

DEBRECENI EGYETEM

Informatikai Kar

**WEB ALAPÚ FOGLALÁSI RENDSZER MEGVALÓSÍTÁSA INGYENES  
SZOFTVEREKKEL**

Témavezető:

Pánovics János

Egyetemi tanársegéd

Készítette:

Kósa Anita

Programtervező matematikus

DEBRECEN

2008

## Tartalomjegyzék

Bevezetés.....	3
WEB történet.....	4
A távoli adatkommunikáció kialakulása.....	4
Azonosítás az Interneten.....	5
A WWW születése.....	5
Webszerverek (APACHE).....	6
A webszerver definíciója, feladata.....	6
Az APACHE.....	6
A HTTP protokoll.....	9
Bevezetés a HTML-be.....	12
A HTML definíciója, funkciója.....	12
Stíluslapok.....	14
Bevezetés a CSS-be.....	14
Dinamikus tartalomgenerálás.....	16
A PHP története.....	16
A PHP és a C.....	19
Adatbázis technológiák (MySQL).....	21
Relációs adatbázisok kezelése.....	21
Adatbázis tervezés.....	23
A program részletes bemutatása.....	26
Összefoglalás.....	35
Függelék.....	37

## 1. Bevezetés

A web az internet-felhasználók többségének mindennapjaiban jelen van. Célom az volt, hogy diplomamunkám keretében az informatika egy aktuális, dinamikusan fejlődő területével, a web alapú alkalmazásfejlesztési technológiákkal és technikákkal foglalkozzak.

Diplomamunkám során a web fejlesztés témakörén belül, nagyon elterjedt, kiforrott és ingyenes szoftverek segítségével valósítok meg egy web alapú foglalási rendszert, amely egy hotel szobafoglalást (bejelentkezést/bejelentkeztetést) valósít meg.

A program elkészítéséhez a PHP, a MySQL és az Apache szoftverek legújabb generációját használtam. Azért esett választásom rájuk, mert segítségével adatbázisból dinamikusan lehet előállítani tartalmat és felhasználói irányból is képesek adatokat fogadni, valamint feldolgozni. A tárgyalt technológiák mára fejlesztési alapnak számítanak a web világában, egyre több alkalmazástípus számára fejlesztenek ki interaktív webes felületet. A MySQL az 5-ös verzió megjelenésével vállalati szintű, professzionális adatbázis kezelővé vált. (pl.: a Google hatalmas méretű adatbázisát is MySQL szolgálja ki)

### *Mi a széles körű elterjedés oka?*

Bármiféle kliens oldali szoftver telepítése nélkül, csupán egy böngésző és internetkapcsolat felhasználásával a világ bármely pontjáról elérhető az így kifejlesztett alkalmazás. Sok olyan előnyt tapasztaltam meg én is az elkészítés során, melyek valóban alátámasztják elterjedésüket. Kitérek ezekre az előnyökre is a későbbiekben.

### *Miért választottam web alapú foglalási rendszert?*

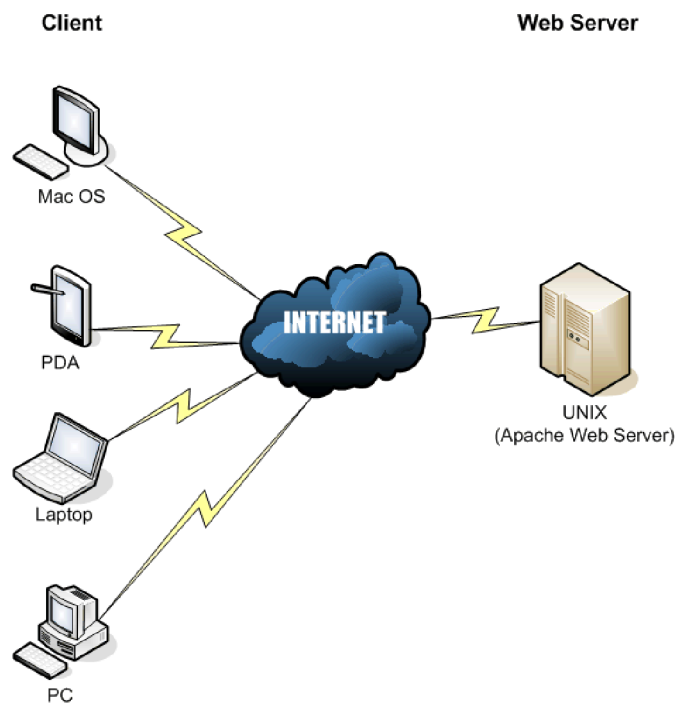
Azért, mert mindhárom technológia képességeit magában hordozva, kézzel fogható módon demonstrálja a korszerű webes technológiákban rejlő lehetőségeket:

1. felhasználóbarát webes felület
2. interaktivitás – kétirányú kommunikáció
3. adatbázis alapú adatfeldolgozás és tárolás
4. szabványos megjelenítési felület, ahol elválik egymástól a tartalom és annak megjelenési módja: XHTML, CSS

## 2. WEB történet

### 2.1. A távoli adatkommunikáció kialakulása

Nagy számú számítógép és egyéb eszközök egy közös kommunikációs hálózatba való összekapcsolását nevezzük internetnek. Az internetre kapcsolt eszközök között a kommunikációt a TCP/IP (Transmission Control Protocol / Internet Protocol) protokoll készlet teszi lehetővé. 1969-ben az Amerikai Egyesült Államok Védelmi Minisztérium ARPA (Advanced Research Projects Agency) hivatala hozta létre a TCP/IP protokollt, ekkor még csak kísérleti céllal. 1975-ben szabványosították és ettől kezdve beszélhetünk kettő illetve több gépen futó program interneten keresztül történő kommunikációjáról, mely során az operációs rendszer egy alacsony szintű interfészt biztosít, ahhoz, hogy a csatlakoztatott gépek közös jelkészlettel kommunikáljanak. Helyi hálózatokat szokás kialakítani intézményeken belül, melynek általában csak egyetlen csomópontja csatlakozik fizikailag az internethez.



1. ábra: Kliens/szerver architektúra az interneten

## **2.2. Azonosítás az interneten**

Az internethez kapcsolódó számítógépeket, egy egyedi 32 bites IP címmel azonosítják, melyet ponttal elválasztott 4 darab 8 bites számként (oktet) ábrázoljuk. A gépek ezen kívül még szöveges nevet is kapnak azonosításuk megkönnyítésére, ezt nevezzük tartománynévnek (FQDN – Teljes minősített domain név, pl.: www.unideb.hu.). Egy adott hálózati csomóponthoz kettő vagy több tartománynév is rendelhető egyszerre. Az FQDN-en belül balról az első tag mindig a gazdagép neve (pl.: www, mail, delfin, stb.), amit az egyre bővebb tartományok, számítógépcsoportok, angolul domain-ek követnek. Az utolsó tartomány megadja, hogy a gazdagép mely felsőszintű szervezethez tartozik és egyben ez a legnagyobb tartománynév is. (Top Level Domained, pl.: hu, com, org, eu, stb.)

## **2.3. A WWW születése**

1989-ben Tim Berners-Lee vezetésével a CERN-nél (Conseil European pour la Recherche Nucleaire, Európai Részecskefizikai Laboratórium) egy új technológiát terjesztettek elő az interneten történő információcseréhez, melyet World Wide Webnek neveztek el. A cél az volt, hogy egy olyan új rendszert tervezzenek, mely segítségével bárki a világ bármely pontjáról dokumentumokat, adatokat tudjon letölteni illetve keresni számos dokumentumszolgáltató adatbázisából.

1990 végére teljesen kifejlesztették és megvalósították az ötleteket egy NeXT típusú számítógépen. 1991-ben már más platformokon is jól működött a technológia így elérhetővé tették az egész világ számára. A web valójában protokollok, technológiák és szoftverek összességét jelenti, amelyek megtalálhatóak a legtöbb internetre csatlakoztatott számítógépen.

A web szolgáltatás kliens/szerver konfigurációban működik. A kommunikációt a kliens kezdeményezi, kérést indít a szerveren tárolt valamilyen információ megszerzése érdekében, melyre a szerver válasz formájában eljuttatja a klienshez a kért információt. A szerver által adott választ a számítógépen, a kliensgépen található böngészők jelenítik meg. Ma már számos grafikus böngésző közül válogathatunk, a leggyakrabban használtak a Microsoft Internet Explorer, Netscape vagy a Mozilla Firefox.

## 3. Webszerverek (APACHE)

### 3.1. A webszerver definíciója, feladata

A webszerverek olyan programok, amelyek egy kliens oldali böngésző irányából érkező kérésre valamilyen információt, dokumentumot juttatnak vissza. A böngészők általában már meglévő dokumentumokat kérnek le, de előfordulhat az is, hogy valamilyen programot szeretnének futtatni a szerveren, az eredményt pedig egy dinamikusan generált tartalom formájában várják vissza. A mai böngészők már nem csak ennyire képesek, számon tartják, hogy milyen dokumentumokat is kértünk le a közelmúltban és ha egymás után többször is ugyanarra lennének kíváncsiak, akkor nem szükséges a szervertől ismételt lekérdezni az információt.

Lényegében a webszerverek az információszolgáltatók, mi akik a webet használjuk, pedig az információ fogyasztói vagyunk.

A böngészők és a szerverek közötti hálózati kommunikációt az URL-ek segítségével érhetjük el. Az URL általános formátuma:

*séma : objektumcím*

A *séma* a kommunikációs protokoll nevét tartalmazza, a mi számunkra a HTTP protokoll lesz fontos a web támogatottsága miatt. Az *objektumcím* a távoli webszerver internetes címét, illetve a lekért dokumentum szerveren belüli helyét azonosítja, szorosan a sémához kötődik. A különböző sémák különböző objektumcímeket használnak. Alapértelmezettként a webszerverrel a 80-as TCP porton keresztül lehet kommunikálni, a webszerver konfigurációs állományában más port is megadható, ezt azonban azt az URL-ben a gazdanév után : *portszám* formában kell megadni. Vannak olyan karakterek, melyeket az URL nem tartalmazhat pl. szóköz, kettőspont, pontosvessző.

Ha a használatuk elkerülhetetlen, akkor a % jel után a karakter kétjegyű hexadecimális kódjának megadásával tehetjük meg.

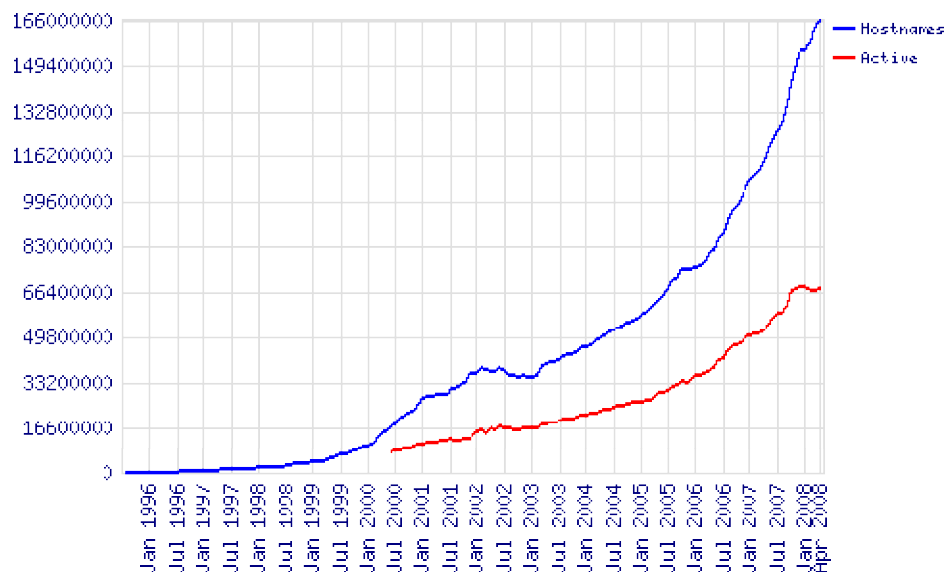
### 3.2. Az APACHE

Webszerverek közül az egyik leggyakrabban alkalmazott szerverszoftver az APACHE. Legnagyobb erőnei többek közt, hogy a legtöbb számítógéplatformra ki lett fejlesztve, ingyenes, nyílt forráskódú és moduláris felépítésű.

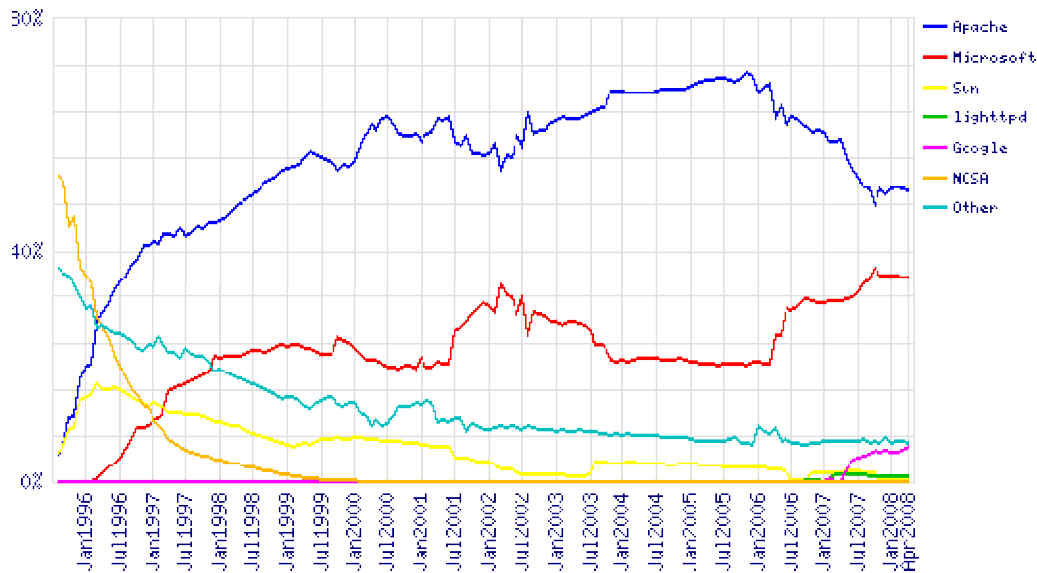
Nevét a keletkezése után kapta, ugyanis kezdetben nem volt több, mint a httpd nevű NCSA (National Center for Supercomputer Applications ) szerver kiegészítése néhány új eszközzel. 1994-ben Rob McCool a szerver írója távozott az NCSA-tól, így egy ideig a fejlesztése megállt. A webfejlesztők továbbra is tartották egymással levélben a kapcsolatot és rendszeresen küldték egymásnak a javított, finomított kódrészleteket (patcheket). 1995. februárjában nyolc fővel pedig elindult az akkor már célirányos fejlesztés. A szoftver akkori állapotában lényegében egy foltozott (patchy) verzió volt.

Az Apache-ot webes alkalmazások kiszolgálása mellett azok készítése és fejlesztése esetén is előszeretettel használják Linux vagy Windows platformokon, MySQL alapú adatbázissal, PHP vagy PERL szerveroldali programozási nyelvekkel integráltan. Az Apache jelenleg rendkívül széleskörű beépített eszközrendszerrel rendelkezik, melyeket akár saját magunk által készített modulok segítségével is bővíthetünk. Talán ezért is vált annyira közkedvelt szerverré, alkalmazásával szinte minden webes igény, követelmény kielégíthető. Ma már annyira számottevő a modularitás, hogy apróbb feladatokat kivéve, szinte már minden lényeges feladatot előre megírt, jól bevált modulok segítségével végez. Így az Apache-ban rejlő moduláris technológiából más kommunikációs protokollok (pl. SMTP) is profitáltak.

Az alábbi két diagram a Netcraft internetszolgáltató vállalat felmérését mutatja az interneten alkalmazott Apache webserverek számát tekintve:



2.ábra: Apache webservert alkalmazó internetes oldalak számának alakulása 1995 és 2008 között



3. ábra: A legelterjedtebb webserverek piaci megoszlás szerint 1995 és 2008 között

#### Hasznos modul funkciók:

- Virtuális-szerver támogatás: egy webservert és egy számítógép segítségével több
- webhely kiszolgálása valósul meg. A felhasználói oldalon úgy tűnik, hogy több egymástól független szerverről van szó. Megtévesztő lehet az is, hogy különböző név és IP-cím adható a virtuális szervereknek. Szerver oldalon megkönnyíti a webhelyek adminisztrálását, karbantartását.
- URL rugalmas átírása (átírányítás – rewrite modul)
- Hibakezelés: A hibákat könnyen felismeri, értelmes, felhasználóbarát hibaüzeneteket ad vissza.
- Szerver oldali scripting nyelv modulszintű integrálhatósága (pl.: PHP)
- Sokrétű tartalom megjelenítés: Képes felismerni a böngésző adottságait és ennek megfelelően megjeleníteni a kért oldalt, akár keretek nélküli, egyszerűbb változatban, ha a böngészőnk szöveges módú lenne.

Az általam fejlesztett webes rendszerhez kapcsolódó Apache webservert konfigurációs állományából az 1. táblázatban részletezem pár konfigurációs beállítás funkcióját:

<p>ServerRoot: A könyvtár szerkezet legfelső szintje melyben a webserverek konfigurációs és naplóállományai találhatóak:</p> <pre>ServerRoot "C:/Program Files/Apache Software Foundation/Apache2.2"</pre> <p>Listen: Lehetővé teszi, hogy az Apache-ot egy adott IP címhez és/vagy porthoz rendeljük.</p> <pre>Listen 127.0.0.1:80 (localhost, 80-as TCP port)</pre> <p>DSO: A Dinamikusan Megosztott Objektumok lehetővé teszik külső modulok használatát anélkül, hogy a alapszoftver kódját újra kellene fordítani. Pl.: a PHP integrálása az Apache szerverbe:</p> <pre>LoadModule php5_module "c:\\Program Files\\PHP\\php5apache2_2.dll"</pre> <p>DocumentRoot: A dokumentumgyökér az az alapkönyvtár, melyből a webes dokumentumok kiszolgálásra kerülnek. Alapértelmezésben minden kérés ebből a könyvtárból kérdezi le az adott dokumentumot.</p> <pre>DocumentRoot "C:/Program Files/Apache Software Foundation/Apache2.2/htdocs"</pre>
--

1. táblázat: Az Apache szerver alapvető konfigurációs paraméterei

### 3.3. A HTTP protokoll

A web technológia kliens/szerver architektúrát alkalmaz, ahol a két vagy több gép között a TCP alapú HTTP (HyperText Transfer Protocol) protokoll biztosítja az információcserét. Kialakításakor szempont volt a gyorsaság és az egyszerűség, melyek ma már elengedhetetlen feltételei egy jól működő infokommunikációs rendszernek.

A kliens egyszerre több kérdést is küldhet a kiszolgálónak, ami minden kérdést külön, egymástól függetlenül kezel és a kért dokumentum továbbítása után lezárja a kapcsolatot. Így mindenki számára közel egyforma kiszolgálási gyorsaság biztosítható.

A HTTP kommunikáció 4 lépésben megy végbe:

1. Kapcsolat megnyitása.
2. Kérés elküldése.
3. Válasz megadása.
4. Kapcsolat lezárása.

A sikeres kapcsolat eredményeképpen a kért dokumentum eljut a klienshez. A kliens egy konkrét weboldal lekérésekor annak minden egyes eleméhez külön kapcsolatot épít ki a webservert irányába.

A kommunikáció két fázisa: a kérés és a válasz. Ezek további két részből tevődnek össze: a fejlécből és a törzsből. A fejléc a kommunikációról, a törzs pedig azon adatokról szolgáltat információt melyekre a kommunikáció irányul.

A HTTP-kérés:

Általános alakja:

*HTTP metódus*    *URL-tartomány része*    *HTTP-verzió*

*Fejlécmezők*

*Üres sor*

*Üzenettörzs*

```
GET /index.html /HTTP-1.1
```

1. példa: GET metódus szintaktikája

A leggyakrabban használt HTTP-kérések:

GET	Visszaadja az aktuális dokumentum tartalmát.
POST	A megadott adatokkal kitölti az aktuális dokumentumot.
HEAD	Fejléc információk megadása.
PUT	Kicseréli a feltüntetett adatokra a megadott dokumentumot.
DELETE	Dokumentum törlése.

2. táblázat: HTTP - kérések

Web alkalmazások fejlesztése során a leggyakrabban az első kettő, azaz a GET és a POST metódusokat használjuk.

GET

A HTTP alapú kliens/szerver kommunikáció során az elérni kívánt dokumentumokat a GET metódus segítségével juttatjuk el a webservert alkalmazáshoz. Az általa lekért adatok

lehetnek statikus vagy valós időben előállított dinamikus tartalmú dokumentumok. A tartalomgenerálás módja a kliens elől rejtve marad, mivel a webszerveren dinamikusan előállított tartalmak is már mint statikus oldalak jelennek meg a kliens oldali böngészőben.

Ebből számos előny származik, egyrészt biztonsági szempontból előnyös, hogy a szerveren futó kódok a kliens oldalán nem jelennek meg, így azok rosszindulatú manipulálására sincs lehetőség közvetlenül.

Másrészt kliens oldalon nem kell foglalkozni a kódot futtató környezettel, kizárólag a következő fejezetben ismertetésre kerülő HTML leírónyelv értelmezésére kell felkészíteni a böngészőt.

## POST

A dinamikusan generált web tartalmak megjelenésével a HTTP POST metódus alkalmazása előtérbe került. Mivel a felhasználó irányából a szerver felé történő információtovábbítás a korszerű web alapú rendszereken alapvető igényként jelenik meg. Ezek az információk (mint továbbítandó paraméterek) többnyire bizalmas jellegűek ezért a kommunikáció során a kezelésük speciális metódust igényel. Erre a feladatra a POST metódus hatékonyan alkalmazható, mert a kliens által továbbított információ (ellentétben a GET metódus által alkalmazott URL-en keresztül történő paraméterátadással) rejtett módon kerül továbbításra a szerverhez.

Alkalmazási példa POST metódusra: tetszőleges beléptető oldalon megadott jelszó továbbítása a szerver irányába. Webes vásárláskor bankkártya információk megadása.

A HTTP-válasz:

Általános alakja:

*Állapotsor*

*Válasz fejléczmezők*

*Üres sor*

*Választörzs*

A web technológiát megközelíthetjük a felhasználó szemszögéből is. A kliens oldalán mindössze magára a számítógépre, internet kapcsolatra és egy kliensoldali szoftverre van szüksége. A kliensoldalon a hálózati kapcsolat meglétéén túl a web technológia megjelenítésére

alkalmas szoftverkörnyezet alapját képező böngésző (browser) helyes megválasztása alapvető fontosságú, hiszen ennek tudásától és megbízhatóságától függ, hogy a szerverrel való kommunikáció mennyire fog sikeresen lezajlani. A kliensoldali szoftvereken, egy böngészőszoftvert kell érteni, illetve a hozzá kapcsolódó számos egyéb szoftverterméket (plugin-t). Pl. Java JRE, Flash player, Video player, PDF viewer plugin.

A különböző HTML szabványok (HTML 4.01, XHTML 1.0), a CSS és JavaScript technológiák a böngésző alapszoftverébe vannak beépítve, hiszen elengedhetetlenek a webes tartalmak feldolgozásához és helyes megjelenítéséhez.

A fejlesztés során megcélzott munkafázisok: háttéradatbázis kifejlesztése, felhasználó - alapú hozzáférés, adminisztrátori és ügyféloldali webes interfészek kialakítása, valamint több ponton megvalósított alkalmazásbiztonsági beállítások elvégzése.

## 4. Bevezetés a HTML-be

### 4.1. A HTML definíciója, funkciója

A HTML (HyperText Markup Language) egy olyan leírónyelv, melyet kifejezetten a weboldalak megjelenítéséhez fejlesztettek ki, 1990 óta használják. HTML kódot nagyon egyszerűen akár egy alap szövegszerkesztővel is írhatunk, de vannak olyan HTML szerkesztők, amelyben a gyakran használt HTML tag-ek előállítására bizonyos billentyűkombinációk használhatóak, ez valamelyest meggyorsítja a fejlesztést. A szabványos HTML dokumentumot bármely HTML kompatibilis böngészővel megtekinthetjük. A nyelv létrejöttének fő célja, hogy jól felépített és szervezett dokumentum-struktúrát lehessen készíteni. Ami a dokumentumformázást illeti, rengeteg lehetőség nyílik meg számunkra a HTML használatával.

Az elmúlt évtizedben a HTML-nek számos verziója látott napvilágot, melyekkel egyre összetettebb dokumentum szerkezeteket lehetett létrehozni. A nyelv fejlődésében jelentős mérföldkö volt a 4.0-ás verzió megjelenése (1997 nyarán), mely fejlett űrlap és táblázatkezeléssel rendelkezett, hivatalosan is támogatta a keretek használatát. Eközben kliens oldalon elindult a böngészők közötti verseny a felhasználókért, mely egészen odáig vezetett,

hogy bizonyos gyártók eltérve a HTML 4.0 ajánlásától saját tag-eket és formázó elemeket vezettek be. Válasz lépésként létrehoztak egy robusztus és letisztult HTML formátumot, ez a XHTML (XML alapú HTML), mely lényegében egy letisztított 4.0-ás verzió XML (eXtensible Markup Language) adottságokkal integrálva. Így tehát kompatibilis minden XML és HTML alapú böngészővel. A jelenlegi web fejlesztési trendek egyértelműen az XHTML irányába mutatnak. A W3C szervezet nagy erővel fejleszti az XHTML 2.0-ás verzióját, a magasabb szintű XML integráltság elérése érdekében.

Az alábbi, általam kitalált minta kódrészlet a HTML 4.0-ás szabványú dokumentumszerkezet főbb részeit mutatja be:

```
<HTML>
<HEAD>
<TITLE>Az én weboldalam</TITLE>
<META HTTP-EQUIV="Expires" CONTENT="31-DEC-2009 12:00:00
GMT">
<LINK REL=HOME HREF="/tags/index.html">
<META NAME=AUTHOR CONTENT="Kósa Anita">
<META NAME="keywords" CONTENT="css web url">
</HEAD>

<BODY>
<H1>Az én weboldalam</H1>
<TABLE border="1" summary="A hotel vendégei számára biztosított webes
regisztrációs form.">
  <TR>
    <TD>Vezetéknév*: </TD>
    <TD><INPUT name="vez_nev" type="text" size="40" maxlength="40"
/> </TD>
  </TR>
  <TR>
    <TD>Keresztnév*: </TD>
    <TD><INPUT name="ker_nev" type="text" size="40" maxlength="40"
/> </TD>
  </TR>
  <TR>
    <TD>Foglalás kezdete: </TD>
    <TD><SELECT name="kezdet_ev">
      <OPTION> 2008 </OPTION>
      <OPTION> 2009 </OPTION>
      <OPTION> 2010 </OPTION>
    </SELECT>
```

```
</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

2. példa: HTML 4.0 kódrészlet

## 5. Stíluslapok

### 5.1. Bevezetés a CSS-be

Bár a HTML mint leírónyelv elsősorban a tartalommal foglalkozik mégis számos HTML tag rendelkezik a megjelenítést befolyásoló attribútummal. Már megszületése után nem sokkal felmerült az igény a tartalom és az azt megformázó kódok, attribútumok szétválasztására, különálló kezelésére. Erre válaszként született meg a W3C által kifejlesztett CSS 1.0 (Cascading Style Sheets) stíluslapszabvány. Ha belegondolunk, hogy egy alapszintű weboldal is már akár több tucat HTML fájlt tartalmazhat, logikus lépésnek tűnik a megjelenésüket befolyásoló stílusok egységesítésének lehetősége.

A HTML 4.0-ban a stílussal kapcsolatos attribútumok jelentős része elavultnak tekinthető a CSS szabványú stíluslapokkal szemben.

A stíluslapokat 3 szinten alkalmazhatjuk:

- belső (a belső stíluslapok kizárólag egy adott tag tartalmára vonatkoznak)
- dokumentum szintű (ezen a szinten megadott stíluslap a HTML teljes törzsében érvényes bizonyos tag-ekre)
- külső (külső önálló állományként megadott stíluslapok, melyek tetszőleges HTML dokumentumban alkalmazhatóak)

Az utóbbi külső szint nyújtja a legkényelmesebb tartalomformázási lehetőséget, ezért mostanra a CSS alkalmazásának legelterjedtebb formájává vált.

Az alábbi példa szemlélteti a belső szintű alkalmazást:

```
<p style="font-size: 24pt; color: blue; background-color: red">  
Ez a szöveg kék színnel és piros háttérrel jelenik meg!  
</p>
```

3. példa: Belső szintű CSS definíció

Példa a dokumentum szintű alkalmazásra:

```
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">  
<style type="text/css">  
body.hatter {  
    background-color: #c85535;  
}  
body.torzs {  
    background-color: #ffffde;  
    color: #000;  
    border-style: solid;  
    border-width: 0px;  
    font-family: Tahoma;  
    font-size: 10pt;  
}  
p.proba {  
    color: blue;  
    font-weight: bold;  
}  
</style>  
</head>  
<body class="hatter">  
<p class="proba">Ez itt kék színnel jelenik meg!</p>  
</body>  
</html>
```

4. példa: Dokumentum szintű CSS definíció

Példa a külső szintű alkalmazásra:

```
<link rel="stylesheet" type="text/css" href = "text/css  
http://localhost/hotel/hotel.css">  
</link>
```

5. példa: Külső szintű alkalmazás

## 6. Dinamikus tartalomgenerálás

### 6.1. A PHP története

1994-ben Rasmus Lerdorf nevéhez fűződik a PHP elődjének a megalkotása. Néhány egyszerűbb eszközből, ötletből indult ki, melyeket saját célra fejlesztett ki (pl.: kíváncsi volt, hányan olvassák el az önéletrajzát és írt egy olyan programot, melynek segítségével nyomon tudta követni az olvasottságot). Később mások is felfigyeltek ezekre a funkciókra, és megkérték Rasmust, hogy használhassák ők is az általa kifejlesztett megoldásokat a saját honlapjukon.

A nagy érdeklődésre való tekintettel továbbfejlesztette a programját, majd leírást készített hozzá és egy parancscsomagot állított össze belőle, valamint elindított egy levelezési listát is. 1995 tavaszán programja a Personal Home Page Tools nevet kapta, ez volt az 1.0-ás verzió. Az egyre különbözőbb, összetettebb funkciók iránti igény hatására Rasmus készített egy C nyelven megírt implementációt, mely már alkalmas volt alapszintű dinamikus webalkalmazás fejlesztésére és adatbázissal való kommunikációra. Az új csomag új nevet is kapott, ez volt a Form Interpreter.

Ezután a PHP fejlesztése rohamos léptekben haladt tovább.

### PHP 3

1998-ban már PHP 3.0-ról beszélünk, ekkor kapta a most is érvényes „PHP: Hypertext Preprocessor” elnevezést. A 3.0-ás verziót az Andi Gutmans és Zeev Suraski vezette fejlesztő csoport hozta létre, mely az 1995-ös első kiadás sokadik átírása, újraírása volt.

A PHP alkalmazása iránti igény tovább növekedett, egyre több webhely használta és használja, melyek száma ma is folyamatosan csak növekszik. Valószínűleg nagy szerepe van ebben annak, hogy a szoftver nyílt forráskódú és ingyenesen letölthető az internetről. Bár kizárólag emiatt nem döntenének tömegesen mellette a fejlesztők. Más oka is van, mégpedig megbízhatósága, rugalmassága és teljesítménye, illetve könnyen tanulhatósága. A PHP nyelv ugyanis a C, a PERL és a JAVA nyelv eszközeiből alakította ki szintaxisát. Így már elegendő a három nyelv valamelyikének ismerete ahhoz, hogy ne tűnjön számunkra teljesen ismeretlennek a PHP nyelvezet. Mivel nyílt forráskódú szkriptnyelvről van szó, így sok hobbi programozó önkéntesen, díjazás nélkül járult hozzá a fejlesztéséhez, hiszen nagyon

sokan foglalkoztak, foglalkoznak vele, a hibákat jelzik vagy jobb esetben akár javíthatják is. Ezen önkéntesek is nagyban hozzájárultak ahhoz, hogy mára a PHP népszerű és látványos funkciók és szolgáltatások egyre szélesedő spektrumát nyújtja a fejlesztők számára.

#### **PHP 4**

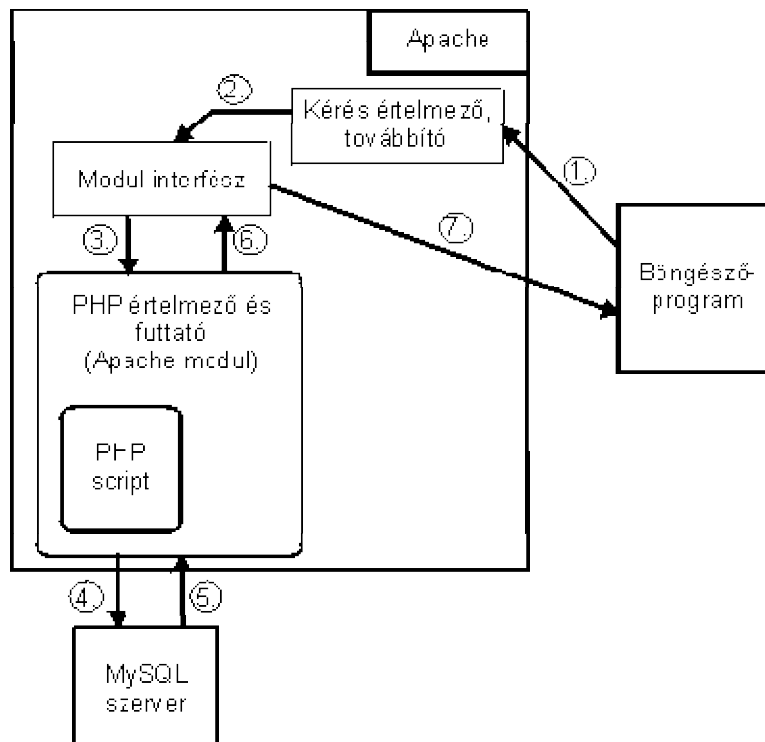
Hivatalosan 2000 májusában adták ki, az előző verzióhoz képest jelentős változásokat tapasztalhatunk. Maga a PHP nyelv nagyon sokat bővült, ettől a verziótól támogatja a modulok írását, a korábbi verziókban használt támogatásokból is modulok készültek, melyek PHP 4.0-ba már alapmodulként lettek beépítve. Fejlesztették az objektumorientált eszközkészletet, a fordítás és kód optimalizálást. Már nem csak az Apache-csal működött együtt, szélesebb lett a támogatott webserverek köre is. A kliens oldalon megadott adatok kezelése biztonságosabbá vált HTTP session-ök, kimenet puffereles alkalmazásával. Az egységesített adatfolyam-kezelés (Stream) is a bevezetett új technológiák között szerepel.

#### **PHP 5**

Hosszú fejlesztés és számos korábbi verzió előzte meg a PHP 5-öt, míg végül 2004 júliusában megjelenhetett. Alapja a Zend Engine 2.0 nevezetű motor, mely tartalmazza az új szolgáltatásokat, valamint az új objektumorientált modellt.

A PHP előnyei:

- Az operációs rendszertől való függetlenség, platformok közötti átjárhatóság pl.: ugyanazt a kódot használhatják a programozók Windows-on és Unix-on
- Az egyszerű, nagyon könnyen elsajátítható szintaxis
- Nagyon jó dokumentáció
- Ingyenes, nyílt forráskódú
- Gyorsaság (gyorsabb az ASP-nél)
- Funkcionalitás (nagyobb teret biztosít a programozóknak a webfejlesztés során, azzal, hogy a kódja közvetlenül beágyazódik a HTML-be.)



4. ábra: PHP illesztése Apache webserververhez dinamikus modulként

A PHP, mint HTML-be ágyazható nyelv rendelkezik olyan speciális tulajdonságokkal, amelyek elsőre ugyan nem tűnnek különlegesnek, mégis nagymértékben megkönnyítik a web alapú alkalmazások fejlesztését. A PHP kódban a kliensoldal felől érkező HTTP, POST vagy GET változók globális tömbként (PHP terminológia: superglobals) elérhetőek anélkül, hogy bármilyen külön műveletet, függvényhívást kellene ehhez végrehajtanunk. C-ben viszont, ha szerver oldali célokra szeretnénk alkalmazni a nyelvet, külön függvénykönyvtárakat kell megismernünk, melyek képesek feldolgozni a HTTP környezeti változókat.

Népszerűségének köszönhetően igen kitűnő, más nyelvekhez nehezen beszerezhető, speciális függvénykönyvtárak állnak a fejlesztők rendelkezésére, ezekből csak ízelítőül említtem a PDF, SWF (Flash multimédiás formátum), GIF/JPEG/PNG állományok olvasásának és generálásának lehetőségét, amelyek adatbázisokra épített alkalmazások esetén nagyon sokoldalúan használható rendszerré teszik a PHP-t.

A PHP további erőssége, és mint szerveroldali nyelv, elterjedésének másik fő oka az, hogy az

adatbázisszerverek rendkívül széles skáláját támogatja: dBase, IBM DB2, Informix, Ingres, Interbase, Frontbase, MS-SQL, MySQL, ODBC, Oracle, PostgreSQL, Sybase.

## 6.2. A PHP és a C

A PHP nyelv legtöbb elemének a szintaxisa a C-hez hasonló, vagy szinte teljesen azonos, sok esetben még működésük is megegyezik ennek megfelelően. A számos hasonlóság ellenére alapvető működésbeli és funkcióbeli különbségek is jelentkeznek, míg az eredeti ANSI C egy általános célú eljárásorientált programozási nyelv, addig a PHP-t beágyazható nyelvként egy kifejezetten speciális területre, a webes alkalmazások szerveroldali programozásához fejlesztették ki. Szoros kapcsolata a web alapú technológiákkal mai napig alapvető jellemzője a nyelvnek. A kód futtatásának módjában is jelentkezik különbség, mivel a PHP fordítás helyett parancsonként értelmezve (interpreterként) hajtja végre az utasítás sorozatot. Mára a nyelv számos elemet integrált a hagyományos programozás technikai megoldásokból pl.: végezhet előfordítást és objektumorientált eszközrendszerrel is kiegészült.

Az alábbiakban összehasonlítom a két nyelvet néhány alapvető nyelvi eszköz alapján.

### Tömbök:

A PHP a számindexekkel jelölt és az asszociatív tömb használatát is lehetővé teszi, ezen felül támogatja a többdimenziós tömböket. A tömböt számmal és névvel is indexelhetjük, definiálására és értékadására az `array()` függvényt használhatjuk.

*pl.: \$mintaTomb=array("Első", "Második", "Harmadik", "Negyedik");*

C-ben csak egész számmal indexelhetünk tömböt.

Az asszociatív tömb indexei karaktersorozatok.

*pl.: \$minta=array(Elem1=>"Első", Elem2=>"Második"); print \$minta[Elem1];*

PHP-ban a többdimenziós tömb a tömbök tömbjeként képzelhető el.

A C nem ismeri a többdimenziós tömböt, mint tárolási formát, helyette olyan egydimenziós tömböket használ, amelynek elemei egydimenziós tömbök.

Mindkét nyelv őrzi a tömbelemekre történő hivatkozás klasszikus szintaxisát.

*pl.: PHP: print "\$mintaTomb[2]";*

*C: print("%d", int\_tomb[1]);*

### Karaktorsorozatok:

A PHP gazdag eszközkészlettel rendelkezik (függvények, operátorok) a karaktorsorozatok formázására, kezelésére, melyek között találhatunk C-ből átvett függvényeket pl.: `printf`, `strlen`, de hasonlóan van olyan függvény amely segítségével karakter eltávolítható, vagy bizonyos karaktert másra tudunk cserélni.

### Típuskonverzió:

PHP változók létrehozásakor nem kötelező megadni annak típusát, mivel a felvett érték alapján állapítja meg azt a futtatókörnyezet.

Az alábbi adattípusokat ismeri a nyelv: `integer`, `double`, `string`, `boolean`, `object` és `array` (tömb), tartalmaz olyan függvényt (`gettype()`), amellyel megállapítható egy változó adattípusa, illetve egy változó beállítható a kívánt adattípusra a `settype()` segítségével. Adattípus ideiglenes beállítását is támogatja, ehhez a változó elé kell írni a kívánt adattípus nevét zárójelk között. pl.: `$szam=1.2659`; `$intszam=(integer)$szam`; `print $intszam`; (az eredmény 1 lesz)

### Változók:

PHP-ban a változók neve elé kell írni a '\$' karaktert, a típusát nem szükséges megadni. A nyelv támogatja az úgynevezett dinamikus változó használatát, melynek érdekessége, hogy egy változónevet egy másik változóban tárolunk. Vannak olyan helyzetek, amikor kifejezetten hasznos lehet a dinamikus változó.

Hagyományos értékadás:

*pl.:* `$nev="Anita"`;

Dinamikus értékadás:

*pl.:* `$seged="nev"`; `$$seged="Anita"`;

A `$seged` a "nev" karaktorsorozatot tartalmazza, a `$(seged)` változót úgy tekintjük, mintha egy '\$' karakter lenne, amit a `$seged` követ, vagyis a "nev". Tehát mindkét változómegadás esetén ugyanazt az eredményt kapjuk.

### Állandók:

Az ANSI C-hez hasonló a konstansok létrehozásának módja, annyi eltéréssel, hogy PHP-ban definiálásukat a `define()` függvény segítségével végezzük. Amikor egy állandót

definiálunk és értéket kap, akkor az a későbbiek során már nem módosítható. Az állandókra a nevükkel hivatkozunk a program során.

#### Kifejezések, operátorok:

PHP-ban klasszikus módon, infix jelöléssel az operandusokat operátorok segítségével kötjük össze (pl.:  $a + b$ ), melyek együttesét nevezi a PHP kifejezésnek. 4 operátortípust különböztet meg: értékadó, matematikai, összehasonlító, logikai.

A C kifejezések esetén a következő operátorokat használhatjuk: egyoperandusú, kétoperandusú és értékadó operátor.

#### Vezérlő szerkezetek:

PHP-ban a vezérlés a következő négy utasítással hajtható végre: if utasítás, switch utasítás, while és for ciklus. A C ugyanezen utasításainak szintaktikájával teljesen megegyező.

#### Függvények:

PHP-ban vannak belső és felhasználó által definiált függvények. A belső függvények a nyelvbe beépítettek, bárki számára elérhetőek, míg a felhasználói függvények csak a létrehozó felhasználóknak.

Függvény definíció PHP-ban: *function függvéynév([arg1],..., [argn]){...}*

C-ben: *[visszatérési\_érték\_típusa] függvéynév([típus arg1], ..., [típus argn]){...}*

Az évek során mindkét nyelv folyamatosan fejlődött és fejlődik ma is, ezért az összetettebb nyelvi eszközrendszerek összehasonlítása terjedelmi okokból nem célja dolgozatomnak.

## 7. Adatbázis technológiák (MySQL)

### 7.1. Relációs adatbázisok kezelése

A MySQL egy többfelhasználós, többszálú, SQL alapú relációs adatbázis-kezelő rendszer. Az 1960-as évek végén E. F. Codd fejlesztette ki a relációs adatbázisrendszerek eredeti elképzelését, amely Codd matematikai adatelméletén alapszik. Egy relációs adatbázis adattáblák gyűjteménye, melyben tetszőleges számú sora és oszlopa lehet egy táblának. A

táblák oszlopai névvel rendelkeznek, így tudunk majd később hivatkozni az oszlopokra. Ez alapján tudunk lekérdezni, módosítani, vagy akár feltölteni táblákat, táblák oszlopait. Jellemző, hogy a táblákhoz elsődleges kulcsot rendelünk, amely egy speciális értékeket tartalmazó oszlopa a táblának, segítségével egyértelműen azonosíthatóak a tábla sorai.

A sorokat a relációs adatbázis-modell terminológiájában rekordoknak nevezzük.

Az SQL egy olyan lekérdező nyelv, amely a relációs adatbázisok elérését és módosítását teszi lehetővé.

A MySQL az egyik legismertebb nyílt forráskódú, ingyenes adatbázis-kezelő programcsomag. Elterjedésében valószínűleg nagy szerepet játszik az, hogy költséghatékony és egyszerű megoldásokat nyújt a manapság elterjedt dinamikus webfejlesztéshez.

Az adatbázisom kialakítása során egy adatbázis-hozzáférési rendszert kellett kidolgozni. Ezt természetesen már a feladat elején meg kell tervezni, de sokszor előfordulhat, hogy egy újabb funkció illetve újabb szempontok felmerülése miatt ezt a tervet módosítani kell. Mindenesetre nem kétséges, hogy ezzel foglalkozni kell, elsősorban azért, mert az előzetesen felállított adatbázis-hozzáférés alapján dől el, hogy mennyi adathoz és milyen lehetőségekkel férnek hozzá az adatbázishoz kapcsolódó felhasználók. Ez azt is jelenti, hogy egy rosszul felmért illetve összeállított adatbázis-hozzáférési szabályrendszer esetén adataink nagy része elveszhet egy esetleges illetéktelen behatolás esetén.

Az általános gyakorlat szerint fel kell készülnünk arra, hogy egy adatbázis szervert többen is használhatnak. Ragaszkodnunk kell ahhoz, hogy az adatbázis adminisztrátora (root) az alapvető adatbázis-hozzáféréseket megfelelően állítsa be, és jogosulatlan felhasználók ne férhessenek hozzá a mi adatbázisunkhoz (pl. egy másik adatbázis adminisztrátorai ne tudjanak kapcsolódni a mi adatbázisunkhoz).

Nem szabad figyelmen kívül hagynunk a MySQL szerver hálózati elérhetőségének szabályozását – hiszen ha bárki elérheti, akkor már csak a MySQL felhasználókezelése védheti meg adatainkat az illetéktelen hozzáférésektől, ez pedig önmagában kevés lehet. A MySQL-hez való hozzáférés, vagyis az adatbázis-kapcsolat kialakításának feltétele több kritériumhoz is köthető, a programom fejlesztése során a MySQL kizárólag localhost-ról fogadta a kapcsolódási kérélmeket. Az egyik, amit meg kell fontolnunk, hogy a MySQL

szervert futtató platformon (az én esetemben Windows platformon) mely felhasználó futtassa a MySQL szolgáltatást (vagyis magát a MySQL szervert). Az optimális megoldás az, ha külön *mysql* felhasználót hozunk létre erre a célra megfelelően beállított jogokkal, akinek kizárólagos írási és olvasási joga van az adatbázisokat tartalmazó alkönyvtárakra.

Annak ellenére, hogy az adatbázis csak közvetett módon, szerveroldali kód végrehajtásával érhető el a felhasználó számára, mindenképp gondoskodni kell a MySQL szintű biztonságról is, az adatbázis-hozzáférés korlátozásával. Ehhez létre kellett hozni a PHP számára egy dedikált SQL felhasználót, melynek jogait a hotel adatbázisra korlátoztam. Így a PHP is az adatbázisszerver felhasználójaként (SQL User) férhet hozzá az adatbázis adataihoz, e jogosultságnak megfelelően kérheti le, módosíthatja azokat.

Az adatbevitelt ellenőrizhetjük bizonyos mértékig adatbázis szinten is, mégsem érdemes főleg adatbázis terhelést előidézni, így, mint korábban írtam, törekedni kell arra, hogy a hibás adatok már el se jussanak az adatbázis-kezelőig. Minden, a felhasználók által felvitt adatot előzetesen, még PHP kód szinten ellenőrizni kell, ezt tettem én is.

## 7.2. Adatbázis tervezés

A *foglalások* táblában tárolom a feldolgozott foglalások adatait. Minden foglalás egyedi azonosítót kap *foglalas\_id*, ezen túlmenően ebben a táblában megjelenik külső kulcsként az *ugyfel\_id* mező, mely az *ugyfelek* nevű táblával teremt kapcsolatot.

Az *ugyfelek* tábla tartalmazza minden olyan ügyfél adatait, akik már legalább egyszer foglaltak szállást a webes rendszeren keresztül. Az *ugyfel\_id* egy auto increment mező (egyben a tábla elsődleges kulcsa), melynek értékét az adatbázis szerver állítja be minden egyes *insert* esetén eggyel növelve azt.

A *szobak* tábla tartalmazza a szálloda teljes szobalistáját, illetve az egyes szobákhoz tartozó felszereltséget.

```
CREATE TABLE `felhasznalok` (  
  `user_id` int(11) NOT NULL auto_increment,  
  `user_name` varchar(25) default NULL,  
  `password` varchar(255) default NULL,  
  PRIMARY KEY (`user_id`)
```

```

) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;

CREATE TABLE `foglalások` (
  `foglalás_id` int(11) NOT NULL auto_increment,
  `ugyfel_id` int(11) NOT NULL,
  `fidopont` datetime default NULL,
  `kezdet` date default NULL,
  `vege` date default NULL,
  `ellatas` varchar(30) default NULL,
  `szobaszam` int(11) NOT NULL,
  `spec_igények` varchar(40) default NULL,
  `bk_típus` varchar(15) NOT NULL,
  `bk_szam` varchar(19) NOT NULL,
  `bk_lejarat` varchar(7) NOT NULL,
  `egyeb` varchar(255) default NULL,
  PRIMARY KEY (`foglalás_id`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8;

CREATE TABLE `szobák` (
  `szobaszam` int(11) default NULL,
  `szobatispus` enum('egyagyas','ketagyas','franciaagyas') default NULL,
  `kabeltv` tinyint(1) default NULL,
  `legkondi` tinyint(1) default NULL,
  `barszekreny` tinyint(1) default NULL,
  `furdokad` tinyint(1) default NULL,
  `szobaszef` tinyint(1) default NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `ugyfelek` (
  `ugyfel_id` int(11) NOT NULL auto_increment,
  `nev` varchar(25) default NULL,
  `e_mail` varchar(30) default NULL,
  `tel` varchar(23) default NULL,
  `cim` varchar(120) default NULL,
  `szemelyi` varchar(15) default NULL,
  PRIMARY KEY (`ugyfel_id`)
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=utf8;

```

#### 6. példa: Adatbázis táblák létrehozása

Mint ahogy korábban említettem a PHP egyik nagy előnye, hogy fejlett adatbáziskezelő függvénytárral rendelkezik, így a MySQL utasításokat PHP kódból is egyszerűen végre tudjuk hajtani. Nem kell mást tenni, mint egy PHP változónak sztringként megadni a végrehajtandó SQL utasítást idézőjelek között. Nagyon fontos figyelni arra, hogy PHP-ban, ha sztringen belül van szükségünk egy változó értékére (pl. SESSION változók értékére vagyunk kíváncsiak) egy adott adatbázis lekérdezés során, akkor kapcsos zárójelek közé kell tenni a változót. Az így kapott formátumot pedig aposztrófok közé helyezzük abban az esetben, ha

karakterlánc típusú mezővel operálunk az SQL utasítás adott pontján.

(pl.: nev= '{\$\_SESSION['megrendelo\_neve']}')

Az alábbi példákat a program elkészítése során alkalmazott SQL utasítások közül válogattam:

A 7. példa során lekérdezem az *ugyfelek* tábla azon ügyfeleit, akiknek neve és személyigazolvány száma megegyezik a webes felületen megadott megrendelő névvel és személyigazolvány számal.

```
$query="SELECT * FROM ugyfelek WHERE  
nev='{$_SESSION['megrendelo_neve']}' AND  
szemelyi='{$_SESSION['szemszam']}';"
```

7. példa: Ügyfelek táblából lekérdezés

A 8. példában a *szobak* táblába két sor beszúrására hozok példát. A *szobak* tábla feltöltését az adatbázis táblák létrehozása után megtettem, arra alapozva, hogy egy hotel szobáinak a szobaszáma, illetve felszereltsége ritkán változik.

```
INSERT INTO szobak VALUES (1,'egyagyas',1,0,0,0,0);  
INSERT INTO szobak VALUES (8,'ketagyas',1,1,1,1,1);
```

8. példa: A szobak tábla egy-egy sorának létrehozása

A 9. példa a *foglalások* táblába vesz fel egy újabb foglalást, ha a felhasználó által megadott adatok rendben vannak, az ellenőrzések sikeresen végrehajtottak. Itt is nagyon kellett arra figyelni, hogy a tábla mezői milyen típusúak. Az *ugyfel\_id* egy INTEGER típusú oszlop, így az ide beszúrandó értéket a következőképpen kell megadni '{\$\_SESSION['ugyfel\_id']}'. Az *ellatas* egy VARCHAR típusú oszlop, az ide beszúrandó értéket '{\$\_SESSION['ellatas']}' formában kell megadni.

```
$query="INSERT INTO foglalasok (ugyfel_id,fidopont,kezdet,vege,ellatas,  
szobaszam,spec_igenyek,bk_tipus,bk_szam,bk_lejarat,egyeb) VALUES  
({$_SESSION['ugyfel_id']},{$_SESSION['fidopont']},  
'{$_SESSION['kezdet]}', '{$_SESSION['vege]}',  
'{$_SESSION['ellatas]}', '{$_SESSION['elso_szabad]}',  
'{$_SESSION['spec_igenyek]}', '{$_SESSION['bk_tipus]}',  
'{$_SESSION['bk_szam]}', '{$_SESSION['bk_lejarat]}',  
'{$_SESSION['egyeb]}')";
```

9.példa: Új foglalás felvétele a *foglalas* táblába

## 8. A program részletes bemutatása

Az elkészített webes alkalmazás legfontosabb feladata az ügyfelek és a szálloda közötti foglalási ügymenet felgyorsítása, hatékony menedzselése. Ahhoz, hogy ennek megfeleljen a fejlesztett alkalmazás, mind az ügyfél, mint pedig a foglalást menedzselő recepciós személyzet számára hatékony felületet kell nyújtania. Már a tervezési fázisban kellő figyelmet kell fordítani a felhasználók által bevitt adatok ellenőrzésére. Két ponton van lehetőségünk adatellenőrzésre: a kliens oldalon, még mielőtt az adatok továbbításra kerülnének a szerverhez vagy magán a szerveren az adatbázisba történő behelyezést megelőzően.

Természetesen mindkét megoldásnak vannak előnyei és hátrányai. A kliens oldali adatellenőrzés megakadályozza, hogy a nem megfelelő adat eljusson a szerverig, viszont az ellenőrző kód a kliens oldalon olvashatóvá válik. Ennek esetleges hibáit kihasználva a rendszeren biztonsági rés keletkezhet, ezért annak, aki ezt a megoldást választja, törekednie kell a tökéletes, mindenre kiterjedő adatellenőrzésre. A szerver oldali ellenőrzés esetén a kód rejtve marad a felhasználó elől, azonban hibás adatok is eljuthatnak a szerverig ellenőrzés nélkül. Ebben az esetben nagy mennyiségű beérkező adat a rendszer működését veszélyeztetheti, így gondoskodni kell a megfelelő rendszervédelemről. Az általam megvalósított foglalási rendszerben a szerver oldali adatellenőrzést alkalmazom, mivel biztonsági szempontból kézben tartható és naplózható a kódok futtatása, az adatbázis nincs közvetlen kapcsolatban a felhasználóval.

A program gerincét az "index.php" nevezetű fájl adja, amiben a program működését vezérlem. A portál megjelenését és stílusát HTML és CSS használatával készítettem el. Bal oldalon egy menüsor található (nevezzük ezt bal-oldalsávnak), ahol a menükre kattintva a hozzájuk tartozó megfelelő oldalak jelennek meg. A bal-oldalsávban megjelenítendő menüsört a "menupontok.dat" fájlban határozom meg, mely tartalmazza a menü nevét, a megjeleníteni kívánt oldal kódját tartalmazó fájl nevét kiterjesztés nélkül, valamint egy rövid leírást a kapcsolódó funkcióról.

Az "index.php" tartalmazza azt a kódrészt, amely megvalósítja azt a funkciót, melynek eredményeképpen a menükre kattintva a megfelelő generált oldalak jelennek meg.

```

<?php
$menuPontok = file('menupontok.dat');
foreach ($menuPontok as $menuPont) {
    if (trim($menuPont) != ""){
        $itemFields = explode("\t", trim($menuPont));
        $ref = $itemFields[1];
        $menuPont[$ref]['linkcim'] = $itemFields[0];
        $menuPont[$ref]['oldalcim'] = $itemFields[2];
        $newLink = "<a href=\"{$_SERVER['PHP_SELF']}?
                    page={ $itemFields[1]}\>";
        $newLink .= "{ $itemFields[0]} </a>\n";
        if (isset($_REQUEST['page']) &&
            ($itemFields[1] == $_REQUEST['page'])){
            $newLink = "<div class=\"act\">{$newLink}</div>";
        }
        print "<div class=\"menu\">{$newLink}</div>";
    }else
        print "<hr width=\"50%\" align=\"center\">\n";
    }
?>

```

10. példa: Részlet az index.php-ból (menüpontok generálása)

A bal oldali menük közül részletesen ismertetem azokat, amelyek dinamikus tartalomra hivatkoznak:

Interaktivitás és adatfeldolgozás szempontjából a portál egyik legfontosabb menüpontja a szobafoglalás menüpont. A látogatónak, amennyiben szobát szeretne foglalni, első lépésként ki kell töltenie a megjelenő űrlapot (form-ot). A kötelezően kitöltendő mezők csillaggal vannak megjelölve.

## Hotel Diploma\*\*\*

[Magunkról](#)

[Szolgáltatásaink](#)

[Szobafoglalás](#)

[Árak](#)

[Elérhetőségek](#)

[Üzenet](#)

[Adminisztráció](#)

Vezetéknév\*:

Keresztnév\*:

Személyigazolvány szám\*:

Foglalás kezdete\*:

Foglalás vége\*:

**Szoba típusa:**

Szoba típusa\*:

Szobák száma:

Ellátás:

**Felszereltség:**

Kábel Tv

Légkondicionáló

Fürdőkád

Bárszekrény

Szobaszéf

Speciális igények:

kisállat  gyerek ágy  mozgássérült szoba

Egyéb kérés:

5. ábra: Foglalási űrlap

Amennyiben a kötelező mezőket elmulasztja kitölteni a felhasználó, vagy hibás adatot ad meg, akkor értelmes, a hiba okát megnevező hibaüzenetet kap eredményül. A sikeres kitöltés után, ellenőrzöm, hogy van-e a kívánt igényeknek megfelelő és a kért szobatípussal megegyező szabad szoba a megadott időintervallumban.

```

$seged=" ";
if($_SESSION['felszereltség'][0]=="TRUE") $seged.=" AND s.kabeltv=TRUE ";
if($_SESSION['felszereltség'][1]=="TRUE") $seged.=" AND s.legkondi=TRUE ";
if($_SESSION['felszereltség'][2]=="TRUE") $seged.=" AND s.furdokad=TRUE ";
if($_SESSION['felszereltség'][3]=="TRUE") $seged.=" AND s.barszekreny=TRUE";
if($_SESSION['felszereltség'][4]=="TRUE") $seged.=" AND s.szobaszef=TRUE ";

$lekerdez="SELECT DISTINCT(s.szobaszam) FROM szobak s, foglalások f
WHERE s.szobatispus='{$_SESSION['szobatispus']}'"$seged."AND

```

```

NOT(f.kezdet>{'$_SESSION['kezdet']}' OR f.vege<={'$_SESSION['vege']'}) AND
s.szobaszam<>f.szobaszam";
$szabad_szobak=mysql_query($lekerdez);
$szabad=mysql_num_rows($szabad_szobak);

```

11.példa: Szabad szobák meghatározása

Ha van szabad szoba, akkor ellenőrzöm, hogy a megrendelő foglalt-e már valaha szobát a hotelben:

```

if($szabad>0){
    $query="SELECT * FROM ugyfelek WHERE
        nev={'$_SESSION['megrendelo_neve']}' AND
        személyi={'$_SESSION['szemszam']}";
    $letezik=mysql_query($query);
    $letezik_db=mysql_num_rows($letezik);
    if($letezik_db>0){
        $query="SELECT * FROM ugyfelek WHERE
            nev={'$_SESSION['megrendelo_neve']}' AND
            személyi={'$_SESSION['szemszam']}";
        $tarol=mysql_query($query);
        $_SESSION['ugyfel_id']=mysql_result($tarol,0,'ugyfel_id');
        $_SESSION['nev']=mysql_result($tarol,0,'nev');
        $_SESSION['e_mail']=mysql_result($tarol,0,'e_mail');
        $_SESSION['tel']=mysql_result($tarol,0,'tel');
        $_SESSION['cim']=mysql_result($tarol,0,'cim');
        $_SESSION['szemelyi']=mysql_result($tarol,0,'szemelyi');
        include("pages/form_lista.php");
    }
    else {
        include("pages/form_kitolt.php");
    }
}

```

12. példa: Ügyfél ellenőrzése

Ha nem, akkor megjelenik egy további űrlap, amin meg kell adnia a szobafoglalás végső hitelesítéséhez szükséges adatait. A már korábban bekért adatok ezen az űrlapon már megjelennek, a hiányzóakra pedig újabb vizsgálatok vonatkoznak.

## Hotel Diploma\*\*\*

<a href="#">Magunkról</a>	Megrendelő neve:	Második Ügyfél
<a href="#">Szolgáltatásaink</a>	Email cím:	<input type="text" value="masodik@ugyfel.hu"/>
<a href="#">Szobafoglalás</a>	Telefonszám*:	<input type="text" value="12345666"/>
<a href="#">Árak</a>	Számlázási cím:	<input type="text" value="Debrecen"/>
<a href="#">Elérhetőségek</a>	Személyigazolvány szám:	43256654747
<a href="#">Üzenet</a>	Foglalás kezdete:	2008-06-17
<a href="#">Adminisztráció</a>	Foglalás vége:	2008-06-23
	<b>Szoba típusa:</b>	Szoba típusa: ketagyas
	Szobák száma:	1
	Ellátás:	Reggeli
	<b>Fizetési mód:</b>	
	Bankkártya típusa:	
	EuroCard/MasterCard:	<input type="radio"/> Visa: <input checked="" type="radio"/> Maestro: <input type="radio"/> American Express: <input type="radio"/>
	Bankkártyaszám*:	<input type="text" value="2222"/> <input type="text" value="3333"/> <input type="text" value="4444"/> <input type="text" value="5555"/>
	Lejárat*:	<input type="text" value="2008"/> <input type="text" value="10"/>
	CVV code*:	<input type="text" value="1111"/>
	<b>Felszereltség:</b>	
	Kábel Tv, Fürdőkád,	
	Speciális igények:	
	kisállat,	
	Egyéb kérés:	
	<input type="button" value="Küld"/>	

6. ábra: További adatok megadása

Ha a foglaláshoz szükséges minden adatot sikerült megadni, akkor a szobafoglalás érvényessé válik és rögzítésre kerül.

Ha igen, akkor a korábbi foglalása során már megadott személyes adatait nem kell ismételtelen megadnia, az adatbázisból visszakeresem a felhasználót a név és a személyi igazolvány szám alapján és az űrlapon tájékoztató jelleggel megjelenítem a róla tárolt adatokat is. A foglaláshoz még szükséges adatok megadása és ellenőrzése után a tranzakció eredményéről szöveges visszaigazolást kap az ügyfél, az adatbázisba pedig regisztrálásra kerül foglalása.

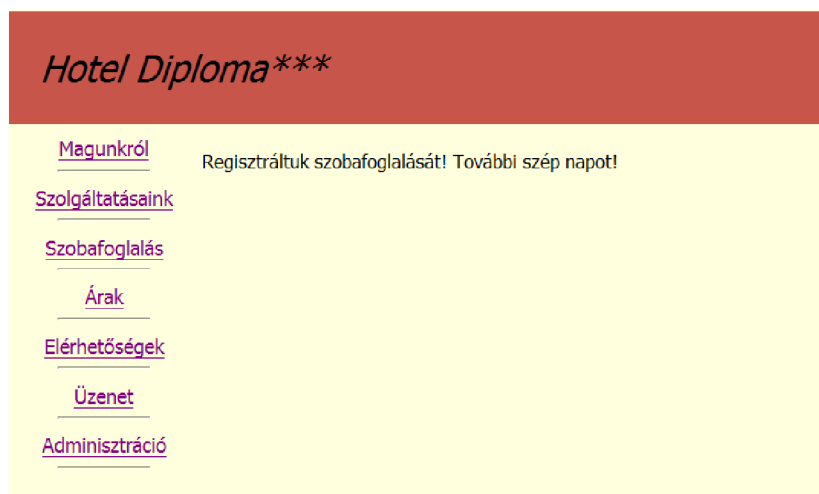
```
if(!empty($_SESSION['tel']) && !empty($_SESSION['bk_tipus']) &&
!empty($_SESSION['bk_szam']) && !empty($_SESSION['bk_lejarat'])){
$query="INSERT INTO ugyfelek (nev,e_mail,tel,cim,szemelyi) VALUES
('".$_SESSION['megrendelo_neve'],'".$_SESSION['e_mail'],'
".$_SESSION['tel'],'".$_SESSION['cim]'),'
".$_SESSION['szemszam]')";
```

```

mysql_query($query,$kapcsolat);
$_SESSION['ugyfel_id']=mysql_insert_id();
$query="INSERT INTO foglalasok
(ugyfel_id,fidopont,kezdet,vege,ellatas,szobaszam,spec_igenyek,
bk_tipus,bk_szam,bk_lejarat,egyeb) VALUES
({$_SESSION['ugyfel_id']},{$_SESSION['fidopont']},
'$_SESSION['kezdet']}', '$_SESSION['vege']}',
'$_SESSION['ellatas']}', {$_SESSION['elso_szabad']},
'$_SESSION['spec_igenyek']}', '$_SESSION['bk_tipus']}',
'$_SESSION['bk_szam']}', '$_SESSION['bk_lejarat']}',
'$_SESSION['egyeb']}')";
mysql_query($query,$kapcsolat);
echo "Regisztráltuk szobafoglalását! További szép napot!";
}

```

13. példa: Szobafoglalás rögzítése adatbázisban



7. ábra: Visszajelzés a szobafoglalás eredményéről

Az üzenet menüre kattintva lehetőség van a közvetlen kapcsolatfelvételre (szabványos SMTP alapú email küldésével), melynek segítségével lehet ajánlatot kérni, érdeklődni, észrevételeket leírni vagy akár panaszt tenni. Ez az üzenet továbbításra kerül az üzemeltető által megadott email címre.

8. ábra: Üzenetküldés

Az Adminisztráció menüponthoz egy olyan adminisztrációs felület tartozik, ahol a hozzáférési jogosultsággal rendelkező személyek (felhasználónév, jelszó megadásával) a foglalásokról kaphatnak részletes információt.

9. ábra: Bejelentkezés az adminisztrációs felületre

Amennyiben sikertelen a belépés, a képernyőn megjelenik a következő üzenet:

*Hibás felhasználónév vagy jelszó! Kérem próbálja újra!*

Ha sikeres a belépés, akkor foglaltsággal kapcsolatos információkat kaphatunk az adott keresési feltételeknek megfelelően: foglalási időintervallum megadása, szűrés szobatípusra, valamint egy konkrét foglaláshoz kapcsolódó személyi adatok lekérdezése. A kereséshez szükséges megadni a foglalás kezdetét, a foglalás végét és a szobatípust.



The screenshot shows the search interface of the Hotel Diploma\*\*\* website. The page has a red header with the text "Hotel Diploma\*\*\*". Below the header, there is a navigation menu on the left with links: "Magunkról", "Szolgáltatásaink", "Szobafoglalás", "Árak", "Elérhetőségek", "Üzenet", and "Adminisztráció". The main content area is titled "Ügyfél keresése:" and "Foglalások listája:". It contains a search form with the following fields: "Foglalás kezdete:" (Start of booking) with three dropdown menus for year (2008), month (01), and day (01); "Foglalás vége:" (End of booking) with three dropdown menus for year (2008), month (01), and day (01); and "Szoba típusa:" (Room type) with a dropdown menu set to "Egyágyas" (Single). A "Küld" (Send) button is located below the form.

10. ábra: Keresés az adatbázisban

Ha a megadott paramétereknek megfelelően volt foglalás, akkor megjelenítésre kerül.

**Hotel Diploma\*\*\***

[Magunkról](#)

[Szolgáltatásaink](#)

**Beérkezett foglalások**

[Szobafoglalás](#) Eddigi foglalások száma: 2

<a href="#">Árak</a>	Ügyfél azonosító	Foglalás időpontja	Tartózkodás kezdete	Tartózkodás vége	Ellátás	Szobaszám	Speciális igények	Bankkártya típusa	Bankkártyaszám	Lejárat dátuma	Egyéb igények
<a href="#">Elérhetőségek</a>	15	2008-04-20 20:04:56	2008-06-04	2008-06-09	Reggeli	11		visa	1111-1111-1111-1111	2009-11	
<a href="#">Üzenet</a>	17	2008-04-26 20:42:50	2008-08-08	2008-08-13	Reggeli	11	kisállat,	visa	8888-8888-8888-8888	2009-08	

[Adminisztráció](#)

11. ábra: A keresés eredménye

Az ügyfél azonosítóra kattintva a személyes adatokat tudom megtekinteni.

**Hotel Diploma\*\*\***

[Magunkról](#)

<a href="#">Szolgáltatásaink</a>	Ügyfél azonosító	Ügyfél neve	E-mail	Telefonszám	Cím	Személyigazolvány szám
<a href="#">Szobafoglalás</a>	15	Kósa Anita	anita.kosa@gmail.com	121212121	Debrecen	223232323

[Árak](#)

[Elérhetőségek](#)

[Üzenet](#)

[Adminisztráció](#)

12. ábra: Személyi adatok listája

## 9. Összefoglalás

A diplomamunkám témájához kapcsolódó webes alkalmazás fejlesztése során igen különböző technológiák illetve technikák együttes alkalmazására volt szükségem. Az elsöre egyszerűnek tűnő felhasználói felületet és könnyű használhatóságot igénylő feladat egy kihívásokkal teli munkává nőtte ki magát. A web fejlődése a mögötte levő technológiák fejlődését jelenti, melyek naprakész ismerete, mint ahogy a dolgozat írása során is megtapasztaltam, alapvető fontosságú a webes alkalmazások fejlesztői számára.

Céлом az volt, hogy diplomamunkám keretében az informatika egy aktuális, dinamikusan fejlődő területével, a web alapú alkalmazásfejlesztéssel foglalkozzak, és korszerű technológiák és technikák segítségével kifejlesszek egy webes foglалási rendszert.

Úgy érzem ezt a célt sikerült elérnem.

A program elkészítéséhez a PHP 5.2.3, a MySQL 5.0.45 és az Apache 2.2.8-as verziókat használtam. Azért esett választásom rájuk, mert segítségükkel adatbázisból dinamikusan lehet előállítani tartalmat és felhasználói irányból is képesek adatokat fogadni, valamint feldolgozni. Előzetes ismereteimnek megfelelően ez a szoftver kombináció jó választásnak bizonyult. Az általam megtervezett akkor még elméleti rendszer megvalósítása során számos fejlesztést segítő funkcióval találkoztam.

## 10. Irodalomjegyzék

### Nyomtatott források:

- [1] Robert W. Sebesta: A World Wide Web programozása (Panem Kiadó, 2005)
- [2] Julie C. Meloni: A PHP, a MzSQL és az Apache használata (Panem Kiadó, 2004)
- [3] R. Allen Wyke, Michael J. Walker, Robert M. Cox: PHP fejlesztők kézikönyve (Kossuth Kiadó, 2002)

### Internetes források:

- [4] [http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html)
- [5] <http://www.prog.hu/cikkek/194/A+HTTP+protokoll+es+a+World+Wide+Web.html>
- [6] [http://www.linuxvilag.hu/content/files/cikk/26/cikk\\_26\\_50\\_51.pdf](http://www.linuxvilag.hu/content/files/cikk/26/cikk_26_50_51.pdf)
- [7] <http://hu.wikipedia.org/wiki/MySQL>
- [8] <http://news.netcraft.com/>

## 11. Függetlenség

### index.php:

```
<?php
    session_start();
    $pagesdir="pages";
    $menupontok= array();
    if(!isset($_REQUEST['page']) || !file_exists("{ $pagesdir}/{ $_REQUEST['page'] }.php"))
        $p = "magunkrol";
    else
        $p = $_REQUEST['page'];
?>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
        <link rel="stylesheet" type="text/css" href="hotel.css">
        <title> Hotel Diploma*** </title>
    </head>
    <body class="hatter">
        <div>
            <table border=0 class="torzs" width="750" height="800" align="center">
            <tr><td class="fejlec" height="100" colspan=2> <I> Hotel Diploma*** </I> </td></tr>
            <tr>
                <td valign="top" width="170" class="menu" >
                    <?php
                        $menupontok = file('menupontok.dat');
                        foreach ($menupontok as $menupont) {
                            if (trim($menupont) != ""){
                                $itemfields = explode("\t", trim($menupont));
                                $ref = $itemfields[1];
                                $menupt[$ref]['linkcim'] = $itemfields[0];
                                $menupt[$ref]['oldalcim'] = $itemfields[2];
                                $newlink = "<a href=\"{$_SERVER['PHP_SELF']}?page={ $itemfields[1]}\">";
                                $newlink .= "{ $itemfields[0]} </a>\n";
                                if (isset($_REQUEST['page']) && ($itemfields[1] == $_REQUEST['page'])){
                                    $newlink = "<div class=\"act\">{ $newlink}</div>";
                                }
                                print "<div class=\"menu\">{ $newlink}</div>";
                            }
                        }
                    else
                        print "<hr width=\"50%\" align=\"center\">\n";
                }
            ?>
            </td>
            <td valign="top" class="tartalom"><?php include ("{$pagesdir}/{ $p }.php"); ?></td>
            </tr>
            </table>
        </div>
    </body>
</html>
```

### magunkrol.php:

```
<p><h3>Diplomamunka keretében készítette: Kósa Anita</h3></p>
```

### **szolgáltatás.php:**

```
<h4 class="kiemelt">BIZTONSÁG: </h4>
<ul class="szin">
  <li>Mágneskártyával működő ajtók </li>
  <li>Szobaszéf </li>
  <li>24 órás recepciós szolgálat </li>
  <li>Térfigyelő rendszer </li>
  <li>Zárt parkoló </li>
</ul>
<h4 class="kiemelt">KOMFORT: </h4>
<ul class="szin">
  <li>Színes TV </li>
  <li>Minibár a szobákban </li>
  <li>Hűtő-fűtő klímaberendezések </li>
  <li>Telefon- és internetkapcsolat minden szobában </li>
  <li>Személyszállító lift </li>
  <li>Szauna </li>
  <li>Transzfer szolgáltatás </li>
</ul>
```

### **form\_elsó.php:**

```
<form method="post" action="<?php print $_SERVER['PHP_SELF']; ?>">
<input name="page" type="hidden" value="foglalás" />
<table valign="top">
<tr><td>Vezetéknév*: </td>
  <td><input name="vez_nev" type="text" size="40" maxlength="40" /> </td></tr>
<tr><td>Keresztnév*: </td>
  <td><input name="ker_nev" type="text" size="40" maxlength="40" /> </td></tr>
<tr><td>Személyigazolvány szám*: </td>
  <td><input name="szemszam" type="text" size="40" maxlength="40" /> </td></tr>
<tr><td>Foglalás kezdete*: </td>
  <td><select name="kezdet_ev">
    <?php
      $datum=getdate();
      for($i=$datum['year']; $i<$datum['year']+3; $i++){ echo "<option> $i </option>";}
    ?>
  </select>
  <select name="kezdet_ho">
    <?php
      for($i=1; $i<13; $i++){
        if($i<10) echo "<option> 0$i </option>";
        else echo "<option> $i </option>";
      }
    ?>
  </select>
  <select name="kezdet_nap">
    <?php
      for($i=1; $i<32; $i++){
        if ($i<10) echo "<option> 0$i </option>";
        else echo "<option> $i </option>";
      }
    ?>
  </td>
```

```

        </select></td></tr>
<tr><td>Foglalás vége*: </td>
    <td><select name="vege_ev">
        <?php
            $datum=getdate();
            for($i=$datum['year']; $i<$datum['year']+3; $i++){ echo "<option> $i </option>"; }
        ?>
    </select>
    <select name="vege_ho">
        <?php
            for($i=1; $i<13; $i++){
                if($i<10) echo "<option> 0$i </option>";
                else echo "<option> $i </option>";
            }
        ?>
    </select>
    <select name="vege_nap">
        <?php
            for($i=1; $i<32; $i++){
                if ($i<10) echo "<option> 0$i </option>";
                else echo "<option> $i </option>";
            }
        ?>
    </select></td></tr>
</table>
<table>
<tr> <td class="kiemelt"> Szoba típusa: </td> </tr>
<tr> <td><p>Szoba típusa*:
    <select name="szobatipus">
        <option value="egyagyas"> Egyágyas </option>
        <option value="ketagyas"> Kétágyas </option>
        <option value="franciaagyas"> Franciaágyas </option>
    </select></p>
    <p>Szobák száma:
    <select name="darab">
        <option> 1 </option>
        <option> 2 </option>
        <option> 3 </option>
        <option> 4 </option>
    </select></p></td></tr>
</table>
<table>
<tr><td><p>Ellátás:
    <select name="ellatas">
        <option> Reggeli </option>
        <option> Félpanzió 3300 Ft/fő/nap</option>
        <option> Teljes panzió 6000 Ft</option>
    </select></p></td></tr>
</table>
<table>
<tr><td class="kiemelt">Felszereltség:</td></tr>
<tr><td><input type="checkbox" name="kabeltv" value="TRUE"> Kábel Tv <br />
    <input type="checkbox" name="legkondi" value="TRUE"> Légkondicionáló <br />

```

```



```

### **foglalas.php:**

```

<?php
$sql_username="hotel";
$sql_password="portal";
$sql_database="hotel";
$kapcsolat=mysql_connect("localhost",$sql_username,$sql_password);
if(!$kapcsolat) die("Nem lehet csatlakozni a MySQL kiszolgálóhoz!");
@mysql_select_db($sql_database) or die( "Nem létező adatbázis.");
$_SESSION['kapcsolat']=$kapcsolat;
$_SESSION['megrendelo_neve']=ucwords($_REQUEST['vez_nev']);
$_SESSION['veznev']=$_REQUEST['vez_nev'];
$_SESSION['megrendelo_neve'].=" ";
$_SESSION['megrendelo_neve'].=ucwords($_REQUEST['ker_nev']);
$_SESSION['kernev']=$_REQUEST['ker_nev'];
$_SESSION['szemzam']=$_REQUEST['szemzam'];
$_SESSION['kezdet']=$_REQUEST['kezdet_ev'];
$_SESSION['kezdet'].=" - ";
$_SESSION['kezdet'].=$_REQUEST['kezdet_ho'];
$_SESSION['kezdet'].=" - ";
$_SESSION['kezdet'].=$_REQUEST['kezdet_nap'];
$_SESSION['vege']=$_REQUEST['vege_ev'];
$_SESSION['vege'].=" - ";
$_SESSION['vege'].=$_REQUEST['vege_ho'];
$_SESSION['vege'].=" - ";
$_SESSION['vege'].=$_REQUEST['vege_nap'];
$_SESSION['szobatipus']=$_REQUEST['szobatipus'];
$_SESSION['darab']=$_REQUEST['darab'];
$_SESSION['ellatas']=$_REQUEST['ellatas'];
for($i=0; $i<5; $i++)
    $_SESSION['felszereltség'][$i]="FALSE";
$_SESSION['vissza_felszereltség']=" ";
if($_REQUEST['kabeltv']==TRUE) {
    $_SESSION['felszereltség'][0]=$_REQUEST['kabeltv'];
    $_SESSION['vissza_felszereltség']="Kábel Tv, ";}
if($_REQUEST['legkondi']==TRUE) {
    $_SESSION['felszereltség'][1]=$_REQUEST['legkondi'];
    $_SESSION['vissza_felszereltség']="Légkondicionáló, ";}

```

```

if($_REQUEST['furdokad']==TRUE) {
    $_SESSION['felszereltseg'][2]=$_REQUEST['furdokad'];
    $_SESSION['vissza_felszereltseg'].="Füüdökád, ";}
if($_REQUEST['barszekreny']==TRUE) {
    $_SESSION['felszereltseg'][3]=$_REQUEST['barszekreny'];
    $_SESSION['vissza_felszereltseg'].="Bárszekrény, ";}
if($_REQUEST['szobaszef']==TRUE) {
    $_SESSION['felszereltseg'][4]=$_REQUEST['szobaszef'];
    $_SESSION['vissza_felszereltseg'].="Szobaszéf";}
$_SESSION['spec_igenyek']=" ";
if($_REQUEST['kisallat']==TRUE) $_SESSION['spec_igenyek'].="kisállat, ";
if($_REQUEST['gyerekagy']==TRUE) $_SESSION['spec_igenyek'].="gyerek ágy, ";
if($_REQUEST['mserult_szoba']==TRUE) $_SESSION['spec_igenyek'].="mozgássérült szoba";
$_SESSION['egyeb']=$_REQUEST['egyeb'];

if ($_SESSION['megrendelo_neve']!=" " && $_SESSION['szemszam']!=" " &&
    $_SESSION['kezdet']<$_SESSION['vege']) {
    $seged=" ";
    if($_SESSION['felszereltseg'][0]=="TRUE") $seged.=" AND s.kabeltv=TRUE ";
    if($_SESSION['felszereltseg'][1]=="TRUE") $seged.=" AND s.legkondi=TRUE ";
    if($_SESSION['felszereltseg'][2]=="TRUE") $seged.=" AND s.furdokad=TRUE ";
    if($_SESSION['felszereltseg'][3]=="TRUE") $seged.=" AND s.barszekreny=TRUE ";
    if($_SESSION['felszereltseg'][4]=="TRUE") $seged.=" AND s.szobaszef=TRUE ";

    $lekerdez="SELECT DISTINCT(s.szobaszam) FROM szobak s, foglalasok f WHERE
        s.szobatispus='{$_SESSION['szobatispus']}'". $seged." AND NOT
        (f.kezdet>'{$_SESSION['kezdet']}' OR f.vege<='{$_SESSION['vege']}') AND
        s.szobaszam<>f.szobaszam";
    $szabad_szobak=mysql_query($lekerdez);
    $szabad=mysql_num_rows($szabad_szobak);

    if ($szabad==0) {
        $lekerdez2="SELECT s.szobaszam FROM szobak s WHERE
            s.szobatispus='{$_SESSION['szobatispus']}'". $seged;
        $szabad_szobak=mysql_query($lekerdez2,$kapcsolat);
        $szabad=mysql_num_rows($szabad_szobak);
    }
    $elso_szabad=mysql_fetch_array($szabad_szobak);
    $_SESSION['elso_szabad']=$elso_szabad[0];

    if($szabad>0){
        $query="SELECT * FROM ugyfelek WHERE nev='{$_SESSION['megrendelo_neve']}' AND
            szemelyi='{$_SESSION['szemszam']}'";
        $letezik=mysql_query($query);
        $letezik_db=mysql_num_rows($letezik);
        if($letezik_db>0){
            $query="SELECT * FROM ugyfelek WHERE nev='{$_SESSION['megrendelo_neve']}'
                AND szemelyi='{$_SESSION['szemszam']}'";
            $tarol=mysql_query($query);
            $_SESSION['ugyfel_id']=mysql_result($tarol,0,'ugyfel_id');
            $_SESSION['nev']=mysql_result($tarol,0,'nev');
            $_SESSION['e_mail']=mysql_result($tarol,0,'e_mail');
            $_SESSION['tel']=mysql_result($tarol,0,'tel');
        }
    }
}

```

```

        $_SESSION['cim']=mysql_result($starol,0,'cim');
        $_SESSION['szemelyi']=mysql_result($starol,0,'szemelyi');
        include("pages/form_lista.php");
    }
    else include("pages/form_kitolt.php");
}
else echo "Az adott időpontban az igényelt szobatípusból nem áll rendelkezésre. Kérem
        jelöljön meg más időpontot vagy szobatípust.";
}
else
    echo "Üres név vagy személyigazolvány szám vagy hibás időpont!";
?>

```

### form\_kitolt.php:

```

<form method="post" action="<?php print $_SERVER['PHP_SELF']; ?>">
<input name="page" type="hidden" value="store_kitolt" />
<table valign="top">
    <tr><td>Megrendelő neve: </td>
        <td><?php echo $_SESSION['megrendelo_neve']; ?> </td></tr>
    <tr><td>Email cím: </td>
        <td><input name="email" type="text" size="40" maxlength="40" /> </td></tr>
    <tr><td>Telefonszám*: </td>
        <td><input name="telszam" type="text" size="40" maxlength="40" /> </td></tr>
    <tr><td>Számlázási cím: </td>
        <td><input name="szamlazasi_cim" type="text" size="40" maxlength="40"/> </td></tr>
    <tr><td>Személyigazolvány szám: </td>
        <td><?php echo $_SESSION['szemszam']; ?> </td></tr>
    <tr><td>Foglalás kezdete: </td>
        <td><?php echo $_SESSION['kezet']; ?></td></tr>
    <tr><td>Foglalás vége: </td>
        <td><?php echo $_SESSION['vege']; ?></td></tr>
</table>
<table>
    <tr><td class="kiemelt"> Szoba típusa: </td></tr>
    <tr><td><p>Szoba típusa: <?php echo $_SESSION['szobatipus']; ?></p>
        <p>Szobák száma: <?php echo $_SESSION['darab']; ?></p></td></tr>
</table>
<table>
    <tr><td><p>Ellátás:<?php echo $_SESSION['ellatas']; ?></p></td></tr>
</table>
<table>
    <tr><td class="kiemelt">Fizetési mód:<br></td></tr>
    <tr><td>Bankkártya típusa: <br></td></tr>
    <tr><td>EuroCard/MasterCard: <input type="radio" value="master" name="kartya_tipus">
        Visa: <input type="radio" value="visa" name="kartya_tipus">
        Maestro: <input type="radio" value="maestro" name="kartya_tipus">
        American Express: <input type="radio" value="aexpress" name="kartya_tipus"></td>
</tr>
</table>
<table>
    <tr><td>Bankkártyaszám*: </td>
        <td><input name="kartya_szam1" type="text" size="4" maxlength="4" />
            <input name="kartya_szam2" type="text" size="4" maxlength="4" />

```

```

        <input name="kartya_szam3" type="text" size="4" maxlength="4" />
        <input name="kartya_szam4" type="text" size="4" maxlength="4" /></td></tr>
<tr><td>Lejárat*:
    <input name="lejarat_ev" type="text" size="4" maxlength="4" />
    <input name="lejarat_ho" type="text" size="2" maxlength="2" /></td>
    <td>CVV code*: <input name="cvv" type="text" size="4" maxlength="4" /></td></tr>
</table>
<table>
    <tr><td class="kiemelt">Felszereltség:</td></tr>
    <tr><td><?php echo $_SESSION['vissza_felszereltség']; ?></td></tr>
    <tr><td>Speciális igények:<br></td></tr>
    <tr><td><?php echo $_SESSION['spec_igények']; ?></td></tr>
    <tr><td>Egyéb kérés:</td></tr>
</table>
<?php echo $_SESSION['egyeb']; ?><br><br>
<input type="submit" value="Küld"/>
</form>

```

### store\_kitolt.php:

```

<?php
$sql_username="hotel";
$sql_password="portal";
$sql_database="hotel";
$kapcsolat=mysql_connect("localhost",$sql_username,$sql_password);
if(!$kapcsolat) die("Nem lehet csatlakozni a MySQL kiszolgálóhoz!");
@mysql_select_db($sql_database) or die( "Nem létező adatbázis.");
$_SESSION['e_mail']=$_REQUEST['email'];
$_SESSION['tel']=$_REQUEST['telszam'];
$_SESSION['cim']=$_REQUEST['szamlazasi_cim'];
$_SESSION['bk_tipus']=$_REQUEST['kartya_tipus'];
$_SESSION['bk_szam']=$_REQUEST['kartya_szam1'];
$_SESSION['bk_szam'].="-";
$_SESSION['bk_szam']=$_REQUEST['kartya_szam2'];
$_SESSION['bk_szam'].="-";
$_SESSION['bk_szam']=$_REQUEST['kartya_szam3'];
$_SESSION['bk_szam'].="-";
$_SESSION['bk_szam']=$_REQUEST['kartya_szam4'];
$_SESSION['bk_lejarat']=$_REQUEST['lejarat_ev'];
$_SESSION['bk_lejarat'].="-";
$_SESSION['bk_lejarat']=$_REQUEST['lejarat_ho'];
$_SESSION['cvv']=$_REQUEST['cvv'];
$fdatum=getdate();
$_SESSION['fidopont']=date('Y-m-d H:i:s', $fdatum[0]);

if(!empty($_SESSION['tel']) && !empty($_SESSION['bk_tipus']) &&
!empty($_SESSION['bk_szam']) && !empty($_SESSION['bk_lejarat'])){
    $query="INSERT INTO ugyfelek (nev,e_mail,tel,cim,szemelyi) VALUES
        ('$_SESSION['megrendelo_neve']', '$_SESSION['e_mail']', '$_SESSION['tel']',
        '$_SESSION['cim']', '$_SESSION['szemszam']')";
    mysql_query($query,$kapcsolat);
    $_SESSION['ugyfel_id']=mysql_insert_id();

    $query="INSERT INTO foglalasok (ugyfel_id,fidopont,kezdet,vege,ellatas,szobaszam,

```

```

spec_igenyek,bk_tipus,bk_szam,bk_lejarat,egyeb) VALUES
({$_SESSION['ugyfel_id'],'$_SESSION['fidopont'],'$_SESSION['kezdet']}',
'$_SESSION['vege'],'$_SESSION['ellatas'],'$_SESSION['elso_szabad']},
'$_SESSION['spec_igenyek'],'$_SESSION['bk_tipus'],'$_SESSION['bk_szam']}',
'$_SESSION['bk_lejarat'],'$_SESSION['egyeb']}");
mysql_query($query,$kapcsolat);
echo "Regisztráltuk szobafoglalását! További szép napot!";
}
else echo "Kérem mindenképp adja meg a telefonszámát, a bankkártyaszámát és a lejáratú időpontot!";
mysql_close($kapcsolat);
?>

```

### form\_lista.php:

```

<form method="post" action="<?php print $_SERVER['PHP_SELF']; ?>">
<input name="page" type="hidden" value="store_lista" />
<table valign="top">
  <tr><td>Megrendelő neve: </td>
    <td><?php echo $_SESSION['nev'];?> </td></tr>
  <tr><td>Email cím: </td>
    <td><?php echo $_SESSION['e_mail']; ?> </td></tr>
  <tr><td>Telefonszám: </td>
    <td><?php echo $_SESSION['tel'];?> </td></tr>
  <tr><td>Számlázási cím: </td>
    <td><?php echo $_SESSION['cim']; ?> </td></tr>
  <tr><td>Személyigazolvány szám: </td>
    <td><?php echo $_SESSION['szemelyi']; ?></td></tr>
  <tr><td>Foglalás kezdete: </td>
    <td><?php echo $_SESSION['kezdet']; ?></td></tr>
  <tr><td>Foglalás vége: </td>
    <td><?php echo $_SESSION['vege']; ?></td></tr>
</table>
<table>
  <tr><td class="kiemelt"> Szoba típusa: </td> </tr>
  <tr><td><p> Szoba típusa: <?php echo $_SESSION['szobatipus']; ?></p>
    <p> Szobák száma: <?php echo $_SESSION['darab']; ?></p></td></tr>
</table>
<table>
  <tr><td><p> Ellátás: <?php echo $_SESSION['ellatas']; ?> </p></td></tr>
</table>
<table>
  <tr><td class="kiemelt">Fizetési mód:<br></td></tr>
  <tr><td>Bankkártya típusa: <br></td></tr>
  <tr><td>EuroCard/MasterCard: <input type="radio" value="master" name="kartya_tipus">
    Visa: <input type="radio" value="visa" name="kartya_tipus">
    Maestro: <input type="radio" value="maestro" name="kartya_tipus">
    American Express: <input type="radio" value="aexpress" name="kartya_tipus"></td>
</tr>
</table>
<table>
  <tr><td>Bankkártyaszám*: </td>
    <td><input name="kartya_szam1" type="text" size="4" maxlength="4" />
      <input name="kartya_szam2" type="text" size="4" maxlength="4" />
      <input name="kartya_szam3" type="text" size="4" maxlength="4" />

```

```

        <input name="kartya_szam4" type="text" size="4" maxlength="4" /></td></tr>
<tr><td>Lejárat*: <input name="lejarat" type="text" size="4" maxlength="4" />
        <input name="kartya_szam" type="text" size="2" maxlength="2" /></td>
        <td>CVV code*: <input name="cvv" type="text" size="4" maxlength="4" /></td></tr>
</table>
<table>
    <tr><td class="kiemelt">Felszereltség:</td></tr>
    <tr><td><?php echo $_SESSION['vissza_felszereltség']; ?> </td></tr>
    <tr><td>Speciális igények:<br></td></tr>
    <tr><td><?php echo $_SESSION['spec_igenyek']; ?> </td></tr>
    <tr><td>Egyéb kérés:</td></tr>
</table>
<?php echo $_SESSION['egyeb']; ?><br /><br />
<input type="submit" value="Küld"/>
</form>

```

### store\_lista.php:

```

<?php
$sql_username="hotel";
$sql_password="portal";
$sql_database="hotel";
$kapcsolat=mysql_connect("localhost",$sql_username,$sql_password);
if(!$kapcsolat) die("Nem lehet csatlakozni a MySQL kiszolgálóhoz!");
@mysql_select_db($sql_database) or die( "Nem létező adatbázis.");
$_SESSION['bk_tipus']=$_REQUEST['kartya_tipus'];
$_SESSION['bk_szam']=$_REQUEST['kartya_szam1'];
$_SESSION['bk_szam'].="-";
$_SESSION['bk_szam']=$_REQUEST['kartya_szam2'];
$_SESSION['bk_szam'].="-";
$_SESSION['bk_szam']=$_REQUEST['kartya_szam3'];
$_SESSION['bk_szam'].="-";
$_SESSION['bk_szam']=$_REQUEST['kartya_szam4'];
$_SESSION['bk_lejarat']=$_REQUEST['lejarat_ev'];
$_SESSION['bk_lejarat'].="-";
$_SESSION['bk_lejarat']=$_REQUEST['lejarat_ho'];
$_SESSION['cvv']=$_REQUEST['cvv'];
$fdatum=getdate();
$_SESSION['fidopont']=date('Y-m-d H:i:s', $fdatum[0]);

if(!empty($_SESSION['bk_tipus']) && !empty($_SESSION['bk_szam']) &&
!empty($_SESSION['bk_lejarat'])){
    $_SESSION['ugyfel_id']=mysql_insert_id();
    $query="INSERT INTO foglalasok (ugyfel_id,fidopont,kezdet,vege,ellatas,szobaszam,
    spec_igenyek,bk_tipus,bk_szam,bk_lejarat,egyeb) VALUES
    ({$_SESSION['ugyfel_id']}, {$_SESSION['kezdet']},{$_SESSION['kezdet']},
    '{$_SESSION['vege']}',{$_SESSION['ellatas']}', {$_SESSION['elso_szabad']},
    '{$_SESSION['spec_igenyek']}', '{$_SESSION['bk_tipus']}', '{$_SESSION['bk_szam']}',
    '{$_SESSION['bk_lejarat']}', '{$_SESSION['egyeb']}')";
    mysql_query($query,$kapcsolat);
    echo "Regisztráltuk szobafoglalását! További szép napot!";
}
else    echo "Kérem mindenképp adja meg a bankkártyaszámát és a lejárat időpontot!";

```

```
mysql_close();
session_write_close();
?>
```

### **arak.php:**

```
<h3 class="szin">Áraink: </h3><br />
<table class="szin" border="border">
  <tr><th>Egyágyas szoba </th>
  <th>12900, - Ft/éj </th></tr>
  <tr><th>Kétágyas szoba </th>
  <th>17800, - Ft/éj </th></tr>
  <tr><th>Francia ágyas szoba </th>
  <th>19800, - Ft/éj </th></tr>
  <tr><th>Pótágy </th>
  <th>5100, - Ft/éj </th></tr>
  <tr><th>Háziállat </th>
  <th>2000, - Ft/éj </th></tr>
</table><br />
```

```
<h4 class="szin"><blink> A szobaárak tartalmazzák:</blink></h4>
<ul class="szin">
  <li>Büfé reggeli </li>
  <li>Szauna </li>
  <li>Ingyenes Internet (WiFi) </li>
  <li>Zárt parkoló </li>
</ul>
```

### **elerhetoseg.php:**

```
<h3>Posta- és elektronikus címeink:</h3>
4024 Debrecen, Új utca 7.<br />
Tel:30/467-3005<br />
Fax:30/467-3005<br />
E-mail:kosa.anita@gmail.com<br />
```

### **uzenet.php:**

```
<form method="post" action="<?php print $_SERVER['PHP_SELF']; ?>">
<input name="page" type="hidden" value="uzenet_kuld" />
Itt küldhet számunkra üzenetet kérdéseivel, észrevételeivel kapcsolatban.
<br /><br />
<table valign="top">
  <tr><td>Vezetéknév*: </td>
  <td><input name="nev" type="text" size="40" maxlength="60" /> </td></tr>
  <tr><td>Email cím*: </td>
  <td><input name="email" type="text" size="40" maxlength="40" /> </td></tr>
  <tr><td>Üzenet tárgya*: </td>
  <td><input name="utargy" type="text" size="40" maxlength="40" /> </td></tr>
</table>
<table>
  <tr><td><p>Üzenet típusa:<br /><br /></p></td></tr>
  <tr><td><select name="utipus">
    <option value="Információ"> Információ </option>
    <option value="Ajánlatkérés"> Ajánlatkérés </option>
  </select></td></tr>
</table>
```

```

        <option value="Észrevétel"> Észrevétel </option>
    </select></p></td></tr>
</table><br />
<textarea name="uzenet" rows="10" cols="50"></textarea><br /><br />
<input type="submit" value="Mehet" />
</form>

```

### uzenet\_kuld.php:

```

<?php
$_SESSION['nev']=$_REQUEST['nev'];
$_SESSION['e_mail']=$_REQUEST['email'];
$_SESSION['utargy']=$_REQUEST['utipus']." ".$_REQUEST['utargy'];
$_SESSION['uzenet']=$_REQUEST['uzenet'];
$_SESSION['hotel_email']="info@hoteldiploma.hu"
$fejlec = "MIME-Version: 1.0\r\n";
$fejlec.= "Content-type: text/html; charset=utf8\r\n";
$fejlec.= "From: \"".$_SESSION['nev']." \" <".$_SESSION['e_mail'].>";

if(!empty($_SESSION['nev']) && !empty($_SESSION['e_mail']) && !empty($_SESSION['uzenet']))
{
    if (mail($_SESSION['hotel_email'], $_SESSION['utargy'], $_SESSION['uzenet'], $fejlec))
        echo "Fogadtuk Üzenetét! További szép napot!";
    else
        echo "Üzenetét technikai okokból nem tudtuk fogadni, kérjük próbálja újra később.";
}
else echo "Kérem mindenképp adja meg nevét, email címét és töltse ki az üzenet mezőt!";
?>

```

### auth.php:

```

<form method="post" action="<?php print $_SERVER['PHP_SELF']; ?>">
<input name="page" type="hidden" value="login" />
<h3>Belépés az adminisztrációs oldalra</h3>
    <table valign="top">
        <tr><td>Felhasználónév*: </td>
            <td><input name="felhasznalonev" type="text" size="25" maxlength="25" /> </td></tr>
        <tr><td>Jelszó*: </td>
            <td><input name="jelszo" type="password" size="25" maxlength="25" /> </td></tr>
        <tr><td></td>
            <td><input type="submit" value="Belépés"/> </td></tr>
    </table>
</form>

```

### login.php:

```

<?php
session_start();
$sql_username="hotel";
$sql_password="portal";
$sql_database="hotel";
$kapcsolat=mysql_connect("localhost",$sql_username,$sql_password);
if(!$kapcsolat) die("Nem lehet csatlakozni a MySQL kiszolgálóhoz!");
@mysql_select_db($sql_database) or die( "Nem letezo adatbázis.");

```

```

if (!isset($_SESSION['uid'])) {
    $lekerdez="SELECT user_name, user_id FROM felhasznalok WHERE
        user_name='{$_REQUEST['felhasznalonev']}' AND
        password=MD5('{$_REQUEST['jelszo']}')";
    $eredmeny=mysql_query($lekerdez);
    if ($usr = mysql_fetch_array($eredmeny, MYSQL_ASSOC))
        $auth_result = $usr['user_name'] == $_REQUEST['felhasznalonev'];
    if ($auth_result == 1) {
        $referrer = $_SERVER['HTTP_REFERER'];
        $_SESSION['uid'] = $usr['user_id'];
        include("pages/admin_form.php");
    }
    else {
        unset($_SESSION['uid']);
        include("pages/sikertelen.php");
    }
}
else
    include("pages/admin.php");
?>

```

### admin\_form.php:

```

<form method="post" action="<?php print $_SERVER['PHP_SELF']; ?>">
<input name="page" type="hidden" value="admin" />
<table valign="top">
    <tr><td class="kiemelt">Ügyfél keresése: </td>
        <td><input name="keres_nev" type="text" size="40" maxlength="40"/> </td></tr>
    <tr><td class="kiemelt">Foglalások listája: </td></tr>
    <tr><td>Foglalás kezdete: </td>
        <td><select name="kezdet_ev">
            <?php
                $datum=getdate();
                for($i=$datum['year']; $i<$datum['year']+3; $i++){ echo "<option> $i </option>";}
            ?>
        </select>
        <select name="kezdet_ho">
            <?php
                for($i=1; $i<13; $i++){
                    if($i<10) echo "<option> 0$i </option>";
                    else echo "<option> $i </option>";
                }
            ?>
        </select>
        <select name="kezdet_nap">
            <?php
                for($i=1; $i<32; $i++){
                    if ($i<10) echo "<option> 0$i </option>";
                    else echo "<option> $i </option>";
                }
            ?>
        </select></td></tr>
    <tr><td>Foglalás vége: </td>
        <td><select name="vege_ev">

```

```

<?php
    $datum=getdate();
    for($i=$datum['year']; $i<$datum['year']+3; $i++){ echo "<option> $i </option>"; }
?>
</select>
<select name="vege_ho">
<?php
    for($i=1; $i<13; $i++){
        if($i<10) echo "<option> 0$i </option>";
        else echo "<option> $i </option>";
    }
?>
</select>
<select name="vege_nap">
<?php
    for($i=1; $i<32; $i++){
        if ($i<10) echo "<option> 0$i </option>";
        else echo "<option> $i </option>";
    }
?>
</select></td></tr>
<tr><td><p>Szoba típusa:
<select name="szobatipus">
    <option value="egyagyas"> Egyágyas </option>
    <option value="ketagyas"> Kétágyas </option>
    <option value="franciaagyas"> Franciaágyas </option>
</select></p></td></tr>
</table>
<input type="submit" value="Küld"/>
</form>

```

### admin.php:

```

<?php
$sql_username="hotel";
$sql_password="portal";
$sql_database="hotel";
$kapcsolat=mysql_connect("localhost",$sql_username,$sql_password);
if(!$kapcsolat) die("Nem lehet csatlakozni a MySQL kiszolgálóhoz!");
@mysql_select_db($sql_database) or die( "Nem letezo adatbazis.");
$_SESSION['keres_nev']=$_REQUEST['keres_nev'];
$_SESSION['keres_kezdet']=$_REQUEST['kezdet_ev'];
$_SESSION['keres_kezdet'].="-";
$_SESSION['keres_kezdet'].=$_REQUEST['kezdet_ho'];
$_SESSION['keres_kezdet'].="-";
$_SESSION['keres_kezdet'].=$_REQUEST['kezdet_nap'];
$_SESSION['keres_vege']=$_REQUEST['vege_ev'];
$_SESSION['keres_vege'].="-";
$_SESSION['keres_vege'].=$_REQUEST['vege_ho'];
$_SESSION['keres_vege'].="-";
$_SESSION['keres_vege'].=$_REQUEST['vege_nap'];
$_SESSION['keres_szobatipus']=$_REQUEST['szobatipus'];

if(!empty($_SESSION['keres_nev'])) {

```

```

$query="SELECT * FROM ugyfelek WHERE nev='{$_SESSION['keres_nev']}'";
$keresett=mysql_query($query);
$db_nev=mysql_num_rows($keresett);
?>
<table border="0" cellspacing="2" cellpadding="2">
  <tr><th><font face="Arial, Helvetica, sans-serif">Ügyfél azonosító</font></th>
    <th><font face="Arial, Helvetica, sans-serif">Ügyfél neve</font></th>
    <th><font face="Arial, Helvetica, sans-serif">E-mail</font></th>
    <th><font face="Arial, Helvetica, sans-serif">Telefonszám</font></th>
    <th><font face="Arial, Helvetica, sans-serif">Cím</font></th>
    <th><font face="Arial, Helvetica, sans-serif">Személyig.szám</font></th></tr>
<?php
$keres_ugyfel_id=mysql_result($keresett,0,'ugyfel_id');
$keres_ugyfel_nev=mysql_result($keresett,0,'nev');
$keres_ugyfel_email=mysql_result($keresett,0,'e_mail');
$keres_ugyfel_tel=mysql_result($keresett,0,'tel');
$keres_ugyfel_cim=mysql_result($keresett,0,'cim');
$keres_ugyfel_szemelyi=mysql_result($keresett,0,'szemelyi');
?>
<h3>
  <tr><td><font face="Arial, Helvetica, sans-serif"><?php echo $keres_ugyfel_id; ?></font></td>
    <td><font face="Arial, Helvetica, sans-serif"><?php echo $keres_ugyfel_nev; ?></font></td>
    <td><font face="Arial, Helvetica, sans-serif"><?php echo $keres_ugyfel_email; ?></font></td>
    <td><font face="Arial, Helvetica, sans-serif"><?php echo $keres_ugyfel_tel; ?></font></td>
    <td><font face="Arial, Helvetica, sans-serif"><?php echo $keres_ugyfel_cim; ?></font></td>
    <td><font face="Arial, Helvetica, sans-serif"><?php echo $keres_ugyfel_szemelyi; ?></font>
  </td></tr>
</h3>
<?php
echo "</table>";
}
if (isset($_SESSION['uid'])) {
  $query="SELECT * FROM foglalasok WHERE (kezdet>='{$_SESSION['keres_kezdet']}' OR
    vege<='{$_SESSION['keres_vege']}' ) AND szobaszam IN (SELECT szobaszam FROM
    szobak WHERE szobatipus='{$_SESSION['keres_szobatipus']}' ) ";
  $eredmeny=mysql_query($query);
  $foglalasok_szama=mysql_num_rows($eredmeny);
  echo "<b><center><h2>Beérkezett foglalások</center></h2></b>";
  print "Eddigi foglalások száma: $foglalasok_szama <br>";
}
?>
<table border="0" cellspacing="2" cellpadding="2">
  <tr><th><font face="Arial, Helvetica, sans-serif">Ügyfél azonosító</font></th>
    <th><font face="Arial, Helvetica, sans-serif">Foglalás időpontja</font></th>
    <th><font face="Arial, Helvetica, sans-serif">Tartózkodás kezdete</font></th>
    <th><font face="Arial, Helvetica, sans-serif">Tartózkodás vége</font></th>
    <th><font face="Arial, Helvetica, sans-serif">Ellátás</font></th>
    <th><font face="Arial, Helvetica, sans-serif">Szobaszám</font></th>
    <th><font face="Arial, Helvetica, sans-serif">Speciális igények</font></th>
    <th><font face="Arial, Helvetica, sans-serif">Bankkártya típusa</font></th>
    <th><font face="Arial, Helvetica, sans-serif">Bankkártyaszám</font></th>
    <th><font face="Arial, Helvetica, sans-serif">Lejárat dátuma</font></th>
    <th><font face="Arial, Helvetica, sans-serif">Egyéb igények</font></th></tr>

```

```

<?php
    $i=0;
    while ($i < $foglalasok_szama) {
        $ugyfel_id=mysql_result($eredmeny,$i,'ugyfel_id');
        $fidopont=mysql_result($eredmeny,$i,'fidopont');
        $kezdet=mysql_result($eredmeny,$i,'kezdet');
        $vege=mysql_result($eredmeny,$i,'vege');
        $ellatas=mysql_result($eredmeny,$i,'ellatas');
        $szobaszam=mysql_result($eredmeny,$i,'szobaszam');
        $spec_igenyek=mysql_result($eredmeny,$i,'spec_igenyek');
        $bk_tipus=mysql_result($eredmeny,$i,'bk_tipus');
        $bk_szam=mysql_result($eredmeny,$i,'bk_szam');
        $bk_lejarat=mysql_result($eredmeny,$i,'bk_lejarat');
        $egyeb=mysql_result($eredmeny,$i,'egyeb');
    }
    <h3>
        <tr><td><font face="Arial, Helvetica, sans-serif"><a href="<?php print $_SERVER['PHP_SELF'];
        print "?ugyfel_id={$ugyfel_id}&page=szemelyiadat"?>"><?php echo $ugyfel_id; ?>
        </a></font></td>
        <td><font face="Arial, Helvetica, sans-serif"><?php echo $fidopont; ?></font></td>
        <td><font face="Arial, Helvetica, sans-serif"><?php echo $kezdet; ?></font></td>
        <td><font face="Arial, Helvetica, sans-serif"><?php echo $vege;?></font></td>
        <td><font face="Arial, Helvetica, sans-serif"><?php echo $ellatas; ?></font></td>
        <td><font face="Arial, Helvetica, sans-serif"><?php echo $szobaszam;?></font></td>
        <td><font face="Arial, Helvetica, sans-serif"><?php echo $spec_igenyek;?></font></td>
        <td><font face="Arial, Helvetica, sans-serif"><?php echo $bk_tipus;?></font></td>
        <td><font face="Arial, Helvetica, sans-serif"><?php echo $bk_szam;?></font></td>
        <td><font face="Arial, Helvetica, sans-serif"><?php echo $bk_lejarat;?></font></td>
        <td><font face="Arial, Helvetica, sans-serif"><?php echo $egyeb; ?></font></td></tr>
    </h3>
    <?php
        ++$i;
    }
    echo "</table>";
?>

```

### szemelyiadat.php:

```

<?php
    $sql_username="hotel";
    $sql_password="portal";
    $sql_database="hotel";
    $kapcsolat=mysql_connect("localhost",$sql_username,$sql_password);
    if(!$kapcsolat) die("Nem lehet csatlakozni a MySQL kiszolgálóhoz!");
    @mysql_select_db($sql_database) or die( "Nem letezo adatbazis.");
    $_SESSION['l_ugyfel_id']=$_REQUEST['ugyfel_id'];
    $query="SELECT * FROM ugyfelek WHERE ugyfel_id='{$_SESSION['l_ugyfel_id']}'";
    $keresett=mysql_query($query);
    $db_nev=mysql_num_rows($keresett);
?>
<table border="0" cellspacing="2" cellpadding="2">
    <tr><th><font face="Arial, Helvetica, sans-serif">Ügyfél azonosító</font></th>
        <th><font face="Arial, Helvetica, sans-serif">Ügyfél neve</font></th>
        <th><font face="Arial, Helvetica, sans-serif">E-mail</font></th>

```

```

        <th><font face="Arial, Helvetica, sans-serif">Telefonszám</font></th>
        <th><font face="Arial, Helvetica, sans-serif">Cím</font></th>
        <th><font face="Arial, Helvetica, sans-serif">Személyigazolvány szám</font></th></tr>
<?php
    $keres_ugyfel_id=mysql_result($keresett,0,'ugyfel_id');
    $keres_ugyfel_nev=mysql_result($keresett,0,'nev');
    $keres_ugyfel_email=mysql_result($keresett,0,'e_mail');
    $keres_ugyfel_tel=mysql_result($keresett,0,'tel');
    $keres_ugyfel_cim=mysql_result($keresett,0,'cim');
    $keres_ugyfel_szemelyi=mysql_result($keresett,0,'szemelyi');
?>
<h3>
    <tr><td><font face="Arial, Helvetica, sans-serif"><?php echo $keres_ugyfel_id; ?></font></td>
        <td><font face="Arial, Helvetica, sans-serif"><?php echo $keres_ugyfel_nev; ?></font></td>
        <td><font face="Arial, Helvetica, sans-serif"><?php echo $keres_ugyfel_email; ?></font></td>
        <td><font face="Arial, Helvetica, sans-serif"><?php echo $keres_ugyfel_tel; ?></font></td>
        <td><font face="Arial, Helvetica, sans-serif"><?php echo $keres_ugyfel_cim; ?></font></td>
        <td><font face="Arial, Helvetica, sans-serif"><?php echo $keres_ugyfel_szemelyi; ?></font>
        </td></tr>
</h3>
<?php
    echo "</table>";
?>

```

#### **sikertelen.php:**

```

<?php
    echo "Hibás felhasználó név vagy jelszó! Kérem próbálja újra!";
?>

```

#### **menupontok.dat:**

```

Magunkról      magunkrol      Szállodánk története
Szolgáltatásaink      szolgaltatas      Vendégeink számára elérhető szolgáltatások
Szobafoglalás  form_elso      Online szobafoglalás
Árak  arak      Aktuális szobaárak
Elérhetőségek  elerhetoseg      A szálloda elérhetőségei
Üzenet  uzenet      Üzenet küldése a hotel számára
Adminisztráció  auth      A beérkezett foglalások adminisztrációja

```

#### **hotel.css:**

```

.hatter {
    background-color: #c85535;
}
.torzs{
    background-color: #ffffde;
    color: #000;
    border-style: solid;
    border-width: 0px;
    font-family: Tahoma;
    font-size: 10pt;
}

```

```
.proba{
    border-style: solid;
    border-width: 0px;
    border-color: #cc5535;
}
.fejlec{
    text-align: top;
    padding: 28px;
    font-size: 20pt;
    background-color: #c8554a;
}
.menu{
    padding-top: 5px;
    font-size: 11pt;
    text-align: center;
    line-height: 100%;
}
.act{
    text-decoration: none;
    font-weight: bold;
    background-color: #ffa465 ;
    color: #000;
}
.tartalom{
    text-align: top;
    padding-top: 20px;
}
.kiemelt{
    padding-top: 8px;
    color: #280000;
    font-size: 11pt;
    font-weight: bold;
}
.szin{
    color: #280000;
}
}
```