöltek törlése: az összes bejelölt tételt törli a kosárból
- megrendelés: ezzel rendelhetjük meg a termékeket a kosárból

6.8.7. A kosár tartalma csak a session lejáratí időig él vagy amíg a felhasználó ki nem jelentkezik vagy be nem zárja az ablakot mert onnantól nem fogja tudni elérni a kosarát, ilyenkor a teljes kosár tartalma törlődik. A session lejáratí idő 24 perc, ha addig a felhasználó nem lép át másik oldalra törlődik a kosár, és a felhasználó kiléptetődik, de azt, hogy újra be kell jelentkeznie, csak akkor veszi észre, amikor egy következő oldalt szeretne megnyitni.

The screenshot shows a web browser window with the address bar containing a local URL. The page title is 'DVD Kalauz webáruház'. The main content area is titled 'A kosár' and features a table with the following items:

Sorsz.	Cim	Ár (ha van akcióval)	Összeg	Db	
1	A tökéletes pasi	950 Ft	950 Ft	1	<input type="checkbox"/>
2	Ragadozók	2152 Ft	2152 Ft	1	<input type="checkbox"/>

Below the table, it states 'A teljes összeg: 3102 Ft'. There are buttons for 'Kijelöltek törlése' and 'Megrendelés'. A sidebar on the left lists 'Legjobb filmek' including 'Final fantasy', 'Ragadozók', 'Hell Ride', and 'Fehér zaj 2. - A...'. Navigation links like 'Főoldal', 'Filmek', and 'Keresés' are also present.

A kosár

6.9. Megrendelés

6.9.1. A kosár ablakból juthatunk tovább ide a Megrendelés gombbal, ha van valami a kosárban.

6.9.2. A megrendelés pontos adatait veszi fel az űrlap. Regisztrációnál kitöltött adatokat adja meg alapértelmezésként az egyes mezőkhöz. A következő adatokat/mezőket jeleníti meg: Név, Város, Irányítószám, Utca, Házsám, Telefonszám, E-mail cím, Felhasználó név, az aktuális jelszó a rendelés megerősítéséhez. A mezők közül kötelező kitölteni a következőket: Város, Irányítószám, Utca, Házsám és jelszó, ezeket a mezők meg is kell, hogy legyenek jelölve. A többi mező értékét nem lehet megváltoztatni, csak tájékoztatás. A többi érték is alapértelmezett, azaz ez rugalmasságot ad, hogy ne regisztrációnál beállított címre lehessen csak szállítani. Az oldal tartalmaz egy Megrendelés feliratú gombot és egy Vissza feliratú linket a kosár tartalmához.

6.9.3. A megrendelésnél a beállított mező értékek valamint a kosár teljes tartalmának összes tétele letárolásra kerül az adatbázisba, az összes aktuális adattal, ezek nem fognak frissülni egy esetleg árváltoztatással, megmaradnak olyanak amilyenek a megrendelés pillanatában voltak.

6.9.4. A megrendeléssel a rendelés a Felvéve állapotba fog kerülni. Ez azt jelenti, hogy amíg ki nem szállítják adott időn belül a rendelés összes dvd-jét az adott címre és ott át nem veszik addig ez az állapot marad. Majd ha megtörtént az átvétel és a fizetés akkor az adminisztrátor állítja át Leszállítva-ra

- 6.9.5. A megrendeléssel mindig növekszik egy érték a rendelésben részt vett filmeknél, ami a rendelések számát fogja mutatni. Ennek figyelembe vételével generálódik a menü ablakkeretben a Legkelendőbb filmek toplistája.
- 6.10. Levél küldés funkció
- 6.10.1. A menüben a Levél menüponttal érhető el csak a bejelentkezett felhasználók részére.
- 6.10.2. Ha a felhasználónak valamilyen gondja vagy kérdése van, itt küldhet levelet erről.
- 6.10.3. Az oldal egyetlen nagy szerkesztő dobozból és egy Elküld feliratú gombból áll.
- 6.10.4. A levelet a webáruház címére küldi el az űrlap
- 6.11. Kilépés menüpont
- 6.11.1. A menüben a Kilépés menüponttal érhető el csak a bejelentkezett felhasználók részére. Rákattintva az adott session-t lezárja az oldal, a felhasználót kilépteti és kosara kiürül.
- 6.12. Beállítások
- 6.12.1. A menüben a Kilépés menüponttal érhető el csak az adminisztrátor számára
- 6.12.2. A webáruház beállításai lehetőségeit tartalmazza, melyek a következők:
- Új film leválogatás kezdeti dátuma
 - Új film leválogatás befejező dátuma
 - Új film leválogatás darabszáma: a főoldalon megjelenő új filmek maximuma
 - Akciós film leválogatás kezdeti dátuma:
 - Akciós film leválogatás befejező dátuma:
 - Akciós film leválogatás darabszáma: a főoldalon megjelenő akciós filmek maximuma
 - Új jelszó: az adminisztrátor új jelszava
- 6.12.3. Valamint megjelenik az aktuális jelszót kérő mező, egy Rögzít feliratú gomb és egy Vissza feliratú link, ami visszavisz a főoldalra. Az új jelszó mezőn kívül mindegyik mezőt kötelező kitölteni. Erről tájékoztat is az oldal.
- 6.12.4. A főoldalon a középső keretben megjelenik az „Új film leválogatás darabszáma”-nak megfelelő vagy kevesebb (ha nincs annyi) új film, melyeket véletlenszerűen választ ki az oldal az „Új film leválogatás kezdeti dátuma” és „Új film leválogatás befejező dátuma” zárt intervallumból. A főoldalon a középső keretben megjelenik az „Akciós film leválogatás darabszáma”-nak megfelelő vagy kevesebb (ha nincs annyi) akciós film, melyeket véletlenszerűen választ ki az oldal az „Akciós film leválogatás kezdeti dátuma” és „Akciós film leválogatás befejező dátuma” zárt intervallumból.
- 6.12.5. A módosítások rögzítéséhez meg kell adni a jelenlegi adminisztrátori jelszót. Ha nem az oldal üzenetet ír erről.
- 6.13. Felhasználók rendelései
- 6.13.1. A baloldali menüben csak az adminisztrátor számára elérhető funkció
- 6.13.2. A rendelések szintén a munkaterületi részben zajlik.
- 6.13.3. Megjelenik egy kereső/szűrő űrlap, mellyel a rendelések állapotára adhatunk ki feltételt egy kombinált listában, ami lehet mind, felvéve és leszállítva alapértelmezett értékken a mind jelenik. Szűrhetünk még a dátum szerint, két szerkesztő doboz található meg, mellyel dátum intervallumot adhatunk meg a

határokkal beleértve, azaz zárt intervallumot, ha az intervallum egyik értéke hiányzik, akkor csak a másik értéket veszi figyelembe. Ezek a mezők nincsenek kitöltve alapértelmezetten üresek, így minden rendelést átenged. Valamint lehetőség van csak egy adott felhasználó rendeléseit kiírni, egy másik kombinált listában fel van sorolva az összes felhasználó név, alapértelmezésként a „mindenki” szerepel. A keresési/szűrési feltételek itt is szűkítők, azaz ÉS-el kapcsolódnak össze. A keresés a keresés feliratú gombbal indítható el.

6.13.4. A munkaterületen a másik rész a rendelések listáját tartalmazza. Az alapértelmezett értékekkel leszűrve, azaz minden felhasználó minden rendelése megjelenik, dátum szerint sorba rendezve. Itt található négy nyomógomb:

- a) Mindent kijelöl törlésre: minden rendelést megjelöl az esetleges későbbi törlésre
- b) Mindent kijelöl leszállításra: az összes rendelést megjelölhetjük az esetleges későbbi leszállítás állapot beállítására
- c) Kijelöltek törlése: kijelölt rendelések törölhetők vele
- d) Kijelöltek leszállítása: a megjelölt rendelések állapotát lehet átállítani leszállított-ra

6.13.5. Az oldal legalján található egy Vissza feliratú link, ami a főoldalra visz.

6.13.6. A rendelések rendelési dátum szerint sorba rendezve követik egymást. Minden rendeléshez a következő adatok jelennek meg: név, irányítószám város, utca, házszám, majd következnek a rendelés tételei, melynek a végén a rendelés összértéke, majd egy jelölő négyzet, amivel törlésre lehet bejelölni a rendelést és egy másik jelölő négyzet, amivel leszállított-ra lehet állítani az állapotát a rendelésnek, ha az adott rendelést még nem szállították ki. Valamint szerepel a dátum és a rendelés állapota.

6.13.7. A rendelés tételei a következőképpen jelennek meg: az első oszlop a sorszám, aztán a film címe, a rendelt mennyiség a filmből, az egység ár akcióval egészre kerekítve, majd az adott tétel részlet összege, azaz a mennyiség szorozva az egység árral.

6.13.8. Egyszerre vagy csak törlésre lehet kijelölni az összes rendelést vagy csak leszállításra az összes még nem szállított rendelést.

The screenshot shows the 'A felhasználók rendelései' page. At the top, there's a navigation menu with links like 'Főoldal', 'Filmek', 'Keresés', 'Beállítások', 'Rendelések', 'Új DVD', 'Felhasználók', and 'Kilépés'. Below the navigation, there's a search form for user orders. The search criteria are: 'Azon felhasználói rendelések listája aminek az állapota mind a rendelés dátuma pedig (éééé-hh-nn óó:pp:mm), (éééé-hh-nn óó:pp:mm) közé esett és a esetlegesen az adott felhasználói név: mindenki'. There are buttons for 'Keresés', 'Mindet kijelöl törlésre', 'Mindet kijelöl leszállításra', 'Kijelöltek törlése', and 'Kijelöltek leszállítása'. Below the search form, user details are shown: 'Név: P Zsolt', 'Város: 3400, Mezőkövesd', 'Utca: Arany J. út 38.'. A table of orders is displayed with the following data:

Film cím:	Darab	Egys. Ár	Rész összeg
1. Ragadozók	1	2153 Ft	2153 Ft
			Összesen: 2153 Ft

A felhasználók rendelései

6.14. Új DVD felvitele

6.14.1. A baloldali menüben az Új DVD menüpont felel meg neki, de ez is csak az adminisztrátor számára elérhető funkció

6.14.2. A következő adatok szerepelnek az ablakban, amiket ki kell tölteni: a dvd címe, a rövid leírása, hosszú leírása, előzetese ahol tulajdonképpen egy fájl nevet lehet megadni, ezzel lehet feltölteni az oldalra a film videó előzetesét, ha van, majd ez fizikailag az elozetes nevű könyvtárba fog kerülni. Szerepel még az egység ár, az esetleges akció %-ban, a film megjelenésének dátuma, egy kép, amit szintén ezen az úrlapon kell feltölteni, ami majd fizikailag a kep nevű könyvtárba fog kerülni. Kategória, amibe a film tartozik és melynek az alapértelmezett értéke az Akció, egy megjegyzés mező az adott filmhez, amibe tetszőleges megjegyzést lehet írni. Egy Rögzít feliratú gomb is szerepel, amivel elküldhetjük az űrlapot, valamint egy Vissza feliratú linket is találunk, ami a főoldalra visz.

6.14.3. Ezekből az adatokból a címet, a rövid leírást, az egység árat és a megjelenés dátumát kötelező kitölteni ezt jelzi is az oldal egy csillaggal, ha a felhasználó nem tölti ki ezen adatok valamelyikét, akkor visszakérül ugyanerre az oldalra.

6.15. Létező DVD szerkesztése

6.15.1. Csak az adminisztrátor számára elérhető funkció. A főoldalon, vagy a Keresés menüpont elérésével is vagy a Filmek menüpont segítségével is minden megjelenített DVD-hez lesz egy Szerkesztés feliratú gomb, ennek megnyomásával érhető el ez a funkció.

6.15.2. A középső ablak területben látható majd az eredmény. Az oldal felépítése megegyezik az új dvd létrehozása ablakéval. A különbség csak annyi, hogy itt megjelennek a dvd adatai is alapértelmezett értéként. A már meglévő előzetes és kép esetén szintén jelzi ezt a beviteli mező fölé írt fájlnevekkel és elérési úttal oda. A változtatásnál nem kell újra megadnunk az előzetes fájlját vagy a képfájlt, ha ilyenkor üresen hagyjuk a megfelelő mezőket akkor megmaradnak az eredeti előzetesek és képek.

6.15.3. A rögzít gömb segítségével menthetők a változtatások. A sikerességről az oldal üzenet is kiír az adminisztrátornak.

6.16. Létező DVD törlése

6.16.1. Csak az adminisztrátor számára elérhető funkció. A főoldalon, vagy a Keresés menüpont elérésével is vagy a Filmek menüpont segítségével is minden megjelenített DVD-hez lesz egy Törlés feliratú gomb, ennek megnyomásával érhető el ez a funkció.

6.16.2. A DVD törlése esetén a hozzárendelt összes adat törlődni fog, azaz a már létező előzetese és a hozzárendelt kép is.

6.17. A felhasználók keresése, szűrése

6.17.1. A baloldali menüben a Felhasználók menüpont felel meg neki, de ez is csak az adminisztrátor számára elérhető funkció

6.17.2. Szintén a középső ablakterületbe dolgozhatunk. Ezen az oldalon van lehetőségünk listázni a felhasználókat, ill. szűrni közöttük, valamint törölni felhasználókat.

6.17.3. Az oldal felső részében található egy szűrő űrlap. A következő adatokat várja: a regisztráció állapota, ami egy kombinált lista, alapértelmezésként a mind jelenik meg benne, de lehet választani az aktivált és nem aktivált értékek közül belőle, értelemszerűen az aktivált és a még nem aktivált felhasználókat szűrhetjük ezzel. A regisztráció dátum intervallumának alsó és felső határa, ez az intervallum zárt, azaz a határokat is beleérti. Ha az intervallum valamelyik határa hiányzik, akkor csak a másik értéket veszi figyelembe, vagy ha az is hiányzik, akkor nem történik szűrés a regisztráció dátumára vonatkozólag. felhasználó neve, amelyet egy kombinált listából lehet kiválasztani, ezzel adott felhasználóra kereshetünk rá, a lista alapértelmezett értéke a Mindenki. Valamint megtalálható az űrlapon egy Keresés feliratú gomb, amivel a keresés/szűrés indítható el. A szűrési feltételek szűkítő jellegűek, azaz ÉS értendő közékük.

6.17.4. Az oldal másik része az eredményeket tartalmazza. Alapesetben minden felhasználót megjelenít.

6.17.5. Az alsó űrlapon található gombok a következők:

- a) Mindent kijelöl: minden felhasználó neve melletti jelölő dobozt bejelöli
- b) Kijelöltek törlése: törli a megjelölt felhasználókat a listáról és az az egész rendszerből is

6.17.6. Az alsó űrlapon a következő adatok jelennek meg táblázatban a felhasználóról, az oszlopok a következő adatokat tartalmazzák: egy sorszám, a felhasználó név, a regisztráció állapota(Aktivált), amely oszlopnál az egyes értékek az igen és nem lesznek, a következő oszlop a regisztráció dátumát tartalmazza, majd az utolsó oszlop egy jelölő négyzetet minden felhasználóhoz. A felhasználók táblázata alatt mutatja, hogy hány darab felhasználót talált a keresés. Majd legalul találunk egy Vissza feliratú gombot, ami a főoldalra vezet.

6.17.7. A keresés eredményeként lekért felhasználók felhasználó neve egy link lesz melyre kattintva a felhasználók részletes adatait megadó oldalra jut az adminisztrátor.

6.18. A felhasználók részletes adatai

6.18.1. Szintén adminisztrátorként elérhető funkció, mely a bal oldali menü Felhasználók menüpontja alatt érhető el, ha az itt megjelenő felhasználók valamelyikének a felhasználó nevére kattint az adminisztrátor.

6.18.2. Az oldal egy űrlapot tartalmaz, amin az adatok egy része a következő: Név, Város, Irányítószám, Utca, Házsám, Telefonszám, E-mail cím, Felhasználó név, Megjegyzés a felhasználóhoz. Ezen adatok közül csak a megjegyzés mezőbe írhat az adminisztrátor a többi szerkesztő doboz csak olvasható.

6.18.3. Az űrlapon levő többi adat a rendelésekre vonatkozik

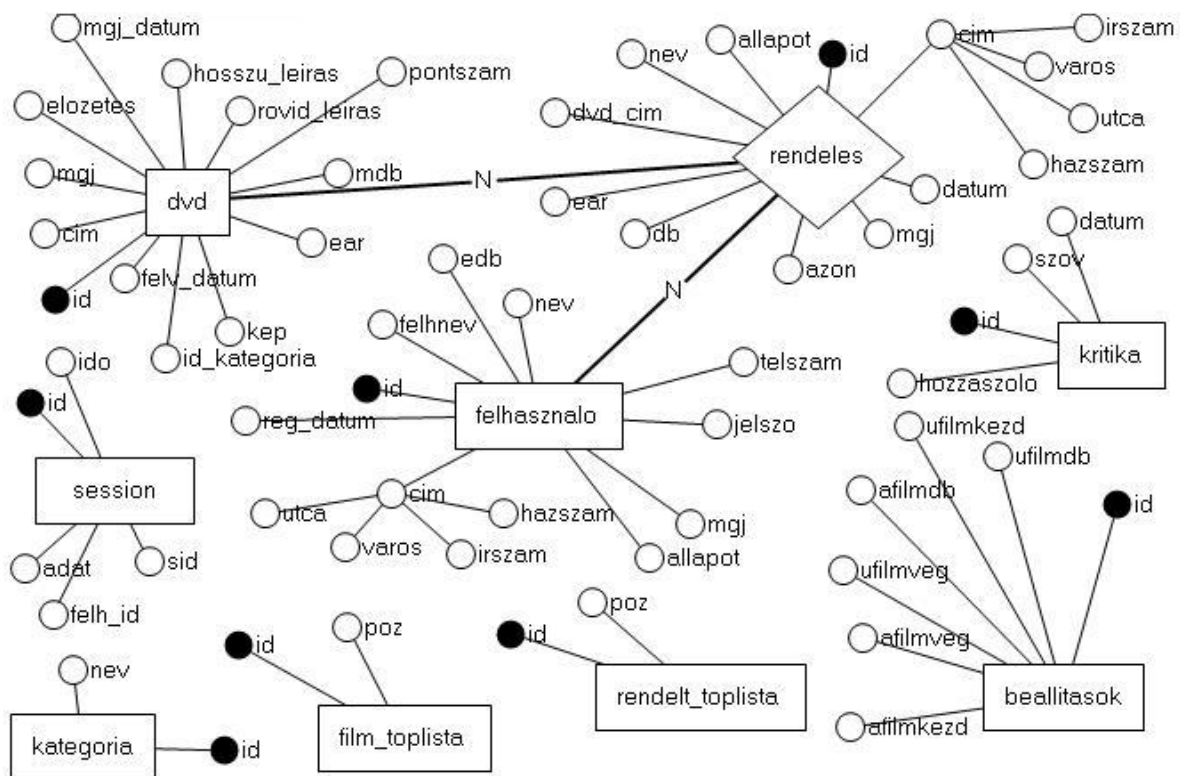
- a) Eddigi rendeléseinek száma (felvett)
- b) Eddigi rendelési összértéke (felvett)
- c) Eddigi rendeléseinek száma (leszállított)
- d) Eddigi rendeléseinek összértéke (leszállított)

6.18.4. Némi statisztikát adnak a fenti adatok.

6.18.5. Az oldal tartalmaz még egy Rögzít feliratú gombot is a megjegyzések tárolására/módosítására, valamint egy Vissza feliratú linket a Felhasználók keresése/szűrése oldalra.

Az adatmodell

ER-Modell

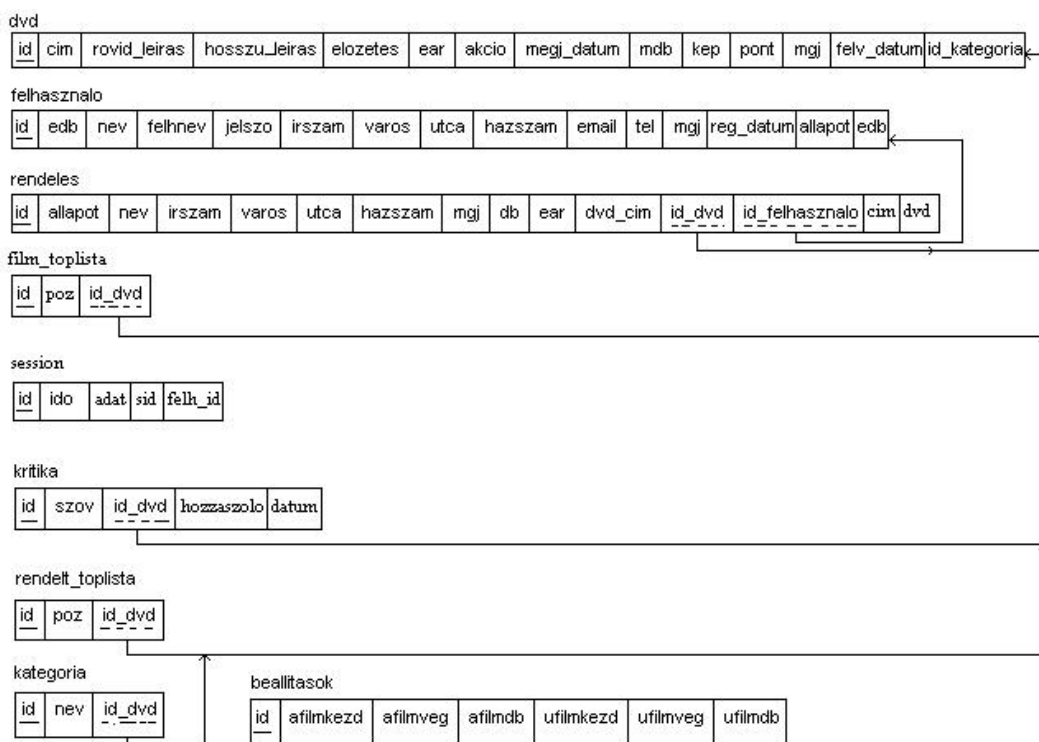


Fogalom szótár:

id	azonosító
mgj	megjegyzés
mgj_datum	megjelenési dátum
mdb	megrendelt darabszám
db	darab
ear	egység ár
irszam	irányítószám
felhnev	felhasználói név
poz	pozíció
afilmkezd	a főoldalon az akciós filmek megjelenési intervallumának kezdete
afilmveg	a főoldalon az akciós filmek megjelenési intervallumának vége
afilmdb	a főoldalon megjelenő akciós filmek darabszáma
ufilmkezd	a főoldalon az új filmek megjelenési intervallumának kezdete
ufilmveg	a főoldalon az új filmek megjelenési intervallumának vége
ufilmdb	a főoldalon megjelenő új filmek darabszáma

Relációs modell

Az előbbi ER-modell relációs modellé átalakítva látható a következő ábrán



A táblák és mezők felsorolása, valamint típusai:

dvd:

id	azonosító,elsődleges kulcs, int, automatikusan növekszik, not null
cim	a dvd címe, varchar(255) , not null
rovid_leiras	a film rövid leirása, varchar(255), default('Ehhez a dvd-hez nincs leírás')
hosszu_leiras	a film hosszú leírása(szerző, tartalom, extrák...), text,default(NULL)
elozetes	a film megtekinthető előzetesének a fájlneve, varchar(255),default(NULL)
ear	a dvd egységára, integer, not null
akcio	a dvd-hez rendelt akció %-a, int, default(0)
megj_datum	a dvd megjelenési dátuma, date, not null
mdb	a dvd-ből már ennyit vettek meg, int, default(0)
kep	a dvd-hez rendelt kép fájlneve, varchar(255), default(NULL)
pont	hány pontot kapott összesen a felhasználóktól, int, default(0)
mgj	egyéb megjegyzések a dvd-hez, text,default(NULL)
felv_datum	a rekord felvételének dátuma, date, not null
id_kategoria	idegen kulcs a kategoria táblára, integer, torles=set null, update=cascade

felhasznalo:

id	azonosító,elsődleges kulcs, int, automatikusan növekszik, not null
edb	a szűrés során a felhasználó által egy oldalon látni kívánt eredmények darabszáma, int, not null, default(10)
nev	a felhasználó neve, varchar(255) , not null
felhnev	felhasználói név(nick), varchar(255) , not null
jelszo	a jelszó, varchar(255) , not null
irszam	irányítószám, smallint,unsigned, default(NULL)
varos	város, varchar(50), default(NULL)
utca	utca, varchar(50), default(NULL)
hazszam	házzszám,smallint,unsigned, default(NULL)
tel	telefonszám,varchar(30) , default(NULL)
email	e-mail, varchar (100) , not null
mgj	egyéb megjegyzések, text, default(NULL)
reg_datum	a regisztráció dátuma, datetime, not null
allapot	0-inaktív, 1-aktív, smallint, not null, default(0)

rendeles:

id	azonosító,elsődleges kulcs, int, automatikusan növekszik, not null
allapot	a megrendelés állapota, varchar(20) , not null, default('felvéve')
nev	a felhasználó neve, varchar(255) , not null
irszam	irányítószám, smallint,unsigned ,not null
varos	város, varchar(50) , not null
utca	utca, varchar(50) , not null
hazszam	házzszám,smallint, unsigned, not null
ear	egységár, integer , unsigned, not null

db	a rendelt darabszám a dvd-ből, int,unsigned, not null, default(1)
mgj	egyéb megjegyzések, text, default(NULL)
id_dvd	idegen kulcs a dvd táblára, integer, not null, CASCADE
Id_felhasznalo	idegen kulcs a felhasznalo táblára, integer, not null, CASCADE
cim	a dvd címe, varchar (255), not null
datum	a rendelés dátuma, datetime, not null
azon	a rendelés azonosítója, varchar(255), not null

film toplista:

id	azonosító,elsődleges kulcs, int, automatikusan növekszik, not null
poz	pozíció a toplistában,smallint, not null
id_dvd	idegen kulcs a dvd táblára, integer, not null

rendelt toplista:

id	azonosító,elsődleges kulcs, int, automatikusan növekszik, not null
poz	pozíció a toplistában,smallint, not null
id_dvd	idegen kulcs a dvd táblára, integer, not null

kritika:

id	azonosító,elsődleges kulcs, int, automatikusan növekszik, not null
szov	a kritika szövege, text, default(NULL)
id_dvd	idegen kulcs a dvd táblára, integer, not null
hozzaszolo	a hozzászóló neve, varchar(255), not null
datum	a bejegyzés dátuma, datetime, not null

kategoria:

id	azonosító,elsődleges kulcs, int, automatikusan növekszik, not null
nev	a kategória neve, varchar(255), not null
id_dvd	idegen kulcs a dvd táblára, integer, not null

beallitasok:

id	azonosító,elsődleges kulcs, int, automatikusan növekszik, not null
afilmkezd	akciós dvd-knél az időintervallum kezdete, not null
afilmveg	akciós dvd-knél az időintervallum vege, not null
afilmdb	akciós dvd-knél a toplista bejegyzéseinek száma,smallint, not null
ufilmkezd	új dvd-knél az időintervallum kezdete, not null
ufilmveg	új dvd-knél az időintervallum vege, not null
ufilmdb	új dvd-knél a toplista bejegyzéseinek száma, not null

Session:

id	azonosító,elsődleges kulcs, int, automatikusan növekszik, not null
ido	a rekord létrehozásának ideje, timestamp, not null
adat	a session adatai, text, not null
felh_id	azon felhasználó azonosítója amihez a rekord tartozik
Sid	a sessuib azonosító, varchar(32), not null

Fejlesztői dokumentáció

Gépi és nyelvi igények

Windows XP rendszer. Olyan számítógép, amin az előbb említett operációs rendszerek futnak. Konfiguráció, amin fejlesztettem:

- processzor: Intel Celeron 800 MHZ
- memória: 384 MB
- videokártya: NVIDIA GeForce 2 MX/MX400
- egér
- billentyűzet

A szerver környezethez szükséges egy operációs rendszer, én Windows XP alatt fejlesztettem. Egy webszerver, én Apache 2.0-t használtam. Az értelmező a PHP 5.2.4. Az adatbáziskezelő a MySQL 5.0. Levelek fogadására és küldésére egy e-mail cím a programban a vmivmivmi@gmail.com –ot állítottam be, amit meg lehet változtatni. Valamint egy SMTP szerver, az oldal a smtp.datanet.hu-t használja erre a célra.

A következő fájlokat bemásoltam a windows\system32 –be:

```
php5apache2.dll
php5ts.dll
php_mysql.dll
libmysql.dll
libmysqli.dll
```

Az apache konfigurációs fájljában (httpd.conf) következő direktívát DirectoryIndex index.html –t átírtam DirectoryIndex index.php –ra.

A php.ini-ben pedig a session.use_trans_sid-t kell átállítani: session.use_trans_sid = 1. Ekkor a url_rewriter.tags-ben beállított összes html tag végére odateszi a session néve,azonosítót. párost. A session név az áruházban: PHPSESSID.

A módosított beállítások a php.ini-ben:

```
session.save_handler = user
session.use_cookies = 0
session.gc_probability = 1
session.gc_divisor = 1
```

A kliens oldalon szükséges egy böngésző program, én Internet Explorer 7.0.5730.13 athasználtam a fejlesztésnél. Ezenkívül szükséges van egy e-mail címre ami tud leveleket fogadni, gmail és freemail-el lett tesztelve.

Algoritmusok szintjeinek leírása

A webáruház több fájlból áll és könyvtárból áll. Három könyvtárt tartalmaz:

- elozetes: a dvd előzeteseket tárolja, kezdetben a próba előzetesek a következők:
 - a_tokeletes_pasi.avi
 - csillagkozi_invazio.avi
 - hell_ride.avi

- hihetetlen_hulk.avi
- leon_a_profi.avi
- White_Noise_2_-_The_light_trailer.avi
- include:az include-ált fájlokat tartalmazza:
 - adatok.php
 - opendb.php
 - session.php
- kep:a dvd-khez feltöltött képeket és a webáruház felületéhez szükséges képeket tartalmazza:
 - a_tokeletes_pasi.jpg
 - csillagkozi_invazio.JPG
 - dvdbolt_kep.JPG
 - feh_zaj2.jpg
 - hell_ride.JPG
 - incrediblehulk_poster2.jpg
 - leon_a_profi.JPG
 - ragadozok.JPG
 - ures.jpg

Az oldal gyökerében található a következő fájlok:

- admin_rendeles.php
- aktivalas.php
- bal.php
- beallitasok.php
- belepes.html
- belepes.php
- elfelejtett.html
- elfelejtett.php
- fejlec.php
- felh_kereses.php
- felh_torles.php
- filmek.php
- fo.php
- hozzaszolas.php
- index.php
- kereses.php
- kilepes.php
- kosar.php
- kozep.php
- level.php
- megrendeles.php
- pontozas.php
- profil.php
- reg.html
- reg.php
- rendelesek.php

- `reszlet.php`
- `reszlet_felh.php`
- `ujdvd.php`

1. `adatok.php`

A webáruház alap adatait tartalmazza.

Változók:

`$host` – a regisztrálás után a felhasználó a megadott e-mail címére kap egy levelet, amiben vagy megerősítheti vagy, törölheti a regisztrációját, ehhez két link lesz, a linkek generálásánál használja az oldal ezt a változót.

`$emailcim` – az oldal e-mail címe a levél küldés oldalon megírt üzenetek erre a címre érkeznek meg

2. `opendb.php`

Szinte minden oldalon használatos ez a fájl. Először csatlakozni próbál a mysql szerverhez a lokális gépen(`localhost`), `root` felhasználóként és a megadott jelszóval. Ha nem tud csatlakozni, akkor ezt közli és leáll a szkript végrehajtása. Siker esetén beállítja a használt adatbázist a webáruház adatbázisára, ami a `dvdbolt`. Majd beállítja, hogy milyen karakterkészletet használ a kliens és a szerver az egymás közti kommunikációban. A karakterkészlet itt `latin2`.

3. `session.php`

A webáruház session-öket használ az adatok tárolására globálisan, azaz az oldalak között az adatbázisban történő adattároláson kívül. Minden egyes oldalon ahol session kezelés szükséges beillesztődik ez a fájl, ami a session kezeléshez szükséges függvényeket tartalmazza. Az alapértelmezett session kezelési módszereket felülbírálja ezekkel a függvényekkel. A session adatait adatbázisba fogja menteni fájlrendszerbeli fájlok helyett. Minden felhasználó, aki az oldalra lép kap sessiont.

A függvények:

`open($save_path, $session_name)`: A törzse üres. A függvény akkor hívódik meg amikor a sessiont elindítottuk a `session_start()`, mint egy konstruktor.

`close()`: Amikor a session művelet lezajlik a szkript végén, meghívódik ez a függvény. Hasonló egy destruktorkhoz egy osztályban.

`function read($id)`: A session olvasásakor hajtódik végre ez a függvény. A `open()` futtatása után fut le ez a függvény. A visszatérési értékének egy stringnek kell lennie. Üres stringgel tér vissza, ha nincs session rekord, amit beolvasson, ha van, akkor pedig a session konvertált stringjével tér vissza. A paramétere a `$id` ami a session azonosítóját tartalmazza. Először indít egy tranzakciót, majd beolvassa a `session` táblából azt a sort, amely az adott session azonosítóhoz tartozik (`$id`). Ha valamilyen hiba történt a tranzakció indításkor vagy a lekérdezésnél, akkor azt jelzi a szkript és leáll, a lekérdezést hiba esetén megpróbálja vizsgálgatni, ha ez sem sikerül azt is jelzi és úgy áll le a futás. Ha nem volt hiba, akkor kommitálja a változtatásokat (ha itt hiba lesz, szintén leáll). Majd megvizsgálja, hogy hány ilyen sort talált, ha nincs ilyen sor, akkor üres stringgel tér vissza a függvény, ha nem 1 ilyen sort talált, akkor hiba történt, ezt jelzi és leáll a futás. Ha csak egy sor van, akkor visszatér a `session` tábla `adat` mezőjének értékével.

`function write($id, $sess_data):` A session adatok adatbázisba írásáért felelős függvény. A session adatok kiírása az aktuális szkript végéig nem történik meg. Egészen addig, amíg a kimeneti folyam le nem zárult a böngészőnek. A paraméterei a `$id` a session azonosító és a `$sess_data` a session adata. Mindkét paraméter string. Először beolvasásra kerül a session azonosítóhoz tartozó összes adat az adatbázis rekordból. Ha nincs ilyen rekord, akkor a függvény létre fogja hozni. Ha olyan felhasználó látogatja az oldalt, aki nincs regisztrálva, azaz nincs felhasználói azonosító a session-ben tárolva akkor az insert parancsnál a felhasználói azonosító mező (`felh_id`) értéke NULL lesz, ellenkező esetben a sessionben tárolt felhasználói azonosító `$_SESSION['felh_id']` lesz. Ezenkívül tárolódik az aktuális idő az `ido` mezőbe, és a session azonosító is a `sid` mezőbe és a session adatok (`$sess_data`) is tárolódnak a `varchar` típusú mezőben (`adat`). Majd a függvény `true`-t, azaz igazat ad visszatérési értéként. Ha már van ilyen azonosítójú session rekord akkor az csak `ido` frissül az adott időre és session adat egyetlen `update sql` paranccsal. Majd a függvény szintén igaz eredménnyel tér vissza. Ha a fenti két eset egyike sem történik akkor hibával leáll a végrehajtás.

`function destroy($id):` Akkor hajtódik végre amikor a sessiont törlik egy `session_destroy()` függvénnyel. A session adatai az adott scriptben ettől még használhatók.

Egyetlen paramétere a session azonosítóját adja. A függvény először nyit egy tranzakciót (hiba esetén jelzi és leáll a végrehajtás), majd végrehajtja a `$id` azonosítójú session törlését egy `delete sql` utasítással. Ha valami hiba történik akkor megpróbálja visszagörgetni, üzenetet küld a hibáról és leállítja a végrehajtást. Ha a visszagörgegnél hiba lép fel akkor szintén üzen és leállítja a végrehajtást. Sikeres törlésnél kommitál és igazzal tér vissza a függvény.

`function gc($maxlifetime):` A session kezelő szemétygyűjtő függvénye. Ha egy session valamilyen oknál fogva nem szűnik meg, akkor adott idő múltán, ha valaki oldalt kér le, amiben van session kezelő kód akkor végrehajtásra kerül adott valószínűséggel ez a hulladékkezelő kód. A `$maxlifetime` tartalmazza a lejáratási idő nagyságát, ami alapértelmezésként `1440 mp` azaz `24 perc`. A valószínűséget `session.gc_probability/session.gc_divisor` `php.ini`-beli értékek hányadosa határozza meg. Ez a függvény minden lejárt idejű session rekordot töröl. A függvény tranzakciót indít, majd törli az összes olyan session rekordot, ami elavult. Ehhez a session tábla `ido` mezőjéhez hozzáadja a `$maxlifetime` -ot ha ez kisebb, mint az aktuális idő akkor törli a rekordot. Ez függvény nem figyeli, hogy kinek a session-je, töröl minden ilyen rekordot. Bármilyen hiba esetén jelzi a hibát a kód és megállítja a futást.

Végül tartalmazza azt a sort amiben a `session_set_save_handler` függvénnyel beállítja a fenti callback függvényeket.

4. admin_rendeles.php

Változók:

`$ki`: Egy string típusú változó. Az `uzenet` nevű függvény használja a kimenet tárolására.

`$er`: Erőforrás típusú változó. Szerte a kódban ez a változó kapja meg a `mysql_query()` függvények által visszaadott erőforrásokat.

\$azon: String típusú. A rendelések egyedi azonosítóját tárolja. Az összehasonlításokhoz a kirak nevű függvényben.

\$j, \$i: Számlálásokra, ciklusokban és ideiglenes integer értékek tárolására használatos változók.

\$osszeg: A kirak függvényben használatos. Az egyes rendelések összértékének tárolására, ill. számítására

\$volt: A kirak függvényben használatos, egész típusú változó. Kapcsolóként, egy dolog első bekövetkezését mutatja.

\$datum: A kirak függvényben használatos, string típusú változó. Az egyes rendelési tételek lekérdezésekor mindig az előző rendelés dátumát tárolja.

\$allapot: A kirak függvényben használatos, string típusú változó. Az egyes rendelési tételek lekérdezésekor mindig az előző rendelés állapotát tárolja.

\$sor: Asszociatív tömb. Szerte a kódban a mysql_fetch_array függvény által lekért rekord mezőit tárolja.

\$kimenet: String típusú. A html kimenetet tároló globális változó.

A következő változók minden függvényen kívül használatosak.

\$dkezd, \$dveg: String változók. Az oldal dátum szerinti szűrő intervallumának kezdeti értékeit tárolják.

\$felhnev: String változó. Az oldal felhasználó név szerinti szűrő kombinált dobozának alapértelmezett értékét tárolja.

\$select: Integer változó. Tárolja, hogy melyik érték az alapértelmezett az állapot szűrésének kombinált listájában.

\$sql: String típusú. Az egyes rendeléseket összeállító lekérdezést tároló változó.

\$kulcs, \$ertek: String ill. egész típusú változók melyek foreach ciklusban használatosak.

Függvények:

`function uzenet($szov, $hova):` Egy üzenetet ír ki az oldalra. Két paramétere van. A \$szov a kiíradó szöveget adja meg. A \$hova pedig, hogy hová vigye a felhasználót a Vissza felíratú nyíl. A függvény a \$ki változóba gyűjti a html kimenetet. A vissza felíratú link kimenetének generálásakor ellenőrzi, hogy a \$hova a fő oldalra mutat-e, mert ha igen akkor az ablak teljes tartalma fog frissülni, azaz beállítja a link target tulajdonságát `_top`-ra. A függvény miután befejezte a kimenet generálást, kilép.

`function kirak(& $kimenet, $sql):` Az egyes rendelések adatainak kimenetét írja hozzá a \$kimenet globális változóhoz. Paraméterek a \$kimenet ami címszerinti melybe fogja bele írni a függvény által tárolt kimenetet és \$sql amelynek megfelelő lekérdezés által kapott rendelések adatait fogja formázottan beleírni a \$kimenet-be. A függvény először tranzakciót nyit, majd lefuttatja a \$sql-ben tárolt lekérdezést. Ha hiba történt valahol akkor azt kiírja és megszakítja a futást. Majd kezdőértékeket ad változóknak. A függvény úgy működik, hogy egy nagy ciklusban elkezd lekérni a lekérdezés által lekért adatbázis rekordokat a \$sor változóba. Minden olyan rendelési tételt egy táblázatba fog írni azokkal a tételekkel, amiknek ugyanaz a rendelés azonosítója. Ezt a cikluson belül a rendelési azonosító és az \$azon változó összehasonlításával végzi. Ha nem egyeznek akkor új rendelésről van szó és új táblázat fejet és rendelési adatokat

generál, valamint lezárja az előző táblázatot az \$összeg és \$j változók 0 értéket vesznek fel. Kivétel az az eset, amikor belépünk a ciklusba, mert akkor az egész kimenetet máshogy kell generálni, ennek a jelzésére szolgál a \$volt változó, ami utána 1 értéket kap. Ha egyezik, akkor ugyanarról a rendelésről van szó és csak generálja a rendelés adatait a megfelelő html táblázat soraiba. Számolja az összárát azáltal, hogy mindig hozzáadja a rendelés részlet összegét az addigi \$összeg változóhoz, a részlet összeg a mennyiség mező szorozva az egység ár mezővel. Majd végül frissíti a \$datum, \$allapot és \$azon változók értékeit az aktuális rekord megfelelő mezőire. A megjelenített adatoknál a megrendelő neve, a város, utca, és a dvd címe mező értékeket 40 karakter szélesre vágja le. A tétel sorszámát a html táblázatban a \$j jelzi a kimenetben. A törlésre és leszállításra jelölő html checkboxokat úgy generálja a függvény, hogy a törlők name tulajdonsága „x” karakter és a \$azon összeillesztésével kap értéket, a leszállításra megjelölőké pedig „y” karakter és a \$azon összeillesztésével kap értéket. Ezen jelölő dobozoknál vizsgálja, hogy \$_POST['kij_torles'] törlő dobozoknál és \$_POST['kij_leszallitas'] a leszállításra jelölő dobozoknál kerülnek kiértékelésre, ha értékük van, akkor az azt jelenti, hogy bejelölte az összes rendelést bejelölte valaki leszállításra ill. törlésre és így ennek megfelelően lesz pipa a jelölődobozokban.

A függvényeken kívül eső kód:

A fájl elején beágyazza a include/opendb.php és include/session.php fájlokat. Ezzel megnyílik az adatbázis kapcsolat ill. biztosítva van a session kezelés. Majd megnyitja a sessiont a session_start()-al. A kereső/szűrő űrlap minden esetben látszani fog az oldal tetején, ha kerestünk már, ha nem, az aktuális ill. előző keresési/szűrési eredményekkel.

Vizsgálatra kerül, hogy a session-ben a felhasználó az admin-e, ha nem akkor üzen erről és átirányítatja a főoldalra egy linkkel. Ezek után alapértelmezéseket állít be. Majd ha már az űrlapot elküldték, azaz létezik a \$_POST['kereses'] akkor ellenőrzi, hogy helyes dátumidő formátumú értékeket adtak-e meg (éééé-hh-nn óó:pp:mm) a szűrő űrlapon form1. Azzal, hogy a \$_POST szuperglobális tömb megfelelő értékeit átalakítja az strtotime függvénnyel php timestamp-é, ha nem sikerül és egyáltalán az űrlap ezen értékét kitöltötték akkor üzenetet ír, hogy nem megfelelő a dátum formátuma. Majd ha helyes a forma akkor a \$dkezd, \$dveg változók megkapják a \$_POST beli változatukat értéként. A \$select is itt kapja meg, hogy milyen kezdeti értéket tároljon az html allapot kombinált listához.(1="mind",2="felvéve",3="leszállítva"), mely értékek lesznek majd az alapértelmezések a kombinált listához. A \$felhnev is itt kapja meg, hogy mi legyen az alapértelmezett felhasználó név a felhasználók kombinált listájában.

Ha valaki megnyomta a törlés gombot valamelyik rendelésnél, akkor egy foreach ciklussal végigjárja a \$_POST-ot a php. Egy delete sql utasítást épít fel, amelyet a \$sql-ben tárol. Ha talál egy „x”-el kezdődő kulcsot akkor levágja a kulcsból a rendelésazonosítót, és hozzá illeszti a törlendő rendelések közé az \$sql változóba „or”-al összekapcsolt azonosító listába. Majd ha az \$sql nem üres akkor végrehajtja a lekérdezést. Ha hiba történt kiírja, majd megállítja a futást. Valamint siker esetén megjeleníti a törölt rendelések számát. Ha az \$sql üres akkor jelzi, hogy nem történt törlés.

Ha valaki megnyomta a törlés gombot valamelyik rendelésnél akkor hasonló dolog történik, mint az előbb csak ott „y” prefixeket keres és update utasítást állít össze, melyben „leszállított”-ra állítja a kiválasztott rendeléseket.

A törlés és leszállítás esetén a kód nem folytatódik. Ellenben elkezd a \$kimenet változóba a kimenet generálását. Generálja a kereső űrlapot (form1) aminek az adatait ugyanez a fájl fogja feldolgozni. Majd a vezérlőket rajta. Felhasználja a \$select, \$dkezd, \$dveg, \$felhnev változókat is. Majd történik egy lekérdezés a felhasználó nevekért. Itt is, ha hiba történik, akkor jelzi a kód, majd leáll a futás. Aztán generálja a 2. űrlapot (form2) aminek az értékeit szintén ez a fájl dolgozza fel.

Majd vizsgálja, hogy ha elküldték-e a kereső űrlapot. Ha igen akkor a keresés/szűrés adatait állítja össze egy select sql utasításhoz. Ha a felhasználó kombinált listában nem a „mindenki”-t választották, akkor lekérdezi a kód a felhasználó táblából az adott név elsődleges kulcsát, majd hozzáilleszti a \$sql változóhoz. Majd vizsgálja a dátumidő intervallumot, ha mindkét értéket megadták az űrlapon, akkor between-el kapcsolja össze őket az sql utasításban, ha csak egy érték van megadva, akkor relációs jellel vizsgálja, hogy a szűrésben csak a nagyobb ill. kisebb dátummal rendelkező rendelések jelenjenek meg. Ha semelyik értéket nem adta meg akkor nem történik dátumra szűrés. Majd ha az állapotra nem a „mind”-et választották ki az űrlapon akkor a kiválasztott értéket („felvéve” vagy „leszállítva”) illeszti az \$sql-hez. Ha bármireszűrés történt akkor azt a \$i változó 1-re állításával jelöli a kód. Ha semmilyen feltételt nem állítottak be akkor egy egyszerű select-et tesz a \$sql-be. A szűrést mindenképpen rendezi az oldal rendelés azonosító, dátum majd a kisbetűsre alakított dvd cím szerint. Ezután meghívja a kirak függvényt az aktuális kimenettel és az \$sql változóval. Majd amikor a függvény visszatér, hozzáilleszti még az oldal lezárását, majd kiírja a \$kimenet változót, azaz elküldi a html kimenetet.

5. aktivalas.php

Változók:

\$hiba: String. Hiba esetén a html kimenete
\$igen: Sting. Ha a felhasználó az aktiváló linkre kattintott.
\$nem: Sting. Ha a felhasználó a törlő linkre kattintott.
\$er: Erőforrás típusú változó. Szerte a kódban ez a változó kapja meg a mysql_query() függvények által visszaadott erőforrásokat.
\$rekszam: Integer. Egy sql lekérdezés által visszaadott rekordok számát tárolja benne az oldal.

A függvényeken kívül eső kód:

A fájl először megjeleníti a kezdeti html kódot (pl beállítja a karakterkészletet latin2-re). Majd következik a php kód. Beágyazza a include/opendb.php és include/session.php fájlokat. Ezzel megnyílik az adatbázis kapcsolat ill. biztosítva van a session kezelés. Majd megnyitja a sessiont a session_start()-al. Majd megkapják a megfelelő html kimenetet az \$igen, \$nem, \$hiba változók. Ezután ellenőrzi, hogy url-ben megkapta-e a szkript a felhasználó nevet(\$_REQUEST['felhnev']) és az állapotot (\$_REQUEST['allapot']), ha nem akkor kiírja a \$hibaváltozót és befejezi a futását.

Majd indít egy tranzakciót és egy lekérdezést az url-ben megkapott felhasználó névre. Ha valahol hiba történik, akkor üzen és befejezi a futását a szkript. Eztán a lekért rekordok számát ellenőrzi, ha nem talált semmit, akkor kiírja a \$hiba értékét és megáll a futás.

A következőkben elindít egy tranzakciót, majd lekérdezi, hogy `$_REQUEST['allapot']` megegyezik-e 1-el. Ha igen akkor a felhasználó aktiválta a loginját, ekkor egy sql update utasítással a `$_REQUEST['felhnev']`, felhasználó nevű rekordnál át állítja az állapotot 1-re azaz aktiváltra, a where záradékban kisbetűs egyenlőséget vizsgál a szkript. Majd kommitálja a tranzakciót. Ha valami hiba történt akkor rollback-elni próbál és jelzi a hibát, majd megáll a futás. Sikereség esetén az `$igen` változó értékét írja ki, ami tartalmaz egy linket a belépési oldalra.

Ha nem egyezik meg 1-el akkor a felhasználó törölte a regisztrációt, akkor nem update-el hanem törli az adott felhasználó nevű rekordot, a where záradékban kisbetűs egyenlőséget vizsgál a szkript. Itt szintén vizsgálja az esetleges hibákat, ha minden rendben megy, akkor a `$nem` változót írja ki és befejezi a végrehajtást.

6. bal.php

Változók:

`$ki`: Egy string típusú változó. Az `uzenet` nevű függvény használja a kimenet tárolására.

`$menu_felh`: String. A regisztrált felhasználó fogja ezt a html kimenetet látni.

`$menu_nfelh`: String. A még nem regisztrált felhasználó fogja ezt a html kimenetet látni.

`$menu_admin`: String. Az adminisztrátor fogja ezt a html kimenetet látni.

`$i`: Integer. Segéd változó.

`$sor`: Asszociatív tömb. Szerte a kódban a `mysql_fetch_array` függvény által lekért rekord mezőit tárolja.

`$er`: Erőforrás típusú változó. Szerte a kódban ez a változó kapja meg a `mysql_query()` függvények által visszaadott erőforrásokat.

`$rekszam`: Integer. Egy sql lekérdezés által visszaadott rekordok számát tárolja benne az oldal.

Függvények:

`function uzenet($szov, $hova="fo.php")`: Egy üzenetet ír ki az oldalra. Két paramétere van. A `$szov` a kiíradó szöveget adja meg. A `$hova` pedig, hogy hová vigye a felhasználót a Vissza felíratú nyíl. Ennek a paraméternek az alapértelmezett értéke a „fo.php”

A függvény a `$ki` változóba gyűjti a html kimenetet. A vissza felíratú link kimenetének generálásakor ellenőrzi, hogy a `$hova` a fő oldalra mutat-e, mert ha igen akkor az ablak teljes tartalma fog frissülni, azaz beállítja a link target tulajdonságát `_top`-ra. A függvény miután befejezte a kimenet generálást, kilép.

`function toplista($tabla, $szov)`: Generálja a toplistát. Két paramétere van. A `$tabla` mutatja meg, hogy melyik táblából kérdezzen le. `$szov` adja meg, hogy mit írjon a táblázat fejlécébe. A függvény közvetlenül a html kimenetbe ír. A html tábla 2 oszlopot tartalmaz az elsőben a sorszámot írja ki (`$i` alapján 1-5), a másodikban a film címét. A lekérdezésben ehhez 2 táblát join-ol össze a `$tabla` és a `dvd` táblát. Ha valahol hiba történt a lekérdezésben, akkor azt jelzi és a szkript futása megáll. Minden eredmény rekordra a táblázatban a film címe link tag-

ek közé lesz téve, melyek a `reszlet.php` fájlra fognak mutatni és annak url-ben átadják a lekérdezett dvd azonosítóját.

A függvényeken kívül eső kód:

Beágyazza a `include/opendb.php` és `include/session.php` fájlokat. Ezzel megnyílik az adatbázis kapcsolat ill. biztosítva van a session kezelés. Majd megnyitja a sessiont a `session_start()`-al.

A függvények definiálása után ellenőrzi, hogy a url-ben kapott-e `felh` nevű változót (`$_GET['felh']`) majd ha igen ellenőrzi, hogy ez szám-e. Ha valamelyik feltétel nem teljesül, akkor az `uzenet` függvény kimetetését írja ki, ami a főoldalra eléréséhez ad egy linket.

A `$_GET['felh']` értékétől függően (0,1,2) rendre a `$menu_nfelh`, `$menu_felh`, `$menu_admin` változókat írja ki. Majd kiírja a `film_toplista` táblával a „Legjobb filmek” fejlécű és `rendelt_toplista` táblával a „Legkelendőbb filmek” html táblával készült toplistát.

7. beallitasok.php

Változók:

`$ki`: Egy string típusú változó. Az `uzenet` nevű függvény használja a kimenet tárolására.

`$rekszam`: Integer. Egy sql lekérdezés által visszaadott rekordok számát tárolja benne az oldal.

`$kimenet`: String típusú. A html kimenetet tároló globális változó.

`$er`: Erőforrás típusú változó. Szerte a kódban ez a változó kapja meg a `mysql_query()` függvények által visszaadott erőforrásokat.

`$sor`: Asszociatív tömb. Szerte a kódban a `mysql_fetch_array` függvény által lekért rekord mezőit tárolja.

`$sql`, `$sql2`: String. Sql lekérdezéseket tartalmazó változók.

Függvények:

`function uzenet($szov, $hova="fo.php")`: Ez teljesen megegyezik a `bal.php`-ben levő `uzenet` függvénnyel.

A függvényeken kívül eső kód:

Hasonlóan mint eddig beágyazza a `include/opendb.php` és `include/session.php` fájlokat. Ezzel megnyílik az adatbázis kapcsolat ill. biztosítva van a session kezelés. Majd megnyitja a sessiont a `session_start()`-al.

Majd ellenőrzi, hogy a session tartalmazza-e a `felhnev`, `jelszo`, `felh_id` változók és a `felhnev` az „admin” szöveget tartalmazza-e. Ha nem akkor jelzi, hogy csak adminisztrátornak érhető el az oldal, linket ad a főoldalra, majd befejezi a szkript a futását.

Eztán egy tranzakció indul és lekérdezi a szkript a `beallitasok` tábla egyetlen sorát. Ha hiba történik valahol akkor erről figyelmeztet, majd a futás megáll. Siker esetén ellenőrzi, hogy valóban 1 sort tartalmaz-e a tábla a `$rekszam` segítségével. Ha nem akkor erről tájékoztat, majd leáll a szkript futása.

Siker esetén elkezd generálni az űrlapot a \$kimenet-be az űrlapot szintén ez a fájl fogja feldolgozni. Ha felhasználó nem küldte el az űrlapot még akkor csak kiírja a kimenetet és befejezi a szkript futását. Ha az űrlapot már kitöltötték, akkor folytatódik a végrehajtás. Leellenőrződik. Hogy létezik-e a felhasználó az adatbázisban. Ha valamilyen hiba történik, akkor a tájékoztatást kapunk a hibáról, és a végrehajtás megáll, ill. akkor is üzen a szkript ha, nem létezik a felhasználó, ilyenkor egy linket kapunk a főoldalra.

A következőkben ellenőrzi, hogy az űrlap megfelelő mezői ki vannak-e töltve (ufilmkezd, afilmkezd, ufilmveg, afilmveg, jelszo), ha nem akkor figyelmeztet, hogy ezeket kötelező kitölteni és visszairányít egy linkkel a beállításokhoz az uzenet függvényvel. Majd ellenőrzi, hogy a darabszámot váró mezőkbe egészet írtak-e és hogy 1-10-ig veszik-e fel az értéküket. A csak számok bevitelét reguláris kifejezéssel ellenőrzi a szkript. Ha nem megfelelőek az értékek, akkor hibaüzenetet kapunk. Eztán a jelszó mezőt kódolja md5 kódolással, ha ez nem egyezik az adatbázisbeli értékkel szintén hibaüzenetet ír és visszairányít.

Ezután tranzakciót indít, törli a beallitasok tábla egyetlen rekordját, majd generál az űrlap adatával egy insert parancsot tartalmazó lekérdezést, amit a \$sql-be tesz. Ha a jelszo2 mezőt is kitöltötték, akkor generálja az \$sql2 változót, ami a felhasználó táblában az adminisztrátornál átírja egy sql update paranccsal a jelszót, md5-el kódolva. Majd végrehajtja az insert-et. Hiba esetén próbál rollback-elni ha nem sikerül, akkor üzen és a szkript nem folytatja a futását tovább. Siker esetén és jelszó módosításkor az update-et is végrehajtja a hiba kezelés hasonló az előbbihez. Majd végül sikeres műveletek esetén kommitálódnak a változtatások, és egy üzenetet küld a sikerességről a szkript egy főoldalra irányító linkkel együtt.

8. belepes.html

Egy statikus html oldal, ami a belépés űrlapját tartalmazza. Az elfelejtett jelszó és a regisztrálás oldalára vezető linkkel. Az űrlapot belepes.php fájl fogja feldolgozni.

9. belepes.php

Változók:

\$rekszam: Integer. Egy sql lekérdezés által visszaadott rekordok számát tárolja benne az oldal.

\$er: Erőforrás típusú változó. Szerte a kódban ez a változó kapja meg a mysql_query() függvények által visszaadott erőforrásokat.

\$sor: Asszociatív tömb. Szerte a kódban a mysql_fetch_array függvény által lekért rekord mezőit tárolja.

\$felh_id: Integer. A felhasználó elsődleges kulcsát fogja tárolni.

\$i: Integer. Számolásra.

Függvények:

function hiba(\$szov): Egy üzenetet ír ki az oldalra, adott html környezetbe ágyazva.

Két paramétere van. A \$szov a kiírandó szöveget adja meg. A \$hova pedig, hogy hová vigye a felhasználót a Vissza a belépés ablakhoz feliratú link. A link a belepes.html oldalra irányít. A függvény közvetlenül a kimenetre ír.

A függvényeken kívül eső kód:

Hasonlóan, mint eddig beágyazza a `include/opendb.php` és `include/session.php` fájlokat. Ezzel megnyílik az adatbázis kapcsolat ill. biztosítva van a session kezelés.

Először is ellenőrzi, hogy a kitöltötték-e a `felhnev` mezőt az űrlapon, ha nem akkor üzen és visszairányít, az űrlapra egy linkkel. Aztán tranzakciót indít és lekéri a `felhasznalo` táblából, hogy létezik-e ilyen felhasználó nevű és jelszavú felhasználó, a `sql where` záradékban kisbetűsen hasonlít össze a jelszó esetén, pedig `md5`-el kódolja az űrlapról kapott jelszavat és így ellenőrzi. Ha hiba történt a lekérdezés vagy tranzakció indításnál jelzi és befejezi a futást a szkript. Siker esetén ellenőrzi a `$rekszam` változóval, hogy hány ilyen felhasználó van. Ha nincs ilyen, akkor üzen (`hiba` fgv) és átirányít egy linkkel.

Majd lekérdezi, hogy hány ugyanilyen felhasználói azonosítójú session van a `session` táblában, hiba esetén üzen és kilép. Ha van már ilyen rekord, akkor az ilyen rekordokat törli. Hiba esetén hasonló zajlik, mint előbb.

Ezután, ellenőrzi, hogy admin lépett-e be, ha igen akkor törli a `film_toplista` és `rendeles_toplista` táblákat, majd lekérdezi az 5 legtöbb pontot kapott filmet, ill. az 5 legtöbbet rendelt filmet. A lekérdezések eredményeit egy tranzakciónyitással megelőzve beírja a megfelelő `film_toplista` ill., `rendeles_toplista` táblába. Az insert-eknél a `$i` értéket, mint sorszámot a `poz` mezőnél felhasználja. Az esetleges hibáknál üzen és kilép, vagy átirányít a szkript.

Majd megnyitja a sessiont a `session_start()`-al, beleírja a `felhnev`, a `jelszo` és `felh_id` mezőket és értéküket és átirányít automatikusan a `fo.php` oldalra, persze generálja hozzá a session azonosítót is urlben.

10. elfelejtett.html

Statikus html oldal. Az elfelejtett jelszó funkció űrlapját tartalmazza. Az űrlapot az `elfelejtett.php` szkript fogja feldolgozni.

11. elfelejtett.php

Változók:

`$hiba`: String. Hibát közöl, html környezetbe ágyazva. Az űrlapra irányító linket tartalmaz.

`$level_szov`: String. A felhasználónak küldött levél html kimenetét fogja tartalmazni.

`$siker`: String. Az új jelszó generálásának sikerességét jelző html kimenet.

`$ujjelszo`: String. Az új jelszót tárolja majd.

`$femail`: Asszociatív tömb. Szerke a kódban a `mysql_fetch_array` függvény által lekért rekord mezőit tárolja.

`$felhnev`: String. A levél szövegének megkonstruálásában felhasznált változó.

A függvényeken kívül eső kód:

Első lépésben ellenőrzi, hogy a `felhnev` és `email` mezők egyáltalán léteznek-e azaz a felhasználó nem közvetlenül jutott erre az oldalra, majd azt is, ha ezek a html mezők nincsenek kitöltve, azaz üres stringet tartalmaznak. Mindkét esetben kiírja a `$hiba` értékét és kilép.

Majd beágyazza a `include/opendb.php` fájtt ezzel kapcsolatot nyitva a MySQL szerver felé. Majd lekérdezi az email és jelszó mezőket a megadott felhasználó nevű rekordot a `felhasznalo` táblából és a lekért sorokat zárolja írásra. A `where` záradékban kisbetűs összehasonlítást végez a felhasználó névre. Majd megvizsgálja a lekért rekordok számát (`$rekszam` változóval), ha nincs találat, akkor ismét kiírja a `$hiba-t` és befejezi a futást. Majd lekérdezi az `$femail`-be a rekord mezőjét, majd összehasonlítja az űrlapon talált email mezőt az adatbázisbelivel, ha nem egyezik (kisbetűs összehasonlítással) akkor szintén `$hiba-t` írja ki és kilép.

Ha minden rendben, akkor új jelszót generál az adatbázis `jelszo` mezőjéből az `md5` függvény segítségével és az `$ujjelszo`-ba teszi. Majd új tranzakciót nyit és felülírja a jelszót az új jelszóval. Hiba esetén `$hiba-t` küldi el a kimenetre és kilép. Ha az `update` nem megy végbe vissza próbálja görgetni a tranzakciót. Majd ha még ebben is hiba történt, üzen erről is, de mindenképpen megáll a végrehajtás.

Majd a `$ujjelszo`, `$felhnev` változók, és `mail` függvény segítségével megpróbál levelet küldeni az űrlapon megadott e-mail címre. A levél típusa `html/text` így `html` levélként megy el, amit küld.

Ha a levelet nem sikerült elküldeni, akkor a jelszó felülírás rollback-elődik. Majd üzenetet kap a felhasználó a sikertelenségről. Ha rendben zajlottak a dolgok, akkor kommitálódik a tranzakció és itt szintén kap üzenetet a felhasználó a sikerességről!

12 fejlec.php

Változók:

`$kimenet`: String. Az oldal `html` kimenetét tárolja.

`$er`: Erőforrás típusú változó. Szerte a kódban ez a változó kapja meg a `mysql_query()` függvények által visszaadott erőforrásokat.

`$rekszam`: Integer. Egy `sql` lekérdezés által visszaadott rekordok számát tárolja benne az oldal.

A függvényeken kívül eső kód:

Beágyazza a `include/opendb.php` és `include/session.php` fájlokat. Ezzel megnyílik az adatbázis kapcsolat ill. biztosítva van a `session` kezelés. Elkezd írni a kimenetet a `$kimenet` változóba. Majd megnyitja a `session` a `session_start()`-al.

Aztán ellenőrzi, hogy a `$_SESSION`-ben léteznek-e a `felhnev`, `jelszo`, `felh_id` változók. Ha nem akkor a felhasználó nem lépett be és ezt jelzi a „Regisztrálatlan” kiírással is a `$kimenet`-ben majd megjeleníti a kimenetet és véget ér a szkript. Ha léteznek akkor ellenőrzi, hogy az adatbázisban is létezik-e ilyen mező értékekkel bíró rekord. Ehhez tranzakciót nyit, majd megírja a lekérdezést, a `where` záradékban kisbetűs `felhnev` ellenőrzést használ, valamint a jelszó `md5`-el kódolva hasonlítja össze, valamint olyan rekordokat keres ahol az `allapot` értéke 1, azaz aktivált felhasználó. Ha sikerül ilyen rekordot találni, akkor egy „1” értékű oszlopot ad eredményként. Majd a lekérdezésnél megvizsgálja az eredmény rekordok számát, ha ez 0, akkor visszairányítja automatikusan a főoldalra a felhasználót, de a `session` azonosítót is kapcsolja az `url`-hez. és a szkript befejeződik. Sikertelenségénél hibát ír ki a lekérdezésnél is és be is fejezi a végrehajtást.

Sikeres lekérdezés esetén megvizsgálja, hogy adminisztrátor lépett-e be, ha igen akkor az „admin” kiírást jeleníti meg a fejlécben. Ha nem akkor egy bejelentkezett felhasználó kérte

le ezt az oldalt. Ha kell, akkor 30 karakterre vágja le a felhasználó nevet. Majd végül a mostani esetben is kiírja a \$kimenet-et a html-nek.

13. felh_kereses.php

Változók:

\$ki: Egy string típusú változó. Az `uzenet` nevű függvény használja a kimenet tárolására.

\$dkezd, \$dveg: String változók. Az oldal dátum szerinti szűrő intervallumának kezdeti értékeit tárolják.

\$felhnev: String változó. Az oldal felhasználó név szerinti szűrő kombinált dobozának alapértelmezett értékét tárolja.

\$select: Integer változó. Tárolja, hogy melyik érték az alapértelmezett az állapot szűrésének kombinált listájában.

\$er: Erőforrás típusú változó. Szerte a kódban ez a változó kapja meg a `mysql_query()` függvények által visszaadott erőforrásokat.

\$sor: Asszociatív tömb. Szerte a kódban a `mysql_fetch_array` függvény által lekért rekord mezőit tárolja.

\$kimenet: String. Az oldal html kimenetét tárolja.

\$i: Integer. A felhasználók számlálásának megjelenítésére a html kimeneten.

\$sql: String típusú. Az felhasználói listát előállító lekérdezést tároló változó.

\$szam: Integer. A lekért felhasználók számát tárolja.

Függvények:

`function uzenet($szov, $hova="fo.php")`: Egy üzenetet ír ki az oldalra. Két paramétere van. A \$szov a kiíradó szöveget adja meg. A \$hova pedig, hogy hová vigye a felhasználót a Vissza felíratú nyíl. Ennek a paraméternek az alapértelmezett értéke a „fo.php”

A függvény a \$ki változóba gyűjti a html kimenetet. A vissza felíratú link kimenetének generálásakor ellenőrzi, hogy a \$hova a fő oldalra mutat-e, mert ha igen akkor az ablak teljes tartalma fog frissülni, azaz beállítja a link target tulajdonságát `_top`-ra. A függvény miután befejezte a kimenet generálást, kilép.

A függvényeken kívül eső kód:

Beágyazza a `include/opendb.php` és `include/session.php` fájlokat. Ezzel megnyílik az adatbázis kapcsolat ill. biztosítva van a session kezelés. Elkezdí írni a kimenetet a \$kimenet változóba. Majd megnyitja a sessiont a `session_start()`-al.

Ez után leellenőrződik, hogy az adminisztrátor lépett be az oldalra vagy egyáltalában is beléptek (`$_SESSION['felhnev']` létezik), ha nem akkor üzen az `uzenet()` fgv visszatérési értékének kiírásával, ami egy linket ad a `fo.php` oldalra, majd befejeződik a szkript futása.

Majd kezdőértékeket kapnak a \$dkezd, \$dveg, \$felhnev, \$select változók.

Eztán ha már az űrlapot elküldték, azaz létezik a `$_POST['kereses']` akkor ellenőrzi, hogy helyes dátumidő formátumú értékeket adtak-e meg (éééé-hh-nn óó:pp:mm) a szűrő űrlapon (`form1`). A szűrő űrlap mindig látszódni fog, ha szűrtek már nem szűrtek

még. Azzal, hogy a `$_POST` szuperglobális tömb megfelelő értékeit átalakítja az `strtotime` függvényvel `php timestamp`-é, ha nem sikerül és egyáltalán az űrlap ezen értékét kitöltötték, akkor üzenetet ír, hogy nem megfelelő a dátum formátuma, majd az ugyanezre az oldalra mutató linket kap a felhasználó. Majd ha helyes a forma, akkor a `$dkezd`, `$dveg` változók megkapják a `$_POST` beli változatukat értéként. A `$select` is itt kapja meg, hogy milyen kezdeti értéket tároljon az `html` állapot kombinált listához. (1="Mind", 2="Aktivált", 3="Nem aktivált"), mely értékek lesznek majd az alapértelmezések a kombinált listához. A `$felhnev` itt kapja meg, hogy mi legyen az alapértelmezett felhasználó név a felhasználók kombinált listájában.

Ezután elkezd generálni a `$kimenet` változóba a `html` kimenetet. Az `html` állapot mező kezdeti értékének megadásánál felhasználja a `$select` értékét.

Majd a felhasználók kombinált listájánál indít egy tranzakciót, majd lekérdezi az összes felhasználó nevet `felhnev` majd `reg_datum` szerint rendezve. Ezek után egy ciklusban sorra lekérdezi lekérdezés sorainak értékeit, majd a `$felhnev` segítségével beállítja az alapértelmezett értéket a felhasználói nevek kombinált listában.

Majd a `$kimenet`-be elkezd létrehozni a törlő űrlapot (`form2`) ami a `felh_torles.php` oldalra küldi el az űrlap adatait feldolgozásra. Ez után vizsgálja, hogy ha elküldték-e a kereső űrlapot. Ha igen akkor a keresés/szűrés adatait állítja össze egy `select sql` utasításhoz. A dátumidő mezőket csak dátumra konvertálja. Ha a felhasználó kombinált listában nem a „mindenki”-t választották, akkor hozzáilleszti ezt a felhasználó nevet a `$sql` változóhoz. Majd vizsgálja a dátumidő intervallumot, ha mindkét értéket megadták az űrlapon, akkor `between`-el kapcsolja össze őket az `sql` utasításban, ha csak egy érték van megadva, akkor relációs jellel vizsgálja, hogy a szűrésben csak a nagyobb ill. kisebb dátummal rendelkező rendelések jelenjenek meg. Ha semelyik értéket nem adta meg akkor nem történik dátumra szűrés. Majd ha az állapotra nem a „Mind”-et választották ki az űrlapon akkor a kiválasztott értéket („Aktivált” vagy „Nem aktivált”) illeszti az `$sql`-hez. Ha, bármire szűrés történt akkor azt a `$i` változó 1-re állításával jelöli a kód. Ha semmilyen feltételt nem állítottak be, akkor egy egyszerű `select`-et tesz a `$sql`-be `where` záradék nélkül. A szűrést mindenképpen rendezi az oldal a felhasználó név, majd a regisztráció dátuma szerint. A szűrés feltételei „and”-el vannak összekapcsolva, így szűkítő jellegű a lekérdezés.

Aztán indít egy tranzakciót és lefuttatja az `$sql`-beli lekérdezést. Itt kapja meg a lekért sorok számát a `$szam`. Hiba esetén üzenetet küld a hibáról, majd leállítja a program futását. Majd ha van érvényes lekért sor, akkor generálja a felhasználók listáját megjelenítő `html` táblázatot a `$i` növekvő változó, az `html` állapot, `reg_datum`, `felhnev` adatbázis mezők segítségével. A felhasználó nevet egyrészt 30 karakter szélesre vágja másrészt linkbe írja, amivel az adott felhasználó részletes adataihoz juthat az adminisztrátor, minden link a `reszlet_felh.php` fájlra mutat és a `felh_id` url változóban az adott felhasználó elsődleges kulcsát adja át. A jelölő dobozoknál figyelembe veszi, hogy előzőleg megnyomták-e az összes jelölő doboz bepipálásra használatos gombot. Ha igen akkor már jelölve generálja a vezérlőt. A jelölő dobozok neve „c” és a felhasználó elsődleges kulcsának összeillesztésével adja meg. Majd a `html` táblázat végére megjeleníti, hogy hány felhasználót sikerült lekérni a szűréssel, megjeleníti a főoldalra mutató Vissza feliratú linket. Majd megjeleníti a `$kimenet` változó tartalmát.

14. felh_torles.php

Változók:

\$ki: Egy string típusú változó. Az `uzenet` nevű függvény használja a kimenet tárolására.

\$kimenet: String. Az oldal html kimenetét tárolja.

\$kulcs, \$ertek: String. A törlendő felhasználóhoz társított jelölődozok neveinek lekérdezésénél használatos ciklusban van szükség rájuk.

\$i: Integer. A törölő sql utasítás generálásánál használt jelző változó.

\$sql: String típusú. A felhasználókat törölő lekérdezést tároló változó.

\$db: Integer. A törölt felhasználók számát tartalmazza majd.

Függvények:

`function uzenet($szov, $hova):` Egy üzenetet ír ki az oldalra. Két paramétere van. A `$szov` a kiíradnó szöveget adja meg. A `$hova` pedig, hogy hová vigye a felhasználót a Vissza felíratú nyíl. A függvény a `$ki` változóba gyűjti a html kimenetet. A vissza felíratú link kimenetének generálásakor ellenőrzi, hogy a `$hova` a fő oldalra mutat-e, mert ha igen akkor az ablak teljes tartalma fog frissülni, azaz beállítja a link target tulajdonságát `_top`-ra. A függvény miután befejezte a kimenet generálást, kilép.

A függvényeken kívül eső kód

A fájl a `felh_kereses.php` törölő űrlapján megjelölt felhasználókat fogja törölni.

Beágyazza a fájl elején a `include/opencv.php` és `include/session.php` fájlokat. Ezzel megnyílik az adatbázis kapcsolat ill. biztosítva van a session kezelés. Majd megnyitja a sessiont a `session_start()`-al. A függvények definiálása után ellenőrzi, hogy a session-ben „admin” szerepel-e mint felhasználó név, ha nem vagy nem szerepel semmi akkor az `uzenet` függvénnyel hibát ír ki majd a kód befejezi a végrehajtást. Ha az admin. lépett az oldalra akkor ellenőrzi, hogy a törlést megerősítő űrlapot megjelenítette-e már (`$_POST['s1']` és `$_POST['s1']` változók léteznek-e).

Ha nem akkor generálja az űrlapot, amin választani lehet, hogy tényleg törli-e a felhasználó(ka)t vagy sem. `S1` nevet kapja az „igen” felíratú gomb, `S2`-öt a „nem” felíratú. Rejtett mező(k)ben tovább adja a törölt felhasználókhöz rendelt jelölő négyzetek neveit („c” és a felhasználó elsődleges kulcsa összeillesztve), hogy legközelebb is megkapja ez a fájl az adatokat, bármit is választott az adminisztrátor.

Ha már választott az admin. akkor a „nem” felíratú gomb választása esetén egyszerűen átirányítja a `felh_kereses.php`-ra az adminisztrátort, az átirányítás automatikus. Valamint hozzáilleszti az aktuális session azonosítóját az url-hez. Ellenőrződik itt, hogy a felhasználó választott-e egyáltalán gombot, ez biztonsági intézkedés, ha nem történt választás, akkor üzen az `uzen` függvénnyel és megállítja a szkript végrehajtását. Az „igen” gomb esetén egy ciklussal végigjárja a `$_POST` változón keresztül átadott törlendő felhasználókat, levágja a „c” prefixet, majd az elsődleges kulcsról hozzáilleszti a `$sql` változóba „or” szócskával kapcsolva. Az `$sql`-ben `delete sql` utasítást generál. Majd miután ez megvan és az `$sql` nem üres, azaz tényleg generált `delete` utasítást akkor tranzakciót indít, lefuttatja a `$sql`-beli lekérdezést. Ha valami hiba történik, akkor ezt jelzi, majd leállítja a szkript futását. Siker

esetén tájékoztat a sikerről és megadja a `$db` segítségével a törölt felhasználók számát és a futás nem folytatódik.

Ha az `$sql` üres akkor arról üzen, hogy nem történt törlés és a szkript végéhez ér a végrehajtás.

15. filmek.php

Beágyazza a fájl elején a `include/opendb.php` és `include/session.php` fájlokat. Ezzel megnyílik az adatbázis kapcsolat ill. biztosítva van a session kezelés. Majd megnyitja a sessiont a `session_start()`-al. Majd ezután egyszerűen átirányítja a felhasználót a `kereses.php` oldalra, de beállítja, hogy `urlben`, megkapja az `sql=1` változó,érték párt, hogy majd az oldal tudja, hogy minden filmet meg kell jelenítenie. Beállítja az `elozo` változót is a `filmek.php`-re. Valamint hozzáadja a session azonosítót is az url-hez.

16. fo.php

Változók:

`$er`: Erőforrás típusú változó. Szerte a kódban ez a változó kapja meg a `mysql_query()` függvények által visszaadott erőforrásokat.

`$kimenet`: String. Az oldal html kimenetét tárolja.

`$rekszam`: Integer. Egy sql lekérdezés által visszaadott rekordok számát tárolja benne az oldal.

A függvényeken kívül eső kód:

A megszokott módon beágyazza a fájl elején a `include/opendb.php` és `include/session.php` fájlokat. Ezzel megnyílik az adatbázis kapcsolat ill. biztosítva van a session kezelés. Majd megnyitja a sessiont a `session_start()`-al.

Ezután elkezd írni a `$kimenet` változót. Három keretet (frame) tartalmazó oldalt akar létrehozni. Egy fejléc és a megmaradó területen 2 függőlegesen felosztott (20%-80%) keretet akar létrehozni. A frame szegélyek nem látszanak és a keretek nem átméretezhetők

A bal oldali keretnél mindenképpen a `bal.php`-t fogja meghívni, a különbség csak a `felh` `urlben` átadott változó értékének megadásában van. Vizsgálja, hogy a session-ben a `felhnev`, `jelszo`, `felh_id` változók be vannak-e állítva.

Ha nem, akkor a bal oldali (`bal` nevű) keretben a `felh` url-ben átadott változó értékét 0-ra állítja, azaz nem regisztrált a felhasználó.

Ha regisztrált felhasználó, akkor indít egy tranzakciót és egy lekérdezést, amiben lekéri azokat a felhasználókat, amik `felhnev`, `jelszo`, `felh_id` session-beli változók illeszkednek a `felhnev`, `jelszo`, `id` oszlopokra. A jelszónál a md5-el kódol és azt illeszti a where feltételbe, a `felhnev` esetén pedig kisbetűs illesztést végez. Ha valahol hiba történt, akkor erről tájékoztat és a szkript befejeződik. Majd `$rekszam` változót felhasználja a lekérde rekordok darabszámának vizsgálatánál. Ha nincsenek az eredményben rekordok, akkor automatikus html átirányítást végez a `fo.php` oldalra és a szkript lezárul

Aztán megvizsgálja, hogy a session-beli felhasználó név az „admin”-e ha igen akkor a `bal.php`-nak átadott `felh` változó 2 értéket kap. Ha sima felhasználó, akkor 1 értéket kap a változó.

A középső (kozep a neve) frame a legnagyobb, a munkaterület. Először megvizsgálja, hogy az oldal kapott-e urlben kozepre változót. Ha igen akkor a középső frame src tulajdonságát beállítja erre ennek a változónak az értékére. Ezzel tetszőleges oldalakat lehet megjeleníteni kívülről a kozep nevű frame-ben.

Majd végül befejezi a kimenetet írását a \$kimenet-be és megjeleníti a változó értékét.

17. hozzaszolas.php

Változók:

\$ki: Egy string típusú változó. A hiba nevű függvény használja a kimenet tárolására.

Függvények:

hiba(\$szov): Egy üzenetet ír ki az oldalra adott html környezetbe ágyazva. A \$szov a kiírandó szöveget adja meg. A függvény ellenőrzi, hogy a \$_POST változóban kapott-e az oldal elozo nevű értéket arról, hogy hová irányítsa a felhasználót a Vissza feliratú link. Ha nem kapott, akkor a főoldalra irányítja, ha kapott, akkor a kapott értékeket adja meg célként. A kimenetet a \$ki változóba gyűjti, majd ezt adja meg visszatérési értéként.

A függvényeken kívül eső kód:

A fájl elején beágyazza az include/opendb.php és include/session.php fájlokat. Ezzel megnyílik az adatbázis kapcsolat ill. biztosítva van a session kezelés. Majd megnyitja a sessiont a session_start()-al. Majd a függvények után ellenőrzi, hogy a session-ben átadott felhnev változó létezik-e, ha nem akkor üzen erről a hiba függvényt felhasználva és a szkript futását megállítja. Ha regisztrált a felhasználó, akkor ellenőrzi, hogy a \$_POST változóban átadott változó értéke (dvd) szám-e, ha nem akkor üzen és kilép. Ellenben indít egy tranzakciót és egy insert sql utasítással elhelyezi a kritika táblába \$_POST-ban átadott dvd azonosítót, a hozzaszolas változó értékét, a felhasználó nevet a session-ből és az aktuális dátumidőt. Majd ellenőrzi, hogy történt-e insert, ha nem akkor üzen erről (hiba függvényel) és kilép. Siker esetén pedig kommitálja a tranzakciót és html átirányítást végez a kozep.php oldalra.

18. index.php

Egy php sort tartalmaz, mely html átirányítást végez a fo.php oldalra, mivel az a webáruház nyitó oldala.

19. kereses.php

Változók:

\$ki: Egy string típusú változó. Az uzenet nevű függvény használja a kimenet tárolására.

\$i: Integer. Ciklusokban, számolásoknál használt segédváltozó.

\$volt: A kirak függvényben használatos, egész típusú változó. Kapcsolóként, egy dolog első bekövetkezését mutatja.

\$er, \$er2: Erőforrás típusú változók. Szerke a kódban ez a változó kapja meg a mysql_query() függvények által visszaadott erőforrásokat.

`$sor`, `$sor2`: Asszociatív tömbök. Szerte a kódban a `mysql_fetch_array` függvény által lekért rekord mezőit tárolják.

`$kimenet`: String. Az oldal html kimenetét tárolja.

`$keres`: String. A keresés/szűrés úrlap keresési feltételeit tárolja, hogy azt később tovább lehessen adni az oldal lapozásával.

`$sql`: String típusú. Az egyes rendeléseket összeállító lekérdezést tároló változó.

`$filmek`: String. Ha a filmek.php oldalról került ide a vezérlés akkor az url-ben kapott sql változót is tovább kell adni a filmek közötti előre/hátra lapozásoknál

`$osszes`: Integer. Egy adott lekérdezés eredményeinek darabszámát tárolja.

`$darab`: Integer. Az egy eredmény oldalon megjelenő filmek darabszámát mutatja.

`$bazis`: Integer. Az összes filmes eredménylapból mutatja az aktuálisat.

`$lapszam`: Integer. Azt tárolja, hogy a lekérdezés eredményei hány eredmény oldalon fognak megjelenni.

`$max`: Integer. Mutatja, hogy hány 10 darab eredménylap van az eredmények lapjai közt.

`$maradek`: Integer. Ha nincs az adott 10 darab eredménylapban 10 darab, akkor azt mutatja, hogy mennyi van benne maximálisan

`$hanyasban`: Integer. Mutatja, hogy a bázis eredmény lap hányas 10-es csoportban van benne.

Függvények:

`function kirak($er, & $kimenet)`: Az eredményeket html formára hozva írja bele a kimeneti változóba. Két paramétere van. `$er` adja egy sql lekérdezés eredmény halmazt ad meg, mint erőforrás típusú változót. A `$kimenet` pedig egy cím szerint átadott string változót vár amihez hozzá fűzi ennek a függvénynek a kimenetét. Az egész függvény lényege szinte egy ciklus, amiben lekéri az `$er` eredmény sorait és négyesével jeleníti meg a lekért film adatait egy html táblázatban soronként. Ezt úgy hajtja végre, hogy az `$i` változót növeli ciklusonként, és ha értéke elérte a 4-et, akkor a kimenetbe beilleszti az új sor html tag-et, majd a `$i-t` 1-re állítja. A ciklusban a `$volt` értéke 1 lesz addig 0 volt. A cikluson belül generálja a film adatait. A `rovid_leiras` rekord mező esetén levágja 48 karakter szélességre a szöveget, majd azt tördeli a `wordwrap` függvénnyel 18 karakter szélességűre, és engedi, hogy az ennél hosszabb egy szóból álló szövegeket törje az adott szélességnél, ezzel megelőzve, hogy a html kinézete elcsússzon. A `cím` mezőnél hasonlóan jár el csak ott 32 karakterre vágja le, és 16 karakternél töri. Majd felhasználva az `$er2`, `$sor2` változókat tranzakciót indít és lekérdezi az aktuális filmhez rendelt kategória nevet a `kategoria` táblából, majd hozzá írja a kimeneti változóhoz (`$kimenet`). Ha hiba történik, akkor jelzi és megállítja a szkript végrehajtást. Ezután számítja az árat, figyelembe véve az akcióban megadott százaléklábat. Majd ellenőrzi, a `felhnev`, `jelszo`, `felh_id` változókkal, hogy regisztrált felhasználó nézi-e az oldalt. Ha igen akkor generálja a darabszám mezőt és kosár gombot. A darabszám mező neve az „x” és film elsődleges kulcsának összekapcsolásából áll, hogy később ki lehessen nyerni, hogy mely termékeket tette a felhasználó a kosarába. Ha a felhasználó az adminisztrátor, akkor neki kirak egy törlés és egy szerkesztés gombot minden filmhez. A törlés gomb neve az „y” és a film elsődleges kulcsának összekapcsolásából áll, a szerkesztés gombnál hasonló van csak ott „z” az prefix.

A cikluson kívül ellenőrzi a `$volt`-al, hogy volt-e egyáltalán eredmény, ha nem akkor üzen és véget ér a szkript végrehajtás.

A függvényeken kívül eső kód:

A fájl elején a szokásos módon beágyazza az `include/opendb.php` és `include/session.php` fájlokat. Ezzel megnyílik az adatbázis kapcsolat ill. biztosítva van a session kezelés. Majd megnyitja a sessiont a `session_start()`-al. Majd elkezd írni a `$kimenet` változót.

Ezután ellenőrzi, hogy az űrlapot nem beküldték, hanem először nyitották meg az oldalt. Ezesetben a kimenet változóba generálja az űrlapot(`form1`) az adatokat a `kereses.php` fájl fogja megkapni. A kateg nevű kombinált lista feltöltését a kategória táblából végzi egy `select` utasítással. Ha a lekérdezésnél hiba történt valahol, akkor üzen és leállítja a végrehajtást. Végül megjeleníti a `$kimenet` változó értékét.

Ha az űrlapot már elküldték és nem a `filmek.php` oldalról került ide a vezérlés akkor ellenőrzést hajt végre az `armin`, `armax` űrlap mezőkön, hogy valóban számokat tartalmaznak-e, ha egyáltalán ki vannak töltve. Valamint a `datummin`, `datummax` űrlap mezőkön, hogy helyes dátumformátumban adták-e meg az adatokat bennük (éééé-hh-nn).

Ha nem a `filmek.php`-ről került a vezérlés ide (`$_REQUEST['sql']` nem létezik) akkor generálja a keresés eredményeinek űrlapját, ami az adatokat a `kosar.php`-nek adja meg és előző oldalnak, ezt az oldalt állítja be. Majd felépíti a `$keres` változót, amibe a keresés mezőket és értékeiket írja bele úgy, hogy azt később url-ben át lehessen adni. Ezután generálja a keresésnek megfelelő filmeket lekérdező `sql select`et az `$sql` változóba. A film cím és leírás mező kitöltése esetén is kis/nagy betű érzéketlen formában keres. A `like` `sql` operátor segítségével az szövegben keres. A `leiras` űrlap mező kitöltésével a rövid és hosszú leírásban is keres a szövegre, „or” kapcsolattal. Majd vizsgálja a dátumidő intervallumot, ha mindkét értéket megadták az űrlapon, akkor `between`-el kapcsolja össze őket az `sql` utasításban, ha csak egy érték van megadva, akkor relációs jellel vizsgálja, hogy a szűrésben csak a nagyobb ill. kisebb dátummal rendelkező rendelések jelenjenek meg. Ha semelyik értéket nem adta meg akkor nem történik dátumra szűrés. Ezután a `kateg` mezőnek megfelelő kategória azonosítót nyeri ki az adatbázisból és illeszti hozzá az `$sql`-hez. Ha nem sikerül, akkor hibát ír ki és megállítja a futást.

Ha bármire szűrés történt, akkor azt a `$i` változó 1-re állításával jelöli a kód. Ha semmilyen feltételt nem állítottak be, akkor egy egyszerű `select`-et tesz a `$sql`-be, ugyan ezt teszi, ha a vezérlés a `kereses.php` oldalról került ide, ez esetben a `$filmek` változónak értékül adja „`sql=1&`” stringet, különben üres marad. A szűrést mindenképpen rendezi az oldal rendelésazonosító, dátum majd a kisbetűsre alakított dvd cím szerint. Majd végrehajtja az `$sql`-beli lekérdezést, hogy megtudja, hogy hány találat van összesen. Majd beállítja a `$bazis` értékét, vagy a `$_POST`-ban levő értékre vagy 0-ra, ha az első eredménylapon vagyunk. Beállítja az egy oldalon megjelenő filmek darabszámát a `$darab` változóba, 12 regisztrálatlan felhasználók esetén ellenkező esetben a `felhasznalo` tábla `edb` mezőjét kapja meg eredményül. Majd az `$sql` változót kiegészíti a `limit` záradékkal, ami mutatja, hogy a lekérdezést eredményét hányadik talált rekordtól kezdje visszaadni és hány darabot összesen onnantól.

Ezek után a lapozó linkeket jeleníti meg, amik segítségével navigálhat a felhasználó, ha több eredménylapot talált a szűréssel. A linkek generálásánál felhasználja az `$osszes`, `$bazis`, `$lapszam`, `$darab`, `$max`, `$hanyasban`, `$maradek`, `$filmek`

változókat. Itt adja hozzá minden linkhez a `$keres` változót, amivel át tudja adni a következő eredménylapra is a keresés feltételeit, valamint `$filmek`-et is, ami ha a `filmek.php`-ről került ide a vezérlés akkor tartalmazni fogja a `sql` url-ben átadott változót. Az aktuális eredménylapot a 10-es link csoportban(már ha van 10) kiemeli félkövérrel.

Végül lefuttatja a `$sql`-beli lekérdezést, már a `limit`-ekkel kiegészítve. Ha hiba történt, akkor üzen és befejezi a szkript futását. Ellenkező esetben átadja az erőforrás változót és a meglévő kimeneti változót (`$er`, `$kimenet`) a `kirak` függvénynek, majd amikor az visszatér, megjeleníti a `$kimenet` változó tartalmát.

20. `kilepes.php`

Kilépteti a bejelentkezett felhasználókat.

A fájl elején a szokásos módon beágyazza az `include/opendb.php` és `include/session.php` fájlokat. Ezzel megnyílik az adatbázis kapcsolat ill. biztosítva van a session kezelés. Majd megnyitja a sessiont a `session_start()`-al. Majd megszünteti az adott session-t, a `session_destroy()`-a és html átirányítást hajt végre a főoldalra.

21. `kosar.php`

Változók:

`$ki`: Egy string típusú változó. A `hiba` nevű függvény használja a kimenet tárolására.

`$elozo`: String. Mely azt tartalmazza majd, hogy melyik oldalról jutottak erre az oldalra.

`$kulcs`, `$ertek`: String és integer. A `$_POST`-ban átadott dvd-k feldolgozására használja őket a szkript, `foreach` ciklusban.

`$sql`: String típusú. Az egyes rendeléseket összeállító lekérdezést tároló változó.

`$osszeg`: Integer. A kosárban található filmek összértéke.

`$i`: Integer. Számolásokra.

`$ar`: Integer. Az árak számításánál használatos.

`$darab`: Integer. Az egyes tételekben a filmek darabszáma.

`$kimenet`: String. Az oldal html kimenetét tárolja.

Függvények:

`hiba($szov, $szov2="A kosár")`: Egy üzenetet ír ki az oldalra adott html környezetbe ágyazva és adott főcímet adva az oldalnak. A `$szov` a kiírandó szöveget adja meg. `$szov2` egy alapértelmezett értékkel ellátott paraméter, ami a megjelenített főcím szövegét kéri. A függvény ellenőrzi, hogy a `$_REQUEST` változóban kapott-e az oldal `elozo` nevű értéket arról, hogy hová irányítsa a felhasználót a Vissza felíratú link. Ha nem kapott, vagy amit kapott abban is a `fo.php` megtalálható, mint szöveg akkor a főoldalra irányítja, ha kapott, akkor a kapott értéket adja meg célként és a link `target` tulajdonságát `_top`-ra teszi, azaz az egész ablakba fogja megjeleníteni az oldalt. A kimenetet a `$ki` változóba gyűjti, majd ezt adja meg visszatérési értéként.

A függvényeken kívül eső kód:

A fájl elején, a szokásos módon beágyazza az include/opendb.php és include/session.php fájlokat. Ezzel megnyílik az adatbázis kapcsolat ill. biztosítva van a session kezelés. Majd megnyitja a sessiont a `session_start()`-al. A függvény definiálása után, változó értékadás majd vizsgálatok következnek. A `fo.php`-t állítja be az `$elozo` értékeként ha a `$_REQUEST`-ban nem kapott `elozo` nevű változó, ellenkező esetben azt kapja. Ellenőrzi, hogy az oldalt regisztrált felhasználó nyitotta-e meg, ha nem üzen a hiba függvénnyel és kilép. Ellenőrzi, hogy megnyomták-e az űrlapon a „Megrendelés” gombot ha igen akkor html átirányítást végez a `megrendeles.php`-re.

A következő lépésben ellenőrzi, hogy az űrlapon esetleg a törlés gombot nyomták-e meg, ha igen akkor `$_POST`-ot bejárja, hogy talál-e „c” prefixű kulcsot, ha igen akkor kinyeri az elsődleges kulcs értékét az adott dvd-hez és törli a `$_SESSION`-ből az „i” prefixű és megegyező elsődleges kulcsú változót azaz filmet a kosárból.

Majd ellenőrzi, hogy ha az adminisztrátor szerkesztés vagy törlést nyomott az adott dvd esetén, a főoldalon, a keresésnél, a filmeknél vagy a filmek részletei oldalon az adott filmhez. Szerkesztés esetén ehhez „z” prefixű kulcsot keres egy ciklusban a `$_POST`-ban majd html átirányít az `ujdvd.php` fájlra. Kulcsként átadja a prefixtől „megszabadított” `$kulcs`-ot url-ben egy `dvd` nevű változónak, majd beállítja az `elozo` url változót is.

Törlés esetén „y” előtagot keres majd levágva a prefixet, `$kulcs` elsődleges kulcsú dvd-t keres a `dvd` táblában. A keresési eredmény sorát a `$sor` változó kapja meg. Hiba esetén jelzi a hibát és kilép. Majd törli egy sql utasítással az adatbázisból a megadott dvd-t. Ezután a `$sor`-beli adatok szerint törli a filmhez rendelt képet a „kep” könyvtárból, ha kapcsoltak képet hozzá, valamint az előzetessel is hasonló történik csak azt az „elozetes” könyvtárból törli. Sikeresség esetén üzen (a `hiba` nevű függvénnyel) és kilép.

A következőkben a kosárba rakott dvd-eket keresi egy ciklussal a `$_POST`-ban és „x” prefixeket keres. Ha talált ellenőrzi, hogy az átvett darabszám az adott dvd-ből (az elsődleges kulcs) szám-e és hogy 1 és 100 között van-e, ha nem akkor üzen (`hiba`) erről és kilép. Majd ellenőrzi, hogy van-e már ilyen termék a kosárban (`$_SESSION`) ha igen akkor hozzáadja az újabb mennyiséget. Majd a `$_SESSION` változóba teszi a dvd-t most már „i” prefix-szel és értékül kapja a darabszámot.

Ezután elkezd generálni a kimenetet. Majd felépít egy sql lekérdezést a `$sql`-be, session-ben tárolt „i” prefixű változókból amiket „or”-al kötve össze dvd-k elsődleges kulcsára keres rá.

Majd ha létezik a `$sql` akkor végre is hajtja a lekérdezést. Majd generálja a html táblázatot a kosár tételeinek. Az áraknál figyelembe veszi az `akcio` rekord mező százalékklábát. Az értékeket mindig egészre konvertálja. Majd az `$osszeg`-hez minden tétel esetén hozzáadja a tétel részlet összegét, ami a mennyiség szorozva az árral. `$i` változót használja a tételek sorszámának kiírására, majd a jelölőnégyzeteknél figyel arra, hogy ha esetleg már megnyomta a felhasználó a „Mindet kijelöl” gombot akkor megjelöltként jelenítse meg.

Majd befejezi a kimenet generálását, ha létezik a `$_REQUEST['elozo']` változó akkor a Vissza feliratú nyíl oda fog mutatni, ha nem akkor a `fo.php` fájlra. Végül megjeleníti a `$kimenet`-et.

22. kozep.php

Változók:

`$ki`: Egy string típusú változó. Az `hiba` nevű függvény használja a kimenet tárolására.

`$beall`: Asszociatív tömb. A `mysql_fetch_array` függvény által lekért rekord mezőit tárolja

`$i`, `$n`: Integer, Számításoknál, ciklusoknál használatosak.

`$ered`: Tömb a `dvd` tábla elsődleges kulcsait tárolja.

`$max`: Integer. Ideiglenes érték tárolásra használatos változó.

`$sql`: String típusú. Az egyes rendeléseket összeállító lekérdezést tároló változó.

`$kimenet`: String típusú. A html kimenetet tároló globális változó.

`$sor`: Asszociatív tömb. Szerte a kódban a `mysql_fetch_array` függvény által lekért rekord mezőit tárolja.

`$dbszam`: Integer. A lekérdezett filmek darabszáma.

`$eredm`: Integer tömb. A véletlenül kiválasztott DVD-k elsődleges kulcsait tartalmazza.

Függvények:

`hiba($szov)`: Egy üzenetet ír ki az oldalra, adott html környezetbe ágyazva és adott főcímet adva az oldalnak. A `$szov` a kiírandó szöveget adja meg. A függvény ellenőrzi, hogy a `$_REQUEST` változóban kapott-e az `oldal_elozo` nevű értéket arról, hogy hová irányítsa a felhasználót a Vissza feliratú link. Ha nem kapott, vagy amit kapott abban is a `fo.php` megtalálható, mint szöveg akkor a főoldalra irányítja, ha kapott, akkor a kapott értékeket adja meg célként és a link `target` tulajdonságát `_top`-ra teszi, azaz az egész ablakba fogja megjeleníteni az oldalt. A kimenetet a `$ki` változóba gyűjti, majd ezt adja meg visszatérési értéként.

`function filmek($tipus, & $eredmeny)`: A függvény feladata, hogy kiválasszon a `beallitasok` tábla `ufilmdb` vagy `afilmdb` darab filmet véletlenszerűen az `[afilmdb_uzen,afilmdb_veg]` ill `[ufilmdb_uzen,ufilmdb_veg]` dátumidő intervallumból a `dvd` tábla filmjei közül. A paraméterek a `$tipus` ami ha 0 az `ufilmdb`, `ufilmdb_uzen`, `ufilmdb_veg` értékeket tekint, azaz az új filmekről van szó, 1 érték esetén a másik `afilmdb`, `afilmdb_uzen`, `afilmdb_veg` értékeket, azaz az akciós filmekről van szó, melyeknél az `akcio` mező nem 0. `$eredmeny` pedig a kiválasztott elsődleges kulcsokat tartalmazó tömb, ami cím szerinti paraméter. Kezdetben lekérdezi a `beallitasok` tábla egyetlen sorát. Ha több sor van, mint 1 vagy kevesebb akkor üzen és kilép. Majd a `$tipus`-tól függően lekérdezi az összes a szűrésnek megfelelő film elsődleges kulcsát egy `sql select` utasítással. Ha hiba történik, akkor jelez és kilép. Eztán inicializálja a véletlenszám generátort az `srand`-al. Majd a lekérdezett összes elsődleges kulcsot egy ciklusban belerakja az `$ered` tömbbe. Eztán a következő ciklusban `$tipus`-tól függően `$beall['ufilmdb']` vagy `$beall['afilmdb']` darabot melyek indexét a véletlenszám generátor kisorsolta átmásolja az `$eredmeny` tömbbe. Majd végül visszatérési értéként visszaadja a kiválasztott kulcsok számát.

`function kirak($mit, $a, & $kimenet)`: Formázottan jeleníti meg a `$mit` tömbben tárolt elsődleges kulcsú `dvd`-ket. Paraméterek: `$mit`, ezt az előbb említettem, `$a` kulcsok darabszáma, `$kimenet`, ebbe a változóba kell beleírni a html kimenetet.

Kezdetben generálja a `$sql` változóba a `$mit` tömb által tartalmazott kulcsoknak megfelelő dvd-eket lekérdező utasítást. Majd le végre is hajtja ezt a lekérdezést. Ha hiba történt jelzi és kilép. A függvény további része eléggé hasonlít `kereses.php`-beli `kirak` függvényére. Az egész függvény lényege szinte egy ciklus, amiben lekéri az `$er` eredmény sorait és négyesével jeleníti meg a lekért film adatait egy html táblázatban soronként. Ezt úgy hajtja végre, hogy az `$i` változót növeli ciklusonként, és ha értéke elérte a 4-et, akkor a kimenetbe beilleszti az új sor html tag-et, majd a `$i`-t 1-re állítja. Majd felhasználva az `$er2`, `$sor2` változókat tranzakciót indít és lekérdezi az aktuális filmhez rendelt kategória nevet a `katgoria` táblából, majd hozzá írja a kimeneti változóhoz (`$kimenet`). Ha hiba történik, akkor jelzi és megállítja a szkript végrehajtást. A cikluson belül generálja a film adatait. A `rovid_leiras` rekord mező esetén levágja 48 karakter szélességre a szöveget, majd azt tördeli a `wordwrap` függvénnyel 18 karakter szélességűre, és engedi, hogy az ennél hosszabb egy szóból álló szövegeket törje az adott szélességnél, ezzel megelőzve, hogy a html kinézete elcsússzon. A `cím` mezőnél hasonlóan jár el csak ott 32 karakterre vágja le, és 16 karakternél tör. Ezután számítja az árat, figyelembe véve az akcióban megadott százaléklábat. Majd ellenőrzi, a `felhnev`, `jelszo`, `felh_id` változókkal, hogy regisztrált felhasználó nézi-e az oldalt. Ha igen akkor generálja a darabszám mezőt és kosár gombot. A darabszám mező neve az „x” és film elsődleges kulcsának összekapcsolásából áll, hogy később ki lehessen nyerni, hogy mely termékeket tette a felhasználó a kosárba. Ha a felhasználó az adminisztrátor, akkor neki kirak egy törlés és egy szerkesztés gombot minden filmhez. A törlés gomb neve az „y” és a film elsődleges kulcsának összekapcsolásából áll, a szerkesztés gombnál hasonló van csak ott „z” az prefix.

A függvényeken kívül eső kód:

A fájl elején, a szokásos módon beágyazza az `include/opendb.php` és `include/session.php` fájlokat. Ezzel megnyílik az adatbázis kapcsolat ill. biztosítva van a session kezelés. Majd megnyitja a sessiont a `session_start()`-al. A függvény definiálása után, elkezdi írni a `$kimenet` változót. Majd kiválasztja az új filmek esetén a beállítások tábla ennek megfelelő szűrési feltételeinek megfelelő filmeket a filmekkel, majd beleírja a filmeket formázva a `$kimenet` változóba majd az akciós filmekkel is megteszi ugyanezt. Végül megjeleníti a `$kimenet` értékét.

23. level.php

Változók:

`$ki`: Egy string típusú változó. Az `uzenet` nevű függvény használja a kimenet tárolására.

`$kimenet`: String típusú. A html kimenetet tároló globális változó.

Függvények:

`function uzenet($szov, $hova)`: Üzenetet ír ki formázott html környezetben majd a egy linket ad ami a `$hova` fájlra mutat. Paraméterek `$szov` a kiírandó üzenet, `$hova` ide mutat majd a link. A `$ki` változóba írja a kimenetet, majd a végén ezzel tér vissza.

A függvényeken kívül eső kód:

Beágyazza az `include/opendb.php`, `include/session.php` és `include/adatok.php` fájlokat. Majd megnyitja a sessiont. Majd létrehozza a `$kimenet` változóba a levélküldő űrlapot. Az űrlapot ez a fájl dolgozza fel.

Majd ellenőrzi, a `felhnev`, `jelszo`, `felh_id` változók meglétét a sessionben, ha nincsenek, akkor üzenetet küld, mert ekkor nem regisztrált felhasználó lépett erre a lapra.

Eztán ellenőrzi, a `$_POST`-ot, hogy tartalmazza-e az `uzenet` változót, ha nem akkor a megjeleníti a levélküldő űrlapot, majd kilép. Végül megpróbál levelet küldeni a webáruház címére (`$emailcim` tárolja), a levél `html` levél és a `$_POST` `uzenet` változóját tartalmazza. Ha nem sikerül a levélküldés, akkor üzen erről, ha sikerül, akkor a siker tényét közli.

24. megrendeles.php

Változók:

`$ki`: Egy string típusú változó. Az `uzenet` nevű függvény használja a `kimenet` tárolására.

`$kimenet`: String típusú. A `html` kimenetet tároló globális változó.

`$elozo`: String. Megadja, hogy azok az oldalak ahová eljut innen a vezérlés hová linkeljék a Vissza feliratú linkjeiket.

`$er`: Erőforrás típusú változó. Szerte a kódban ez a változó kapja meg a `mysql_query()` függvények által visszaadott erőforrásokat.

`$sor`, `$sor2`, `$sor3`: Asszociatív tömbök. Szerte a kódban a `mysql_fetch_array` függvény által lekért rekord mezőit tárolják.

`$level`: String. Ezt fogja levélben elküldeni az oldal a rendelés visszaigazolása képpen.

`$i`: Integer. Ciklusban használatos számolásra.

`$osszeg`: Integer. Az adott rendelés összértékét tárolja.

`$mikro`: String. A `microtime()` függvény értékét tartalmazza formázva.

`$tomb`: Tömb. Segéd tömb a `$mikro` formázásához.

`$kulcs`, `$ertek`: String és integer. Foreach ciklusban használatos.

`$as`: String. SQL lekérdezést tárol.

Függvények:

`function uzenet($szov, $hova):` Egy üzenetet ír ki az ablakba. Két paramétere van. A `$szov` a kiírandó szöveget adja meg. A `$hova` pedig, hogy hová vigye a felhasználót a Vissza feliratú nyíl. A függvény a `$ki` változóba gyűjti a `html` kimenetet. A vissza feliratú link kimenetének generálásakor ellenőrzi, hogy a `$hova` a fő oldalra mutat-e, mert ha igen akkor az ablak teljes tartalma fog frissülni, azaz beállítja a link `target` tulajdonságát `_top`-ra. A függvény miután befejezte a kimenet generálást, kilép.

A függvényeken kívül eső kód:

A fájl elején beágyazza az `include/opendb.php`, `include/session.php` fájlokat. Majd megnyitja a sessiont.

Majd ellenőrzi, hogy a session tartalmazza-e a felhnev, jelszo, felh_id változók azaz bejelentkezett felhasználó-e, ha nem üzen erről és kilép.

Ellenőrzi, ha nem küldték el még a megrendelés adatait bekérő űrlapot, akkor lekérdezi, hogy létezik-e a felh_id elsődleges kulcsú felhasználó az adatbázisban, ha nem akkor üzenetet ír és befejezi a szkript a futást. Majd generálja az űrlapot a \$kimenet változóba, az űrlapot ugyanez a fájl fogja feldolgozni. Az összes felhasználó adatot kitölti előre a megfelelő beviteli mezőkben amennyiben meg vannak adva azok az adatok. A varos, irszam, utca, hazszam mezőkbe írhat a felhasználó a többi disabled tulajdonságot kap. Majd megjeleníti a kimenetet és kilép.

Ha beküldték az űrlapot, akkor ellenőrzi a fenti négy mező és a jelszo mező meglétét a \$_POST-ban, ha nincs vagy nincsenek kitöltve, akkor üzen és kilép. Az irányító szám és házszám mezőknél ellenőrzi ezután, hogy számokat tartalmaznak-e, ha nem üzenetet ír erről és kilép. Majd ellenőrzi, hogy van-e felh_id elsődleges kulcsú aktív (allapot=1) felhasználó az adatbázisban, ha nincs akkor, üzen erről és kilép. Az esetleges hibáknál is üzenetet ír majd a szkript befejezi a futását. Siker esetén ellenőrzi, hogy a session-beli felhnev megegyezik-e az adatbázisbelivel, valamint a jelszóval is végrehajtja ezt a vizsgálatot ehhez md5 kódolást használ.

Majd elkezd generálni a rendelést igazoló levelet a felhasználónak. Ehhez felhasználja a felhnev rekord mezőt, melyet 30 karakterre vág le.

Majd lekérdezi a mostani dátumidőt, ami a \$sor3['most']-ba kerül. Ezután a \$mikro változóba írja a mikrotime() értékét, majd levágja „0.”-ot az elejétől és a közepéből kiveszi a space karaktert. Ez lesz a rendelés azonosítója minden rendelés tételhez, ez kerül. Tranzakciót indít majd egy foreach ciklussal bejárja a \$_SESSION-t, „i” prefixű változókat keres, mert ezek a kosárban levő dvd-k. Majd lekérdezi minden ilyen az „i” prefixtől megszabított kulcsra az id,cim,ear,akcio mezőket a dvd táblából. Majd összeállítja a \$as mezőbe a rendelést beszűrő sql utasítást a rendeles táblába, az eddigi adatokat felhasználva.. Majd egy sql update-el megnöveli a megvásárolt dvd darabszámával a dvd mdb mezőjét. Ezután lefuttatja a \$as-beli lekérdezést is. Majd ha nem történt beszűrés akkor üzen és kilép. Majd generálja tovább a \$level változót, felhasználva az értékeket.

Ezután egy ciklusban törli a dvd-eket a session-ből. Végül a felhasználó e-mail címére megpróbálja elküldeni a visszaigazoló levelet a \$level szöveggel.

Hiba és sikeres küldés esetén is üzenetet ír a megfelelő szöveggel és a főoldalra irányít át.

24. pontozas.php

Változók:

\$ki: Egy string típusú változó. A hiba nevű függvény használja a kimenet tárolására.

\$er: Erőforrás típusú változó. Szerte a kódban ez a változó kapja meg a mysql_query() függvények által visszaadott erőforrásokat.

Függvények:

function hiba(\$szov): Egy üzenetet ír ki az oldalra. A \$szov a kiíradnó szöveget adja meg. A függvény ellenőrzi, hogy a \$_REQUEST változóban kapott-e az oldal elozo nevű értéket

arról, hogy hová irányítsa a felhasználót a Vissza felíratú link. Ha nem kapott, vagy amit kapott abbaqn is a fo . php megtalálható, mint szöveg akkor a főoldalra irányítja, ha kapott, akkor a kapott értéket adja meg célként és a link target tulajdonságát _top-ra teszi, azaz az egész ablakba fogja megjeleníteni az oldalt. A kimenetet a \$ki változóba gyűjti, majd ezt adja meg visszatérési értéként.

A függvényeken kívül eső kód:

A fájl elején beágyazza az include/opendb.php, include/session.php fájlakat. Majd megnyitja a sessiont. A függvény definiálása után ellenőrzi, hogy regisztrált felhasználó használja-e az oldalt (létezik a felhnev változó a session-ben), majd hogy a \$_POST változó tartalmaz-e dvd nevű kulcsot és hogy ez szám-e. Ha ezek a feltételek nem teljesülnek üzenetet küld, majd kilép.

Ezután egy sql update-el növeli a dvd tábla dvd kulcsú sorának pont mezőjét az űrlapon megadott pont értékkel. Ha valami hiba történik, akkor üzen erről és a szkript befejezi a futását. Ellenkező esetben html átirányítás történik a kozep . php fájlra.

25. profil.php

Változók:

\$ki: Egy string típusú változó. Az hiba nevű függvény használja a kimenet tárolására.

\$er: Erőforrás típusú változó. Szerte a kódban ez a változó kapja meg a mysql_query() függvények által visszaadott erőforrásokat.

\$sor: Asszociatív tömb. Szerte a kódban a mysql_fetch_array függvény által lekért rekord mezőit tárolja.

\$kimenet: String típusú. A html kimenetet tároló globális változó.

\$rekszam: Integer. Egy sql lekérdezés által visszaadott rekordok számát tárolja benne az oldal.

\$sql: String típusú. Az egyes rendeléseket összeállító lekérdezést tároló változó.

\$level: String. Ezt fogja levélben elküldeni az oldal a profil megváltoztatásáról.

Függvények:

function uzenet(\$szov): Egy üzenetet ír ki az oldalra. \$szov mutatja az üzenetet. A kimenetet a \$ki változóba gyűjti, majd azzal tér vissza.

A függvényeken kívül eső kód:

Beágyazza az include/opendb.php, include/session.php fájlakat. Majd megnyitja a sessiont. Ellenőrzi, hogy regisztrált felhasználó használja-e az oldalt ha nem akkor üzen erről és bejezi a szkript futását. Majd lekérdezi a felhasznalo táblából az aktuális felhasználó adatait. Ha hiba történik, akkor üzenetet ír.

Majd ha az űrlapot nem küldték még be, akkor generálja azt a \$kimenet változóba, felhasználva az előző lekérdezés adatait alapértelmezésként ahol csak lehet. Az űrlapot ugyanez a fájl fogja feldolgozni. Ezután megjeleníti a változót és kilép.

Ha beküldték az űrlapot, akkor ellenőrzi, hogy a \$_POST-beli jelszó megegyezik-e az adatbázisbelivel, ha nem üzen erről és kilép. Majd ellenőrzi, hogy az űrlapon az eredmények darabszáma, a név és az e-mailhez rendelt beviteli mezők ki vannak-e töltve (edb, nev,

email), ha nem üzen és kilép, mint az előbb. Majd leellenőrzi, hogy az űrlapon az edb, irszam, hazszam mezők szám értéket kaptak-e meg, ha nem akkor szintén üzen és bejeződik a futás. Ez után ellenőrzi, hogy az edb mező értéket [1,100] intervallumban van, ha nem üzen erről és kilép. Ezután ha az irszam, vagy hazszam mezők valamelyike nincs kitöltve akkor „null” értéket kapnak, hogy az adatbázisban ne vegyenek fel értéket az update-nél. Majd ezután tranzakciót indít és megkonstruálja a \$sql változóba a sql update utasítást az űrlap adataival. Ha az új jelszó mezőt is, kitöltötték akkor azt is hozzáadja md5-el kódolva az \$sql-hez. Hiba esetén közli a hibát és befejezi a futást. Egyébként generálja a visszaigazoló levél tartalmát (\$level). A levélben, ha a jelszót nem változtatják meg csak a változtatás ténye lesz feltüntetve és a felhasználó név. Ha jelszót is változtattak, akkor az új jelszó is szerepel majd ott.

Végül megpróbálja elküldeni a levelet a felhasználó címére, ha majd a küldés eredményességéről üzenetet ír és befejezi a szkript a futását.

26. reg.html

Egy statikus html oldal, ami a regisztráció űrlapját tartalmazza. Az űrlapot reg.php fájl fogja feldolgozni.

27. reg.php

Beágyazza az include/opendb.php, include/adatok.php fájlokat.

Majd a \$hiba, \$hiba2, \$siker, \$level változóba generál html üzeneteket. Majd ellenőrzi a beküldött űrlapról a nev, tel, emial, felhnev, jelszo, jelszo2 kötelezően kitöltendő meglétét, ha nem akkor üzen a \$hiba kiírásával és kilép.

Majd ellenőrzi, hogy az irányítószám és házzszám mezők szám értékűek-e feltéve, hogy ki vannak töltve, ha nem akkor üzen és kilép. Ha ezen mezők üresek, akkor értékül „null” stringet kapják, az sql insert utasításhoz.

Majd egy sql lekérdezéssel ellenőrzi, hogy ilyen felhasználó névvel regisztrált-e valaki, ha az eredményben a rekordok darabszáma nem 0 akkor üzenetet ír arról, hogy a ez a felhasználó néve foglalt már majd kilép.

Ezután tranzakciót indít és beszúr egy új rekordot a felhasználó táblába az űrlap adataival. Hiba esetén üzenetet ír és kilép. Siker esetén erről is üzen. Majd megkonstruálja a regisztrációról a levelet, melyben megjelennek az aktiválás és deaktiválás linkek, amik az aktivalas.php-ra mutatnak. Itt szerepet kap a \$host változó, ami tartalmazza az oldal domain-jét, ill, ip címét. Ha nem sikerül elküldeni a levelet, akkor üzen erről.

28. rendelesek.php

Változók:

\$ki: Egy string típusú változó. Az hiba nevű függvény használja a kimenet tárolására.

\$azon: String típusú. A rendelések egyedi azonosítóját tárolja. Az összeahasonlításokhoz a kirak nevű függvényben.

\$j, \$i: Számlálásokra, ciklusokban és ideiglenes integer értékek tárolására használatos változók.

\$osszeg: A kirak függvényben használatos. Az egyes rendelések összértékének tárolására, ill. számítására

\$volt: A kirak függvényben használatos, egész típusú változó. Kapcsolóként, egy dolog első bekövetkezését mutatja.

\$datum: A kirak függvényben használatos, string típusú változó. Az egyes rendelési tételek lekérdezésekor mindig az előző rendelés dátumát tárolja.

\$er: Erőforrás típusú változó. Szerte a kódban ez a változó kapja meg a `mysql_query()` függvények által visszaadott erőforrásokat.

\$sor: Asszociatív tömb. Szerte a kódban a `mysql_fetch_array` függvény által lekért rekord mezőit tárolja.

\$kimenet: String típusú. A html kimenetet tároló globális változó.

Függvények:

`function uzenet($szov):` Egy üzenetet ír ki az oldalra. \$szov mutatja az üzenetet. A kimenetet a \$ki változóba gyűjti, majd azzal tér vissza. Egy Vissza feliratú linket tartalmaz a főoldalra.

`function kirak(& $kimenet, $allapot):` Az egyes rendelések adatainak kimenetét írja hozzá a \$kimenet globális változóhoz. Paraméterek a \$kimenet, ami cím szerinti melybe fogja bele írni a függvény által tárolt kimenetet és \$allapot amelynek megfelelő állapotú rendeléseket fog belegenerálni \$kimenet-be.

A függvény először tranzakciót nyit, majd lekéri az aktuális felhasználóhoz tartozó összes olyan rendelést, aminek az állapota megfelel a \$allapot-nak Hiba esetén üzenetet ír és kilép.

A függvény úgy működik, hogy egy ciklusban elkezd lekérni a lekérdezés által lekért adatbázis rekordokat a \$sor változóba. Minden olyan rendelési tételt egy táblázatba fog írni azokkal a tételekkel, amiknek ugyanaz a rendelés azonosítója. Ezt a cikluson belül a rendelési azonosító és az \$azon változó összehasonlításával végzi. Ha nem egyeznek, akkor új rendelésről van szó és új táblázat fejet és rendelési adatokat generál, valamint lezárja az előző táblázatot, valamint az \$osszeg és \$j változók 0 értéket vesznek fel. Kivétel az az eset, amikor belépünk a ciklusba, mert akkor az egész kimenetet máshogy kell generálni, ennek a jelzésére szolgál a \$volt változó, ami utána 1 értéket kap. Ha egyezik, akkor ugyanarról a rendelésről van szó és csak generálja a rendelés adatait a megfelelő html táblázat soraiba. Számolja az összarat azáltal, hogy mindig hozzáadja a rendelés részlet összegét az addigi \$osszeg változóhoz, a részlet összeg a mennyiség mező szorozva az egység ár mezővel. Majd végül frissíti a \$datum, \$allapot és \$azon változók értékeit az aktuális rekord megfelelő mezőire. A megjelenített adatoknál a megrendelő neve, a város, utca, és a dvd címe mező értékeket 40 karakter szélesre vágja le. A tétel sorszámát a html táblázatban a \$j jelzi a kimenetben. Az egyes rendeléseket sorszámozza a \$i-vel.

A függvényeken kívül eső kód:

Beágyazza az `include/opendb.php`, `include/session.php` fájlokat. Majd megnyitja a `session`-t. Ellenőrzi, hogy regisztrált felhasználó használja-e az oldalt, ha nem akkor üzen erről és bejezi a szkript futását.

Majd elkezd generálni a \$kimenet változóba html lapot. Majd a kirak függvénnyel generálja először a „felvéve” állapotú rendeléseket, majd a „leszállítva” állapotúakat a

\$kimenet változóba. Majd hozzáad egy Vissza felíratú linket, ami a főoldalra mutat, majd végül megjeleníti \$kimenet értékét.

29. `reszlet.php`

Változók:

\$er, \$er2: Erőforrás típusú változók. Szerte a kódban ez a változók kapják meg a `mysql_query()` függvények által visszaadott erőforrásokat.

\$sor, \$sor2: Asszociatív tömbök. Szerte a kódban a `mysql_fetch_array` függvény által lekért rekord mezőit tárolják.

\$kimenet: String típusú. A html kimenetet tároló globális változó.

A függvényeken kívül eső kód:

Beágyazza az `include/opencv.php`, `include/session.php` fájlokat. Majd megnyitja a `session`-t. Majd ellenőrzi, hogy `url`-ben átadtak neki `dvd` nevű változót és az egyáltalán egész számot tartalmaz-e, ha nem akkor mind két esetben `html` átirányítást végez a főoldalra.

Majd elkezd generálni a \$kimenet változóba a `html` kimenetet. Először lekéri az adott `dvd`-hez tartozó összes adatot az adatbázisból. Ha hiba történik, akkor jelzi ezt és abbahagyja a futást a kód. Ezután generálja a `dvd` adatait formázottan. A `cim` mezőt tördeli 24 karakterrel a `wordwrap` függvénnyel, a `rovid_leiras`-t 40 karakterrel, a `hosszu_leiras`-t 52 karakterrel amelyben `\n` karaktereket kicseréli sortörés (`
`) tag-ekre. A kategória nevének megjelenítéséhez lekérdezést hajt végre a `katgoria` táblában. Hiba esetén közli a hibát és a szkript befejezi a futását. Az áraknál figyelembe veszi az esetleges akciót, ekkor kiírja az egységárat és az akciós árat is. Majd ellenőrzi, hogy regisztrált látogató nézi-e az oldalt.

Ha igen akkor, ha ez nem az adminisztrátor, akkor \$kimenet-be generálja `form1` nevű űrlapot, amit a `kosar.php` dolgoz majd fel. Hozzáadja a kosár vezérlőit. Azaz a megvásárolandó darabszámot, az ennek megfelelő szerkesztő doboz nevét a „x” és a `dvd` elsődleges kulcsának összeillesztésével adja. Majd egy „Kosárba” felíratú `submit` típusú gombot. Adminisztrátor esetén az űrlapot szintén a `kosar.php` dolgozza fel. Az űrlap tartalmaz két gombot: a törlés gomb nevét „y” és a `dvd` elsődleges kulcsának összeillesztésével adja. A szerkesztés gomb esetén a hasonlóan jár el csak ott „z” a prefix. Majd generál egy rejtett mezőt is, aminek a neve `dvd` lesz és értékékként a `$_REQUEST['dvd']`-t kapja.

Majd a `$_REQUEST['elozo']` alapján generál egy Vissza felíratú linket. Ezután generálja az előzetesre mutató linket, ha van, majd regisztrált felhasználók által adott pontokat.

Regisztrált felhasználók esetén újabb űrlapot generál (`form3`), melyen rádiógombokkal és a „Szerintem” felíratú gombokkal lehet pontokat adni a filmnek. Ezen űrlap adatait a `pontozas.php` fogja feldolgozni. Rejtett mezőként tartalmazza az űrlap a `dvd` elsődleges kulcsát és az `elozo` mező értékeként a `reszlet.php`-t kapja meg.

Regisztrált felhasználó esetén generál egy `form4` nevű űrlapot is, amelyben a kritikát írhat egy szerkesztő mezőbe a felhasználó és elküldheti az „Elküld” gombbal. Az űrlapot a `hozzaszolos.php` fogja feldolgozni. Rejtett mezőként tartalmazza majd a `dvd` elsődleges kulcsát és az `elozo` mező értékeként a `reszlet.php`-t kapja meg.

Ezután külön html táblázatot kezd a dvd kritikáinak. Ehhez lekéri a kritikákat a kritika táblából az adott dvd-hez (hiba esetén üzen és kilép), majd formázottan hozzáadja a \$kimenet-hez. Végül megjeleníti a \$kimenet változó értékét.

30. részlet_felh.php

Változók:

\$ki: Egy string típusú változó. Az uzenet nevű függvény használja a kimenet tárolására.

\$er, \$er2: Erőforrás típusú változók. Szerte a kódban ez a változók kapják meg a mysql_query() függvények által visszaadott erőforrásokat.

\$sor, \$sor2: Asszociatív tömbök. Szerte a kódban a mysql_fetch_array függvény által lekért rekord mezőit tárolják.

\$felvett_db, \$leszallitott_db: Integer változók. A felhasználó „felvéve” ill. „leszállítva” típusú rendeléseinek a számát tárolják.

\$felvett_osszes, \$leszallitott_osszes: Integer változók. A felhasználó „felvéve” ill. „leszállítva” típusú rendeléseinek az összértékét mutatják meg.

\$kimenet: String típusú. A html kimenetet tároló globális változó.

Függvények:

function uzenet(\$szov, \$hova): Egy üzenetet ír ki az oldalra. Két paramétere van. A \$szov a kiírandó szöveget adja meg. A \$hova pedig, hogy hová vigye a felhasználót a Vissza feliratú nyíl. A függvény a \$ki változóba gyűjti a html kimenetet. A vissza feliratú link kimenetének generálásakor ellenőrzi, hogy a \$hova a fő oldalra mutat-e, mert ha igen akkor az ablak teljes tartalma fog frissülni, azaz beállítja a link target tulajdonságát _top-ra. A függvény miután befejezte a kimenet generálást, kilép.

A függvényeken kívül eső kód:

A fájl elején beágyazza az include/opencv.php, include/session.php fájlokat. Majd megnyitja a sessiont. A függvény definiálása után ellenőrzi, hogy a session-ben létezik-e felhasznalo nevű változó ill. az értéket „admin”-e. Ha nem, mind két esetben üzenetet küld (uzenet függvényel) és a szkript befejezi a végrehajtást. Majd ellenőrzi, hogy átadtak-e ennek az oldalnak felh_id nevű változót ill. az értéke szám-e, ha nem ismét üzen és kilép.

Ezután ha az űrlapot még nem töltötték ki, akkor megjeleníti azt. Első lépésben lekérdezi az adatbázisból a felh_id elsődleges kulcsú felhasználót. Hiba esetén üzen és befejezi a végrehajtást. Siker esetén új tranzakciót indít és lekérdezi a rendeles táblából a felhasználó „felvéve” állapotú rendeléseinek a számát. Itt kap értéket a \$felvett_db. Majd hasonlóan lekérdezi a „leszállítva” állapotú rendeléseket, itt pedig a \$leszallitott_db kap értéket.

Ezután lekérdezi az adatbázisból a felhasználó „felvéve” állapotú rendeléseinek összértékét, itt kap értéket a \$felvett_osszes. Majd lekérdezi az adatbázisból a felhasználó „leszállítva” állapotú rendeléseinek összértékét, itt kap értéket a \$leszallitott_osszes.

Majd generálni kezdi a \$kimenet értékét. Az űrlapot ugyanez a fájl fogja feldolgozni. Az alapértelmezett értékeket ahol létezik, kitölti. Minden szerkesztő doboz az űrlapon disabled állapotot kap, kivéve a megjegyzés mezőt (mgj), csak ezt szerkesztheti az adminisztrátor. Végül megjeleníti a \$kimenet értékét és véget ér a szkript végrehajtása.

Ha az űrlapot beküldték, akkor tranzakciót indít és megpróbálja egy sql update-el frissíteni a felhasználó megjegyzés mezőjét. Hiba esetén üzen és befejezi a végrehajtást. Sikeres módosítás esetén hasonlóan jár el, de itt a `reszlet_felh.php`-ra mutató linket ad vissza az üzenet. Valamint url-ben átadja a `felh_id` változót és annak értékét.

31. `ujdvd.php`

Változók:

\$ki: Egy string típusú változó. Az `uzenet` nevű függvény használja a `kimenet` tárolására.

\$er, \$er2: Erőforrás típusú változók. Szerte a kódban ez a változók kapják meg a `mysql_query()` függvények által visszaadott erőforrásokat.

\$sor, \$sor2: Asszociatív tömbök. Szerte a kódban a `mysql_fetch_array` függvény által lekért rekord mezőit tárolják.

\$kimenet: String típusú. A html kimenetet tároló globális változó.

\$fajl: String. A dvd-hez tartozó előzetes fájl nevét tartalmazza.

\$kep: String. A dvd-hez tartozó kép fájl nevét tartalmazza.

\$kat: Integer. A dvd-hez tartozó kategória elsődleges kulcsát tartalmazza.

\$sql: String típusú. Az egyes rendeléseket összeállító lekérdezést tároló változó.

Függvények:

`function uzenet($szov, $hova="fo.php")`: Egy üzenetet ír ki az oldalra. Két paramétere van. A `$szov` a kiírandó szöveget adja meg. A `$hova` pedig, hogy hová vigye a felhasználót a Vissza feliratú nyíl. Ennek a paraméternek az alapértelmezett értéke a „fo.php”

A függvény a `$ki` változóba gyűjti a html kimenetet. A vissza feliratú link kimenetének generálásakor ellenőrzi, hogy a `$hova` a fő oldalra mutat-e, mert ha igen akkor az ablak teljes tartalma fog frissülni, azaz beállítja a link target tulajdonságát `_top`-ra. A függvény miután befejezte a kimenet generálást, kilép.

A függvényeken kívül eső kód:

A fájl elején beágyazza az `include/opendb.php`, `include/session.php` fájlokat. Majd megnyitja a sessiont. Ez után leellenőrzi, hogy a `felhnev`, `jelszo`, `felh_id` változók szerepelnek-e `$_SESSION`-ban, majd, hogy a `felhnev` értéket „admin”-e, ha nem teljesülnek a feltételek mindkét esetben üzen (`uzenet` függvény) majd befejeződik a végrehajtás.

Ha úgy került a vezérlés erre az oldalra, hogy egy dvd adatait szerkeszteni akarták, akkor ellenőrződik, hogy a `$_REQUEST['dvd']` értéke szám-e, ha nem akkor üzen és befejeződik a végrehajtás. Majd egy `sql select`-ben lekérdezi a dvd összes adatát. Majd generálja a `$kimenet` változóba a dvd szerkesztés űrlapját. Az űrlapot ugyanez a fájl fogja feldolgozni. A beviteli vezérlőkhöz ahol létező adat van azt alapértelmezett értéként a vezérlőben meg is, jeleníti. A `katteg` nevű kombinált lista feltöltéséhez egy tranzakciót indít és lekérdezi az összes sort a `katteg` táblából. Majd itt is összehasonlítás alapján alapértelmezésként a már megadott kategóriát fogja megjeleníteni. A fájl feltöltéshez meg kell

adni, a feltöltött fájl maximális méretét, amit itt 10 MB-ra állít be. Majd rejtett mezőben megadja a `$_REQUEST['dvd']`-t, a mező neve „dvd” lesz. Végül megjeleníti a `$kimenet` értékét és a végrehajtás befejeződik.

Ha az új dvd-t hoz létre az adminisztrátor és még nem kapta meg azt az űrlapot ahol a dvd adatait viheti fel, akkor a szkript megjeleníti neki. Elkezdí írni a `$kimenet` változót az űrlappal. A kombinált listában kategória neveinek megjelenítéséhez lekérdezést hajt végre az adatbázisban. Az űrlapot ugyanez a fájl dolgozza fel. A fájl feltöltéshez meg kell adni, a feltöltött fájl maximális méretét, amit itt 10 MB-ra állít be. Végül megjeleníti a `$kimenet` értékét és itt befejeződik a végrehajtás.

Ha elküldte az űrlapot, akkor ellenőrzi a `$_POST`-ban hogy a cím, `rovid_leiras`, `ear`, `mgj_datum` értéket kaptak-e, ha nem akkor üzen erről és kilép. Majd vizsgálja hogy az `ear`, `akcio` változók értékei egészek-e. Ezután ellenőrzi a `mgj_datum` változó formátumát a helyes formátum (éééé-hh-`nn`), ha nem megfelelő a forma, akkor üzen és kilép.

Majd leellenőrzi, hogy feltöltöttek-e előzetest, ha igen akkor ellenőrzi, hogy megvan-e az ideiglenes könyvtárban és névvel a fájl. Ha igen akkor átmozgatja a fájlt az „elozetes” könyvtárba a feltöltött névre átnevezve. Ha ez nem sikerül akkor üzen és kilép. Itt kap értéket az `$fajl`.

Majd leellenőrzi, hogy feltöltöttek-e képet, ha igen akkor ellenőrzi, hogy megvan-e az ideiglenes könyvtárban és névvel a fájl. Ha igen akkor átmozgatja a fájlt az „kep” könyvtárba a feltöltött névre átnevezve. Ha ez nem sikerül, akkor üzen és kilép. Itt kap értéket az `$kep`.

Ezután lekérdezi és kiválasztja a kategória táblából az űrlapon kiválasztott kategóriának megfelelő elsődleges kulcsot, és bele írja a `$kat` változóba, ha sikerül. Hiba esetén üzenetet küld és befejeződik a végrehajtás.

Majd ellenőrzi, hogy ha az `akcio` szerkesztő mezőnek nem adtak értéket, akkor kapjon „null”-t az insert ill. update miatt.

Ezután ha szerkesztették az űrlapot, és úgy küldték el, akkor tranzakció nyílik és a módosításokról egy update sql utasítás generálódik a `$sql`-be az űrlap adatokkal. Itt a `rovid_leiras` szerkesztő mező string értékét levágja 255 karakter hosszúra. Végül lefuttatja a lekérdezést tartalmazó `$sql`-t az adatbázisban. Hiba esetén és sikeresség esetén is üzen, majd leállítja a végrehajtást.

Ha új dvd-t vettek fel és az űrlapot és úgy küldték el, akkor tranzakció nyílik és a módosításokról egy update sql utasítás generálódik a `$sql`-be az űrlap adatokkal. Itt a `rovid_leiras` szerkesztő mező string értékét levágja 255 karakter hosszúra. Végül lefuttatja a lekérdezést tartalmazó `$sql`-t az adatbázisban. Hiba esetén és sikeresség esetén is üzen, majd leállítja a végrehajtást.

Az oldal fejlesztésének lehetőségei

Ahhoz, hogy a webáruház mindennapi használatban is üzemeltethető legyen szükséges a számla ill. bizonylat kezelés létrehozása, azaz számla kiállítás nyomtatás,... stb.

Az esetleges raktár kezeléssel is lehet fejleszteni az oldalon.

A felhasználói felület dizájnosabbá tételével is növelhető az oldal népszerűsége a felhasználók között. Ilyen szempontból akár CSS stíluslapok alkalmazása is célszerű lehet.

A kategóriák automatikus kezelése is fejlesztési lehetőség, azaz ezek létrehozás, törlése, módosítása. Egyszerre több kategória kiválasztása a filmekhez is hasznos dolog lenne.

Látványosabbá tehető az oldal, ha a film előzetest már az oldalon is meg lehetne tekinteni egy videó lejátszóval.

Az oldalhoz létre lehetne hozni egy fórumot is, amin a felhasználók tanácsokat kérhetnek, ötleteket adhatnak egymásnak film ügyekben, vagy filmekkel kapcsolatos témákról diskurálhatnak.

Az oldalon a pénzügyi kimutatásokat lehetne fejleszteni. Többféle ÁFA kezelését is lehetne beilleszteni, így nem csak bruttó árakkal dolgozna.

Összefoglalás

Sikerült létrehozni egy műsoros DVD-k árusításával foglalkozó webáruházat PHP nyelven, mely MySQL-t adatbázis kezelővel és Apache webserverral. A kitűzött célokat többségében sikerült megvalósítani.

Létrejött az adminisztrációs felület, a regisztrált felhasználói felület, valamint a nem regisztrált felhasználók is kereshetnek a termékek között, nézelődhetnek az oldalon de, vásárolni nem tudnak. Sikerült megvalósítani, hogy az oldal leveleket küldjön e-mail címekre például a regisztráció megerősítésére, az egyes rendelésekről, stb.. Azonban volt olyan internetes levelező (pl. mail.yahoo.com) melyre nem érkeztek meg helyesen a levelek.

Az oldal elég jól használható a céljára. Könnyen átlátható és használható és informatív felhasználói felületet kapott.

Nem sikerült megvalósítani a dizájnosabb felhasználói felületet de erre már nem lett volna időm, így kicsit szegényesebb külsővel rendelkezik.

Az oldalt két internetes böngészőn lett kipróbálva az Internet Explorer és Mozilla Firefox és helyesen jelenítette meg az áruházat.

Ezen webáruház elkészítése után nagyobb rálátással rendelkezem a webes programozás területére, mind a lehetőségekre, mind a buktatókra és adott egy kis bevezetést a webes technológiákba. Így számomra ez a dolgozat elkészítése mindenképpen pozitív eredménnyel zárult.

Irodalomjegyzék

- [1] Peter Moulding (2002): *PHP Haladóknak Fekete Könyv*, Perfact-Pro Kft., Budapest.
- [2] Julie C. Meloni (2003): *A PHP, a MySQL és az Apache használata*, Panem Kiadó, Budapest
- [3] MySQL 5.0, manual.chm
- [4] <http://steve82.extra.hu>
- [5] <http://weblabor.hu/cikkek/munkamenetkezeles1>
- [6] http://hu.wikipedia.org/wiki/Apache_HTTP_Server
- [7] <http://hu.wikipedia.org/wiki/MySQL>
- [8] <http://hu.wikipedia.org/wiki/PHP>