

SZAKDOLGOZAT

**Téma: Az informatika iskolai
alkalmazása**

Debrecen

2008

DEBRECENI EGYETEM

INFORMATIKAI KAR

Könyvtári kölcsönző-program készítés PHP alapokon

Témavezető:

Dr. Rutkovszky Edéné
Egyetemi tanársegéd

Készítette:

Kész Béla
Informatika tanár szakos
kiegészítő, levelező hallgató

Debrecen

2008

Tartalomjegyzék

I. Bevezetés	5
II. Előzmények	6
<i>A könyvtár és a számítógép kialakulása és elterjedése</i>	6
<i>A könyvtár és a számítógép funkciói</i>	7
III. A fejlesztés folyamán alkalmazott technológiák áttekintése	9
<i>Az Internet</i>	9
<i>A Web</i>	11
<i>HTML</i>	13
<i>CSS</i>	13
<i>JavaScript</i>	14
<i>Kliens-szerver architektúra</i>	14
<i>Webszerver</i>	16
<i>Apache</i>	17
<i>DBMS</i>	18
<i>MySQL</i>	18
<i>A PHP</i>	19
IV. A program részletes bemutatása	21
Környezet:	21
Telepítési útmutató	22
Az adatbázis:	24
Az alkalmazás:	26
<i>Nyitóoldal:</i>	27
<i>Bejelentkezés</i>	28
<i>Regisztráció</i>	29
<i>Bejelentkezés után</i>	31
SZEMÉLYES ADATOK	32
<i>Módosítás</i>	32
<i>Jelszó csere</i>	34
KÖNYVTÁR	35
<i>Keresés</i>	35
<i>Előjegyzéseim</i>	38

<i>Könyveim</i>	39
<i>Tartozásaim</i>	40
RÓLUNK	41
<i>Bemutakozás</i>	41
<i>Kapcsolat</i>	41
ADMINISZTRÁCIÓ	42
<i>Új könyv</i>	42
<i>Könyvek</i>	45
<i>Felhasználók</i>	47
<i>A megfelelő jogosultságok kezelése:</i>	48
Összegzés	49
Köszönetnyilvánítás	50
Irodalomjegyzék:	51

I. Bevezetés

Informatikus hallgatóként témába vágónak találtam egy olyan webes alkalmazás kifejlesztését, melyet bármely leendő munkahelyemen akár már felvételi elbeszélgetéskor felmutathatok, és hasznos referenciaként szolgálhat számomra. Továbbá bármely iskolában egy ezzel a témával foglalkozó tárgyat oktatta a diákok betekintést nyerhetnek a legújabb webes technológiák egy területébe.

Emiatt gondoltam egy mini könyvtári alkalmazás kifejlesztésére, amit akár egy ilyen tárgyat oktatta a szakdolgozatom segítségével bemutathatok lépésről lépésre.

Az általam elkészített webes alkalmazás segítségével az érdeklődők (akik lehetnek az iskola diákjai, tanárai, vagy a könyvtárba beiratkozott emberek) otthonról a böngészőjük segítségével betekintést nyerhetnek a könyvtár könyvei közé. Továbbá a könyvtár adminisztrátora naprakész információkat kaphat a könyvtár bármely könyvéről, beiratkozott felhasználójáról, valamint egyszerűen és gyorsan regisztrálhat könyveket a rendszerbe.

II. Előzmények

A könyvtár és a számítógép kialakulása és elterjedése

A könyvtár tulajdonképpen egyidős a civilizációval, a kommunikáció második forradalmával, az írással, együtt születik meg. Egy olyan megőrző intézmény lett, ahol a megszerzett tudást rendszerezve őrzik, és tovább adják az utókornak. Már az ókorban is léteztek könyvtárak (Ninive, Egyiptom, Görögország, Róma), ahol papirusztekercseket, agyagtáblákat gyűjtöttek.

A történelem során különböző könyvtárak jöttek létre. Előbb magánkönyvtárak voltak, amelyek később nyilvánossá váltak, és a nagyközönség igényeit szolgálták ki. A könyvtár mindig a társadalom igényeit elégítette ki, és ahhoz alkalmazkodott. A középkorban például beszűkült, visszaszorult a kolostorok falai közé, és vallási célokat szolgált. Aztán a reformáció és a polgárosodás korában demokratizálódott, ekkor jelennek meg a közkönyvtárak. Ehhez persze szükség volt a kommunikáció új, immár harmadik forradalmára a nyomtatás feltalálására. Egyre több új könyv és folyóirat jelent meg, amit már sokan értettek. A könyvtár felvállalta az átalakulást, és az új eszmék terjesztőjévé vált. A 19. századra az addigi enciklopédikus könyvtárak már csak bizonyos pontig lehettek enciklopédikusak, mert már nem lehetett összegyűjteni mindent, ami megjelent. Így alakultak ki a szakkönyvtárak. Más könyvtári épületekre volt szükség, a könyvtárosság pedig ekkor vált igazán szakmává.

Tehát a könyvtár sikeresen alkalmazkodott, és biztosította fennmaradását. A 20. század új technikai forradalma emlékeztet a nyomtatás megjelenése után kialakult helyzetre. Még rövidebb idő alatt még több információhoz tudunk jutni, ha értünk a számítógéphez, pontosabban az Internethez. Ahogy Tóth Gyula írja, egyfajta második analfabetizáció alakult ki, hiszen most nem az írást, olvasást kell megtanulnunk, hanem a számítógép használatát. Most nincs új könyvtártípus, hiszen ő az elektronikus könyvtárat csak a hozzáférés új lehetőségének tekinti. A könyvtár azonban most is folyamatosan változik, alakul, ahogy eddig minden korban.

A számítógép története nem nyúlik vissza olyan távolra, mint a könyvtaré. Története csak a 20. században kezdődött, annak is a második felében. Az első

programozható számítógép modelljét ugyan már az 1930-as években elkészítették, az áttörést mégis a II. világháború hozta meg. Az első tisztán elektronikus számítógépet a lövedékek röppályájának kiszámítására építették 1943 és 1946 között Aberdeenben majd a Pennsylvanai Egyetemen. A gép neve ENIAC (Electronic Numerical Integrator And Calculator) volt. 18 ezer elektroncsövet tartalmazott, elhelyezéséhez 30 méternél hosszabb teremre volt szükség.

Az első generáció igazi számítógépe az EDVAC (Electronic Discrete Variable Computer) volt, amely Neumann János elvei szerint készült. A kettes számrendszert használta, külön vezérlő és végrehajtó egységgel rendelkezett, és belső programvezérlése volt. Ma is ezeken az elveken működik a számítógép. A számítógépek második generációját a tranzisztor megjelenésétől szoktuk számítani, amit kapcsolóelemként alkalmaztak a rövid életű elektroncső helyett. Ezáltal jelentősen megnőtt a számítógépek élettartama. Magasabb szintű programozási nyelvek, eszköz független módszerek alakultak ki, és megjelent a szabványosítás is. A harmadik generáció nagy vívmányai az operációs rendszerek és az integrált áramkörök. A 16 bites memória helyett egyre gyakrabban használnak 64 bitet, és a gép már 1 millió műveletet tudott elvégezni egyetlen másodperc alatt. A mai számítógépek már negyedik generációs számítógépek. Méretük jelentősen lecsökkent, gyorsabbak és magasabb fokú integráltságot valósítanak meg, mint elődjeik. A japánok 1991-re már meghírdették az ötödik generációt is, de a szuperszámítógépek még ma sem készültek el. De ahogy fejlődik a technika, biztosak lehetünk benne, hogy hamarosan ezek is megjelennek. A számítógép tehát az elmúlt 50 évben bámulatos karriert futott be. A technika fejlődésével maga is változott, és olyanná alakult, amilyennek ma ismerjük.

A könyvtár és a számítógép funkciói

A könyvtár fő funkciói: őrzi az ismeretanyagot, rendszerezi és szolgáltatja. Nem könyveket gyűjt, hanem tudást halmoz fel. A könyvtár célja minden korszakban ugyanaz marad: beszerez, őrzi és hozzáférhetővé teszi a tudást és az információt és segíti azokat, akik hozzá akarnak férni. Tehát információszolgáltatást végez. Ezzel jutunk el a könyvtári funkció lényegéhez. Tehát ennek az intézménynek tudomást kell szereznie a dokumentumról, meg kell szereznie, tárolnia kell, és azok rendelkezésére kell bocsátania, akiknek a dokumentumra szüksége van. Természetesen csak azoknak lehet rá igénye, akik tudnak a dokumentumról, és olyan ismereteket kapnak róla, amely

nyomán el tudják dönteni, hogy szükségük van-e az abban foglalt információkra. Ez egy újabb funkcióhoz vezet: a dokumentumok létéről és főbb jegyeiről szóló információt rögzítenie kell, és hozzáférhetővé kell tenni (feldolgozás).

A számítógép legfőbb funkciója szintén az információszolgáltatás, csak tágabb körben és gyorsabban. Ahhoz, hogy a számítógép elvégezhesse ezt a funkcióját, hálózatba kell kötnünk, mert egymagában a gép csak saját adattárát használhatja. Ha hálózatba kapcsoljuk, akkor mi is elérhetjük mások gépeinek az adattárát, és mások is a miénket. Többféle hálózat létezik, de számunkra a legfontosabb az Internet. E hálózaton belül szinte bármilyen információ elérhető, még akkor is, ha a világ másik felén lévő számítógépen található. Népszerűsége a 90-es évek elején nőtt meg igazán, amikor megjelent a WWW (World Wide Web). Grafikus felülete nyomán az Internet könnyebben kezelhetővé és jobban élvezhetővé vált. Az Internet megjelenése megváltoztatta a könyvtár működését, funkcióit kitágította. A korábbi statikus gyakorlattal szemben, amikor a könyvtár és használója pontosan tudta, hogy hol és hogyan kereshet a többé-kevésbé jól feltérképezhető adatbázisokban, az Interneten további keresésre alkalmas adatbázisok és maguk az elsődleges vagy annak álcázott, szintén kereshető információk korábban elképzelhetetlen mennyiségben állnak rendelkezésre. Az Internet abban a korszakban van, hogy gazdagabb minden könyvtárnál, a könyvtári funkció, gyűjtő és szolgáltató részének, soha nem ismert bőségét valósítja meg. Legnagyobb problémája azonban, hogy a keresőmotorok és a meta adatok ellenére a rajta lévő információ rendezetlen. Ezért nem lehet könyvtárhoz hasonlítani, mert a könyvtárnak egyik alapfunkciója, hogy rendezetten bocsátja az olvasók rendelkezésére az információkat. Találó az a mondat, hogy amit egy jó könyvtárban maximum 3 óra alatt meg lehet találni, azt a weben akár 3 napig is lehet keresni. Másrészt a weben lévő adatok gyakran nem pontosak, nem hiteles dokumentumok, mert semmiféle szakmai ellenőrzése nincsen. Mindezek ellenére az Internet sok-sok jó, és hasznos anyagot rejt. Ebben áll a könyvtárnak egy új funkciója: segítenie kell az olvasót az Internet használatában, illetve az internetek keresztül már hozzá kell tudnunk férni a könyvtári adatbázisokhoz.

III. A fejlesztés folyamán alkalmazott technológiák áttekintése.

Az Internet

Az **internet** egy nemzetközileg elterjedt, angol eredetű szóból ered (internetwork), magyarul: „hálózatok hálózata”. Az egész világot körülölelő számítógép-hálózat, hatalmas rendszer, amely számítógép-hálózatokat fog össze. Célja az, hogy az egymástól teljesen különböző hálózatok egymással átlátszó módon tudjanak elektronikus leveleket cserélni, állományokat továbbítani.

Az internet úgynevezett TCP/IP-alapú hálózat. Mivel ez a protokollkészlet több hálózatnak is alapja, ezért a globális hálózatot helyi hálózatok, intranetek, különböző távolsági hálózatok alkotják. Mindeközben az adatok a legkülönbözőbb fizikai közegekben utazhatnak telefonvonalak, különböző hálózati kábelek vagy kommunikációs műholdak segítségével. Röviden: az internet nem fizikai hálózat, hanem annak módja, ahogy az egymástól különböző hálózatokat összeköti, hogy egymással kommunikálni tudjanak.

Mivel az internet egymástól különböző hálózatokat köt össze, a felhasználó bátran választhat bármilyen eszközt a munkája elvégzéséhez, az adatokat a hálózaton keresztül egységesen tudja kezelni. A legfontosabb adaléka az, hogy az üzenetszórásos médiumokkal ellentétben itt a felhasználó választhatja meg, hogy milyen információt akar megszerezni. Ugyanígy bárkiből válhat információforrás.

Protokollok

Az Internetre kötött számítógépek vagy a TCP (**T**ransmission **C**ontrol **P**rotocol) vagy az UDP (**U**ser **D**atagram **P**rotocol) protokollt használják, amint azt az alábbi ábrán is látni lehet:

APPLICATION (HTTP,ftp,telnet,...)
Transport (TCP,UDP,...)
Network (IP,.....)
Link (device driver,...)

1. ábra: Hálózati rétegek

Ha egy magas szintű nyelven programot írunk (pl. JAVA, PHP, C++ stb.), ami használja a hálózatot, akkor a felhasználói réteget programozzuk. Ha két program megbízhatóan akar egymással kommunikálni a hálózaton, létrehozunk egy kapcsolatot, és ezen keresztül küldik az adatokat.

TCP- (Transmission Control Protocol - átviteli vezérlő protokoll), egy kapcsolat alapú protokoll, ami az adatok megbízható folyamat adja két számítógép között, vagyis garantálja, hogy az adat, amit küldünk, helyes sorrendben érkezik meg a vevőhöz.

UDP- egy olyan protokoll, ami az adatok olyan független csomagjait továbbítja egyik számítógépről a másikra, amiket adatsomagoknak hívunk, és nincs garancia a megérkezésükre. Az UDP nem kapcsolat alapú, mint a TCP.

IP-cím (Internet Protocol cím) -- egy egyedi hálózati azonosító, amelyet az Internet Protocol segítségével kommunikáló számítógépek egymás azonosítására használnak. Minden, az internetre kapcsolt számítógépnek van IP-címe, de egy-egy konkrét cím nem kötődik feltétlenül egy-egy géphez: egyes gépeknek több címük is lehet (ilyenkor a különböző címek rendszerint a számítógépek különböző hálózati eszközeit azonosítják), vagy több gép osztozhat egy címen (például NAT vagy proxy használata esetén), vagy a gép IP-címe rendszeresen változhat. Ez utóbbi eset különösen a lakossági internet szolgáltatón keresztül kapcsolódó otthoni számítógépekre jellemző.

Az IP-címek 32 biten ábrázolt egész számok, amelyeket hagyományosan négy darab egybájtos, azaz 0 és 255 közé eső, ponttal elválasztott számmal írunk le a könnyebb olvashatóság miatt (például: *192.168.42.1*). A címek felépítése hierarchikus: a szám bal oldala (vagy szakmai nevén a *legnagyobb helyi értékű bájtt*) a legfelső szintet jelenti, és jobbra haladva az ez alatti szinteket kapjuk meg, például egy szolgáltatót, a szolgáltató alatti ügyfeleket, és az ügyfelek alatti egyes számítógépeket.

Az internetet használó számítógépek minden tranzakció során (például egy weboldal megjelenítésekor) megadják IP-címüket, ami révén elvileg be lehet azonosítani az adott gép helyét és tulajdonosát, illetve egy adott géppel végzett tevékenységekről sok információt lehet gyűjteni. A gyakorlatban a számítógépek jelentős része az internetszolgáltatójától rendszeresen új IP-címet ad, így azonosítása csak a szolgáltató együttműködésével lehetséges, aki csak rendőri megkeresésre és más jól meghatározott esetekben adhatja ki. Ezzel együtt az IP-cím számos országban, többek között Magyarországon is személyes adatnak számít.

A Web

A **Web vagy Világháló** (angol eredetiben *World Wide Web, WWW*) az interneten működő, egymással úgynevezett hiperlinkek-kel összekötött dokumentumok rendszere. A rendszert web böngésző program segítségével lehet elérni. Ez a program képes megjeleníteni az egyes dokumentumokat, „weblapokat”. A felhasználó a lapokon található hiperlinkek segítségével további lapokat kérhet le, amelyeken újabb hiperlinkek lehetnek. A rendszer „háló”- jellegét is ez adja; a dokumentumok a háló csomópontjai, míg a hiperlinkek a háló szálai, amelyeken keresztül egy vagy több lépésben tetszőleges csomóponthoz eljuthatunk.

A Web alapelveit Tim Bernes-Lee, a CERN részecskefizikai kutatóközpont munkatársa dolgozta ki 1989-ben. Eredeti célja a különböző intézményekben világszerte dolgozó kutatók közötti automatizált információ megosztás volt. Az alapötlet egy globális információs hálózat létrehozása volt a számítógépek, a számítógépes hálózat és a hipertext képességeinek ötvözésével.

A Világháló három szabványra épül:

- Az *Uniform Resource Locator* (URL), leírja, milyen egyedi „címmel” kell rendelkeznie az egyes oldalaknak;
- A *hipertext átviteli protokoll* (*Hyper Text Transfer Protocol*, HTTP), megadja, hogyan küld egymásnak információt a böngésző és a kiszolgáló,
- *hipertext leíró nyelv* (*Hyper Text Markup Language*, HTML), az információkódolás eljárása mellyel az oldal sokféle eszközön megjeleníthetővé válik.

URL- az interneten található erőforrásokra tud hivatkozni. Két fő komponensből áll:

- Protokollazonosító (Protokoll identifier)
- Erőforrás neve (Resource name)

Például: <http://java.sun.com> – Ez egy URL, ami a Java weboldalára mutat, amit a Sun Microsystems üzemeltet. Ahol is a http-a protokollazonosító, majd az erőforrás neve következik, amit kettősponttal választunk el

HTML – később bővebben kifejtem.

HTTP (*Hyper Text Transfer Protocol*) egy információátviteli protokoll a világhálón. Az eredeti célja a HTML lapok publikálása és fogadása volt. A HTTP fejlesztését a World Wide Web Consortium koordinálta.

A HTTP egy kérdés-válasz alapú protokoll kliensek és szerverek között. A kommunikációt mindig a kliens kezdeményezi. A HTTP klienseket gyűjtőnéven user agent-nek is nevezik. A user agent jellemzően, de nem feltétlenül web böngésző.

A HTTP általában a TCP/IP réteg felett helyezkedik el, de nem függ tőle. A HTTP implementálható más megbízható átviteli réteg felett is akár az interneten akár más hálózaton.

HTML

A HTML (angolul: Hyper Text Markup Language - hiper szöveges jelölő nyelv) egy leíró nyelv, melyet weboldalak készítéséhez fejlesztettek ki, és mára már internetes szabvánnyá vált a W3C (World Wide Web Consortium) támogatásával.

HTML általában szöveges állományokban található meg olyan számítógépeken, melyek az internethez kapcsolódnak. Ezek az állományok tartalmazzák azokat a szimbólumokat, amelyek a megjelenítő programnak leírják, hogyan is kell megjeleníteni illetve feldolgozni az adott állomány tartalmát. Megjelenítő program lehet egy web böngésző (angolul: web browser), levelező program (mint például: Microsoft Outlook stb.), valamint egyéb eszközök, például mobiltelefon.

CSS

A CSS (angolul Cascading Style Sheets) a számítástechnikában egy stílusleíró nyelv, mely a HTML vagy XHTML típusú strukturált dokumentumok megjelenését írja le. Ezen kívül használható más, hasonló struktúrájú dokumentumok leírására is(például:XHTML).

A CSS-t a weblapok szerkesztői és olvasói egyaránt használhatják, hogy átállítsák vele a lapok színét, betűtípusait, elrendezését, és más megjelenéshez kapcsolódó elemeit. A tervezése során a legfontosabb szempont az volt, hogy elkülönítsék a dokumentumok struktúráját (melyet HTML vagy egy hasonló leíró nyelvben lehet megadni) a dokumentum megjelenésétől (melyet CSS-sel lehet megadni). Az ilyen elkülönítésnek több haszna is van, egyrészt növeli a weblapok használhatóságát, rugalmasságát és a megjelenés kezelhetőségét, másrészt csökkenti a dokumentum tartalmi struktúrájának komplexitását. A CSS ugyancsak alkalmas arra, hogy a dokumentum stílusát a megjelenítési módszer függvényében adja meg, így elkülöníthető a dokumentum formája a képernyőn, nyomtatási lapon, hangos böngészőben (mely beszédszintetizátor segítségével olvassa fel a weblapok szövegét).

JavaScript

A JavaScript ma az egyetlen jelentős kliensoldali szkriptnyelv. Weboldalak milliói használják űrlapok helyes kitöltésének ellenőrzéséhez, sütik kezeléséhez, a felhasználói élmény növeléséhez. Sajnos sok olyan oldal van ahol az alkalmazott JavaScript kód öncélú, sőt meglehetősen zavaró az oldal használója számára.

Kliens-szerver architektúra

A kliens-szerver kifejezést először az 1980-as években használták olyan számítógépekre (PC-kre) amelyek hálózatban működtek. A ma ismert modell a 80-as évek végén vált elfogadottá. A kliens-szerver szoftver architektúra egy sokoldalú, üzenetalapú és moduláris infrastruktúra, amely azért alakult ki, hogy a használhatóságot, rugalmasságot, együttműködési lehetőségeket és bővíthetőséget megnövelje a centralizált, nagygépes, időosztásos rendszerekhez képest.

A **kliens** (angolul client) olyan számítógép, amely hozzáfér egy (távoli) szolgáltatáshoz, amelyet egy számítógép hálózathoz tartozó másik gép nyújt.

Jellemzői:

- Kéréseket, lekérdezéseket küld a szervernek
- A választ a szervertől fogadja.
- Egyszerre általában csak kisszámú szerverhez kapcsolódik
- Közvetlenül kommunikál a felhasználóval, általában egy GUI-n (Graphical User Interface = Grafikus felhasználói felület) keresztül.

A **kiszolgáló** vagy **szerver** (angolul server), olyan (általában nagyteljesítményű) számítógépet, illetve szoftvert jelent, ami más gépek számára a rajta tárolt vagy előállított adatok felhasználását, a kiszolgáló hardver erőforrásainak (például nyomtató, háttértárolók, processzor) kihasználását, illetve más szolgáltatások elérését teszi lehetővé.

Jellemzői:

- Passzív, a kliensektől várja a kéréseket
- A kéréseket, lekérdezéseket feldolgozza, majd visszaküldi a választ
- Általában nagyszámú klienshez kapcsolódik egyszerre

- Általában nem áll közvetlen kapcsolatban a felhasználóval

A kiszolgálókat többféleképpen csoportosíthatjuk, például:

- a funkciójuk szerint, például webkiszolgálók, FTP- kiszolgálók, adatbázis-kiszolgálók;
- a kiszolgált kör alapján, például internetes kiszolgálók, intranetes kiszolgálók;
- a teljesítményük alapján.

A kliens-szerver olyan architektúra, amely elválasztja egymástól a klienst és a szervert, és az esetek nagy többségében egy számítógép hálózaton alakítják ki. A hálózat klienseit és szervereit más néven csomópontnak (angolul node) is nevezhetjük. A kliens-szerver architektúra legalapvetőbb formájában mindössze kétfajta csomópont van, a kliens és a szerver. Ezt az egyszerű architektúrát **kétszintűnek** (angolul two-tier) hívják.

Bonyolultabb architektúrák is léteznek, amelyek 3 különböző típusú csomópontból állnak: kliensből, alkalmazás szerverből (application server) valamint adatbázis szerverből (database server). Ezt **háromszintű** (three-tier) architektúrának hívják, és a leggyakrabban alkalmazott a kliens-szerver megoldások közül. Amelyek kettőnél több szintet tartalmaznak **többszintű** (multi-tiered) és m-szintű (m-tiered) architektúrának is nevezzük.

A háromszintű kiépítésben az alkalmazásszerverek azok, amelyek kiszolgálják a kliensek kéréseit, és az adatbázisszerverek az alkalmazásszervereket szolgálják ki adatokkal. Ennek a rendszernek nagy előnye a bővíthetőség.

A több szintű kiépítés előnye, hogy egyensúlyozza és elosztja a feldolgozásra váró adatmennyiséget és munkát a több és gyakran redundáns, specializált csomópont között. Ez javítja a rendszer teljesítményét, és a megbízhatóságát is, hiszen a feladatok párhuzamosan több szerveren is elvégezhetőek. Hátránya, hogy nagyobb az adatátviteli forgalom a hálózaton és, hogy nehezebben programozható illetve tesztelhető egy kétszintű architektúránál, mert több eszközt kell összehangolni a kliensek kéréseinek kiszolgálásához.

A kliens-szerver architektúra előnyei:

Lehetővé teszi, hogy a feladatokat elosszuk olyan számítógépek között, amelyek csak a hálózaton keresztül érintkeznek egymással, ami megkönnyíti a karbantartás elvégzését. Megoldható például, hogy javítsunk, frissítsünk, áthelyezzünk vagy akár kicseréljünk egy szervert anélkül, hogy klienseire ez bármilyen hatással lenne.

Az összes adat a szerveren vagy szervereken tárolódik, amelyek általában sokkal erőteljesebb biztonsági ellenőrzéssel rendelkeznek, és jobban tudják szabályozni az erőforrásokhoz és adatokhoz való hozzáférést.

Mivel az adattárolás centralizált, könnyebb frissíteni az adatokat, mint amennyire ez egy P2P rendszerben lehetséges lenne.

Web szerver

A **webkiszolgáló/web szerver** egy kiszolgáló, mely elérhetővé teszi a helyileg (esetleg más kiszolgálón) tárolt weblapokat a HTTP protokollon keresztül. A HTTP web szerverekhez web böngészőkkel lehet kapcsolódni.

Egy webszerver kéttípusú lehet:

1. egy erre a célra kialakított számítógép;
2. egy számítógépes program, mely a háttérben futva biztosítja a weblapok elérését.

Bár a webszerverek többnyire különböznek a részletekben, az alapvető funkcióik azonosak. Minden webszerver HTTP kéréseket fogad a hálózatról, és HTTP válaszokat küld vissza. A HTTP válasz az esetek többségében egy HTML dokumentum, de lehet még egyszerű szöveges fájl, kép, vagy más típusú fájl is.

A webszerverek a kienstől kapott kérésekben többek között URL címet (pl. www.pelda.com) kapnak, melyet aztán majd a maguk módján értelmeznek.

Ha nem létezik egy adott kérés, akkor 404-es hibakóddal tér vissza, ha hozzáférhető, akkor beolvassa, elvégzi rajta az esetleges további műveleteket, majd elküldi a kliensnek.

A legnépszerűbb HTTP webserverek a következők:

- Apache HTTP Server az Apache Software Foundation terméke.
- Internet Information Services (IIS) a Microsoft terméke.
- Sun ONE a Sun Microsystems terméke.

Apache

Az **Apache HTTP Server** (röviden **Apache**) egy nyílt forráskódú webkiszolgáló alkalmazás, szabad szoftver, mely kulcsfontosságú szerepet játszott a World Wide Web elterjedésében. Az Apache egy robosztus, erőteljes és rugalmas web szerver, amit többek között a következő operációs rendszerekhez készítették el: Unix, Linux, Mac OS X és Microsoft Windows.

1996 óta az Apache az Interneten legszélesebb körben használt webkiszolgáló. 2008 februárjában az Apache a Föld webhelyeinek 50,93%-át szolgálja ki.

Az Apache sok szabványt támogat, melyeknek nagy része fordított modulok formájában áll rendelkezésre a mag kiegészítéseként. Ezek a modulok sok területet lefednek a kiszolgáló oldali programnyelv támogatástól kezdve a hitelesítési sémákig. Az ismertebb, támogatott programnyelv modulok a mod_perl, a mod_python és a PHP.

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

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

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.

DBMS

Az **adatbázis kezelő rendszer** (angolul Database Management System, **DBMS**) több felhasználós, hálózatos környezetben, adatbázisokhoz való hozzáférést biztosító szoftveralkalmazás. Eredetileg csak nagy cégek alkalmaztak adatbázis kezelőket, melyek nagy mennyiségű adatot kezeltek nagy teljesítményű számítógépekkel, de mára ezek elterjedt komponensei a cégek számítástechnikai rendszereinek.

Az adatbázis-kezelők meghatározó része a logikai adatbázis kialakítására koncentrál. A logikai adatbázisokat szerkezetük, felépítési és működési sajátosságaik alapján különböztetjük meg, amelyet összefoglalóan, tömören adatmodellnek nevezünk. Értelemszerűen tehát az adatbázis-kezelőket az általuk kezelt, vezérelt, valamint megvalósított adatmodellek alapján osztályozzuk. Ennek megfelelően beszélhetünk relációs, objektumorientált és objektumrelációs adatbázis kezelőkről és adatbázisokról.

1986-ban az **SQL**, mint a relációs adatbázisok lekérdezőnyelve az Egyesült Államokban is, és Európában is szabványossá válik.

Manapság a legfontosabb intézményi információs rendszerek alapját maguk az adatbázisok alkotják, ugyanakkor tipikusan objektumorientált programozási nyelveken írt alkalmazások használnak relációs adatbázisokat. Ezt a fajta tudathasadást egy öszvér megoldással igyekeznek napjainkban áthidalni úgy, hogy objektumszerűen lehet elérni relációs adatbázistartalmakat. Erre, mint tudjuk, a logikai adatbázis fogalma eleve lehetőséget ad.

MySQL

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

A szoftver fejlesztője a svéd MySQL AB cég, amelyet 2008 januárjában a Sun felvásárolt.

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

A PHP

A PHP (PHP: Hypertext Preprocessor) egy 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 legtöbb webserverre, 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 webservert egyik legnépszerűbb beépülő modulja.

A PHP legfrissebb változata az 5.2.6 verziószámú, amely 2008. május 1-jén jelent meg.

A **PHP** fejlődése kezdetén csak CGI- programok halmaza volt. Ezeket Lerdorf néhány Perl szkript lecserélésére írta, amelyeket honlapjának karbantartására használt. Később ezeket a programokat kombinálta a szintén általa írt *Form Interpreter* (ürlap-értelmező) alkalmazással - így jött létre a *PHP/FI*, ami már jóval, szélesebb funkcionalitással bírt. Az új, C nyelven megírt változat képes volt adatbázisokhoz kapcsolódni és segítségével egyszerű dinamikus weboldalak is létre lehetett hozni. Lerdorf **1995. június 8-án adta ki a PHP első nyilvános változatát**, hogy külső segítséggel gyorsabbá tegye a hibák megtalálását és a kód további fejlesztését. Az új verziót **PHP 2**-nek nevezte el és már megtalálhatóak voltak benne a mai PHP alapvető tulajdonságai: a Perl-éhez hasonló változók, az ürlapok kezelése és a HTML-kód beszúrásának lehetősége. A PHP szintaktikája is hasonló volt a Perl-éhez, de annál jóval korlátoltabb, egyszerűbb és kevésbé egységes volt.

1997-ben két izraeli fejlesztő újraírta az értelmezőt, ezzel megteremtve a *PHP 3* alapját - ekkor született meg a PHP új neve, a *PHP: Hypertext Preprocessor* rekurzív rövidítés is. Ők alapították meg a *Zend Technologies*-t is, ami máig aktívan ellenőrzi a PHP fejlesztését.

A **PHP 4** 2000. május 4-én jelent meg. Ezt követte 2004. július 13-án a következő nagy mérföldkőnek számító, az új *Zend Engine II*-n alapuló **PHP 5**. Az ötös verzió sok újítást tartalmazott: fejlettebb objektum-orientált programozási lehetőségeket, a PDO (*PHP Data Objects*) adatbázis-absztrakciós kiterjesztést, és sok teljesítményt növelő javítást is.

A PHP működése.

A PHP oldalak elkészítésénél a HTML-t gyakorlatilag csak, mint formázást használják, ugyanis, azon 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 web szervernek.

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 pl. a bejelentkezés, az adatbázis kezelés, file kezelés, kódolás, adategyeztetés, kapcsolatok létrehozása, e-mail küldése, adatfeldolgozás, dinamikus listakészítés stb. Minden olyan esetben, ahol nagyszámú ismétlődő feladatsort kell végrehajtani (pl. 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.

IV. A program részletes bemutatása.

Környezet:

A fejlesztést Windows operációs rendszer alatt végeztem és ezen kívül a XAMPP integrált telepítő csomag segítségével történt. Mivel egy szerver alkalmazás telepítése sokszor nem könnyű feladat, ezért döntöttem egy előre csomagolt és minden szükséges alkalmazást telepítő és bekonfiguráló program, a XAMPP mellett.

A telepítő csomagok nagy hátránya viszont az, hogy bár többnyire jól működnek, egy esetleges hiba esetén a megszüntetése eléggé bajos lehet, mivel kevésbé tudunk a csomag működésébe belelátni.

A XAMPP for Windows 1.6.5 változata a következő szoftvereket telepíti és konfigurálja:

- Apache HTTPD 2.2.6+Openssl 0.9.8g
- MySQL 5.0.51
- PHP 5.2.5
- PHP 4.4.7
- phpMyAdmin 2.11.3
- FileZilla FTP Server 0.9.24
- Mercury Mail Transport System 4.52

Az első négy szoftver már bemutatásra került, a phpMyAdmin egy egyszerű kezelőfelület a MySQL adatbáziskezelő-rendszerhez, a FileZilla FTP Server lehetőséget nyújt a fájljaink eléréséhez, módosításához távoli gépekről. A Mercury Mail Transport System pedig egy SMTP servert üzemeltet be számunkra, mely lehetővé teszi mailek küldését és fogadását a számítógépünkre.

Telepítési útmutató:

Software üzembe helyezésének lépései:

1. A XAMPP letöltése a következő webhelyről:

http://sourceforge.net/project/downloading.php?groupname=xampp&filename=xampp-win32-1.6.8-installer.exe&use_mirror=garr

2. A letöltött/mellékelt (xampp-win32-1.6.8-installer.exe) fájl feltelepítése: semmi extra beállítás nem szükséges, a szokásos módon.
3. XAMPP Control Panel elindítása, itt indíthatjuk el vagy állíthatjuk le a PHP serverünket és a MySQL adatbázisunkat.

- Indítsuk el az Apache-ot. (Ütközés lehet a Skype programmal, mert az is a 80-as portot szeretné használni, tehát vagy leállítjuk a Skype-ot, vagy a beállításaiban kikapcsoljuk a 80-as port használatát (ki kell jelentkezni, aztán Eszközök/Beállítások -> kikapcsolni a 80-as port használatát és egy másik portot beállítani pl.:3293))
- Indítsuk el az MySQL-t.
- Ha nem service-ként futtatjuk az Apachot akkor mindig el kell indítanunk, ha használni akarjuk!

4. XAMPP felhasználói beállítások:

- A server elérése (a böngészőnkbe írjuk) : localhost
- <http://localhost/security/index.php> ezen belül:
<http://localhost/security/xamppsecurity.php>
- Jelszó beállítása a XAMPP-hoz, illetve a MySQL adatbázis root felhasználójának.

5. A MySQL adatbázisunk létrehozása:

- <http://localhost/phpmyadmin/>
- Új adatbázis létrehozása: konyvtar

- Adatbázis beimportálása: Import menüpont alatt, a szöveges fájl pedig az sql.txt fájlunk.
- Jogosultság hozzárendelése: Kezdőlap -> Privilegiumok -> Új felhasználó hozzáadása -> Felhasználó név : konyvtar_admin
Jelszó : konyvtar_pwd

Az összes privilegium megadása. -> Végrehajt.

6. A PHP script elhelyezése a web serverünkön:

- A konyvtar mappa bemásolása a xampp/htdocs mappába, ha a telepítésnél a felkínáltat választottuk, akkor ide: C:\xampp\htdocs\

7. Elérés:

- <http://localhost/konyvtar/>
- Alapból elkészült egy admin fiók: E-mail : kisszati@citromail.hu
Jelszó : 123456
user fiók : E-mail : user@freemail.hu
Jelszó : useruser

8. Elérés kívülről:

- a_gep_ip_cime/konyvtar

Az adatbázis:

Az adatbázis 6 alaptáblából áll: *felhasználó*, *könyv*, *lakhely*, *session*, *előjegyzések* és *kölcsönzések*. A felhasználó táblában tárolom a regisztrált felhasználók adatait, azokét, akik már legalább egyszer valamikor regisztráltak a rendszerben. Ezen táblában megjelenik egy *id_lakhely* mezőnév, mely külső kulcsként hivatkozik a lakhely táblára. Ugyanis a lakhely táblában vannak a felhasználók lakhely adatait. A könyv tábla tartalmazza a könyvtárban megtalálható összes könyvet.

```
--
-- Tábla szerkezet: `felhasznalo`
--
CREATE TABLE `felhasznalo` (
  `id` bigint(20) NOT NULL auto_increment,
  `e_mail` varchar(30) collate utf8_hungarian_ci NOT NULL,
  `jelszo` varchar(35) collate utf8_hungarian_ci NOT NULL,
  `vez_nev` varchar(30) collate utf8_hungarian_ci NOT NULL,
  `uto_nev` varchar(30) collate utf8_hungarian_ci NOT NULL,
  `mobil` varchar(15) collate utf8_hungarian_ci NOT NULL,
  `vezetokes` varchar(15) collate utf8_hungarian_ci default NULL,
  `id_lakhely` bigint(20) NOT NULL,
  `cim` varchar(50) collate utf8_hungarian_ci NOT NULL,
  `hirlevel` tinyint(4) default '1',
  `admin` tinyint(4) default '0',
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_hungarian_ci AUTO_INCREMENT=7 ;

-----
--
-- Tábla szerkezet: `konyv`
--
CREATE TABLE `konyv` (
  `id` bigint(20) NOT NULL auto_increment,
  `cim` varchar(200) character set utf8 collate utf8_hungarian_ci NOT NULL,
  `szerzo` varchar(200) character set utf8 collate utf8_hungarian_ci NOT NULL,
  `kiado` varchar(50) character set utf8 collate utf8_hungarian_ci NOT NULL,
  `kategoria` varchar(50) character set utf8 collate utf8_hungarian_ci NOT NULL default 'egyeb',
  `ev` varchar(4) character set utf8 collate utf8_hungarian_ci NOT NULL,
  `isbn` varchar(30) character set utf8 collate utf8_hungarian_ci NOT NULL,
  `oldalszam` int(4) default NULL,
  `egyeb` text character set utf8 collate utf8_hungarian_ci,
  `darabszam` bigint(10) default '1',
  `ar` int(7) unsigned NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 ROW_FORMAT=DYNAMIC AUTO_INCREMENT=15 ;

-----
--
-- Tábla szerkezet: `lakhely`
--
CREATE TABLE `lakhely` (
  `id` bigint(20) NOT NULL auto_increment,
  `ir_szam` varchar(7) collate utf8_hungarian_ci NOT NULL,
  `helyseg` varchar(35) collate utf8_hungarian_ci NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_hungarian_ci AUTO_INCREMENT=7 ;

-----
--
-- Tábla szerkezet: `session`
--
CREATE TABLE `session` (
  `sid` varchar(32) collate utf8_hungarian_ci NOT NULL,
  `user_id` bigint(20) NOT NULL,
  `last_update` varchar(20) collate utf8_hungarian_ci NOT NULL,
  PRIMARY KEY (`sid`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_hungarian_ci;
```

2. ábra: Felhasználó, könyv, lakhely és session táblasémák.

A session tábla szolgálja az authentication (bejelentkeztetés) funkció alapját. A tábla elsődleges kulcsa a felhasználó session azonosítója (sid). Továbbá tartalmaz egy user_id mezőt mely az adott sessionhoz tartozó felhasználó azonosítója. Végül szerepel még egy last_update mező, melynek értéke a felhasználó tevékenységének legfrissebb pontos dátuma, mely a biztonságot szolgálja, egy adott tétlenségi időkorlát letelte után törölődik a felhasználó authentication-ja.

Továbbá van még két tábla: *előjegyzések* és *kölcsönzések*. Ezekben van eltárolva az éppen előjegyzett vagy már kikölcsönzött könyvek. Az előjegyzések tábla id mezője a tábla elsődleges kulcsa (primary key), mely egy auto_increment mező, vagyis értékét a MySQL adatbázis servere állítja be, minden egyes insert után eggyel növelve a legutóbbi értéket.

```
--
-- Tábla szerkezet: `elojegyzesek`
--
CREATE TABLE `elojegyzesek` (
  `id` bigint(20) NOT NULL auto_increment,
  `id_felhasznalo` bigint(20) NOT NULL,
  `id_konyv` bigint(20) NOT NULL,
  `datum` varchar(25) collate utf8_hungarian_ci NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_hungarian_ci AUTO_INCREMENT=94 ;

-----
--
-- Tábla szerkezet: `kolcsonzesek`
--
CREATE TABLE `kolcsonzesek` (
  `id` bigint(20) NOT NULL auto_increment,
  `id_felhasznalo` bigint(20) NOT NULL,
  `id_konyv` bigint(20) NOT NULL,
  `datum` varchar(25) collate utf8_hungarian_ci NOT NULL,
  `idotartam` varchar(2) character set ucs2 collate ucs2_hungarian_ci NOT NULL default '30'
  COMMENT 'napokban',
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_hungarian_ci AUTO_INCREMENT=16 ;

-----
```

3. ábra: Előjegyzések és kölcsönzések táblasémák

Adatbázis biztonság:

Mostanában a dinamikus tartalmat szolgáltató web alkalmazások sarokkövének számítanak az adatbázisok. Mivel nagyon kényes, titkos adatok tárolására szolgálhatnak ezek az adatbázisok, erősen megfontolandó, miképp védjük meg ezeket.

Információk tárolásához vagy visszakereséséhez csatlakozni kell az adatbázishoz, egy érvényes lekérdezést kell küldeni, az eredményt ki kell tudni olvasni, és le kell zárni a kapcsolatot.

Az adatbázisban tárolt adatok védelme igényli a legnagyobb körütekintést. Az adatok titkosítása kellőképp enyhíti ezt a veszélyt. Olyan teljesen rejtett adatok esetén, amelyeknek nyílt ábrázolásukra nincs szükség, mert nem lesznek kiírva a hashelés technikáját ajánlott alkalmazni.

PHP-ban több függvény is segítségünkre áll e technika megvalósítására. Általában az MD5 titkosítást szokták alkalmazni. Lássunk rá egy példát hogyan is működik:

```
<? php
```

```
$valtozo = "admin";// A kapott érték
```

```
$valtozo_titkositott = md5 ($valtozo);
```

```
print($valtozo_titkositott);
```

```
?>
```

Kimenet: 21232f297a57a5a743894a0e4a801fc3

A titkosítás csak egyirányú, szinte visszafejthetetlen. Éppen ezért alkalmas a felhasználók azonosítására. Adott szövegre mindig ugyanazt a 32 karakterből álló sorozatot adja. Regisztráláskor a titkosított jelszót tároljuk, majd bejelentkezéskor a felhasználó által megadott szöveget md5() függvénnyel átkonvertáljuk, és ezt hasonlítjuk össze az adatbázisban tárolt azonosítóval.

Az alkalmazás:

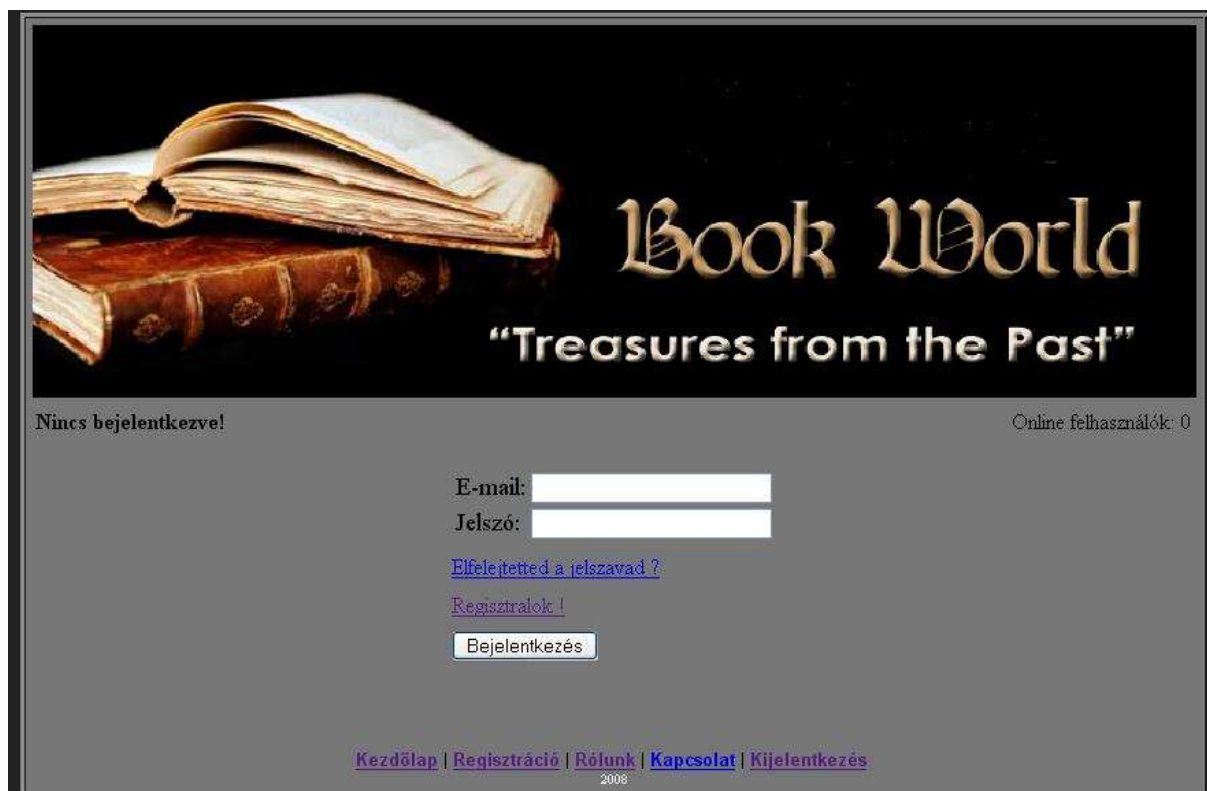
Az elkészült webes alkalmazás legfőbb feladata a könyvtár és a könyvtárt használó emberek – vagy nevezzük őket usereknek – közötti kölcsönzési folyamat felgyorsítása illetve hatékony gördülékeny menedzselése. Ahhoz, hogy a fejlesztett alkalmazás ennek megfeleljen, mind a user mind pedig a könyvtárat üzemeltető admin számára hatékony, könnyen átlátható felületet kell biztosítania.

További alapvető cél, hogy egy olyan alkalmazást fejlesszünk, ami megkönnyíti egy bármilyen méretű könyvtárral rendelkező intézmény dolgozóinak munkáját azzal, hogy a könyveket csak előjegyzés után lehessen kivenni. Erre azért van szükség, hogy gördülékenyebben történhessen a kölcsönzési folyamat.

Nyitóoldal (4. ábra):

Az oldalra látogatót az 4.-es ábrán látható kép fogadja. Kéttípusú felhasználót különböztethetünk meg user és admin, attól függően, hogy milyen funkciót tölt be az alkalmazás „életében”. Usernek nevezhetjük az oldalra látogató érdeklődőt és a már valaha regisztrált embert is. Admin pedig, aki karbantartja a könyvtárt, új könyveket tud regisztrálni, a már kikölcsönzött könyveket törli az illető kölcsönzései közül.

Az oldalra látogatónak első lépésben regisztrálnia kell. E nélkül a lépés nélkül senki sem tud böngészni a könyvtári könyvek között. Ha már regisztrált, akkor az email címe és a megadott jelszava alapján tud belépni a könyvtárba



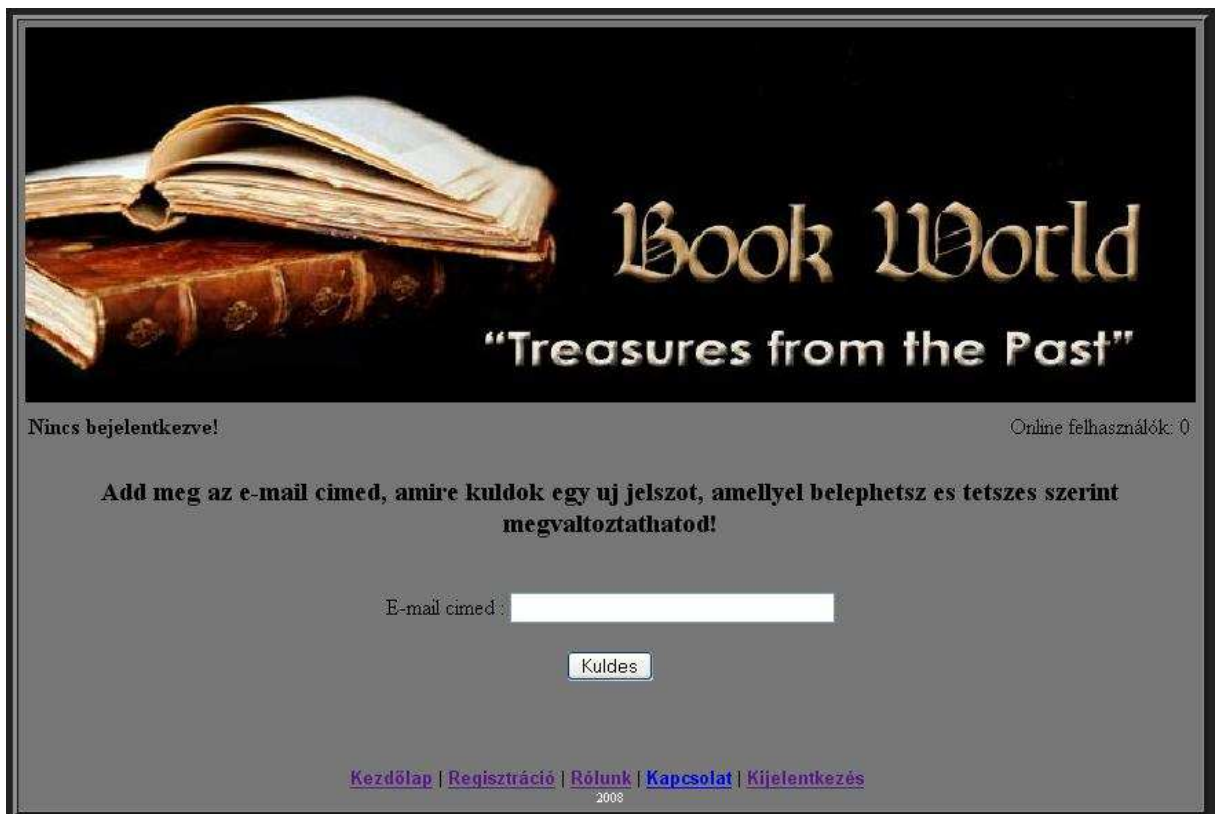
4. ábra: Nyitóoldal

Bejelentkezés

Sok problémára kell felkészülni egy komoly alkalmazás megírásakor. Fontos, hogy kellően fel legyünk készülve minden hiba esetén, amit a felhasználó pontatlansága, vagy emberi mivolta okozhat. Programozó berkekben ezt úgy emlegetik, hogy egy „jó programnak bolond biztosnak kell lennie!”.

Ha a begépett email címet vagy jelszót rosszul adja meg valaki, akkor a **„Rossz felhasználónév vagy jelszó!”** hibaüzenet jelenik meg, a beviteli mező fölött.

Az is gyakran előfordulhat, hogy valaki elfelejti a regisztráláskor megadott jelszavát. Ekkor kell az **„Elfelejtetted a jelszavad?”** feliratú gombra kattintani. Ekkor a következő oldalon találjuk magunkat, amit az 5. ábra mutat. Itt értelemszerűen kitöltve az email címünkre titkosítva kapunk egy új jelszót. Ehhez szükségünk van egy Mail serverre, melyet biztosít számunkra a XAMPP csomag: ez a Mercury Mail Transport System 4.52



Nincs bejelentkezve! Online felhasználók: 0

Add meg az e-mail címed, amire küldök egy új jelszót, amellyel belephetsz és tetszes szerint megváltoztathatod!

E-mail címed :

Küldés

[Kezdőlap](#) | [Regisztráció](#) | [Rólunk](#) | [Kapcsolat](#) | [Kijelentkezés](#)

2008

5. ábra: Elfelejtett jelszó

Regisztráció (6. ábra)

Ha valaki még nem járt az oldalon, akkor kötelező regisztrálnia, ha szeretné a könyvtári alkalmazást használni. A regisztráló gombra kattintva, a 3. ábrán látható képtárul a felhasználó szeme elé:

Nincs bejelentkezve! Online felhasználók: 0

A *-gal jelolt mezok kitoltese kotelezo !!!

Vezetéknév: [text box] *

Utónév: [text box] *

E-mail: [text box] *

Jelszó: [text box] min 6, max 12

Jelszó, újra: [text box] *

Mobil: [text box] *

Vezetékes: [text box] *

Irányítószám: [text box] *

Helység: [text box] *

Utca, házszám: [text box] *

Hírlevél: ☐

Regisztráció

[Kezdőlap](#) | [Regisztráció](#) | [Rólunk](#) | [Kapcsolat](#) | [Kijelentkezés](#)

2008

6. ábra: Regisztrációs űrlap (Form)

Ki kell tölteni ezt a regisztrációs űrlapot (Form-ot). A kötelezően kitöltendő mezők csillaggal vannak jelölve, ezen kívül a kis és nagybetűket nem veszem figyelembe az eltároláskor. Amennyiben a kötelező mezőket elmulasztja kitölteni a felhasználó, vagy hibás adatot ad meg, akkor értelmes a hiba okát megnevező hibaüzenetet kap eredményül: „A * -gal jelölt mezők kitöltése kötelező!!!”

A fenti form forráskódja PHP-ban a következőképpen néz ki (7. ábra):

Ezt egy user_form nevű függvénnyel valósítottam meg, mely a php felhasználásával generálja a html formot. Lényegében a függvényben egy html nevű, string típusú változót generálok, melynek értéke a megjelenítendő form lesz. A form POST módon küldi át az adatait ugyanezen oldalnak (megjelenítő oldal-- lásd 6. ábra)

```
function user_form()
{ $html="
<center>
<h2>A *-gal jelolt mezok kitoltese kotelezo !!! </h2></br>
<form method=post action="$_SERVER['PHP_SELF'] ." >
<table border=0>
<tr>
<td> vezetéknév:</td>
<td> <input type=text name=vez_nev size=25 value=".$this->vez_nev ." >
<font size=5>*</font> </td>
</tr>
<tr>
<td> utónév: </td>
<td> <input type=text name=uto_nev size=25 value=".$this->uto_nev ." >
<font size=5>*</font> </td>
</tr>
<tr>
<td> E-mail: </td>
<td> <input type=text name=e_mail size=25 value=".$this->e_mail ." >
<font size=5>*</font> </td>
</tr>
<tr>
<td> Jelszó : </td>
<td> <input type=password name=jelszo1 size=20><font size=3> min 6, max 12</font>
</td>
</tr>
<tr>
<td> Jelszó, újra: </td>
<td> <input type=password name=jelszo2 size=20><font size=5>*</font> </td>
</tr>
<tr>
<td> Mobil: </td>
<td> <input type=text name=mobil size=25 value=".$this->mobil ." >
<font size=5>*</font> </td>
</tr>
<tr>
<td> vezetékes : </td>
<td> <input type=text name=vezetekes size=25 value=".$this->vezetekes. " ></td>
</tr>
<tr>
<td> Iranyitoszam: </td>
<td> <input type=text name=ir_szam size=5 value=".$this->ir_szam ." >
<font size=5>*</font> </td>
</tr>
<tr>
<td> Helység: </td>
<td> <input type=text name=helyseg size=25 value=".$this->helyseg ." >
<font size=5>*</font> </td>
</tr>
<tr>
<td> Utca, házszám: </td>
<td> <input type=text name=cim size=25 value=".$this->cim ." >
<font size=5>*</font> </td>
</tr>
<tr>
<td> Hírlevél: </td>
<td> <input type=checkbox name=hirlevel> </td>
</tr>
<tr>
<td></td>
<td> <br> <input type=submit value="\Regisztráció\"> </td>
</tr></table>
</form></center>";
return $html;
}
```

7. ábra: User_form php kódja

Bejelentkezés után (8. ábra)

Sikeres bejelentkezés vagy regisztrálás után a felhasználók a 8. ábrán látható oldalra kerülnek:

A képernyő négy, jól elkülöníthető területre (sávra) osztható: Header (Fejléc), Left Body (Törzs bal oldala=Menü), Center Body (Törzsközpont), Footer (Lábléc).

1. Header(Fejléc)

A könyvtár logóját ábrázoló kép, mely az index.php-ra mutat. Alatta pedig a bejelentkezett felhasználó adatait, illetve az Online felhasználók számát mutatja.

2. Left Body (Törzs bal oldala=Menü)

A **Menü** a következő menüpontokat tartalmazza:


- **Személyes adatok**
- **Könyvtár**
- **Rólunk**
- **Adminisztráció** (ez a menüpont csak az adminisztrátori jogkörrel rendelkező személyek felületén látható)

3. Center Body (Törzsközpont)

Az éppen aktuális oldal törzse. Pl. az index.php-n mindig a három legújabb könyv adatai láthatók.

4. Footer (Lábléc).

Az oldal legfőbb pontjaira mutató linkeket tartalmazza. Segítik a tájékozódást.



Treasures from the Past

Bejelentkezve: Kesz Vagyok, **adminisztrátor** ([kijelentkezés](#))
Online felhasználók: 1

Személyes adatok

Újdonságaink :


Könyvtár

Rólunk

Adminisztráció


Varkij András: **Gyerek mondókák** (456 oldalas)

- 4566 , 100 holdas pagony kiadó
- Kategória : Gy100
- ISBN : 4646468484
- Kölcsönözhető : igen.**
- Egyéb, tartalom : Kicsi gyerekek számára

Előjegyzem : 


ROBERT MERLE: **Mesterségem a halál** (34 oldalas)

- 1239 , Irodalmi kiadó
- Kategória : Irodalom
- ISBN : I111
- Kölcsönözhető : nincs a könyvtárban!**
- Egyéb, tartalom : A koncentrációs táborok működése és kialakulása a II. Világháború alatt.

Előjegyzem : 

Maksa: **Kalkulus** (555 oldalas)

- 2000 , Unideb kiadó
- Kategória : Matematika
- ISBN : M230
- Kölcsönözhető : igen.**
- Egyéb, tartalom : Tömény gyönyör :)

Előjegyzem : 

[Kezdőlap](#) | [Regisztráció](#) | [Rólunk](#) | [Kapcsolat](#) | [Kijelentkezés](#)

2008

8. ábra: Bejelentkezés után

Nézzük a menüpontokat sorrendben részletesen is:

SZEMÉLYES ADATOK

Módosítás (9. ábra)

Lehetőségünk van rá, hogy módosításokat végezzünk a regisztráció után bármikor, ha valamely személyes adatunk megváltozik. A vezetéknév és az email cím rögzített, azokon módosítani csak az adminisztrátornak van lehetősége. A mobil és vezetékes számunkat, az irányítószámunkat és a címünket viszont bármikor megváltoztathatjuk. (az már más kérdés, hogy talán pont ez utóbbi adatok módosítása estén lehet visszaélések elkövetni egy esetleges lopáskor, de most feltesszük, hogy ez nem fordulhat elő mondjuk, ha egy kis faluban történik a könyvtári alkalmazás bevezetése)



Bejelentkezve: Kesz Vagyok, **adminisztrátor** ([kijelentkezés](#)) Online felhasználók: 1

Személyes adatok <div>Módosítás</div> <div>Jelszó csere</div> Könyvtár Rólunk Adminisztráció	ID: 5 Vezetéknév: Kesz Utónév: Vagyok E-mail: kisszati@citromail.hu Mobil: 06301234545 * Vezetékes : 06301234545 Irányítószám: 4027 * Helység: Debrecen * Utca, házszám: Egyetem * Hírlevél: Nem <div>Módosít</div>
--	---

[Kezdőlap](#) | [Regisztráció](#) | [Rólunk](#) | [Kapcsolat](#) | [Kijelentkezés](#)
2008

9. ábra: Adatmódosítás

Ha sikeresen megtörtént a módosítás, akkor a következőt fogjuk látni:

Bejelentkezve: Kesz Vagyok, **adminisztrátor** ([kijelentkezés](#)) Online felhasználók: 1

Személyes adatok <div>Módosítás</div> <div>Jelszó csere</div> Könyvtár Rólunk Adminisztráció	<p>A módosításokat végrehajtottuk.</p> <p style="text-align: center;">Vissza</p>
--	--

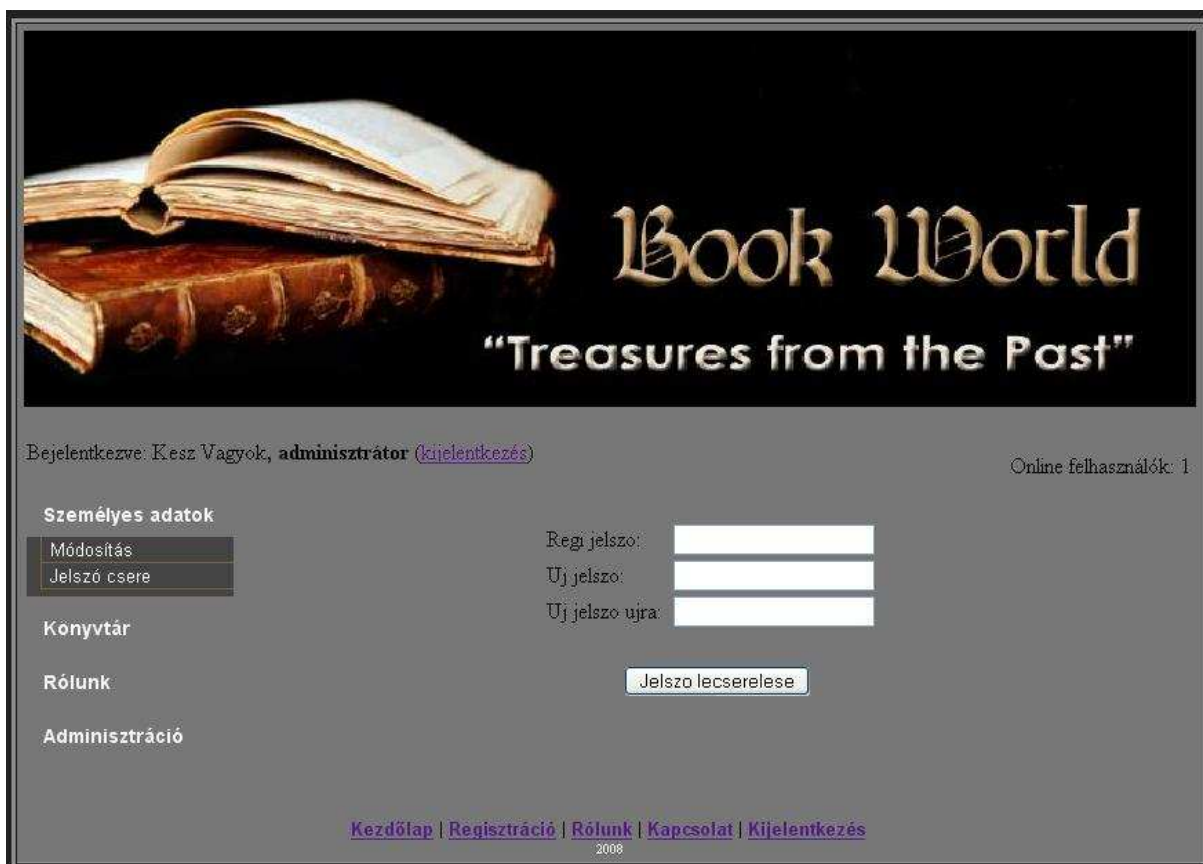
[Kezdőlap](#) | [Regisztráció](#) | [Rólunk](#) | [Kapcsolat](#) | [Kijelentkezés](#)
2008

10. ábra: Sikeres módosítás

Jelszó csere (11. ábra)

Jelszó cserére is szükség lehet néha, ezt segíti elő ez az opció.

A jelszót az elgépelés elkerülése végett kétszer kell megadni, csak abban az esetben lesz sikeres a jelszómódosítás, ha mindkét esetben azonos jelszót adunk meg. Ha a jelszó lecserélése gomb megnyomásával megerősítjük jelszó módosítási szándékunkat, akkor módosításra kerül az adatbázisban a rekord megfelelő mezője.



The screenshot shows the 'Book World' website interface. At the top, there is a banner with an image of an open book and the text 'Book World' and 'Treasures from the Past'. Below the banner, the user is logged in as 'Kesz Vagyok, adminisztrátor' with a link to 'Kijelentkezés'. The 'Online felhasználók: 1' is also shown. The main content area is divided into two columns. The left column contains a 'Személyes adatok' section with 'Módosítás' and 'Jelszó csere' buttons, and a 'Könyvtár' section with a 'Rólunk' button. The right column contains a 'Jelszó' section with three input fields for 'Regi jelszo:', 'Uj jelszo:', and 'Uj jelszo ujra:', and a 'Jelszo lecserelese' button. At the bottom, there is a footer with links for 'Kezdőlap', 'Regisztráció', 'Rólunk', 'Kapcsolat', and 'Kijelentkezés', and the year '2008'.

11. ábra: Jelszócsere

KÖNYVTÁR

Keresés (13. ábra)

A keresés fogalma bármilyen programot is írjunk, majd hogyan minden fejlesztés folyamán felmerül. Minden programozó találkozott már a keresés megvalósításának problémájával. Egy beiratkozott felhasználó általában nem találomra szeretne a könyvek között keresgélni, hanem van egy meghatározott célja, adott című és szerzőjű mű után kutat.

Keresni tudunk cím, szerző, kategória, kiadó, kiadás dátuma szerint vagy ezek bármilyen kombinációja szerint, viszont egy darab karakternek legalább szerepelnie kell valamelyik mezőben, mert különben értelmes hibaüzenettel válaszol a rendszer

Példák keresésre:

1. Nem írunk be egy karaktert sem, ekkor ezt olvashatjuk: **„Szükséges kitöltened valamelyik mezőt”**
2. 'a' karaktert beírjuk a Cím mezőbe, ekkor, mindazon könyveket kilistázza a program az adatbázisból, melyeknek a címében valahol szerepel egy 'a' karakter: Java, A kép stb. (Ha egy 'A' nagy a karaktert írunk be akkor is hasonló lesz a helyzet, mivel nem vesszük figyelembe a kis és nagy betűket a keresés folyamán)
3. Beírhatunk teljes címet is, ekkor a kis és nagybetűk nem lesznek megkülönböztetve:
vagyis, ha a cím mezőbe beírjuk a következőket: „JAVA”, java, ja, Java ugyanazt fogja eredményezni a keresés.

A következő kódrészlet, a program forráskódjából való. Nézzük meg részletesen is, hogyan lehet megvalósítani PHP-ban egy keresést (12. ábra):

```

function keres($adatok)
{
    $sql="SELECT id FROM konyv WHERE ";
    if(!empty($adatok['cim'])){
        $sql.="cim LIKE '%$adatok[cim]%' AND ";
    }
    if(!empty($adatok['szerzo'])){
        $sql.="szerzo LIKE '%$adatok[szerzo]%' AND ";
    }
    if(!empty($adatok['kategoria'])){
        $sql.="kategoria LIKE '%$adatok[kategoria]%' AND ";
    }
    if(!empty($adatok['kiado'])){
        $sql.="kiado LIKE '%$adatok[kiado]%' AND ";
    }
    if(!empty($adatok['ev']) || $adatok['ev']=='0' ){
        $sql.="ev LIKE '%$adatok[ev]%' AND";
    }

    $sql=substr($sql,0,strlen($sql)-4);
    if(strlen($sql)>24){
        $result=$this->db->query($sql);
        if($result->size()>0){
            while($sor=$result->fetch()){
                $this->book->lekerdez($sor[0]);
                $talalatok.=$this->book->megjelenit();
            }
        }else{
            $talalatok."<h4>Nincs találat!<h4>";
        }
    }else{
        $talalatok."<h4>Muszály kitöltened valamelyik mezőt!<h4>";
    }

    return $talalatok;
}

```

12. ábra: Keresés PHP-ban

A függvény hívásakor paraméterként megkapja a keresés form által post-olt adatokat (\$_POST[]) vagyis a post tömböt. Ez alapján egy sql nevű string típusú változót generálunk melynek where záradékát kitöltött mezők alapján készítjük. Aztán lefuttatjuk a lekérdezést és visszaadjuk a megfelelő hibaüzeneteket, illetve ha van találat, akkor az egyes könyvek adatait fűzzük össze egy html kóddá mely a törzsben kerül megjelenítésre.

Bejelentkezve: Hi Viki, adminisztrátor ([kijelentkezés](#)) Online felhasználók: 1

Személyes adatok

Könyvtár

- Keresés
- Előjegyzéseim
- Könyveim
- Tartozásaim

Rólunk

Adminisztráció

Cím:

Szerző:

Kategória:

Kiadó:

Kiadás dátuma:

[Kezdőlap](#) | [Regisztráció](#) | [Rólunk](#) | [Kapcsolat](#) | [Kijelentkezés](#)

2008

13. ábra: Keresés menüpont

Előjegyzéseim (15. ábra)

Előjegyezni könyvet keresés után lehet, a könyv címe után megjelenik az Előjegyzem felirat és alatta egy zöld pipa jel, ennek megnyomásával lehet. Mindezek után a „**Kéréseket előjegyeztük, ha holnap délig nem veszed át a könyvet, törlésre kerül!**” feliratot olvashatjuk.

Előjegyzéseim almenüpont alatt megnézzük a már előjegyzett könyveinket, sorban kilistázva. Mindenképpen szükség van előjegyzésre ugyanis a program elkészítése kezdetén, az volt az alapvető cél, hogy egy olyan alkalmazást fejlesszünk, ami megkönnyíti egy nagy könyvtárral rendelkező intézmény dolgát azzal, hogy az ott dolgozók csak előjegyzett könyveket adhatnak ki. Ha időközben már módosultak az igényeink, és netalán nem lenne szükségünk valamelyik előjegyzett könyvre, akkor annak törlését itt tehetjük meg.

Minden sor végén van egy jelölő mező vagy négyzet, aminek a segítségével az adott előjegyzést törölhetjük a listánkból. Ha nem pipálunk ki egy sort sem, és úgy próbáljuk az összes mezőt törölni, akkor törlődnek az előjegyzések, és a „**Nincs előjegyzés regisztrálva**” felirat lesz olvasható. Ha kipipáljuk őket szintén az előbbi események fognak megtörténni.

Ezt a menüpontot az alábbi függvény segítségével valósítottam meg:

```
function elojegyez($id_konyv)
{
    global $hibauzenet;
    $id_felhasznalo=$this->auth->session->get('user_id');
    $result=$this->db->query("SELECT id FROM konyv WHERE id=$id_konyv AND darabszam>0 ");
    if($result->size()==1){
        $result1=$this->db->query("UPDATE konyv SET darabszam=darabszam-1 WHERE id=$id_konyv");
        $datum=date("Y-m-d H:i:s");
        $result=$this->db->query("INSERT INTO elojegyzesek (id_felhasznalo, id_konyv, datum)
                                VALUES ('$id_felhasznalo', '$id_konyv', '$datum')");
        if($result!=false && $result1!=false){
            $hibauzenet="<h4>kéréseket előjegyeztük,
                        ha holnap délig nem veszed át a könyvet, törlésre kerül !<h4>";
        }else{
            $hibauzenet="<h4>kéréseket nem sikerül regisztrálnunk, lehet már elfogyott !<h4>";
        }
    }else{
        $hibauzenet="<h4>kéréseket nem sikerül regisztrálnunk, lehet már elfogyott !<h4>";
    }
}
```

14. ábra: Előjegyzés függvény forráskódja PHP-ban,




15. ábra: Előjegyzéseim

Könyveim (16. ábra)

A „Nincs kikölcsönzött könyve nálunk” feliratot olvashatjuk, ha nincs folyamatban lévő kölcsönzésünk, ellenkező esetben pedig a könyveket láthatjuk kilistázva.

Soronként megjelenik a szerző, cím, az érvényesség, amely megmutatja nekünk dátum szerint, hogy melyik naptári napon jár le a kölcsönzés, illetve a lejárat napokban mérve, továbbá még egy mező, a Késedelmi díj, mely forintban mutatja a díj összegét.



Bejelentkezve: Kész Vagyok, **adminisztrátor** ([kijelentkezés](#)) Online felhasználók: 1

Személyes adatok

Könyvtár

- Keresés
- Előjegyzéseim
- Könyveim
- Tartozásaim

Rólunk

Adminisztráció

Szerző	Cím	Érvényesség	Lejárat	Késedelmi díj
Fazekas István	Valószínűségi számítás	2008 November 24. 13:47:3	még 30 nap	0 Ft
Pici	Java 1	2008 November 24. 13:47:3	még 30 nap	0 Ft

[Kezdőlap](#) | [Regisztráció](#) | [Rólunk](#) | [Kapcsolat](#) | [Kijelentkezés](#)
2008

16. ábra: Könyveim

Tartozásaim (17. ábra)

Tartozás megállapításakor, minden könyvnél a könyv árának 1%-a tartozási díj van megállapítva. Ha valaki túllépi az egy hónapos kölcsönzési határidőt, annak könyvenként kell az adott könyv árának 1%-át kifizetnie szorozva késedelmi napok számával.



Bejelentkezve: Kész Vagyok, **adminisztrátor** ([kijelentkezés](#)) Online felhasználók: 1

Személyes adatok

Könyvtár

- Keresés
- Előjegyzéseim
- Könyveim
- Tartozásaim

Rólunk

Adminisztráció

Szerző	Cím	Késedelmi díj
Fazekas István	Valószínűségi számítás	0 Ft
Pici	Java 1	0 Ft
Ossztartozásod :		0 Ft

[Kezdőlap](#) | [Regisztráció](#) | [Rólunk](#) | [Kapcsolat](#) | [Kijelentkezés](#)
2008

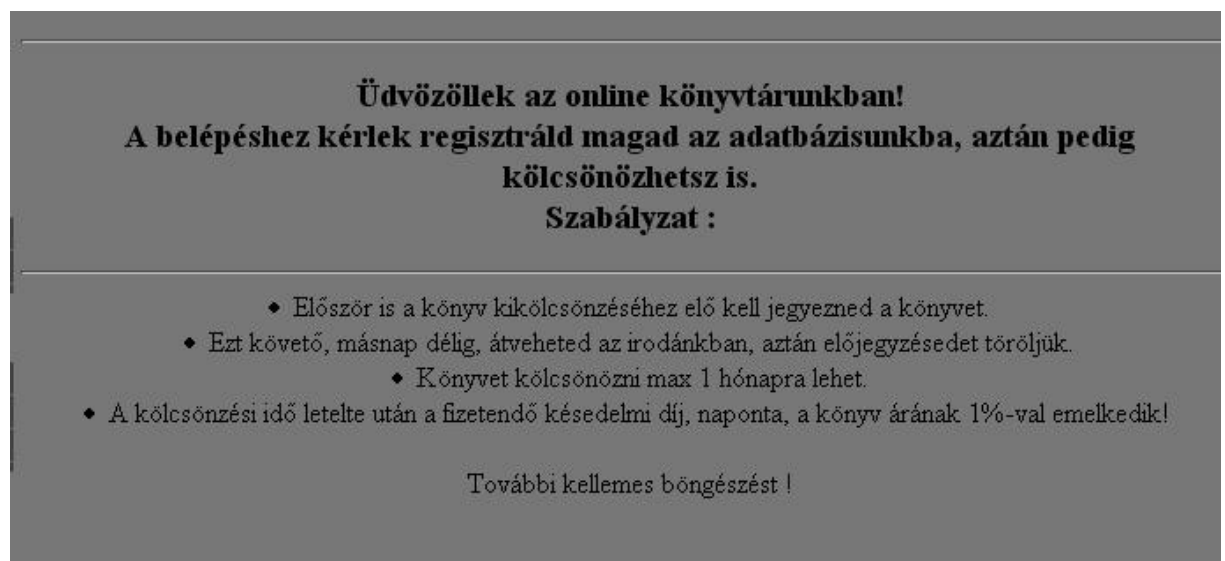
17. ábra: Tartozásaim

RÓLUNK

Bemutakozás (18. ábra)

Ide először is egy üdvözlő szöveget helyeztem el, annak, aki regisztrált és belépett a könyvtárunkba.

Itt lehet olvasni a könyvtár szabályzatát, valamint a belépéssel kölcsönzéssel és az aktualitásokkal foglalkozó információkat. Minden könyvtárnak megvan ugyanis a saját működési is irányítási szabályzata, amivel minden beiratkozott felhasználónak tisztában kell lennie. A jelen esetben a beiratkozás egy éves időtartamot jelent, továbbá egy könyv kikölcsönzése azt jelenti, hogy a könyvet maximum 30 napig tarthatja magánál valaki.



Üdvözöllek az online könyvtárunkban!
A belépéshez kérlek regisztráld magad az adatbázisunkba, aztán pedig kölcsönözhetsz is.
Szabályzat :

- ♦ Először is a könyv kikölcsönzéséhez elő kell jegyezned a könyvet.
- ♦ Ezt követő, másnap délig, átveheted az irodánkban, aztán előjegyzésedet töröljük.
 - ♦ Könyvet kölcsönözni max 1 hónapra lehet.
- ♦ A kölcsönzési idő letelte után a fizetendő késedelmi díj, naponta, a könyv árának 1%-val emelkedik!

További kellemes böngészést !

18. ábra: Bemutakozás

Kapcsolat (19. ábra)

Ha bárkinek valamilyen problémája van a könyvtárral, a könyvtári szolgáltatással, az alkalmazottakkal, vagy akár a szoftverrel, azoknak szolgál hasznos segítséggel ez a menüpont. Lehet benne akár névtelenül is írni, ugyanis a rendszer nem ellenőrzi, hogy beiratkozott felhasználó küldi-e az üzenetet. Vagyis ha valaki fiktív nevet ad meg, a rendszer akkor is elküldi az üzenetet.

Bejelentkezve: Kész Vagyok, **adminisztrátor** ([kijelentkezés](#)) Online felhasználók: 1

Személyes adatok

Könyvtár

Rólunk

- Bemutakozás
- Kapcsolat

Adminisztráció

- Új könyv
- Könyvek
- Felhasználók

Elérhetőségeim :

E-mail: **valami@valami.com**
 Mobil: **06-70-000-0000**
 Cím: **Debrecen, Egyetem tér 1.**

Azonnali üzenet küldése :

Feladó*:
 Tárgy*:
 Üzenet*:

[Kezdőlap](#) | [Regisztráció](#) | [Rólunk](#) | [Kapcsolat](#) | [Kijelentkezés](#)
 2008

19. ábra: Kapcsolat

ADMINISZTRÁCIÓ

Adminisztrátori jogkörrel rendelkező felhasználók, új felhasználókat vehetnek fel, új könyvekkel bővíthetik a könyvtárat, megváltoztathatják a felhasználók és a könyvek adatát. A felhasználók előregisztrált könyveit csakis az ilyen jogkörrel ellátott egyén hagyhatja jóvá vagy törölheti, illetve a tartozásokat az ő felületén keresztül tudja kiegyenlíteni minden felhasználó.

Új könyv (20. ábra)

A könyvtári állomány bővítését az új könyv menüpont segítségével végezhetjük el. Sikeres regisztráció esetén az új könyv bekerül a könyvtáblába az adatbázisban.

A fent elmondottakhoz, az adminisztrátornak egy beviteli űrlapot kell kitöltenie (vagy más néven form-ot, hasonlót ahhoz, amit már a felhasználó regisztráció menüpont alatt láthattunk) (20. ábra)

"Treasures from the Past"

Bejelentkezve: Kész Vagyok, [adminisztrátor](#) ([kijelentkezés](#))

Online felhasználók: 1

Személyes adatok

Könyvtár

Rólunk

Adminisztráció

Új könyv
Könyvek
Felhasználók

A *-gal jelölt mezők kitöltése kötelező !!!

Cím : *

Szerző : *

Kiadó : *

Kategória : *

Kiadás éve : *

ISBN : *

Oldalszám :

Egyéb,
leírás,
tartalom :

Darabszám : *

Ár : *

20. ábra: Új könyv felvétele űrlap

Mint láthatjuk a szokásos módon, a csillaggal jelölt mezők kötelező jellegűek, majd a Regisztráció gombra kattintva lehet jóváhagyni a regisztrációt. Ekkor a „**Sikeres regisztráció!**” felirat lesz olvasható.

Egyébként pedig, ha kihagytuk a cím mezőt, akkor: „**Nincs kitöltve a Cím mező**” felirat, és értelemszerűen, amelyik mező kitöltése kimaradt, annak neve lesz olvasható a hibaüzenetben. Ha mindent jól kitöltöttünk, akkor a regisztráció gombra kattintva, látható lesz a 22. ábrán levő kép. Itt kell jóváhagyni vagy a Javít gombra kattintva módosítani.

Bejelentkezve: Kész Vagyok, [adminisztrátor](#) [\(kijelentkezés\)](#) Online felhasználók: 1

Személyes adatok

Könyvtár

Rólunk

Adminisztráció

- Új könyv
- Könyvek
- Felhasználók

A *-gal jelölt mezők kitöltése kötelező !!!

Cím : *

Szerző : *

Kiadó : *

Kategória : *

Kiadás éve : *

ISBN : *

Oldalszám :

Egyéb, leírás, tartalom :

Darabszám : *

Ár : *

21. ábra: Könyv regisztrálása

Bejelentkezve: Kész Vagyok, [adminisztrátor](#) [\(kijelentkezés\)](#) Online felhasználók: 1

Személyes adatok

Könyvtár

Rólunk

Adminisztráció

- Új könyv
- Könyvek
- Felhasználók

A következő adatok kerülnek rögzítésre :

Cím :	PHP
Szerző :	Ian Thorpe
Kiadó :	Informatika
Kategória :	Informatika
Kiadás éve :	2008
ISBN azonosító :	11236
Oldalszám :	506
Egyéb, leírás, tartalom :	
Darabszám :	55
Ár :	7000

22. ábra: Könyv regisztráció megerősítése

Könyvek (23. ábra)

Az adminisztrátori jogkörrel rendelkező egyén egyik fő feladata a könyvtári állomány bővítése, új könyvvel. Mint tudjuk, egy könyvtárban nem tíz vagy húsz, hanem több száz, több ezer, sőt egy nagyobb könyvtárban több tízezer könyv is található. Ezért fontos az, hogy egy teljesen új, vagy egy olyan könyv kerül-e a könyvtárba melyből már néhány példány megtalálható.

Például hoznak „Java 1.” című könyvet a könyvtárunkba. Ekkor az adminisztrátor:

1. Rákeres az adott könyvre.(24. ábra) (elég egy db. 'j' karaktert beírni a cím ablakba, persze ilyenkor fennáll annak a veszélye, hogy nagyon sok rekordot kidob a kereső, ezért nagy könyvtárak esetén célszerű minél pontosabban megadni a keresett könyvet, de ettől itt most eltekintünk)



BOOK WORLD

"Treasures from the Past"

Bejelentkezve: Kesz Vagyok, adminisztrátor ([kijelentkezés](#))

Online felhasználók: 1

Személyes adatok

Könyvtár

Rólunk

Adminisztráció

- Új könyv
- Könyvek
- Felhasználók

Szerző:

Cím:

Keresés

[Kezdőlap](#) | [Regisztráció](#) | [Rólunk](#) | [Kapcsolat](#) | [Kijelentkezés](#)

2008

23. ábra: Könyvek almenüpont

2. Ha van ilyen rekord, akkor a Módosítást jelölő „ceruzára” kattintva, átírjuk a darabszámot. Aztán pedig, a módosít gombbal jóváhagyjuk.
3. különben ha nincs ilyen könyv. akkor könnyű dolgunk van, mert akkor az „Új könyv”- re kattintva regisztrálunk egy új könyvet.



Treasures from the Past

Bejelentkezve: Kesz Vagyok, [adminisztrátor](#) ([Kijelentkezés](#)) Online felhasználók: 1

Személyes adatok

Könyvtár

Rólunk

Adminisztráció

- Új könyv
- Könyvek
- Felhasználók

Szerző:

Cím:

ID	Szerző	Cím	Kategória	Módosítás
4	Pici	Java 1	Info	

[Kezdőlap](#) | [Regisztráció](#) | [Rólunk](#) | [Kapcsolat](#) | [Kijelentkezés](#)
2008

24. ábra: Példa keresésre

[Vissza](#)

Személyes adatok

Könyvtár

Rólunk

Adminisztráció

- Új könyv
- Könyvek
- Felhasználók

ID: 4

Cím: *

Szerző: *

Kiadó: *

Kategória: *

Kiadás éve: *

ISBN: *

Oldalszám:

Egyéb, leírás, tartalom:

Darabszám: *

Ár: *

Kölcsönzői:

ID	Név	E-mail	Szerkesztés
5	Kesz Vagyok	kisszati@citromail.hu	

25. ábra: Könyv adatainak módosítása

Felhasználók (26. ábra)

A felhasználók adatainak, kölcsönzéseinek, tartozásainak teljes körű menedzselését is az adminisztrátor végzi a program segítségével. Ezen menüpont alatt minden ehhez szükséges eszköz egy oldalon rendelkezésre áll. Egy felhasználóra történő keresés után, a könyveknél látott módon van lehetőség a felhasználó adatainak illetve kölcsönzéseinek a módosítására. (a szerkesztés alatti „ceruza” ikonra kattintva)

Adminisztráció
Új könyv
Könyvek
Felhasználók

ID: 5
Vezetéknév: Kesz
Utónév: Vagyok
E-mail: kisszati@citromail.hu
Mobil: 06301234545 *
Vezetékes : 06301234545
Iranyitoszam: 4027 *
Helység: Debrecen *
Utca, házszám: Egyetem *
Hírlevél: Nem

Módosít Töröl

Előjegyzései :

ID	Szerző	Cím	Kategória	Jóváhagyás
95	Fazekas István	Valószínűségyszámítás	Matematika	<input type="checkbox"/>

Jóváhagy Összes jóváhagyása

Könyvei :

Szerző	Cím	Érvényesség	Lejárat	Késedelmi díj	Visszavesz
Pici	Java 1	2008 November 9. 20:39:9	még 29 nap	0 Ft	<input type="checkbox"/>
Maksa	Kalkulus	2008 November 9. 20:39:9	még 29 nap	0 Ft	<input type="checkbox"/>

Visszavesz

26. ábra: Egy felhasználó az adminisztrátor szemszögéből

Lássunk egy példát a jobb megérthetőség kedvéért:

A Kész Valaki nevű felhasználó szerdai napon előjegyzi otthonából a Fazekas István: Valószínűség számítás című könyvét. Másnap bemegy a könyvtárba, azzal a céllal, hogy ki szeretné kölcsönözni. Az adminisztrátornak (könyvtárosnak) megmondja a nevét. Ő rákeres a Kész Valaki nevű felhasználóra. (26. ábra) Eztán a Kész Valaki sor végén van egy olyan opció, hogy szerkesztés, ez alatt a ceruza ikonra kattintva rögtön az ábrán látható kép tárul az adminisztrátor szeme elé.

Itt sorrendben a következőket teheti:

- Módosíthatja egy felhasználó adatait.
- Jóváhagyhat előjegyzést, vagyis eztán a könyv ténylegesen is kikölcsönzésre kerül
- Könyvet vehet vissza.

A megfelelő jogosultságok kezelése:

Minden egyes oldal létrejötte előtt példányosítanunk kell a Page osztályt, melynek konstruktorában meghívódik a jogosult() metódus, mely egy logikai értékkel tér vissza attól függően, hogy a bejelentkezett felhasználó szerepköre kielégíti-e az oldal létrejöttekor definiált jogosultságot. Ha kielégíti, akkor megjelenik az oldal, különben pedig az alábbi Error page:



ERROR

Az oldal megtekintéséhez előbb be kell jelentkezned !

E-mail:

Jelszó:

[Elfelejtetted a jelszavad ?](#)

[Regisztrálok !](#)

Biztonsági kiléptetéskor szintén a fenti hibakód látható. Mivel a rendszer biztonsági okok miatt egy előre definiált időkorlát letelte után kijelentkezteti a felhasználót, tehát további műveletekhez ismét be kell jelentkeznie.

Összegzés

Egyetemi éveim alatt, ahogy egyre több tapasztalatot szereztem programozás terén, rájöttem arra, hogy egy programot mindig lehet tovább bővíteni, új funkcionalitást bevinni mindig maradnak benne továbbfejlesztési lehetőségek.

Egy bővebb könyvtár manapság már tartalmaz külön olvasótermet, ahol a látogatók a szabad polcokról levett könyveket benn a könyvtár falai között olvashatják el. Tartozhat levéltár is a könyvtárhoz. A levéltárba külön engedély szükséges az oda belépő usereknek, lehet külön belépési díjat kiróni a könyvtári olvasóknak illetve a levéltári olvasóknak, sőt olvasó és olvasó között is különbséget tehet a rendszer, gondolok itt a tanár, diák, felnőtt belépőkre.

Le lehet védeni könyveket, vagyis lesznek olyan könyvek, amit nem lehet kölcsönözni, csak a könyvtárban olvasni, vagy még esetleg azt sem.

Itt jegyezném meg, hogy mivel nem nagyon kiterjedt és nagy funkcionalitással bíró alkalmazásról van szó, emiatt nem tartottam fontosnak SÚGÓ menüpont beiktatását, viszont célszerű, és nagy alkalmazások készítése esetén akár kötelezőnek is mondható ennek megléte.

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani témavezetőmnek, aki segítségemre volt tanácsaival és szakszerűen útbaigazított mindig, amikor arra szükségem volt.

Köszönöm az informatika kar oktatóinak, hogy az egyetemi éveim alatt próbáltak minél több tudást átadni a számomra.

Továbbá köszönettel tartozom szerető családomnak, akik minden percen melletttem álltak és támogattak mind az egyetemi éveim alatt, mind azt megelőzően.

Irodalomjegyzék:

- **Peter Moulding (2002):** PHP Haladóknak - Fekete Könyv.
Perfact-Pro Kft. Budapest.
- **Nagy Gusztáv (2008):** Web programozás.
Kiskapu Kiadó. Budapest
- **George Schlossnagle (2004):** PHP fejlesztés felsőfokon.
Kiskapu Kiadó. Budapest.
- **G. Reese, R. J. Yarger, T. King (2003):** A MySQL kezelése és használata.
Kossuth Kiadó. Budapest.
- **Nagy Gusztáv (2004):** A PHP, a MySQL és az Apache használata.
Panem Kiadó, Budapest
- <http://hu.wikipedia.org/wiki>
- <http://prog.hu>
- <http://www.php.net>
- <http://www.mysql.com>