

# ***SZAKDOLGOZAT***

*Sipos Márton*

*Debrecen*

*2009*

**Debreceni Egyetem**  
**Informatikai Kar**

**Web alapú alkalmazás fejlesztése**

Témavezető:

Dr. Rutkovszky Edéné

Egyetemi tanársegéd

Készítette:

Sipos Márton

Programtervező informatikus

Debrecen

2009

## Tartalomjegyzék

Bevezetés.....	4
A PHP.....	4
A MySQL.....	6
Előrevetítés az alkalmazásra és a szakdolgozatra .....	7
Az alkalmazás.....	8
Az alkalmazásról röviden.....	8
Az Apache-ről .....	8
Az alkalmazás futtatása .....	9
Az alkalmazás bemutatása, részenként.....	12
Nyitóoldal.....	12
Menü.....	14
Főcím.....	15
Bejelentkezés.....	15
Kijelentkezés .....	19
Regisztráció .....	19
Profil.....	25
Főoldal.....	27
Fórum .....	29
Beleolvasás a fórumba .....	31
Hozzászólás a fórumban.....	33
Adminisztrátori műveletek a fórumban.....	35
Megfigyelő oldal .....	37
Megfigyelés beküldése.....	38
Megfigyelések megjelenítése .....	39
Információk a felhasználóról.....	42
Adminisztrátori műveletek a megfigyelőoldalon.....	42
Webshop.....	43
Böngészés a webshopban .....	44
Lehetőségek bejelentkezett felhasználóknak és a megrendelések .....	47
Adminisztrátori műveletek a webshopban .....	54
Összefoglalás.....	57
Irodalomjegyzék.....	58
Köszönetnyilvánítás .....	59

## Bevezetés

Napjainkra az internet mindennapi életünk részévé fejlődött, többségünk nap, mint nap használja azt. Sokaknak munkájuk elengedhetetlen részévé vált, és még többeknek nyújt sokféle szolgáltatást, melyek megkönnyítik elintéznivalójuk lebonyolítását, és segítséget nyújtanak a szórakozáshoz, kikapcsolódáshoz. Ilyen szolgáltatás például a levelezés, kapcsolattartás egyszerűsítésére és gyorsítására az e-mail, a chat-alkalmazások, vagy valósídejű hang illetve mozgókép átvitel, valamint a közösségi oldalak; vásárlások lebonyolítására jöttek létre az online boltok, webshop-ok; banki tranzakciókat végezhetünk a bank internetes oldalán keresztül; ismereteinket naprakészen tarthatjuk a hírportálok és szakportálok segítségével; és segíthetnek a kikapcsolódásban a különböző humoroldalak vagy videó megosztók. Ezen felsorolás csak néhányat tartalmaz az interneten elérhető rengeteg szolgáltatás közül, tehát sok mindent valóban gyorsabban elintézhetünk az internet segítségével.

Az interneten elérhető szolgáltatások többségét web böngésző segítségével érhetjük el. Kezdetben, még mikor az internet nem volt széles körben elérhető, az oldalakat HTML-ben kódolták. Ez akkor a célnak megfelelő volt, kiszolgálta az igényeket. Később azonban szinte robbanásszerűvé vált az internet elterjedése, ezáltal az igények is megnöttek. Egyrészt divatosabb, szebb oldalakat kezdtek el készíteni, és ezzel akár a látogatószámot is növelni lehetett, másrészt szükségszerűvé vált, hogy az oldalak többnyire ne csak statikusan működjenek, hanem legyenek dinamikusak, személyre szabottan kommunikáljanak a felhasználóval, aki műveleteket hajthasson végre az oldalon. Ezen igény kielégítésére különböző programnyelvek jöttek létre, mint például a PHP.

**A PHP** (Hypertext Preprocessor) 1995-ben jelent meg. 1994-ben Rasmus Lerdorf tervezte, de a további fejlesztést már a PHP Group végzi. De Facto szabvány, nincs hivatalos specifikációja. Dinamikus weboldalak készítéséhez használatos, nyílt forráskódú, szabad szoftver. Imperatív, objektumorientált szkriptnyelv. Széles körben alkalmazható, mivel

platform független. Mivel Lerdorfnek eredetileg néhány Perl szkript lecserélése volt a célja, ezért a PHP-ben megtalálhatók belőle bizonyos vonások, mint például a változók.

A PHP programokat általában HTML oldalba építve futtatják, ritkán HTML nélkül, parancssoros programként is alkalmazzák. A következő három területen alkalmazzák legfőbbképpen a PHP-t:

- *Szerver oldali programozás:* ez a leghagyományosabb és legfőbb területe a PHP alkalmazásának. Itt három fő dologra lesz szükségünk: egy PHP elemzőre / értelmezőre, egy web szerverre, és egy böngészőre. A web szervert egy megfelelően beállított PHP-vel kell futtatnunk. A szerver által és a böngésző segítségével tekinthetjük meg a PHP program eredményét, kimenetét. Mindezt elvégezhetjük az otthoni gépünkön is.
- *Parancssoros programozás:* szerver és böngésző nélkül is készíthetünk így PHP alkalmazásokat, mindössze a PHP értelmezőre lesz szükség. Ez hasznos lehet például feladatok ütemezésére, vagy akár egyszerű szövegfeldolgozásra.
- *„Grafikus” alkalmazások:* itt a grafikus szó azt jeltené, hogy a program rendelkezik egy grafikus felhasználói felülettel. Habár a PHP nem a legmegfelelőbb nyelv ilyen alkalmazások készítésére, de ha már elég jól ismerjük a nyelvet, és használni szeretnénk a PHP fejlettebb képességeit a kliens-oldali alkalmazásainkban, akkor a PHP-GTK lesz segítségünkre.

A PHP további területeken is széleskörű választási lehetőségeket biztosít:

- *Operációs rendszerek:* a PHP a legtöbb, illetve a legelterjedtebb operációs rendszereken működtethető, mint például: Linux, sok Unix változaton (HP-UX, Solaris, OpenBSD), Microsoft Windows, Mac OS X, RISC OS, és még valószínűleg sok máson is.
- *Web szerverek:* ma a PHP már a legtöbb web szervert támogatja, mint például: Apache, Microsoft Internet Information Server, Personal Web Server, Netscape és iPlanet szerverek, Oreilly Website Pro, Caudium, Xitami, OmniHTTPd, és még sok mást is. A PHP CGI feldolgozóként is képes működni.

PHP programozáshoz készítettek különböző fejlesztői eszközöket, melyek megkönnyítik a programozó munkáját, de PHP program megírható a legegyszerűbb szövegszerkesztővel is.

Egy egyszerű példa a HTML-be ágyazott PHP-ről:

```
<html>

    <head></head>

    <body>

        <?php

            echo "Helló Web!";

        ?>

    </body>

</html>
```

Tehát PHP kódot HTML kódban bárhol elhelyezhetünk. PHP kód kezdetét a következőképpen jelezhetjük: <?php illetve <? , PHP kód végét pedig ?> -el.

A PHP lehetőséget biztosít arra, hogy adatbázissal kapcsolatot hozzunk létre. PHP programok esetében elmondható, hogy leginkább a MySQL adatbázis-kezelőt szokták használni.

**A MySQL** egy többfelhasználós, többszálú, SQL-alapú relációs adatbázis-kezelő szerver. Elterjedésének legfőbb okai közt szerepel a könnyű kezelhetőség, beállíthatóság, valamint nyílt forráskódú. A MySQL adatbázisok adminisztrációja történhet parancssorosán, de létezik rá más eszköz is, mint például a PHP nyelven írt phpMyAdmin, mely szintén nyílt forráskódú. A MySQL a platformok nagyon széles körén futtatható. Ma a SUN tulajdona.

Példa MySQL adatbázishoz való kapcsolódásra PHP-ből:

```
$kapcsolat=mysql_connect("host", "felhasznalonev", "jelszo");
```

```

if ( ! $kapcsolat )

    die( "Nem lehet kapcsolodni a MySQL kiszolgálóhoz!" );

mysql_select_db("adatbazis")

    or die ( "Nem lehet megnyitni a(z) 'adatbazis'
    adatbazist " .mysql_error() );

```

Meghívjuk a `mysql_connect()` beépített függvényt, melynek paraméterként megadjuk a host-ot, a csatlakozáshoz szükséges felhasználónevet, és hozzá tartozó jelszót. A függvény eredménye a `$kapcsolat` változóba kerül. Amennyiben nem sikerül a kapcsolódás, a `die` leállítja a program futását, miután kiírta a paraméternek megadott karaktersorozatot. Ha sikerült a kapcsolatot létrehozni, akkor a `mysql_select_db()` függvényt hívjuk meg, melynek paraméternek megadjuk annak az adatbázisnak a nevét, mellyel dolgozni szeretnénk. Ha nem sikerült az adatbázis kiválasztása, akkor a `die` rész fut le az előbb leírt módon, különben a `$kapcsolat` változó segítségével ezután elérhetjük és módosíthatjuk az adatbázist.

### **Előrevetítés az alkalmazásra és a szakdolgozatra:**

Mivel napjainkra jelentősen megugrott a dinamikus oldalak, web alapú alkalmazások száma, és ez még növekvő tendenciát is mutat, ezért gondoltam arra, hogy megismerkedek a PHP programnyelvvvel. Készítettem egy web alapú alkalmazást, mely részben az időjárással, meteorológiával kapcsolatos. Szabadidőmben szeretek az időjárással foglalkozni, azért esett a választásom erre a témakörre. Az alkalmazás három fő részből áll: található benne fórum, időjárást megfigyelő oldal, valamint online bolt (webshop).

A szakdolgozat további részében szó lesz az apache-ról, valamint az általam készített alkalmazást fogom részletesen bemutatni. A külső szemlélet mellett bemutattva annak technikai, technológiai hátterét, az alkalmazott algoritmust.

## **Az alkalmazás**

### **Az alkalmazásról röviden:**

Az alkalmazás egy időjárással, meteorológiával kapcsolatos weboldal, mely az „IdőjárásCentrum” nevet viseli. Az oldal célja, hogy az emberekkel megkedveltesse a meteorológiát, illetve az érdeklődők számára hasznos információkat nyújtson. Erre szolgál a megfigyelő oldal valamint a fórum. Továbbá az online boltban megvásárolhatják azokat az eszközöket, melyekkel pontos méréseket, megfigyeléseket végezhetnek, és egyéb kiegészítők közül válogathatnak.

Az oldal mindkét fél számára hasznos. Egyik oldalról hasznos a felhasználónak, mert kikapcsolódást jelenthet számára, bővítheti ismereteit, társaloghat emberekkel, segítséget kap az elkövetkezendő rövidebb időszak időjárására vonatkozóan, ezáltal tervezheti meg akár szabadidejét, és ha tovább szeretne lépni környezetének megfigyelésében, megtalálhatja és megvásárolhatja a számára legmegfelelőbb eszközt az online boltból. Másik oldalról hasznos az oldal üzemeltetőjének, mert a plusz funkciókkal (fórum és megfigyelő oldal) több látogatóra számíthat, ezáltal növekedhet a forgalma a webshop-ban.

Az alkalmazás készítésekor a PHP és MySQL 5-ös verzióját használtam, valamint szükség volt még egy web-szerverre (Apache).

### **Az Apache-ről:**

Ha egy PHP kódot, illetve egy PHP kódot tartalmazó oldalt szeretnénk megtekinteni, akkor a böngészőnk nem kapja meg egyből az oldal forráskódját, hanem azt a kódot először egy kiszolgáló illetve egy szerver feldolgozza azt. A forrásban található HTML kódok változatlanok maradnak, de a PHP kód nem kerül átküldésre, hanem azt feldolgozva, annak az eredménye HTML kódban lesz továbbküldve a kliensnek, tehát a kliens nem látja az eredeti forráskódot.



PHP tartalmú oldal készítése során a programozónak szüksége van arra, hogy bizonyos időközönként megtekintse addigi munkája eredményét, de ehhez szüksége van egy olyan eszközre, mely feldolgozza PHP kódját. Ekkor a programozó saját számítógépére is telepíthet ilyen eszközt, egy web-szervert. A legelterjedtebb web-szerver az Apache.

Az Apache nyílt forráskódú, szabad szoftver. C programozási nyelvben íródott, platformfüggetlen webkiszolgáló alkalmazás.

### **Az alkalmazás futtatása:**

Az alkalmazás futtatásához 9 darab adatbázis-táblára van szükség:

- aktualis\_ido
- egyeni\_efore
- tarsalgo
- webshop\_forum
- felhasználok
- eszlesek
- fooldal
- termek
- megrendesek

Ezek a táblák rendre a következőképpen épülnek fel:

- 'aktualis\_ido', 'egyeni\_efore' és 'tarsalgo' táblák:
  - *nev* VARCHAR(99)
  - *hozzaszolas* VARCHAR(999)
  - *datum* DATETIME
  - *idobelyeg* LONG
  - *kulcs* INT NOT NULL AUTO\_INCREMENT, PRIMARY KEY
- 'webshop\_forum' tábla:
  - *nev* VARCHAR(99)
  - *hozzaszolas* VARCHAR(999)
  - *datum* DATETIME

- *idobelyeg* LONG
- *id* LONG
- *kulcs* INT NOT NULL AUTO\_INCREMENT, PRIMARY KEY
- 'felhasznalok' tábla:
  - *azonosito* VARCHAR(99)
  - *loginjelszo* VARCHAR(99)
  - *telepules* VARCHAR(99)
  - *eszlelohely* INT
  - *kinezet* VARCHAR(99)
  - *regido* DATETIME
  - *belido* DATETIME
  - *bemutakozas* VARCHAR(99)
- 'eszlelesek' tábla:
  - *azonosito* VARCHAR(99)
  - *telepules* VARCHAR(99)
  - *eszlelohely* INT
  - *felho* INT
  - *datum* DATETIME
  - *idobelyeg* LONG
  - *komment* VARCHAR(999)
  - *homerseklet* INT
  - *kulcs* INT NOT NULL AUTO\_INCREMENT, PRIMARY KEY
- 'fooldal' tábla:
  - *cim* VARCHAR(99)
  - *leiras* VARCHAR(999)
  - *idobelyeg* LONG
- 'termekek' tábla:
  - *id* LONG
  - *nev* VARCHAR(99)
  - *tipus* VARCHAR(99)
  - *leiras* VARCHAR(999)
  - *garancia* INT

- *ar* INT
- *kep* VARCHAR(99)
- 'megrendelesek' tábla:
  - *id* LONG
  - *nev* VARCHAR(99)
  - *tipus* VARCHAR(99)
  - *leiras* VARCHAR(999)
  - *garancia* INT
  - *ar* INT
  - *kep* VARCHAR(99)
  - *megrendelo* VARCHAR(99)
  - *kosar\_id* LONG
  - *leadva* LONG
  - *fizetve* INT
  - *kezbesitve* INT
  - *darab\_termek* INT
  - *cim* VARCHAR(199)

Az alkalmazás összesen 16 darab PHP fájlból tevődik össze, és ezek mellett szerepel még egy PHP fájl, melynek `tables_create.php` a neve, melyet ha lefuttatunk, akkor eldobja (törli) az előbb felsorolt nevű táblákat, majd a megfelelő adatszerkezettel létrehozza azokat. Ebben a fájlban található még olyan kódsorozat is, mely a webshopot tölti fel példatermékekkel. Ezen műveletek előtt készítenünk kell egy adatbázist, melynek be kell állítani a csatlakozáshoz szükséges felhasználónevet és jelszót. Ezután a szükséges adatokat meg kell adni a `connect.php` fájlban. Ebben a fájlban található egy `$felhasznalo` változó, melynek értékül meg kell adni az adatbázishoz való csatlakozáshoz szükséges felhasználónevet. A `$jelszo` változónak értékül kell adni az adatbázishoz való csatlakozáshoz szükséges jelszót. A `$adatbazis` változónak értékül kell adni annak az adatbázisnak a nevét, amelyet használni szeretnénk. A `$kapcsolat` változó értékadásánál szereplő `mysql_connect()` függvény első paraméterénél kell megadni a `host`-ot (mely

helyi gép esetén maradhat localhost). Az alkalmazás a <http://martinus333.fw.hu> címen érhető el.

Az alkalmazás készítése folyamán minden PHP fájl a `connect.php` fájl alapján próbál kapcsolatot létesíteni az adatbázissal. Ennek a módszernek az előnye, hogy az esetlegesen bekövetkező bármilyen adatmódosulás esetén, mely az adatbázissal való kapcsolatteremtéshez szükséges, csak egyetlenegy fájlban kell módosítani, ezáltal csökkenthető a karbantartás ideje, és sok hibalehetőséget el tudunk ezáltal kerülni.

### **Az alkalmazás bemutatása, részenként:**

#### *Nyitóoldal*

Az alkalmazás böngészőből történő megnyitása egyet jelent az `index.php` indításával. Tehát indíthatjuk úgy, hogy az elérési útvonal után megadjuk az `index.php`-t, de ha csak magában az elérési útvonalat adjuk meg, alapértelmezésként mindig az `index.php` fog elindulni.

Ha megtekintjük az `index.php` kódját, azt láthatjuk, hogy a következőképpen kezdődik:

```
<?php  
  
    session_start();  
  
?>
```

Ennek a résznek mindig a kód legelején kell szerepelnie, ha már valamilyen kód után helyezük el, hibához vezet.

A `session_start()`; függvénnyel elindíthatunk egy munkamenetet, és kapunk egy munkamenet azonosítót. Ha az oldalt újratöltenénk (frissítenénk), akkor ugyanazt az azonosítót kapnánk. A működéshez engedélyezni kell a böngészőben a süti használatát.

Munkamenetre azért van szükség, mert a HTML esetében, ha egyik oldalról átkerülünk egy másik oldalra, semmilyen információt nem tudunk átvinni, ugyanis a HTML állapotmentes. Ha azonban munkamenetet indítottunk, akkor egy oldalon olyan információkat lehet kiolvasni, amiket egy előző, vagy akár az azt megelőző oldalon állítottunk be, adtunk meg. Ehhez viszont minden oldalon el kell indítani a munkamenetet. Így az alkalmazásom minden egyes PHP oldalának kezdetén megtalálható a munkamenet indítása.

Az alkalmazás megnyitásakor az index oldal töltődik be, melyet alább láthatunk:

The screenshot shows the homepage of 'Időjárás Centrum'. At the top right, there is a login section with the text 'Bejelentkezés', an 'Azonosító:' input field, a 'Jelszó:' input field, and buttons for 'Regisztráció' and 'Belépés'. The main content area is divided into three columns:

- FÓRUM:** A fórumban megtekintheti regisztrált felhasználóink eszmecseréjét a jelenlegi, illetve a közeljövő időjárásáról, vagy kötetlen társalgásukat. Egy regisztrációt/bejelentkezést követően Ön is hozzászólhat az adott témákhoz!
- MEGFIGYELÉSEK:** A megfigyelő oldalon vizuálisan megtekintheti Magyarország aktuális (elmúlt 30 perc) időjárását, elolvashatja a megfigyelők meglátását, valamint a megfigyelő nevére kattintva információkat tudhat meg róla. Regisztrációt/bejelentkezést követően Ön is küldhet megfigyeléseket!
- WEBSHOP:** A webshopban lehetősége van különböző meteorológiai eszközök, műszerek vásárlására, melyek segítik a mindennapokban tájékoztatni Önt az időjárás paramétereiről. Különböző kiegészítők közül is válogathat. Leadott megrendeléseit a leadást követő első 8:00-ig módosíthatja.

Below each column, there are links to news items: 'hír 5' (hír 5 törzs 5), 'hír 4' (hír 4 törzs 4), and 'hír 3' (hír 3 törzs 3). On the left side, there is a navigation menu with links for 'Főoldal', ' Fórum', 'Megfigyelő oldal', and 'WebShop'.

Az oldalt három fő részre tagoltam:

- Főcím
- Menü
- Tartalom

A tagolást HTML keretek, FRAME-ek segítségével oldottam meg. A felső (főcím) és a bal oldali (menü) keretekben mindig ugyanaz a PHP fájl van megnyitva (focim.php és menu.php), a harmadik keretben viszont kezdéskor a tartalom.php fájl, utána a felhasználó utasításának megfelelő PHP fájl töltődik be.

Tehát az index oldal csak a keretek definiálását tartalmazza. Végezetül még egy <NOFRAMES> HTML tag is található, melynek abban az esetben van szerepe, ha a böngésző nem támogatja a kereteket. Ez esetben egy erre utaló üzenet jelenik meg, és nem töltődik be az oldal.

## Menü

A menü négy hivatkozást tartalmaz:

- Főoldal
- Fórum
- Megfigyelő oldal
- WebShop

Ezen hivatkozásra kattintva nyithatjuk meg sorra a:

- tartalom.php
- forum.php
- megfigyelo.php
- webshop.php

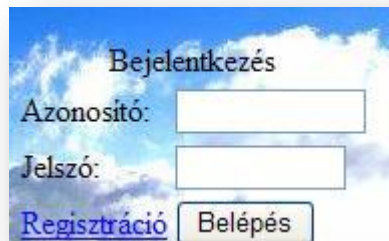
oldalakat, melyek a 'tartalom' keretbe töltődnek be. A menu.php fájlban definiálva van, hogy a megjelenés a 'stiluslap.css'-ben megadottak szerint történjen. Eszerint a linkek szövegei kék színűek, ha az egérmutató felettük áll, pirosak, és ha megelőzőleg rá lett kattintva, akkor zöld színűek. A CSS fájlok segítségével átláthatóbbá tesszük a kódot úgy, hogy nem ott tároljuk le a megjelenésre / stílusra vonatkozó információkat, hanem egy külön fájlba. Mindemellett a fejlesztést is gyorsítja azáltal, hogy ha egyszer definiáltunk egy fajta megjelenést egy dologhoz, azt megadhatjuk egy másik dologhoz is.

## Főcím

A `focim.php` tartalmaz egy háttérképet melyre ha rákattintunk, a 'tartalom' keretbe betöltődik a `tartalom.php` fájl. Ezt egy `onclick` eseményhez megadott java szkripttel valósítom meg:

```
onclick = "javascript:window.open('tartalom.php','tartalom')"
```

A lényegi rész a főcím jobb oldalán helyezkedik el:



Itt van a felhasználónak lehetősége a rendszerbe történő bejelentkezésre, kijelentkezésre, illetve regisztráció indítására. Mivel a `focim.php` fájl mindig betöltve marad a fenti keretrészben, ezért a felhasználó az alkalmazás bármelyik részén is járjon, ezt a részt mindig látni fogja.

### Bejelentkezés:

Minden olyan felhasználó be tud jelentkezni, aki már regisztrált az oldalon. Sikeres bejelentkezéshez meg kell adnia a felhasználónevét (azonosítóját) valamint a hozzá tartozó jelszót. Ezután a 'Belépés' gombra kattintva elküldi az előbb megadott adatokat. Az adatok elküldése a `$_POST` változó segítségével történik. Hasonló még hozzá a `$_GET` változó. Ezek használata például a következő:

Létre kell hozni egy HTML FORM-ot. Ez a `<form>` és `</form>` tagekkel alakítható ki. A `<form>` tag-ben megadható az `action=` után annak az oldalnak a neve, ahová a FORM végeztével ugrani szeretnénk, valamint megadható (`method=`) hogy POST-ot vagy GET-et szeretnénk-e használni:

```

<form name="login" action="focim.php" method="POST">

    <input type="text" name="azonosító" value="" size="10" />

    <input type="password" name="jelszó" value="" size="10" />

    <input type="submit" value="Belépés" name="belépés" />

</form>

```

Azon az oldalon, ahová ugrottunk, a `$_POST` illetve a `$_GET` változókból olvashatóak ki az előző oldalon beleírt értékek. Mindkét változó asszociatív tömbként viselkedik, így az alkalmazásban a bejelentkezéskor az azonosító és jelszó megadását követően az adatokat a következőképpen érhetjük el:

- Azonosító : `$_POST['azonosító']`
- Jelszó : `$_POST['jelszó']`

Ha a `$_POST['azonosító']` -nak van nem üres értéke, akkor kapcsolódás történik az adatbázishoz:

```

include("connect.php");

$kapcsolat=connect();

```

Ezután a `$kapcsolat` változó segítségével érhetjük el az adatbázist.

Ha sikeres volt az adatbázishoz való csatlakozás, akkor a megadott azonosítóhoz tartozó értékek lekérdezésre kerülnek a 'felhasznalok' táblából. Például a megadott azonosítóhoz tartozó jelszó lekérdezése:

```

$tmp=$_POST['azonosító'];

$parancs="SELECT loginjelszo FROM felhasznalok WHERE
azonosito='$tmp'";

$eredmeny = mysql_query( $parancs, $kapcsolat) or die ( "Nem
sikerült a bejelentkezés: " .mysql_error() );

```



A `$parancs` változóban letárolom az SQL utasítást, majd azt végrehajtom a `$kapcsolat` változó által definiált adatbázison a `mysql_query()` függvény segítségével. Amennyiben a függvény futása nem volt sikeres, félbeszakad a program futása, melyet hibaüzenet kísér. Ha sikeres volt a futás, akkor a `$eredmeny` változó fogja tárolni a lekérdezés eredményét, melyből a következőképpen olvashatjuk ki az információkat:

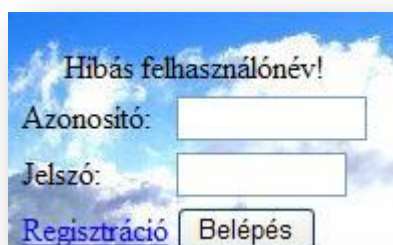
```
$egy_sor = mysql_fetch_row( $eredmeny );
```

Tehát a `$egy_sor` nevű változóba megkaptuk a megadott azonosítóhoz tartozó jelszót az adatbázisból. Ezután összehasonlíthatjuk ezt a bejelentkezési űrlapon megadott jelszóval. Ha megegyeznek (és a felhasználó nincs kizárva), az azonosítót értékül adjuk a munkamenet szuper-globális változójának, a `$_SESSION` változónak:

```
$_SESSION['felhasznalonev']=$_POST['azonosító'];
```

Ha az azonosítóhoz tartozó jelszó lekérdezésekor nem kaptunk vissza pontosan egy darab eredményt, akkor egy hibakód kerül beállításra, melynek értéke 1 lesz, mely arra enged következtetni, hogy a megadott azonosító nem megfelelő. Ha a bejelentkezéskor megadott azonosítóhoz pontosan egy darab jelszó létezik, de az nem egyezik meg a megadott jelszóval, akkor a hibakód 2 lesz. Ha az előző három esetből egy sem következett be, akkor az valamilyen egyéb hibára utal, így a hibakód 3 lesz. Végezetül, ha a felhasználó ki van zárva (a jelszó egy speciális karaktersorozat), akkor a hibakód 4 lesz. Ha a bejelentkezés sikeres volt, a hibakód nulla marad.

Ezt követően kerülnek kezelésre a hibakódok. Ha a hibakód értéke 1-el egyenlő, akkor csak annyi történik, hogy kiíratjuk, hogy a megadott azonosítóval probléma van:



Ha a hibakód értéke 2-vel egyenlő, akkor csak annyi történik, hogy kiíratjuk, hogy a megadott azonosítóhoz tartozó jelszó hibás:



Hibás jelszó!

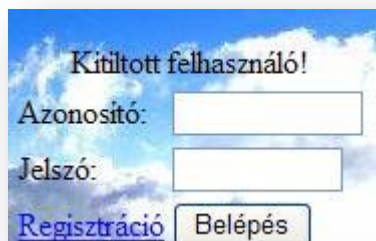
Azonosító:

Jelszó:

[Regisztráció](#)

Ha a hibakód értéke 3-al egyenlő, akkor csak annyi történik, hogy kiíratjuk, hogy hiba történt.

Ha a hibakód értéke 4-el egyenlő, akkor csak annyi történik, hogy kiíratjuk, hogy az adott felhasználó ki van tiltva az oldalról:



Kiltott felhasználó!

Azonosító:

Jelszó:

[Regisztráció](#)

Sikeres bejelentkezéskor még az alkalmazás adatbázisba menti, hogy az adott időpontban a felhasználó sikeresen belépett az oldalra.

Ha nem történt semmilyen hiba, a bejelentkezés után a következőt láthatja a bejelentkezett felhasználó:



Belépve: [martinus](#)

**Kijelentkezés:** Kiírásra kerül a bejelentkezett felhasználó azonosítója, valamint megjelenik egy gomb, melyre kattintva kijelentkezhet az oldalról. Ebben az űrlapban el van helyezve egy rejtett mező:

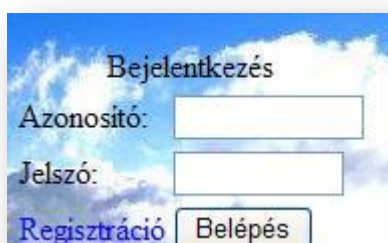
```
<input type="hidden" name="logout" value="1" />
```

Ennek a rejtett mezőnek az a szerepe, hogy ha a felhasználó rákattint a kijelentkezés gombra, akkor az űrlap elküldi a `$_POST` változóban, hogy a `logout` értéke 1-el egyenlő. Az adott oldal ezt figyeli, és ha a `logout` értéke egyenlő 1-el, akkor a super-globális változóból törli a felhasználó nevét, ezáltal lehet tudni, hogy ezután nincs bejelentkezett felhasználó. Tehát a kijelentkezés gombra kattintva visszkapjuk az alap bejelentkezési képet, és a felhasználó bejelentkezett státusza megszűnik.

Az előző képen is látható, hogy ha a felhasználó sikeresen bejelentkezett, akkor kiírásra kerül az azonosítója. Ez a kiírt azonosító egyben egy hivatkozás is, melyre kattintva a 'tartalom' keretbe betöltődik a `profil.php` fájl. Ennek a segítségével a felhasználó megtekintheti a profilját, adatlapját, illetve lehetősége van az ott megadott adatokat módosítani. Mivel a lap hasonló a regisztrációhoz, ezért a regisztráció tárgyalása után fogok visszatérni a profilmódosításhoz.

### **Regisztráció:**

A bejelentkezési résznél található egy Regisztráció nevű hivatkozás:



Bejelentkezés

Azonosító:

Jelszó:

[Regisztráció](#)

Az oldalra látogató személynek lehetősége van az oldalon saját profilt létrehozni a regisztráció segítségével. Miután a felhasználó már egyszer sikeresen regisztrált, utána lesz lehetősége az alkalmazásba történő bejelentkezésre. A bejelentkezésre pedig azért lehet szüksége, mert az alkalmazásban bizonyos funkciókat csak bejelentkezett felhasználó vehet igénybe.

A regisztrációra kattintva a `regist.php` fájl töltődik be a 'tartalom' keretbe. A `regist.php` fájl:

Felhasználónév:

Jelszó:


Jelszó:  (A két beírt jelszónak meg kell egyeznie.)

Település:

Észlelési hely: 0 

1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26
27	28	29	30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49	50	51	52
53	54	55	56	57	58	59	60	61	62	63	64	65
66	67	68	69	70	71	72	73	74	75	76	77	78
79	80	81	82	83	84	85	86	87	88	89	90	91
92	93	94	95	96	97	98	99	100	101	102	103	104
105	106	107	108	109	110	111	112	113	114	115	116	117

Bemutakozás:

Oldal megjelenési stílusa: classic 

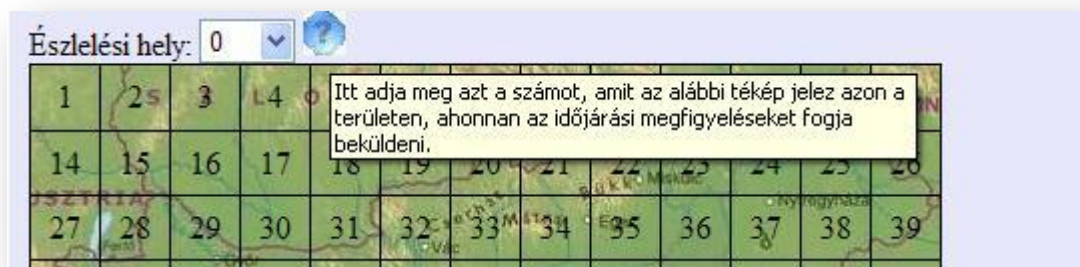
Kijelentem, hogy jelen weboldalt rendeltetésszerűen használom, abban semmiféle kárt nem okozok, és semmilyen részét nem használom fel más célra, más weboldalhoz. Megfigyelések beküldésekor valós adatokat közlök, ellenkező esetben regisztrációm törlésre kerülhet.

Igen, elfogadom.

A regisztrációs űrlapon a regisztrálónak meg kell adnia egy felhasználónevet. Ez lesz a regisztrációt követően a felhasználóneve, az azonosítója, mellyel majd be tud jelentkezni az oldalra, és ez fog megjelenni az oldal azon részein, ahol valamilyen tevékenységet végzett.

Ezt követően meg kell adnia egy jelszót. Ezen jelszóval, és az előbb megadott felhasználónév együttesével tud majd a felhasználó bejelentkezni az oldalra. A regisztrációs űrlapon a választott jelszót kétszer kell megadnia. Erre azért van szükség, hogy ha kiválasztott egy jelszót, de azt véletlenül rosszul gépelte be, akkor ez a hiba kiderüljön azáltal, hogy még egyszer meg kell adnia ugyanazt a jelszót (ugyanis sokkal kisebb az esélye annak, hogy valaki kétszer ugyanúgy elgépeljen valamit). Így el lehet kerülni azt a kellemetlenséget, hogy sikeres regisztrációt követően a felhasználó ne tudjon bejelentkezni az oldalra, mert tudtán kívül más jelszót adott meg.

Továbbá a felhasználónak meg kell adnia egy településnevet. Azt a települést kell megadnia, ahol általában tartózkodik a felhasználó, illetve ahonnan majd az időjárással kapcsolatos megfigyeléseit szeretné majd beküldeni. Emellett ki kell választania egy észlelési helyet. Ezt egy legördülő listával teheti meg. Itt meg kell adnia egy számot. Ennek a számnak meg kell egyeznie a legördülő lista alatt található térképen elhelyezett számok egyikével, mégpedig azzal, amelyik a térkép azon felületét fedi le, ahonnan a felhasználó az időjárással kapcsolatos megfigyeléseit szeretné majd beküldeni. Tehát például egy budapesti személynek a 45-ös számot kell megadnia. Az oldal segítséget nyújt a felhasználónak, elmagyarázza, hogy mit jelent pontosan az, hogy egy számmal kell megadni az észlelési helyet, még hozzá úgy, hogy a legördülő lista mellett el van helyezve egy kis kép, mely egy kérdőjelet ábrázol, melyre az egeret mozgatva jelenik meg a segítséget nyújtó szöveg:



Ezt követően a felhasználónak lehetősége van megadni (opcionális) egy bemutatkozó szöveget, melyben tetszőleges információt közölhet magáról. Ez a szöveg a megfigyelő oldalról lesz elérhető, ahol ha megfigyelést küld, akkor a megfigyelése mellett megjelenik a neve, és erre kattintva egyéb információk mellett meg fog majd jelenni ez a bemutatkozó szöveg is.

A felhasználónak ki kell választania az oldal megjelenési stílusát. Jelenleg két elérhető stílus van:

- Classic
- Modern

Ezek rendre a következő két stílusdefiníciós fájlt definiálják:

- classic.css
- modern.css

Ez azért lényeges, mert a két stílus meglehetősen különböző kinézetet biztosít az oldalnak. Ha a felhasználó a regisztrációkor kiválasztja (vagy később a profiljában módosítja) valamelyik stílust, akkor ha bejelentkezik, utána az általa kiválasztott stílusnak megfelelően fog az oldal pompázni a tartalom keretben. Ezáltal többen is magukénak érezhetik az oldalt. Akik a hagyományos, jól megszokott színeket kedvelik, minden bizonnyal a `classic` beállítás mellett fognak dönteni, de akiknek már túl megszokottá, esetleg unalmassá váltak már a klasszikus színek, beállíthatják a `modern` stílust, ezáltal jobban is érezhetik magukat az oldalon. Mindenki kedvére választhat, hogy hogyan jelenjenek meg az oldalak, melyik beállítás mellett érzi jobban magát. Az alkalmazás alapértelmezésként a klasszikus stílust használja, így ha egy nem bejelentkezett felhasználó használja, akkor a `classic.css` fájl szerint jelennek meg az oldalak. A helyes stíluslap alkalmazása a következőképpen történik:

Az adatbázis '`felhasznalok`' táblájában minden egyes felhasználónak el van mentve az általa kiválasztott stílus neve. Minden egyes PHP fájl legelején meg van adva, hogy az az oldal egy külső fájl alapján kezelje az oldal kinézetét:

```
<link rel="stylesheet" type="text/css" href="">
```

ahol a `href` részben kell megadni a fájl (`css` fájl) nevét. Ez előtt egy PHP kódrésszel megvizsgálom, hogy éppen van-e felhasználó bejelentkezve, tehát a `$_SESSION['felhasznalonev']`-t vizsgálom. Ha ez üres, akkor nem történik semmi, különben lekérdezi az adott felhasználónévhez tartozó elmentett stílus nevét az adatbázisból.

A `href` részben az eredetileg megadott '`.css`' karakterek elé egy PHP kódrésszel kiíratom a lekérdezett stílus nevét, így teljes lesz a külső stíluslapra történő hivatkozás. Természetesen csak akkor, ha történt lekérdezés (tehát van bejelentkezett felhasználó). Ez a PHP kódrész tartalmaz egy olyan részt is, mely akkor hajtódik végre, ha nem történt lekérdezés (tehát nincs bejelentkezett felhasználó). Ebben az esetben ez a kódrész a '`classic`'-ot írja a '`.css`' elé. Így érhetjük el, hogy a bejelentkezett felhasználó által kiválasztott stílus jelenjen meg, illetve ha nincs bejelentkezve a felhasználó, akkor alapértelmezésként a klasszikus stílust használjuk.

Példa a klasszikus stílusra:



<u>FÓRUM</u>	<u>MEGFIGYELÉSEK</u>	<u>WEBSHOP</u>
A fórumban megtekintheti regisztrált felhasználóink eszmecseréjét a jelenlegi, illetve a közeljövő időjárásáról, vagy kötetlen társalgásukat. Egy regisztrációt/bejelentkezést követően Ön is hozzászólhat az adott témákhoz!	A megfigyelő oldalon vizuálisan megtekintheti Magyarország aktuális (elmúlt 30 perc) időjárását, elolvashatja a megfigyelők meglátását, valamint a megfigyelő nevére kattintva információkat tudhat meg róla. Regisztrációt/bejelentkezést követően Ön is küldhet megfigyeléseket!	A webshopban lehetősége van különböző meteorológiai eszközök, műszerek vásárlására, melyek segítik a mindennapokban tájékoztatni Önt az időjárás paramétereiről. Különbféle kiegészítők közül is válogathat. Leadott megrendeléseit a leadást követő első 8:00-ig módosíthatja.
hír 5 hír 5 törzs 5	hír 4 hír 4 törzs 4	hír 3 hír 3 törzs 3

Példa a modern stílusra:

<u>FÓRUM</u>	<u>MEGFIGYELÉSEK</u>	<u>WEBSHOP</u>
A fórumban megtekintheti regisztrált felhasználóink eszmecseréjét a jelenlegi, illetve a közeljövő időjárásáról, vagy kötetlen társalgásukat. Egy regisztrációt/bejelentkezést követően Ön is hozzászólhat az adott témákhoz!	A megfigyelő oldalon vizuálisan megtekintheti Magyarország aktuális (elmúlt 30 perc) időjárását, elolvashatja a megfigyelők meglátását, valamint a megfigyelő nevére kattintva információkat tudhat meg róla. Regisztrációt/bejelentkezést követően Ön is küldhet megfigyeléseket!	A webshopban lehetősége van különböző meteorológiai eszközök, műszerek vásárlására, melyek segítik a mindennapokban tájékoztatni Önt az időjárás paramétereiről. Különbféle kiegészítők közül is válogathat. Leadott megrendeléseit a leadást követő első 8:00-ig módosíthatja.
hír 5 hír 5 törzs 5	hír 4 hír 4 törzs 4	hír 3 hír 3 törzs 3

(A szakdolgozat további részében szereplő képek a klasszikus stílusnak megfelelőek lesznek.)

És végezetül található egy rövid szabályzat, melyet el kell fogadnia a felhasználónak, ha az oldal tagja szeretne lenni. Az elfogadást úgy teheti meg, hogy a szabályzat után bejelöli a jelölőnégyzetet az „Igen, elfogadom” szöveg előtt.

Ha végzett a felhasználó az adatok megadásával, akkor a 'Regisztrál' gombra kattintva véglegesítheti regisztrációját. Ekkor nem történik más, mint a megadott adatok \$\_POST változóba történő írása és átadása a 'regist\_2.php' számára, ugyanis ez az oldal fogja kiértékelni a bevitt adatokat, és véghezvinni a szükséges lépéseket, műveleteket. Kiolvassa a \$\_POST azon tartalmát, amire szükségünk lesz a regisztráció végrehajtásához, tehát az előző oldalon megadott adatokat. Ahhoz, hogy a regisztráció sikeres legyen, tehát a felhasználó adatai bekerüljenek az adatbázisba, az oldal a következőket ellenőrzi előtte:

- *felhasználónév*: a megadott felhasználónév akkor elfogadható, ha
  - o nem üres
  - o legalább 4, legfeljebb 18 karakter hosszúságú
  - o még nem szerepel ilyen az adatbázis, felhasználók nevű táblájában
- *jelszó*: a megadott jelszó akkor elfogadható, ha
  - o nem üres
  - o legalább 4, legfeljebb 18 karakter hosszúságú
  - o a két megadott jelszó ugyanaz, karakterről karakterre megegyeznek
- *település*: a településnek nem szabad üresnek lennie
- *észlelési hely*: az észlelési helynek nem szabad nullának lennie
- *kijelentés*: a kijelentésnek el kell lennie fogadva

Amennyiben ezeknek a feltételeknek mindegyike teljesül, a megadott adatok mindegyike elmentésre kerül az adatbázisba, és megjelenik a 'Sikeres regisztráció' szöveg, utalva arra, hogy a regisztráció folyamán minden rendben zajlott, sikeresen tagja lett az illető az oldalnak. Azonban ha nem teljesül kivétel nélkül a feltételek mindegyike, nem módosul az adatbázis, nem lesznek elmentve az adatok, hanem csak az adott feltétel teljesítetlenségére figyelmezteti a felhasználót a megfelelő szöveg kiírásával.

Például ha semmit nem adunk meg a regisztrációs űrlapon, a következő kép/üzenet fog minket fogadni:



Nem adott meg felhasználónevet!  
Nem adott meg jelszót!  
A sikeres regisztrációhoz el kell fogadnia a feltételeket!  
Nem adta meg a városa nevét!  
Nem választotta ki az időjárás megfigyelési helyét!  
A felhasználónévnek minimum 4, maximum 18 karakter hosszúnak kell lennie!  
A jelszónak minimum 4, maximum 18 karakter hosszúnak kell lennie!

Az üzenet(ek) alatt található egy 'Vissza' nevezetű gomb. Ha a felhasználó használja ezt a gombot, akkor visszakerül a regisztrációs oldalra, de a felhasználónév, jelszó, település és észlelési hely mezők már ki lesznek töltve az előzőleg megadott adatokkal. Ezeket természetesen lehet módosítani. Ezáltal kényelmesebbé lehet tenni az oldal használatát, hiszen ha valamilyen hiba történik a regisztráció közben, nem kell mindent előről kezdenie a felhasználónak.

### **Profil:**

A regisztrációs felülethez nagyban hasonlít a profil felülete, mint az már korábban említettem. Tekintsük most meg a `profil.php` és `profil_2.php` fájlokat:

`profil.php`:

Felhasználónév: martinus

Jelszó:

Jelszó:  (Akkor kell megadni ezen jelszókat, ha jelszavát módosítani szeretné. (A két jelszónak egyeznie kell!))

Település: Debrecen

Észlelési hely: 50 

1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26
27	28	29	30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49	50	51	52
53	54	55	56	57	58	59	60	61	62	63	64	65
66	67	68	69	70	71	72	73	74	75	76	77	78
79	80	81	82	83	84	85	86	87	88	89	90	91
92	93	94	95	96	97	98	99	100	101	102	103	104
105	106	107	108	109	110	111	112	113	114	115	116	117

Bemutakozás:

Oldal megjelenési stílusa: classic 

Jelszó:  A sikeres módosításhoz meg kell adnia jelszavát!

Itt a felhasználó megtekintheti, hogy éppen milyen adatok tartoznak hozzá az adatbázisban. Ezt egy olyan felületen teheti meg, mint amilyen a regisztrációs felület, csak itt a hozzá tartozó értékekkel vannak automatikusan kitöltve a mezők. Ezáltal gyorsabbá tehető a profilmódosítás, mert ha a felhasználó megtekinti adatait, csak át kell írnia a kívánt tényezőt, és a 'Módosítás' gombra kattintva készen is van. A regisztrációs felülethez képest különbség, hogy itt már nem kell még egyszer elfogadni a feltételeket, a felhasználónév ki van írva, és ezt már nem lehet módosítani, két ugyanolyan jelszót csak abban az esetben kell megadni, ha a felhasználó módosítani szeretné jelszavát, valamint a sikeres profilmódosítás érdekében meg kell adnia jelenlegi jelszavát.

A 'Módosítás' gombra kattintva az űrlapban szereplő minden érték a \$\_POST változóba kerül, és továbbadódik a profil\_2.php fájl számára. Ez az oldal is hasonlóképpen működik, mint a regist\_2.php, tehát ellenőrzéseket hajt végre az

űrlap adatain, és jelen űrlap adataihoz tartozó feltételek ugyanazok, mint a regisztrációs űrlap adataira vonatkozó feltételek. Ha a feltételek mindegyike kivétel nélkül teljesül, az adott felhasználóhoz tartozó bejegyzések a 'felhasznalo' táblában felülírásra kerülnek, majd ha ez sikerült, egy erre vonatkozó üzenet, a 'Sikeres módosítás!' fogja tájékoztatni a felhasználót arról, hogy adatait sikeresen megváltoztatta. Amennyiben nem teljesült minden egyes feltétel, a nem teljesülő feltételekhez tartozó hibaüzenetek jelennek meg, és semmilyen érték nem lesz felülírva az adatbázisban. Itt is szerepel egy 'Vissza' gomb, de itt ha ezt használjuk, az eredeti, az adatbázisban szereplő értékek fognak megjelenni.

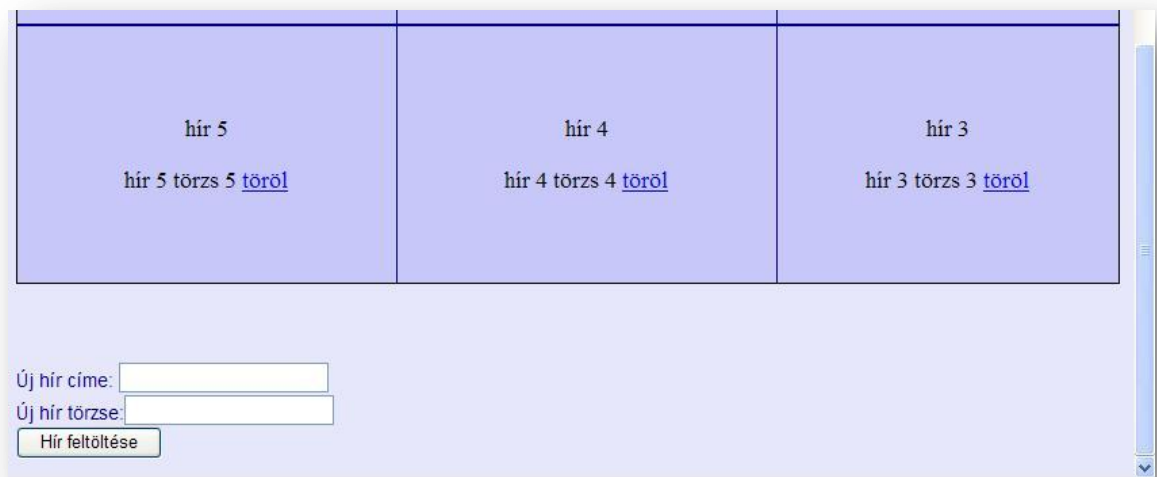
## Főoldal

<u>FÓRUM</u>	<u>MEGFIGYELÉSEK</u>	<u>WEBSHOP</u>
<p>A fórumban megtekintheti regisztrált felhasználóink eszmecseréjét a jelenlegi, illetve a közeljövő időjárásáról, vagy kötetlen társalgásukat. Egy regisztrációt/bejelentkezést követően Ön is hozzászólhat az adott témához!</p>	<p>A megfigyelő oldalon vizuálisan megtekintheti Magyarország aktuális (elmúlt 30 perc) időjárását, elolvashatja a megfigyelők meglátását, valamint a megfigyelő nevére kattintva információkat tudhat meg róla. Regisztrációt/bejelentkezést követően Ön is küldhet megfigyeléseket!</p>	<p>A webshopban lehetősége van különböző meteorológiai eszközök, műszerek vásárlására, melyek segítik a mindennapokban tájékoztatni Önt az időjárás paramétereiről. Különböző kiegészítők közül is válogathat. Leadott megrendeléseit a leadást követő első 8:00-ig módosíthatja.</p>
<p>hír 5 hír 5 törzs 5</p>	<p>hír 4 hír 4 törzs 4</p>	<p>hír 3 hír 3 törzs 3</p>

A főoldal nem más, mint a 'tartalom.php'. Ez egy két soros és három oszlopos táblázat. Az első / felső sor a főoldal kivételével ugyanazokat a hivatkozásokat, linkeket tartalmazza, mint ami a menüben szerepel, de itt alattuk található egy szöveges leírás, mely röviden bemutatja az adott menüpontokon belül levő lehetőségeket.

A második / alsó sorban hírek helyezkednek el, pontosabban a három legfrissebb hír. Baloldalon a legfrissebb, onnan jobbra pedig a régebbi híreket láthatjuk. Ezen hírek

mindegyike a ' fooldal ' nevű adatbázis-táblában van letárolva. Minden hírnek van egy címe, és egy törzse / leírása, valamint az adatbázisban minden hír rendelkezik egy időbélyeggel, melynek segítségével meg lehet határozni a három legfrissebb hírt. Hírt hozzáadni, illetve törölni csak az oldal adminisztrátora tud. Az oldal adminisztrátorának az „admin” a felhasználóneve illetve azonosítója. Az alkalmazás onnan tudja, hogy az adminisztrátorral van dolga, hogy a bejelentkezett felhasználó neve az „admin”.



Mint az a fentebbi képen látható, ha az oldal adminisztrátora bejelentkezik, akkor a főoldalon a hírek mellett megjelenik egy hivatkozás (töröl), melyre ha rákattint az adminisztrátor, törlésre kerül az adatbázisból a kiválasztott hír, így a főoldalról is eltűnik.

A törlés a következőképpen kerül megadásra:

Az adatbázisból kiolvasásra kerül a három legfrissebb hír, annak minden elemével együtt (cím, törzs, időbélyeg). Ha az adminisztrátor van bejelentkezve, akkor a hír után egy hivatkozás kerül kihelyezésre ( `<a href...>` ), mely szintén a főoldalra, a ' tartalom.php ' -re mutat. A hivatkozásban plusz adatok vannak megadva, a fájl neve után egy kérdőjel megadása után van erre lehetőségünk. Ha több adatot szeretnénk megadni, ' & ' jellel kell azokat elválasztanunk. Az így megadott adatok a \$ \_GET változóba kerülnek. Tehát a hivatkozásba információt helyezünk el a törlendő hírrel kapcsolatban. Például:

```
<a href="tartalom.php?torol=<? print $hir_1[2] ?>">töröl</a>
```

Az első hírnél az előbbi hivatkozás kerül megadásra. A linkre kattintás után a `$_GET['torol']` értéke a `$hir_1[2]` kattintás előtti értéke lesz, vagyis az első hír időbélyege. A `'tartalom.php'` egyik kódrésze pedig arra az esetre van, ha a `$_GET['torol']` értéke nem üres, akkor az abban szereplő értéknek megfelelő időbélyegzőjű hírt törli az adatbázisból. Természetesen ez a kódrész tartalmaz egy olyan feltételt is, hogy az adminisztrátor legyen bejelentkezve, mert ellenkező esetben előfordulhatna, hogy esetleg egy leleményes felhasználó a címsorban kiadja a megfelelő hivatkozást, és így akár bárki törölhetne. Erre persze kicsi az esély, de így még biztonságosabb. Az alkalmazás minden részénél, ahol szükség van rá, vizsgálat történik arra vonatkozóan, hogy egy adott művelet végrehajtásánál tényleg az a felhasználó van-e bejelentkezve, akire vonatkozik a művelet.

Az adminisztrátornak lehetősége van még hírt felvenni a főoldalra, így az adatbázisba. Ehhez csak ki kell töltenie az előző képen látható, a hírek alatt megjelenő űrlapot. Itt meg kell adnia a hír címét, és a hír törzsét (leírását), majd a `'Hír feltöltése'` gombra kattintva az alkalmazás automatikusan hozzáilleszti az aktuális időbélyeget, és hozzáadja az új hírt az adatbázishoz. Ezután az megjelenik a főoldalon is.

## Fórum

A főoldalon, vagy a menüben a Fórum menüpontra kattintva juthatunk el a fórumba. Ez a `forum.php` fájl.

<a href="#">Aktuális időjárás</a>	11	Utolsó hozzászólás:
Leírás: A jelenlegi időjárási események, paraméterek leírása, tárgyalása.	hozzászólás	martinus_ : 2009-03-05 11:16:02
<a href="#">Egyéni előrejelzések</a>	7	Utolsó hozzászólás:
Leírás: Saját gondolatok, egyénileg készített előrejelzések.	hozzászólás	admin : 2009-04-07 14:11:23
<a href="#">Társalgó</a>	5	Utolsó hozzászólás:
Leírás: Kötetlen beszélgetés, bármilyen témáról.	hozzászólás	martinus : 2009-04-07 14:08:22

A fenti kép tárul elénk a fórum megnyitásakor. A fórumban megtekinthető a regisztrált felhasználók eszmecsereje az aktuális, illetve a közeljövő időjárásáról, illetve kötetlen társalgásuk. A fórumban három téma közül lehet választani:

- *Aktuális időjárás:* Itt lehet közölni, megtekinteni, illetve megtárgyalni az aktuális időjárást, az észlelt eseményeket, megfigyeléseket. A felhasználók itt érdekes információkat szerezhetnek egymástól.
- *Egyéni előrejelzés:* Itt írhatják le a regisztrált felhasználók saját véleményüket, meglátásukat arról, hogy milyen időjárási paraméterek, események következhetnek be a közeljövőben. Megoszthatják a többiekkel, hogy ezen eredményekre milyen úton következtetnek, és hogy milyen ismeretekre van szükség egy helyes előrejelzés készítésére, ezáltal is minden felhasználó újabb és újabb érdekes ismeretre tehet szert.
- *Társalgó:* Itt történhet a kötetlen beszélgetés, nincs témamegkötés. Tetszőleges témakörben cseveghetnek egymással a felhasználók, legyen az akár időjárás, sport, hírek, vagy akár egy hétvégi buli részletei.

Mindhárom témához külön-külön tábla tartozik az adatbázisban, ezek rendre a következők:

- aktualis\_ido
- egyeni\_efore
- tarsalگو

Mindhárom tábla a következő mezőket tartalmazza:

- nev VARCHAR(99)
- hozzaszolas VARCHAR(999)
- datum DATETIME
- idobelyeg LONG
- kulcs INT

A kulcs nem lehet NULL, és mindemellett a kulcs elsődleges kulcs a táblában, melynek meg van adva az AUTO\_INCREMENT, ennek segítségével a kulcs értéke mindig 1-el automatikusan növekszik egy új hozzászólás adatbázisba történő rögzítésekor.

A fórum oldalon mindhárom téma mellett további két rész látható még. Az egyik azt jelzi, hogy az adott témához összesen mennyi hozzászólás került már megírásra, a másik pedig azt, hogy az adott témában melyik felhasználó volt az utolsó hozzászólás írója, és ez mikor lett másodpercre pontosan megírva.

**Beleolvasás a fórumba:** Egy fórumtémát megnyitva a következőt láthatjuk, például a társalgó esetében:



The screenshot shows a forum thread titled "Társalgó". It contains five replies, each by the user "martinus". Each reply is displayed in a blue header bar with the following format: "X - martinus ::: YYYY-MM-DD HH:MM:SS ::: válasz". Below each header, the text of the reply is shown, followed by "tars X" where X is the reply number. The replies are numbered 1 to 5 from bottom to top. At the bottom right of the thread, there are two dropdown menus: one for page number (set to 1) and one for posts per page (set to 25). There are also buttons for "oldal listázása -> Listázz!" and "OK". At the bottom center, there is a message: "Új hozzászólás küldéséhez be kell jelentkeznie!"

Az oldal legfelső részén kiírásra kerül annak a témának a neve, amelyikbe éppen beléptünk. Alatta található a hozzászólások. Mindegyik hozzászólás kiírása szerkezetileg két részre van bontva. Az első részben először kiírásra kerül, hogy az aktuális témában az adott hozzászólás hányadikként szerepel, tehát hányadik hozzászólás az az adott témában. Utána megjelenítésre kerül a hozzászólás szerzője, illetve annak felhasználóneve, majd láthatjuk az adott hozzászólás elküldésének pontos dátumát és időpontját. Ezek után még az aktuális felhasználó állapotától függően további funkciók fognak megjelenni. A második részben jelenik meg a hozzászólás szövege.



Az kilistázott hozzászólások alatt, az oldal jobb oldalán látható egy oldalválasztó. Ennek segítségével legördülő listában kiválaszthatjuk, hogy egy oldalon mennyi hozzászólást szeretnénk látni, és kiválaszthatjuk, hogy hányadik oldalt szeretnénk megtekinteni. Miután kiválasztottuk a kívánt értéket, az 'OK' illetve 'Listázz!' gombra kattintva érvényesíthetjük akaratunkat. Az hogy mennyi hozzászólást szeretnénk látni egy oldalon, eltárolásra kerül a \$\_SESSION szuper-globális változóban, így ha egy másik fórumtémát választunk, ott is érvényes lesz az előzőleg kiválasztott érték mindaddig, amíg be nem zárjuk a böngészőt. Alapértelmezés, hogy 25 darab hozzászólás kerüljön egy oldalra, de tetszés szerint 1-től 50-ig választhat a felhasználó.

A két értékből (a kiválasztott oldalszámból, és a hozzászólások egy oldalra eső darabszámából), valamint az aktuális téma hozzászólásainak darabszámából könnyedén meghatározhatóak azok a hozzászólások, melyeket ki szeretnénk listázni, és mivel minden téma táblájában van egy INT típusú elsődleges kulcs, mely egyesével automatikusan növekszik a hozzászólások rendjében, ezért könnyedén kilistázhatóak a kívánt hozzászólások.

Így a forum.php fájl elején meghatározódik, hogy mettől meddig vagyunk kíváncsiak a hozzászólásokra, és mindegyik tábla esetében lekérdezésre kerülnek ezen hozzászólások. A hozzászólások táblánként (témánként) különböző névvel ellátott változóba kerülnek.

A fórum főoldalán, ahol ki lehet választani, hogy melyik témát szeretnénk megnyitni, a témák hivatkozásába be van építve egy plusz információ. Ha valamelyikre rákattintunk, akkor el lesz küldve a \$\_GET változóban egy olyan érték, mely megegyezik az adott témának megfelelő adatbázis-tábla nevével. Például a társalgó esetében az adatbázistábla neve: 'tarsalگو', és a hivatkozás a következőképpen néz ki:

```
forum.php?oldal=tarsalگو
```

Ez azért lényeges, mert a hozzászólások kiíratásakor a PHP dinamikus változóját veszem igénybe. Ejsünk néhány szót a PHP dinamikus változójáról:

Változókat azért definiálunk, mert szükségünk van egy olyan tárhelyre, melybe értéket szeretnénk eltárolni. A PHP-ben változót úgy tudunk megadni, hogy kiválasztunk egy tetszőleges nevet (melyben szerepelhet kis-és nagybetű, szám, aláhúzás jel, de nem kezdődhet számmal), és elé írunk egy '\$' jelet. A PHP lehetőséget nyújt arra a dinamikus változóval,



hogy egy változóban értéként egy másik változó nevét tároljuk el. Nézzük a következő példát:

Példa:

Tegyük fel, hogy egy változóban letároljuk a választott téma nevét, tehát:

```
$tema = "tarsalgo";
```

Ekkor a `'print $tema;'` kiírja nekünk, hogy `'tarsalgo'`. De ugyanezt az eredmény elérhetjük a következőképpen is:

```
$tema = „tarsalgo”;
```

```
$t = „tema”;
```

Ha ekkor kiadjuk a `'print $$t;'` parancsot, akkor szintén a `'tarsalgo'` -t kapjuk eredményül. Figyeljük meg, hogy kiíratáskor a változó neve előtt kettő darab `'$'` szerepel. Ezzel adjuk meg ugyanis, hogy mi dinamikus változót szeretnénk használni. Tehát az első `'$'` -jel definiál egy változót, melynek neve a `$t` változó értéke lesz.

Tehát én is ezt a technikát alkalmaztam a hozzászólások kiíratásánál. A `$_GET` változóban kerül átadásra a választott téma neve, és mivel a hozzászólásokat tároló változók is a saját témanevükkel van elnevezve, így dinamikus tömböt alkalmazva pontosan elérjük azt a változót, így azokat a hozzászólásokat, melyekre ténylegesen szükségünk van.

### **Hozzászólás a fórumban:**

Ha a felhasználó nincs bejelentkezve, akkor olvashatja a fórumot, de nem tud ahhoz hozzászólni. Erre figyelmezteti is az oldal a lap alján, a hozzászólások és az oldalválasztó után:

**Új hozzászólás küldéséhez be kell jelentkeznie!**

Ha bejelentkezett a felhasználó, akkor a fórumban a hozzászólások olvasása mellett lehetősége nyílik hozzászólás írására is. Ekkor az előbbi üzenet nem jelenik meg, hanem

helyette egy többsoros szövegbeviteli mező jelenik meg, mely alatt található a 'Küldés' gomb. A szövegbeviteli mezőbe tetszőlegesen bevitt szöveg elküldése után az oldal helyén egy üzenet tájékoztatja a felhasználót a hozzászólás elküldésének sikerességéről, és ha sikeresen megtörtént, azután az általa megadott formában megjelenik a fórum aktuális témájában a hozzászólása. A Küldés gombra kattintva a szöveg a `$_POST` változó segítségével a `send.php` fájl számára adódik át, mely a megfelelő kiegészítő információkkal együtt elmenti azt az adatbázisba, és ez az oldal értesíti a felhasználót a mentés sikerességéről. Az üzenet alatt helyezkedik el egy Tovább gomb, melyre kattintva a fórum főoldalára kerül a felhasználó.

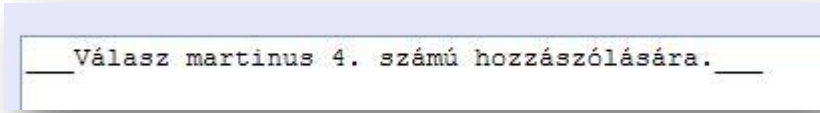
Ha a felhasználó be van jelentkezve, akkor plusz eszközök állnak rendelkezésére:



Minden hozzászólás mellett megtalálható a 'válasz' link, (bár ez megjelenik nem bejelentkezett állapotban is, de rákattintva semmi nem történik), és azon hozzászólások mellett, melyeknek a szerzője a bejelentkezett felhasználó volt, megjelenik a 'szerkesztés' link. Ha a szerkesztés linkre kattintunk, akkor az oldal aljára ugrást követően (ezt úgy érhetjük el, hogy a linkbe egy '#' jelet követően megadjuk a szövegdoz nevét) bemásolódik a hozzászólás szövege a szövegbeviteli mezőbe, és mivel a szerkesztés link márt tartalmazta az aktuális hozzászólás időbélyegét, itt is tudásunk birtokában lesz az, melyet a szöveg szerkesztését követően a küldés gombra kattintva továbbadunk a `send.php` részére. A `send.php` észleli, hogy ez bizony egy hozzászólás módosítása lesz, így nem fut le az új hozzászólás adatbázisba illesztésére vonatkozó rész, hanem egy olyan utasítást fog kiadni, mely az adott téma táblájában módosítja az adott időbélyegzőjű hozzászólás szövegét. Ezután egy szöveg jelenik meg az oldalon, mely tájékoztatja a felhasználót a művelet sikerességéről.

Ha a felhasználó a válasz linkre kattint, akkor az előzőekhez hasonlóan az oldal aljára ugrás történik, ahol a szövegbeviteli mező található, és mivel a válasz link tartalmazza az adott

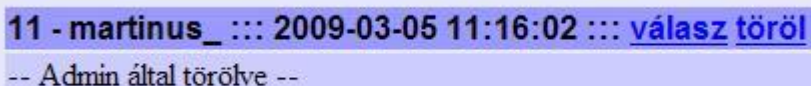
hozzászólás szerzőjének a felhasználónevét, valamint az adott hozzászólás sorszámát az adott témában, ezért ezen információkból egy szöveg automatikusan legenerálásra kerül, mely bemásolódik a szövegdobozba. Például:

A screenshot of a forum reply box. The box is light blue with a white border. Inside, the text reads: "Válasz martinus 4. számú hozzászólására. \_\_\_\_".

Válasz martinus 4. számú hozzászólására. \_\_\_\_

Ezután begépelhetjük a hozzászólásunkat, és a küldés gombra kattintva ugyanazon folyamatok fognak végbemenni, mint az alap hozzászólás küldésekor.

**Adminisztrátori műveletek a fórumban:** Amennyiben az oldalra bejelentkezik az adminisztrátor, és bármelyik témát nyissa is meg a fórumban, minden hozzászólás mellett pluszban megjelenik neki a válaszoláshoz és szerkesztéshez hasonló link, mely a törlés lesz.

A screenshot of a forum post header. The text is displayed on a blue background with white text. It reads: "11 - martinus\_ ::: 2009-03-05 11:16:02 ::: [válasz töröl](#)  
-- Admin által törölve --".

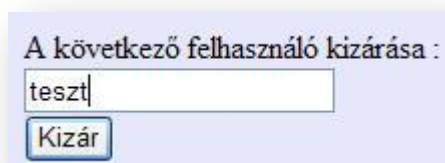
11 - martinus\_ ::: 2009-03-05 11:16:02 ::: [válasz töröl](#)  
-- Admin által törölve --

Ez a link tartalmazza a téma nevét, a hozzászóló felhasználónevét, valamint a hozzászólás időbélyegét, és ha az adminisztrátor rákattint, akkor a `forum.php` fájlban a törlésre vonatkozó algoritmusrész fog végbemenni. Ekkor az adott témának megfelelő táblában, az adott felhasználónevű és időbélyegzőjű hozzászólás szövege törlődik, és új értéke az 'Admin által törölve' lesz, valamint a hozzászóláshoz tartozó felhasználónév az aláhúzás jellel kiegészül, ez egyrészt jelzi, hogy a hozzászólás törölve lett, másrészt pedig a felhasználó később már tudja ezt a hozzászólást módosítani, mivel a letárolt felhasználónév, és a felhasználó azonosítója nem fog megegyezni.

Ha az oldal adminisztrátora be van jelentkezve, akkor a fórum főoldalának alján, valamint a témák oldalának alján is megjelenik egy 'Felhasználó kizárása' link. Erre ha ritkán is, de szükség lehet. Ha például egy felhasználó olyan hozzászólást küld be a fórumba, mely bármilyen okból kifolyólag nem állja meg a helyét, sért valakit vagy valamit, akkor a hozzászólás melletti töröl linkre kattintva eltávolítható a hozzászólás. De amennyiben egy

felhasználó sorozatosan követ el ilyen, vagy ehhez hasonló hibákat, vagy az alkalmazás más részén cselekszik helytelenül, akkor az adminisztrátor mérlegelhet. Amennyiben úgy dönt, hogy a felhasználó olyan súlyos hibát / hibákat vétett, melyek már nem elfogadhatóak, vagy sorozatosan tesz helytelent, az oldal biztonsága és a fegyelmezettség megtartása érdekében kizárhatja a felhasználót. Ekkor veheti igénybe a 'Felhasználó kizárása' linket. Ha rákattint, betöltődik a 'kizar.php' oldal.

A kizárás első lépése, hogy egy űrlapban meg kell adni a kizárni kívánt felhasználó nevét.

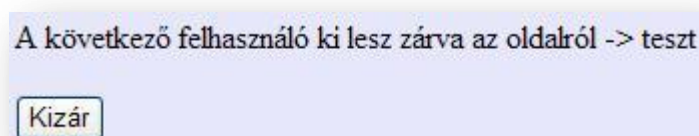


A következő felhasználó kizárása :

Azért választottam azt a módot, hogy be kelljen gépelni a felhasználó nevét, mert így kisebb az esélye annak, hogy véletlenül (félrekattintás) egy másik felhasználót zárjunk ki.

Miután megadtuk a felhasználó nevét, és rákattintottunk a kizár gombra, az oldal megerősítést fog kérni. Kiírja azt a felhasználónevet, melyet előtte adtunk meg, és ismételten rá kell kattintani a 'Kizár' gombra:



A következő felhasználó ki lesz zárva az oldalról -> teszt

Miután rákattintottunk, az oldal leellenőrzi, hogy van-e ilyen nevű felhasználó az adatbázisban. Ha nincs, megjelenik egy üzenet, mely tájékoztatja az adminisztrátort erről, különben az adott felhasználó jelszavát felülírja egy 19 hosszúságú karaktersorozattal, és értesíti az adminisztrátort a felhasználó sikeres kizárásáról. Mivel maga a felhasználó benne marad az adatbázisban, így ugyanazzal a felhasználónévvel nem lehet újabb regisztrációt végrehajtani, de ezzel a felhasználónévvel a továbbiakban belépésre sincs lehetőség. Egyrészt azért, mert a felhasználónak semmilyen információja nincs az új jelszóról, másrészt pedig a bejelentkezés folyamata során az alkalmazás figyeli a jelszót, és ha a bejelentkezni kívánó

felhasználó azonosítójához tartozó adatbázisbeli jelszó megegyezik a kizáráskor adandó jelszóval, a bejelentkezési folyamat nem fog végbemenni, nem tud a felhasználó bejelentkezni, és értesítést kap a felhasználó a kizárásról. Az az eset se fordulhat elő, hogy egy felhasználó a regisztráció folyamán esetleg ugyanazt a jelszót adja meg, mint ami a kitiltáskor használatos, mert a kitiltáskor használandó jelszó 19 karakter hosszú, de a regisztrációkor maximum 18 karakter hosszú jelszót lehet megadni.

### Megfigyelő oldal

A főoldalon a megfigyelések, vagy a menüben a megfigyelő oldal linkre kattintva juthat el a felhasználó a megfigyelő oldalra. Ekkor a tartalom keretbe betöltődik a `megfigyelo.php` fájl.

A megfigyelő oldal azt a célt szolgálja, hogy a felhasználók meg tudják tekinteni Magyarország aktuális időjárását. Erre mind vizuális, mind szöveges formában lehetősége van. Ezen információkat szintén a felhasználók biztosítják, ők küldik be az oldalra. Lehetőség van beküldeni az aktuális égbépet, és az aktuális hőmérsékletet, valamint a megfigyelése mellét szöveges leírást is mellékelhet, melyben tovább részletezheti az aktuális időjárást.

Az oldal bal felső sarkában egy térkép helyezkedik el, melyen Magyarország, és szűk környezete látható. Ugyanez a térkép jelenik meg a regisztráció folyamán is azzal a különbséggel, hogy a regisztrációnál a térkép fel van osztva kis területekre, mely területek sorra meg vannak számozva. Itt lesz szerepe a regisztrációkor megadott, vagy később módosított 'Észlelési hely' -nek. A megfigyelő oldalon látható térkép ugyanúgy, ugyanolyan formában fel van osztva területekre, mint ahogy az a regisztrációs oldalon, csak itt sem a keretek, sem a számok nem látszanak az áttekinthetőség érdekében. Az alkalmazás az adatbázisból kilistázza azon megfigyeléseket, melyek az elmúlt 30 percen kerültek beküldésre. Azért csak ezeket listázza ki, hogy a térképen valóban az aktuális időjárás látszódjon, ne maradjanak rajta régi értékek. Még mielőtt ezeket az értékeket elhelyezné a térképen, ellenőrzésre kerül, hogy ezen értékek közt van-e olyan, melyet ugyanazon területen kellene megjeleníteni. Ha van, akkor az adott területre a legutoljára beküldött adatokat fogja

figyelembe venni az alkalmazás, a többivel nem foglalkozik, így egyértelműen egy területen maximum egy ikon fog megjelenni.

### **Megfigyelés beküldése:**

Megfigyelést beküldeni csak bejelentkezett felhasználó tud. Ha a felhasználó nincs bejelentkezve, akkor a megfigyelő oldalon mindent meg tud tekinteni, de a megfigyelés beküldésére szolgáló helyen egy üzenet fog neki megjelenni, mely informálja ezen tényről:

**Ha megfigyelést szeretne küldeni, előbb be kell jelentkeznie!**

Ha a felhasználó be van jelentkezve, akkor lehetősége adódik megfigyelés beküldésére:



Az aktuális égbépet egy legördülő listából lehet kiválasztani. A következők közül választhatunk:

- Derült
- Gyengén felhős
- Közepesen felhős
- Erősen felhős
- Borult
- Eső
- Zivatar
- Száraz zivatar
- Havazás
- Köd

Hőmérsékletet szintén legördülő listából választhatunk, melyben az értékek -30 -tól +50-ig szerepelnek. Kezdetben / alapértelmezésként a '---' karaktersorozat szerepel kiválasztottként, ha ezen nem módosítunk, akkor az alkalmazás azt úgy veszi, hogy nem adtunk meg hőmérsékleti adatot. Továbbá opcionálisan még megadhatunk egy szöveges kiegészítést a megfigyeléshez, melyben közölhetünk még további paramétereket, vagy bármilyen más információt közölhetünk az időjárásról. A 'Küld' gombra kattintva a megadott adatokat a \$\_POST változó segítségével továbbküldjük a megfigyel.php fájl számára, mely ekkor betöltődik. A küldésre kattintva a felhasználónak bizonyos adatai is átküldésre kerülnek, melyek szükségesek a sikeres megfigyeléshez, de ezt nem kell megadnia a felhasználónak, hanem a profilban szereplő értékek automatikusan beillesztésre kerülnek. Ezt a felhasználó nem látja, mert ezen értékek rejtett mezők alapértelmezett értékei lesznek.

A megfigyel.php a megkapott szöveges épképet megfelelő számkóddá alakítja (IF utasítások segítségével), valamint a többi megkapott értéket is lementi változókba. A getdate() beépített függvénnyel lekérdezi az aktuális dátumot és időt, egy másik változóba pedig a time() beépített függvény állítja elő az aktuális időbélyeget. Ezen adatok összessége végül egy megfigyelésként elmentésre kerül az 'eszlelesek' adatbázis-táblába. Ez a tábla tartalmaz egy 'kulcs' nevű INT típusú mezőt is, mely elsődleges kulcs a táblában, és érvényes rá a NOT NULL valamint az AUTO\_INCREMENT megszorítás, mely szerint nem vehet fel NULL értéket, és új bejegyzés rögzítésekor 1-el automatikusan növekszik az értéke.

A művelet sor végén a megfigyel.php fájl egy szöveges üzenetet ír ki a felhasználó számára a megfigyelés elmentésének sikerességéről.

### **Megfigyelések megjelenítése:**

Az megfigyelés beküldését követően az adott megfigyelés megjelenik a térképen mindaddig, amíg:

- a megfigyelés időpontjától számított 30 perc el nem telik. Amennyiben a 30 perc eltelté után töltjük be az oldalt, a térképen már nem fog megjelenni az adott megfigyelés.

- ugyanazon megfigyelési területről nem érkezik újabb megfigyelés. Amennyiben újabb megfigyelés érkezik tetszőleges felhasználótól, akkor az kerül megjelenítésre az oldal betöltésekor.

Tehát ha megfigyelést küldünk be, az megjelenik a térképen, és a térképen a megfigyelés felé mozdítva az egeret, információk jelennek meg a megfigyeléssel kapcsolatban:



A térkép megfelelő pontján (ami a felhasználó profiljában meg van adva megfigyelési helynek) megjelenik az megfigyeléskor beküldött égbépnak megfelelő ikon. A beküldéskor az alkalmazás az égbépet egy számként mentette el, mely szám pontosan megegyezik az adott égbéphez tartozó ikon / képfájl nevével, így megjelenítéskor csak egyszerűen a `' .jpg'` karaktersorozat elé ki kell íratnunk azt a számot, mely az adott megfigyeléshez tartozik. Így megkapjuk a helyes ikon teljes fájlnevét. Az égbép nevének lekérése mellett a megfigyeléshez tartozó további érték is lekérdezésre kerülnek, és ezek közül néhányat meg is jeleníthetünk, ha az egérmutatót a térképen a megfigyelés égbép-ikonja fölé mozdítjuk. Ezek kötőjellel vannak elválasztva. Az értékek sorra:

- hőmérséklet, amennyiben tartozik az adott megfigyeléshez ilyen adat. Ha nem tartozik, akkor a ennek helyén nem jelenik meg semmi, hanem a következővel indul az információsáv:
- égbép megnevezése



- a megfigyelést beküldő felhasználó azonosítója
- a megfigyelés pontos helye, városnév alapján
- a megfigyelés beküldésének pontos dátuma és időpontja

Ezen értékeket a kép beillesztésekor az `<img>` tag-en belül az `alt=""` részben adhatjuk meg, ez esetben fognak megjelenni ezek az információk részben, ha a képre mozdítjuk az egérmutatót.

A megjelenített égbélikon azonban nem más, mint egy hivatkozás megjelenítendő része, tehát ennek a képnek a megjelenítése a `<a>` és `</a>` HTML részek közt szerepelnek. Így ha a térképen az égbélikonra kattintunk, a `<a>` HTML részben megadott hivatkozás fog érvénybe lépni, melyet a `'href='` részben adhatunk meg. Az alkalmazásban itt ez a rész egy olyan hivatkozás, mely alapján a térkép alatt szereplő megfigyelési lista azon részére ugorhatunk, ahol az adott megfigyelés szerepel. Ennek a helynek a beazonosítására a megfigyeléshez tartozó időbélyeg nyújt segítséget. Jelen alkalmazás esetében például:

```
<a href="#"<?php print $idobelyeg[$z-1]; ?>">
```

ahol a `'$z-1'` index azonosítja az adott megfigyelést.

Így ha a térképen rákattintunk egy égbélikonra, akkor az oldal a lentebb szereplő megfigyelések közül oda ugrik, ahol szerepel az adott megfigyelés, és ott is megnézhetjük a beküldött valamint a beküldő adatait, és ha van, elolvashatjuk a megfigyeléshez fűzött szöveget. Természetesen ezt manuálisan is megtehetjük az oldal görgetésével, de így minden bizonnyal gyorsabb.

Egy példa: a térkép alatti megfigyelések listájából egy megfigyelés:



Tehát itt is megtekinthetjük a megfigyelést beküldő felhasználó azonosítóját, helyét, dátumát, időpontját, az égbélikon és hőmérsékletet, valamint ez alatt megjelenik az észleléshez mellékelt szöveg is.

### **Információk a felhasználóról:**

Itt a felhasználó neve egyben egy hivatkozás is, helyesebben fogalmazva egy HTML <a> részében elhelyezett 'onclick' esemény kezelése. Ha a felhasználó nevére egérrel történő kattintás megy végbe, akkor az 'alert()' segítségével egy ablak jelenik meg, melyben a felhasználóról jelennek meg a következő adatok:

- a felhasználó neve
- a felhasználó regisztrációjának pontos dátuma és időpontja
- a felhasználó utolsó belépésének pontos dátuma és időpontja
- a felhasználó által az oldalra beküldött megfigyelések darabszáma
- a felhasználónak a bemutatkozó szövege, melyet a regisztrációkor adott meg, illetve melyet később módosíthat a profiljában.



### **Adminisztrátori műveletek a megfigyelőoldalon:**

Amennyiben az oldalra az adminisztrátor van bejelentkezve, a megfigyelő oldalon egy plusz link jelenik meg számára:



Ez pedig a töröl nevezetű link. A töröl link hivatkozásában szerepel az adott megfigyeléshez tartozó felhasználónév, és a megfigyelés időbélyege, ezáltal egyértelműen beazonosítható a

törölni kívánt megfigyelés az adatbázisban. Ha erre a linkre rákattint az adminisztrátor, akkor az adott megfigyelés törlésre kerül az adatbázisból. (Természetesen ilyenkor az adott felhasználó által beküldött megfigyelések darabszáma is csökken 1-el).

A megfigyelő oldal alján a fórumhoz hasonló módon itt is megtalálhatjuk az oldalválasztót, ahol beállíthatjuk, hogy az alkalmazás egy oldalra mennyi megfigyelést listázzon ki az oldal aljára, illetve, hogy a kilistázott megfigyelések hányadik oldalára vagyunk kíváncsiak. A megfigyelési lista elején szerepelnek a legfrissebb megfigyelések, ezt követően egyre régebbiek szerepelnek, így az utolsó oldalon találhatóak a legrégebben beküldöttek.



Tehát a fórumhoz hasonló módon az aktuális oldalszámból, az egy oldalra kilistázandó megfigyelések számából, és az összes megfigyelés számából egyértelműen meghatározhatóak azon megfigyelések, melyet a felhasználó meg szeretne tekinteni. Ezután ezen értékek segítségével, és az adatbázis-táblában található kulcsok segítségével kerülnek lekérésre az adatok.

## WebShop

Az alkalmazás webshop része online boltot valósít meg. A felhasználók nézegethetnek, válogathatnak a különféle termékek és kiegészítők között, melyek többnyire az időjáráshoz, meteorológiához kapcsolhatók. Különféle eszközök állnak rendelkezésre a meteorológiai paraméterek méréséhez, de megtalálhatók a fényképezőgépek és videokamerák is, melyekkel megörökíthetnek akár egy megszokott, akár egy nem hétköznapi mondható időjárási jelenséget. A webshopban ezen kívül a termékekhez sokféle kiegészítő fellelhető, ezáltal is

megkönnyítve a felhasználónak az optimális vásárlást. A következő kategóriák lelhetők fel a webshopban:

- Összes kategória
- Kábelek, csatlakozók
- Időjárás állomások
- Villámdetektorok
- Csapadékmérők
- Videokamerák
- Fényképezők
  - o kiegészítők
  - o gépek
- Szélmérők
- Hőmérők

Megrendeléseit otthonról is elintézheti a számítógép előtt ülve, és megrendelésének leadását követően hamarosan házhoz szállítják csomagját.

### **Böngészés a webshopban:**

A főoldalon vagy a menüben a WebShop linkre kattintva a tartalom keretbe betöltődik a `webshop.php` fájl, melyet követően az oldal tetején a következőt láthatjuk (nem bejelentkezett felhasználó esetében):

## Összes kategória

Kábelek, csatlakozók

Időjárás állomások

Villámdetektorok

Csapadékmérők

Videokamerák



Fényképezők

Szélmérők

Hőmérők

Tehát kezdetben linkek vannak felsorolva, mely linkek megadják az elérhető kategóriákat. Legfelül látható egy olyan link, melyre kattintva az adatbázisban található összes termék, kategóriától függetlenül, megjelenítésre kerül. A további linkek bármelyikére kattintva, kivéve a fényképezőket, megjelennek az adott kategórián belüli termékek. A fényképezőkre kattintva egy újabb lista jelenik meg, melyen belül kiválaszthatjuk, hogy fényképezőgépet szeretnénk nézni, vagy a kiegészítőket böngésszük.

Ha megnyitunk egy kategóriát, akkor megjelennek a kategórián belüli termékek, és részleteik:

Név	Típus	Leírás	Garancia	Ár	Kép	
StrikeAlert	Villámdetektor	Nadrágra csíptethető villámdetektor, mely figyelmeztet, ha vihar van 65 kilométeres körzetben.	1	24900		<a href="#">Vélemények</a>
Boltek LD-250	Villámdetektor	Számítógéphez csatlakoztató villámdetektor, mely 500 kilométeres körzetben észleli és jeleníti meg a képernyőn a villámokat.	1	239000		<a href="#">Vélemények</a>

Minden termék esetében láthatjuk annak nevét, típusát, részletes leírását, garanciaidejét, árát, képet a termékről, és egy 'Vélemények' nevű link áll a felhasználó rendelkezésére. A termékek listája ezen felsorolt tulajdonságok bármelyike alapján rendezhető akár növekvő,

akár csökkenő sorrendben is. Ehhez csak rá kell kattintani a kiválasztott tulajdonságra a táblázat legfelső sorában. Ezek mindegyike egy link, melyeknek hivatkozásába be van illesztve, hogy pontosan melyik tulajdonság szerint kell majd rendezni (a linkben a tulajdonságnevek pontosan ugyanúgy vannak megadva, mint ahogy az az adatbázisban is el van nevezve, ezáltal később egyszerűbb lesz kiadni az SQL utasítást). Továbbá ez a hivatkozás tartalmaz még egy olyan információt is, hogy növekvően, vagy csökkenően kell-e majd rendezni. Ennek az információnak a beillesztését is PHP kód végzi, ugyanis a rendezési információ a `$_GET` változóban lesz megadva. A program megnézi, hogy be van-e állítva a rendezési irány. Ha be van állítva 'ASC' értékre, akkor az azt jelenti, hogy előzőleg már volt rendezve a lista, még hozzá növekvően, tehát ha a felhasználó megint kattintana a linkre, akkor ezt az értéket módosítani kell csökkenőre, tehát 'DESC' -re. Minden más esetben 'ASC' -ra kell módosítani. Azért választottam azt, hogy paraméterként pontosan az 'ASC' és 'DESC' karaktersorozatok legyenek átküldve, mert később ezeket egyszerűen beillesztve az SQL utasításba működő SQL utasítást kapunk. Ez szemléltetve egy, az alkalmazásból kimásolt kódrészlettel:

```
. . . &rend=<? if($_GET['rend']=="ASC") print "DESC"; else  
print "ASC"; ?> . . .
```

Tehát így a felhasználó az általa fontosnak vélt szempontok szerint rendezheti a termékeket, ezáltal mintegy szűrve a termékeket, hogy melyek számára a megfelelőbbek.

A termékek kilistázásakor láthatunk egy olyan linket is minden egyes termék mellett, melynek az a neve, hogy 'Vélemények'. Ez a funkció azért nagyon hasznos, mert minden egyes termékhez, külön-külön szövegesen véleményt lehet írni, illetve ezeket a véleményeket elolvasni. Nem bejelentkezett felhasználó ezeket csak olvasni, bejelentkezett felhasználó ezekhez újat írni is tud. Ezáltal a felhasználók megoszthatják másokkal tapasztalatait, vagy akár azt, hogy mit hallottak az adott termékről, mi a véleményük, illetve a leendő vásárlók elolvashatják ezen véleményeket, melyek segíthetnek a helyes döntés meghozatalában.

A 'Vélemények' linkre kattintva nem történik más, minthogy a fórum, a `forum.php` fájl töltődik be. A `forum.php` -nek a `$_GET` változóban átadásra kerül a fórumtéma neve, mely ez esetben 'webshop\_forum' lesz, valamint átadásra kerül a termék ID-je, azonosítója (az adatbázisban minden terméknek egyedi azonosítója van). A fórum

betöltődéskor ezen értékek jelenlétéből tudja, hogy a felhasználó nem a hagyományos fórumot kívánja használni (mely már be lett mutatva a szakdolgozat első felében), hanem a 'webshop\_forum' témába kíván belépni. Az adatbázisban található egy olyan tábla, melynek neve megegyezik a téma nevével (webshop\_forum), valamint a tábla mezői is ugyanazok, mint a hagyományos fórum tábláinak, de itt még pluszban van egy mező, melyben az ID (azonosító) lesz eltárolva. Ezáltal fogjuk tudni, hogy az adott hozzászólás, vélemény melyik termékhez tartozik.

Ha megnyitjuk egy termékhez tartozó véleményeket, akkor az ID alapján lekérdezésre kerül a termék neve, és az oldal tetején ez kiírásra kerül, így a felhasználó mindig tudhatja, hogy éppen melyik termékkel kapcsolatban olvas éppen. Ezt követően időben csökkenő módon kilistázásra kerülnek az adott termékhez tartozó hozzászólások. Mivel a fórumba kerültünk, így ugyanolyan a megjelenítés, mint ahogy az a fórumnál tárgyalva volt. Ugyanaz a működés, mint a fóruból elérhető három témánál (aktuális időjárás, egyéni előrejelzések, társalgó), csak ebben az esetben egy másik témát használunk. Például ugyanúgy lehet hozzászólást küldeni, ugyanúgy jelenik meg minden, ugyanúgy megtalálhatóak a válaszolásra, hozzászólás szerkesztésére, és az adminisztrátornak a hozzászólás törlésére való linkek. Egyetlenegy különbség van, itt nincs oldalválasztó. Itt a hozzászólások egy oldalon jelennek meg. Ennek oka, hogy egy termék esetében nem várható, hogy túl sok hozzászólás lenne hozzá.

### **Lehetőségek bejelentkezett felhasználóknak és a megrendelések:**

Ha a felhasználó bejelentkezik, akkor a webshop megnyitásakor két újabb link jelenik meg számára:

## **KOSÁR (0)**

### **Megrendelések**

### **Összes kategória**

#### **Kábelek, csatlakozók**

#### **Időjárás állomások**

#### **Villámdetektorok**

#### **Csapadékmérők**

#### **Videokamerák**

#### **Fényképezők**

#### **Szélmérők**



#### **Hőmérők**

Megjelenik tehát egy 'Kosár' és egy 'Megrendelések' link. Ha a felhasználó a 'Megrendelések'-re kattint, áttekintheti megrendeléseit. A 'Kosár' linkben kiírásra kerül még egy szám is. Ez a szám jelzi a felhasználónak, hogy a virtuális bevásárlókosarában összesen mennyi termék van. Mivel kezdetben még semmit nem helyezhetett még bele, ezért ez a szám induláskor nulla lesz. Ha nincs semmi sem a kosarában, akkor nincs mit megnézni, ezért ha ekkor kattintunk rá a 'Kosár' linkre, semmi nem fog történni.

Miután be vagyunk jelentkezve, nyissuk meg valamelyik kategóriát a webshopban. A következőt láthatjuk:



## KOSÁR (0)

Név	Típus	Leírás	Garancia	Ár	Kép		
StrikeAlert	Villám-detektor	Nadrágra csíptethető villám-detektor, mely figyelmeztet, ha vihar van 65 kilométeres körzetben.	1	24900		Vélemények	1 darabot Kosárba
Boltek LD-250	Villám-detektor	Számítógéphez csatlakoztató villám-detektor, mely 500 kilométeres körzetben észleli és jeleníti meg a képernyőn a villámokat.	1	239000		Vélemények	1 darabot Kosárba

A termékek kilistázása felett itt is látható a 'Kosár' link, és itt is láthatjuk, hogy mennyi terméket helyeztünk már el benne. Továbbá a táblázatban megjelent egy újabb oszlop, melyben elvégezhetjük a kiválasztott termék kosárba helyezését. Még mielőtt behelyezzük a kosárba, beállíthatjuk, hogy összesen hány darabot szeretnénk az adott termékből. Ez a szám minden termék esetében kezdésnek 1, de a felhasználó ezt módosíthatja. A 'Kosárba' gombra kattintva az adott termék a kosarába kerül, és a fenti 'Kosár' linknél a szám 1-el növekszik. Akkor is csak 1-el növekszik a szám, ha egy adott termékből többet helyeztünk kosárba, mivel a kosárban egy termék csak egyszer szerepel, de minden termékhez hozzá van rendelve (így ki is van írva) egy szám, mely a darabszámot jelzi. Így például, ha egy adott terméknél folyamatosan nyomogatjuk a 'Kosárba' gombot, akkor sem fog növekedni ez a számláló, de a kosárban az adott termék mellett szereplő számláló viszont növekedni fog.

Amikor a felhasználó a kosarába helyezi a legelső terméket, akkor az alkalmazás lekéri az abban a pillanatban érvényes időbélyeget, és ezt fogja beállítani kosárazonosítónak. Ezután tehát annak a kosárnak lesz egy azonosítója. Amennyiben egy időbélyeg érvénye alatt egynél több kosár jönne létre, akkor se fog ütközés történni sehol, mert az alkalmazás minden műveletnél a kosárazonosító mellett az ahhoz tartozó felhasználói nevet is ellenőrzi.

Például helyezzünk egyféle termékből két darabot a kosárba, és ezután kattintsunk a 'Kosár' linkre. Ekkor megtekinthetjük a kosarunk tartalmát, valamint további műveletek állnak rendelkezésünkre.

## KOSÁR (1)

Számlaszám: 123456789-987654321

Név	Típus	Leírás	Garancia	Ár	Kép	Darab		
I_AUDIO D2 TV out	Kábel, csatlakozó	iAudio D2 MP4 lejátszóhoz TV kimeneti kábel.	1	1590		2	<a href="#">Vélemények</a>	<a href="#">Töröl</a>

**3180 Ft**

Nettó: 2650 Ft  
ÁFA összege: 530 Ft

[MÓDOSÍTÁS \(visszahelyezés a kosárba\)](#)

[VÉGLEGESÍTÉS](#)

[TÖRLÉS](#)

A kilistázás előtt és a 'Kosár' link közt megjelenik egy számlaszám. A későbbiekben a felhasználó erre a számlaszámra utalhatja át megrendelése összegét. Továbbá kilistázásra kerül kosarának tartalma. Ugyanazon információkat láthatjuk róla, mint a webshopban való keresgéléskor, de itt megjelenik egy darabszám is. Itt láthatjuk, hogy az adott termékből összesen hány darabot tettünk a kosarunkba. Továbbá megjelenik minden termék mellett egy 'töröl' link is, melyre kattintva az adott terméket eltávolíthatja a felhasználó a kosarából. A táblázat alatt ki van írva, hogy megrendelése összesen mennyibe kerül. Láthatjuk, hogy egy féle, 1590 forint értékű termék van a kosárban, de mivel 2 darabot helyezett bele, ezért a végösszeg  $2 \times 1590 = 3180$  forint lesz. Az alkalmazás további információkét kiszámolja, és kiírja, hogy ezen megrendelésnek mennyi a nettó összege, és mennyi a fizetendő ÁFA.

Mielőtt továbbtérnénk a módosításra, véglegesítésre, és törlésre, meg kell említenem, hogy mi is történik pontosan akkor, amikor a felhasználó behelyez egy terméket a kosarába.

Az adatbázisban szerepel egy 'megrendelesek' nevű tábla. Ez a tábla fogja tárolni a felhasználók megrendeléseit. A táblában a következő mezők szerepelnek:

- *id* LONG – termék azonosítószáma
- *nev* VARCHAR(99) – termék neve
- *tipus* VARCHAR(99) – termék típusa
- *leiras* VARCHAR(999) – termék leírása
- *garancia* INT – termékre vonatkozó garanciaidő
- *ar* INT – termék ára
- *kep* VARCHAR(99) – termékénél megjelenítendő képfájl neve
- *megrendelo* VARCHAR(99) – terméket megrendelő felhasználó azonosítója
- *kosar\_id* LONG – azonosítja, hogy melyik kosárhoz / megrendeléshez tartozik a termék
- *leadva* LONG – amennyiben a megrendelés le lett adva, a megrendelés leadásakor aktuális időbélyeg az értéke, különben nulla
- *fizetve* INT – amennyiben a megrendelés ki lett fizetve, értéke 1, különben 0
- *kezbesítve* INT – amennyiben a megrendelés le lett kézbesítve, értéke 1, különben 0
- *darab\_termek* INT – a termékből hány darab lett az adott megrendelésben megrendelve
- *cim* VARCHAR(199) – a megrendelés kiszállításához szükséges információk

Amikor egy felhasználó belehelyez egy terméket a kosarába, akkor az alkalmazás ezt az esemény rögzíti az adatbázisba. Az előbb bemutatott 'megrendelesek' nevű táblába fog egy bejegyzést létrehozni. Ezzel lehet biztosítani azt, hogy ha a felhasználó termékeket helyezett a kosarába, és valamilyen okból kifolyólag megszakad a folyamat (például véletlenül bezárja a böngészőt, félrekattint, lefagy a számítógépe, áramszünet, vagy csak később szeretné folytatni a megkezdett megrendelést...), akkor azt a felhasználó később újra elővehesse, és ott folytathassa a megrendelését, ahol abbahagyta. Ezzel egyszerűbbé és gyorsabbá válik a felhasználó munkája, és szimpatikusabbá válhat számára az oldal.

Tehát ezért lesz szerepe a 'Módosítás (visszahelyezés a kosárba)' linknek. A webshop nyitóoldalán a 'Megrendelések' linkre kattintva két link fog megjelenni:

- Függőben lévő megrendelések

- Leadott megrendelések

A függőben lévő megrendelésekre kattintva az alkalmazás kilistázza azon megrendeléseit, melyet elkezdett, de nem fejezett be. Ezt az alkalmazás onnan tudja, hogy megnézi a 'megrendelesek' táblát, és ha szerepel benne olyan termék, amelyhez a hozzárendelt felhasználónév pontosan a bejelentkezett felhasználó neve, és a 'leadva' mezőben nulla szerepel, akkor az ahhoz a termékhez tartozó 'kosar\_id' azonosítójú megrendelés biztosan nem lett véglegesítve, leadva. Ezért itt kilistázza a félbeszakadt megrendeléseket, és ha az egyikre rákattintunk, és azon belül választjuk a módosításra vonatkozó linket, akkor az adott megrendelés adatai átkerülnek a kosárba, (az oldal tetején a 'Kosár' linkben a számláló is beállításra kerül), és utána ott folytathatjuk az adott megrendelést, ahol abbahagytuk. Tehát ha terméket teszünk a kosárba, akkor az előzőleg módosításra kiadott kosár tartalmához fogjuk a termékeket hozzárendelni. Amikor félbeszakított megrendelést helyezünk vissza a kosárba, az alkalmazásban csak annyi történik, hogy az adott megrendeléshez tartozó azonosító (kosar\_id) lesz beállítva a \$\_SESSION változóba (ugyanis mindig ebben a szuper-globális változóban tároljuk, hogy éppen melyik kosárral dolgozunk), és ezután az oldal automatikusan már az új azonosítóval fog dolgozni.

Ha felhasználó úgy dönt, hogy minden megrendelni kívánt terméket elhelyezett kosarában, és az adott megrendelést el szeretné küldeni, akkor ezt a 'Véglegesítés' linkre kattintva teheti meg. Egyébként az adott megrendelésnek nem szükséges a kosárban lennie, ha megnyitja a függőben lévő megrendeléseinek valamelyikét, akkor azt a kosárba visszahelyezés nélkül is megteheti. Ha a felhasználó a véglegesítést választja, akkor a 'megrendelesek' táblában az összes olyan termék esetében, ahol a termékhez tartozó felhasználónév megegyezik a bejelentkezett felhasználó nevével, és a megrendelés-azonosító (kosar\_id) is megegyezik az aktuális (kiválasztott) megrendelés azonosítójával, a 'leadva' értéke az aktuális időbélyeg lesz.

Továbbá egy függőben lévő megrendelésben, vagy a kosárban választható még a 'Törlés' link is, melyre kattintva a felhasználó törölheti az adott megrendelését. Ekkor a 'megrendelesek' táblából minden olyan termék eltávolításra kerül, melyhez tartozó felhasználói név megegyezik az éppen bejelentkezett felhasználó nevével, és a hozzá tartozó

megrendelés-azonosító (`kosar_id`) is megegyezik az aktuális (kiválasztott) megrendelés azonosítójával.

A függőben lévő megrendelések mellett a felhasználó megtekintheti a már leadott megrendeléseit is. Egy leadott megrendelést megnyitva további információk jelennek meg a megrendelésen belüli termékeken, és a megrendelés árán, valamint az átutaláshoz használandó számlaszámon kívül:

**Megrendelésszám: martinus1239372994**

**Fizetve: Nem**

**Kézbesítve: Nem**

**Kiszállítási cím: 4030 - Debrecen - Valamilyen utca 33. - Sipos Márton**

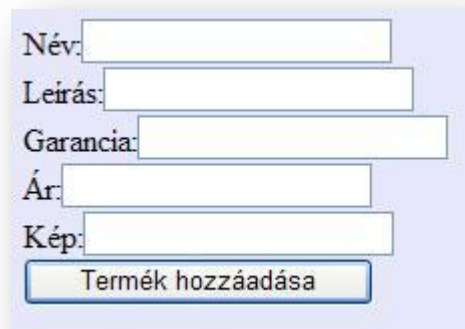
Láthatjuk, hogy mi az aktuális megrendelés száma, megtekinthetjük, hogy átutalásunk megérkezett-e, illetve utánvétel esetén a fizetési státusz be lett-e regisztrálva, láthatjuk, hogy be lett-e regisztrálva, hogy kiszállították a csomagot, valamint megtekinthetjük, hogy milyen címet adtunk meg a kiszállításhoz.

Függőben lévő megrendelést bármikor módosíthatunk, de egy leadott megrendelést csak a leadást követő első 8:00 –ig lehet módosítani, utána az erre vonatkozó linkek is eltűnnek a megrendelésen belül. Ez azért van így, mert például az online bolt minden nap reggel 8:00 –tól kezd a csomagolási és küldési munkálatokat, akkor így lehet konzisztensen tartani a megrendeléseket. Ezt a vizsgálatot az alkalmazás egy függvényhívással teszi. Amikor a módosításra való linkek megjelennek, valamint módosítás előtt a kiválasztott megrendelés leadásának időbélyegzője paraméterként adódik az oldal elején általam definiált `'levanezarva()'` függvénynek, mely meghatározza, hogy a megrendelésen lehet-e módosítást végrehajtani. Amennyiben a függvény a `'true'` karaktorsorozatot adja vissza, az azt jelenti, hogy a megrendelés módosításának ideje lejárt, tehát már nem módosítható a megrendelés, akkor az alkalmazás módosítás kérése esetén nem hajtja azt végre, hanem közli a felhasználóval, hogy letelt az időkorlát. Ha a `'false'` karaktorsorozatot adja vissza, az azt jelenti, hogy a megrendelés még időkorláton belül van, tehát végrehajtható a kért módosítás.

### Adminisztrátori műveletek a webshopban:

Ha az oldalra az bejelentkezik az adminisztrátor, akkor lehetősége nyílik a termékek közül törölni, illetve új terméket adhat hozzá az adatbázishoz, ezzel naprakészen tartva az árukat. Ha az adminisztrátor megnyitja valamelyik kategóriát, minden egyes termék mellett, a táblázat jobb oldalán egy újabb oszlop jelenik meg, melyben egy törlés link található. Ez a hivatkozás tartalmazza a törlendő termék azonosítóját, így az a `$_GET` tömbben átadásra kerül. Ha az alkalmazás észleli, hogy van megadva ilyen azonosító a `$_GET['termeket_torol']` -ben, és ha az aktuális felhasználó az oldal adminisztrátora, akkor lefut egy kódrész, mely eltávolítja az adatbázisból az adott azonosítójú terméket. Ekkor ez a termék a 'termékek' táblából eltűnik, ezért is van előnye annak, hogy egy megrendelés készítése folyamán a termék minden tulajdonsága eltárolásra kerül a 'megrendelesek' táblában. Ha ugyanis egy terméket eltávolít az adminisztrátor, akkor az a leadott és függőben lévő megrendeléseknél is megmarad.

Továbbá az adminisztrátor új terméket vehet fel az adatbázisba. Ha az adminisztrátor van bejelentkezve, akkor minden kategóriában, a termékek kilistázását követően megjelenik egy űrlap, mely erre nyújt lehetőséget:



Név:   
Leírás:   
Garancia:   
Ár:   
Kép:

Itt meg kell adni minden, a termékkel kapcsolatos adatot. A termékről egy képet kell másolni a 'pictures' mappába, majd ennek a képnek kell a nevét (a fájl kiterjesztésével együtt) megadni a 'Kép' mezőbe. A 'Termék hozzáadása' gombra kattintva a `$_GET` változó segítségével átadódnak a megadott értékek, és elmentésre kerülnek az adatbázisba,

mint egy darab termék. Ezután abban a kategóriában fog a termék megjelenni, amelyik kategórián belül használtuk az űrlapot.

Az adminisztrátornak ugyanúgy elérhetőek a függőben levő, és a leadott megrendelések link, melyekre kattintva az adatbázisban szereplő összes megrendelést láthatja. A megrendelések bármelyikére kattintva ugyanúgy jelenik meg az adott megrendelés, mint bármely más felhasználónak, de itt az oldal alján két link jelenik meg:

[Megrendelés befizetésének beregisztrálása](#)  
[Megrendelés kézbesítésének beregisztrálása](#)

Amennyiben az adott megrendelést leadó felhasználó teljesítette fizetési kötelezettségét, az adminisztrátor használhatja a 'Megrendelés befizetésének beregisztrálása' linket. Ekkor az adott megrendeléshez tartozó termékek mindegyikénél (a 'megrendelesek' táblában) a 'fizetve' mező értéke 1-re lesz állítva. Ha a megrendelés kézbesítése megtörtént, az adminisztrátor a 'Megrendelés kézbesítésének beregisztrálása' linkre kattintva az adott megrendeléshez tartozó minden termék esetében a 'kezbesítve' mező értéke 1-re lesz állítva.

Ezen beállítások után a megrendelés megtekintésekor a termékek listázása után a megrendelések részleteinek felsorolásakor már az új értékek fognak megjelenni. Ha az adminisztrátor megtekinti a leadott megrendelések listáját, akkor például a következőt láthatja:

[2009-3-10 12:12:4](#) Fizetve Kézbesítve  
[2009-4-6 9:37:28](#) Fizetve  
[2009-4-10 18:0:42](#)

Fel vannak sorolva a megrendelések, és mindegyik megrendelés mellett megjelenik, hogy az ki van-e már fizetve, illetve sor került-e már a kézbesítésre. Ezáltal az adminisztrátor gyorsan áttekintheti a megrendelések állapotát, így a valamilyen okból elmaradt megrendelések esetében gyorsan intézkedhet. Mivel a megrendelések időpont szerint kerülnek kijelzésre, így

nem zavarja felesleges információ, és gyorsan átláthatja az időpontokat. Ha plusz információra van szüksége, az adott időpontra (linkre) kattintva mindent megtudhat.



## Összefoglalás

Ezzel végére értünk az alkalmazás megismerésének. Megismerhettük az alkalmazás három fő funkcióját: a fórumot, a megfigyelő oldalt és a webshopot. Láthattuk az ezekben rejlő lehetőségeket. Az alkalmazást két szempont szerint ismertük meg: egyrészt az átlagfelhasználó oldaláról, másrészt bepillantást kaphattunk a háttérben zajló folyamatokról, melyet az oldal adminisztrátora végezhet. Az alkalmazás ismertetése folyamán megemlítésre kerültek a legfőbb funkciók algoritmikus megoldásának mikéntje, és esetekben tárgyalásra került egy adott PHP eszköz.

Pár mondatban be lettek mutatva az alkalmazott eszközök: a PHP, a MySQL, és Apache.

Az alkalmazás témájának azért választottam a meteorológiát, mert meglátásom szerint napjainkban egyre nagyobb szerepet kap ez a tudományág. Habár már eddig is nagyon sok mindenhez segítségül hívták a meteorológiát, mint például a mezőgazdaság, vagy bizonyos fontosabb, nagyobb tömegeket érintő események, rendezvények lebonyolításának kapcsán, ezután várhatóan még nagyobb figyelem fog a tudomány ezen ágára szegeződni. Egyrészt mert ezáltal bizonyos dolgokat valószínűleg hatékonyabban, nyereségesebben lehet véghezvinni, másrészt talán azért, mert a napjainkban oly sokat hallott problémáról, mint a globális felmelegedésről és ennek következményeiről tudhatunk meg többet. Meglátásom szerint ma az emberek egyre nagyobb része kezd hajlani annak irányába, hogy valamilyen szinten megismerjék közvetlen környezetüket, így az időjárást, ezért is találtam fontosnak egy ilyen oldal készítését, ahol egy helyen megtalálhatják mindazt, amire szükségük lehet e témakörben.

Napjainkra drasztikusan megnőtt a dinamikus weboldalak iránti igény, és várhatóan a továbbiakban is növekvő tendenciát fog ez az igény mutatni. Ezért is választottam azt, hogy készítek egy dinamikus weboldalt melynek a megvalósításához az egyik legnépszerűbb nyelvet, a PHP-t választottam. Minden dinamikus weboldalra elmondható, hogy adatbázissal állnak kapcsolatban, ez ugyanis leegyszerűsíti, megkönnyíti az adatok kezelését. Az alkalmazás készítésekor a MySQL adatbázis-kezelőt használtam, melyről elmondható, hogy szintén széles körben elterjedt termék. Web-szerverként pedig a szintén csak nagy népszerűségnek örvendő Apache volt segítségemre.

## Irodalomjegyzék

- PHP
  - <http://www.php.net>
  - <http://hu.wikipedia.org/wiki/PHP>
  
- MySQL
  - <http://hu.wikipedia.org/wiki/MySQL>
  
- Apache
  - [http://hu.wikipedia.org/wiki/Apache\\_HTTP\\_Server](http://hu.wikipedia.org/wiki/Apache_HTTP_Server)

## **Köszönetnyilvánítás**

*Köszönöm Dr. Rutkovszky Edéné egyetemi tanársegédnek a dolgozat elkészítéséhez nyújtott segítségét.*