

Dr. Pál László, Sapientia EMTE, Csíkszereda

WEB PROGRAMOZÁS

5.ELŐADÁS

2015-2016

Sütik és munkamenetek kezelése

A HTTP protokoll – Emlékeztető

2

- A HTTP protokoll állapotmentes
- Nem emlékezik az előző kérés adataira
- Függetlenül kezeli a kéréseket
- Gyakran szükséges a kliens azonosítása és a hozzá kapcsolódó adatok tárolása
- Megoldás
 - ▣ Munkamenetek (Session)

Munkamenet példák

3

- Bejelentkezés alapú alkalmazások:
 - ▣ Levelezés
 - ▣ Internet bank
 - ▣ Facebook
- Bejelentkezés nélküli alkalmazások
 - ▣ Webshop – kosár

Állapotmentesség feloldása

4

- az alkalmazásnak kell gondoskodnia az állapot megtartásáról
- Állapot megtartása: az adat kliensenként történő megőrzése
 - ▣ Pld. webshop kosara – kliensenként eltérő kosarak
- Megvalósítás
 - ▣ kliens oldalon
 - ▣ szerver oldalon

Kliens oldali állapottartás

5

- Az adatot a kliens gépén tároljuk
- Minden kéreésnél felküldjük az adatokat a szerverre
- A szerver visszaadja a kliensnek
- Kliens oldali technológiák
 - ▣ URL
 - ▣ Rejtett mező
 - ▣ Süti

Állapottartás URL-ben

6

- URL querystring részében adjuk tovább az adatot
 - Pld. `szamol.php?samlalo=1`
- Hátrányok:
 - Minden linkhez oda kell generálni
 - Sok adat nem fér el benne
 - URL hossza legfeljebb 2kB
 - Feltűnő (zavaró)
 - Könnyen átírható

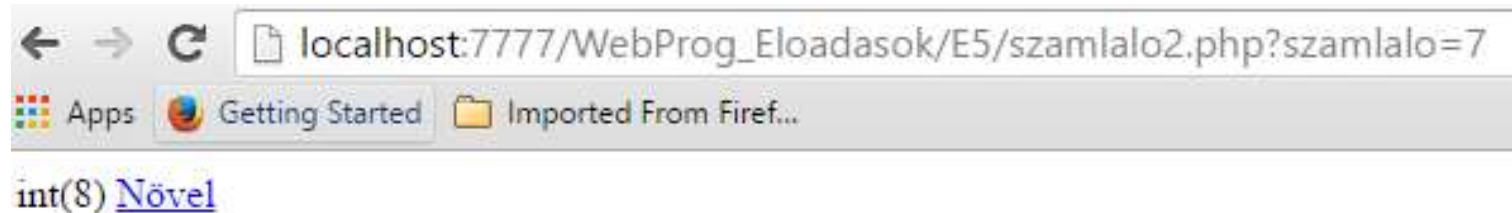
Állapottartás URL-ben

7

□ Példa:

```
<?php
$szamlalo = 0;
if (isset($_GET['szamlalo'])) {
    $szamlalo = $_GET['szamlalo'];
}
$szamlalo += 1;

var_dump($szamlalo);
?>
<a href="szamlalo2.php?szamlalo=<?php echo $szamlalo; ?>">Növel</a>
```



Állapottartás rejtett mezőben

8

□ Rejtett mező:

```
<input type="hidden" name="szamlalo" value="3">
```

□ Előnyök:

- Sok adat
- Nem feltűnő

□ Hátrányok:

- Manipulálható
- Csak űrlapok esetén használható

Állapottartás rejtett mezőben

9

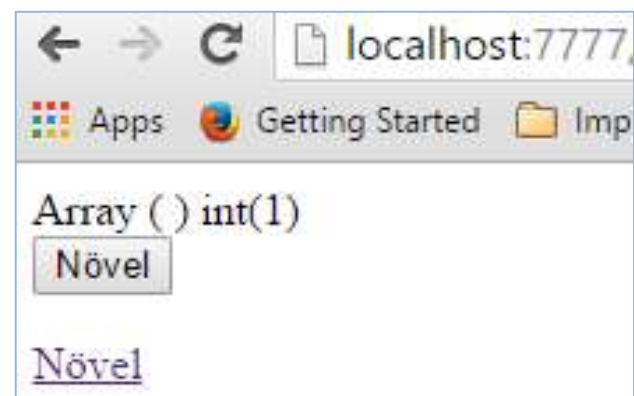
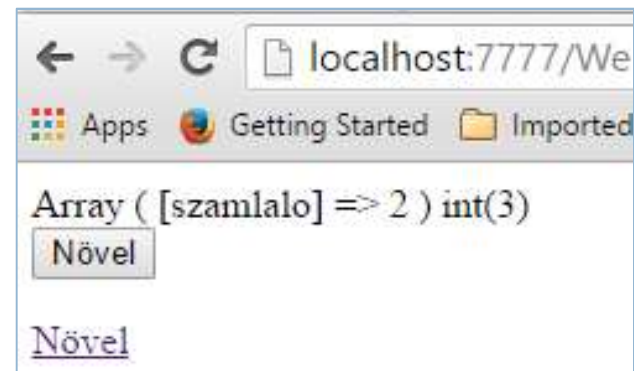
□ Példa:

```
<?php
print_r($_POST);
$szamlalo = 0;
if (isset($_POST['szamlalo'])) {
    $szamlalo = $_POST['szamlalo'];
}
$szamlalo += 1;

var_dump($szamlalo);
?>

<form action="szamlalo3.php" method="post">
    <input type="hidden" name="szamlalo"
        value="<?php echo $szamlalo; ?>">
    <input type="submit" value="Növel">
</form>

<!-- Ez nem működik -->
<a href="szamlalo3.php">Növel</a>
```



10

Állapottartás sütivel

Mi a süti (cookie)?

11

- Olyan kisebb adatmennyiségek, melyek egy-egy változó nevének és értékének tárolására szolgálnak, valamint arról a webhelyről is feljegyzik az információkat, melyről ezek az adatok származnak
- Kliensoldali adattárolást tesznek lehetővé
- A webhelyek általában csak a saját sütijeiket módosíthatják
- Az a webszerver érheti el és írhatja át, mely eredetileg kiküldte őket a kliensgépre

Mire használjuk a sütit?

12

- Kisebb lélegzetű, hasznos, ám nem kritikus fontosságú feladatok esetében érdemes használni
 - ▣ Felhasználói beállítások tárolása a webhelyek megjelenítését befolyásoló, testre szabható paraméterek (színek, betűtípusok) kezelésére
 - ▣ Felhasználói azonosító kulcsok tárolása, melyekkel a felhasználókról tárolt személyes adatokat érhetjük el

Hogyan használjuk a sütit?

13

- A kliens egy sütitet alkalmazó webhelyre látogat
- A webszerver arra utasítja a klienst, hogy későbbi használatra helyben tároljon el egy meghatározott adathalmazt
- A soron következő kérések alkalmával a kliens visszaküldi a megőrzött adatok másolatát
- A visszaküldött adatok alapján a szerver elvégzi a szükséges beállításokat

Hogyan használjuk a sütiket?

14

- A kliens oldalon az adatok tárolása egy előre meghatározott időtartam lejártáig tárolódnak, amit a szerver állít be másodpercben
- A sütik létrehozásuk után már a kliens fennhatósága alá tartoznak, tehát törölhetik ezeket akár a böngészőprogram segítségével is

Sütik létrehozása, beállítása

15

- A sütik kezelésére a PHP több módszert is felkínál. A módszerek között van olyan is ami direkt a `header()` paranccsal hajtódik végre, meg olyan is a mi függvényeket használ (`setcookie()`)
- Megjegyzés: A HTTP protokoll szerint a szerver először ún. fejléctet küld, ennek a fejlécnek lesz része a süti is. A HTML oldal a fejléc után kerül küldésre

Sütik létrehozása, beállítása

16

- A **setcookie** függvény: létrehoz egy sütit, ami a többi header információval együtt kerül az olvasó böngészőjéhez

- Szintaxis:

setcookie(name, value, expire, path, domain);

- **Name:** a süti neve
- **Value:** a névhez rendelt érték
- **Expire:** az az idő intervallum, amíg a süti használható
- **Path:** az elérési út a szerveren
- **Domain:** a domain ahol érvényes a süti

Sütik létrehozása, beállítása

17

- **Példa:** Az alábbi példában létrehozunk egy *user* nevű sütit és az érvényességi időt egy órában határozzuk meg.

```
<?php  
    setcookie("user", "Kiss Istvan", time()+3600);  
?>
```

Sütik kiolvasása

18

- A `$_COOKIE['valtozo']` globális tömb segítségével történik a kiolvasás
- Példa:

```
<?php
setcookie("user", "Kiss Manzo", time()+3600);
?>
<html>
  <body>
    <?php
      if (isset($_COOKIE["user"]))
        echo "Welcome " . $_COOKIE["user"] . "<br />";
      else
        echo "Welcome guest!<br />";
    ?>
  </body>
</html>
```

Sütik kiolvasása

19

Példa: számláló sütivel

```
<?php
    if (!isset($_COOKIE['visits'])) $_COOKIE['visits'] = 0;
    $visits = $_COOKIE['visits'] + 1;
    setcookie('visits', $visits, time() + 3600 * 24 * 365);
?>
<html>
<body>
<?php
    if ($visits > 1) {
        echo("Ez a $visits. látogatásod");
    } else {
        // Elso latogatas
        echo'Üdvözöllek a honlapomon!';
    }
?>
</body>
</html>
```

Sütik kiolvasása

20

□ Előző példa HTTP kérése és válasza:

Kérés

```
▼ Response Headers    view parsed
HTTP/1.1 200 OK
Date: Tue, 17 Nov 2015 08:35:53 GMT
Server: Apache/2.4.16 (Win32) OpenSSL/1.0.1p PHP/5.6.12
X-Powered-By: PHP/5.6.12
Set-Cookie: visits=2; expires=Wed, 16-Nov-2016 08:35:53 GMT; Max-Age=31536000
Content-Length: 189
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```

Válasz

```
Request Headers    view parsed
GET /WebProg_Eloadasok/E5/szamlalo.php HTTP/1.1
Host: localhost:7777
Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, e Gecko) Chrome/46.0.2490.86 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8,hu;q=0.6,und;q=0.4
Cookie: visits=1; firstVisitTime=1447257360; PHPSESSID=a3bdm71f1s78bb46mi51
```

Süti törlése

21

- A süti törlése gyakorlatilag azt jelenti, hogy a sütit lejáratnak állítjuk be (az aktuális időből kivonunk egy számot)

```
<?php  
    setcookie("user", "", time()-3600);  
?>
```

Példa: login ablak sötivel

22

login.html

```
<form name="login" method="post" action="login.php">  
  Username: <input type="text" name="username"><br>  
  Password: <input type="password" name="password"><br>  
  Remember Me: <input type="checkbox" name="rememberme" value="1"><br>  
  <input type="submit" name="submit" value="Login!">  
</form>
```

User Login

Username:

Password:

Remember Me: ☐

Példa: login ablak sötivel

23

login.php

```
$user = 'alma';
$pass = 'alma';
if (isset($_POST['username']) && isset($_POST['password'])) {
    if (($_POST['username'] == $user) && ($_POST['password'] == $pass)) {

        if (isset($_POST['rememberme'])) {
            setcookie('username', $_POST['username'], time()+60*60*24*365);
            setcookie('password', md5($_POST['password']), time()+60*60*24*365);
        }
        else {
            setcookie('username', $_POST['username'], time()- 3600);
            setcookie('password', md5($_POST['password']), time()-3600);
        }
        header('Location: index.php');

    } else {
        echo 'Username/Password Invalid';
    }
} else {
    echo 'You must supply a username and password.';
}
```

Példa: login ablak sütivel

24

index.php

```
$user = 'alma';  
$pass = 'alma';  
  
if (isset($_COOKIE['username']) && isset($_COOKIE['password'])) {  
    if (($_COOKIE['username'] != $user) || ($_COOKIE['password'] != md5($pass)))  
        header('Location: login.html');  
    else  
        echo 'Welcome back ' . $_COOKIE['username'];  
}  
else  
    header('Location: login.html');
```


Kliensoldali megoldások

25

- Adat a kliensen van
- Manipulálható
- Sok adat esetén feleslegesen sok adat megy oda-vissza a kliens és szerver között
- Szerveroldali megoldások
 - ▣ ?????

Szerveroldali megoldások

26

- Tároljuk az adatot a szerveren
 - ▣ nem manipulálható kliens oldalon
 - ▣ nem kell sok adatot küldözgetni
- A kliens megkülönböztetése továbbra is szükséges
 - ▣ ID-t kap, amivel azonosítja magát és hozzáfér a ID-hoz tartozó adatokhoz
- Az ID kliensoldali megoldással közlekedik
 - ▣ süti (alapértelmezett)
 - ▣ url (ha nincs süti)

27

Munkamenet (session) kezelés

Munkamenet - Bevezető

28

- Ismert tény, hogy a HTTP protokoll állapotmentes, azaz a kliens nem követhető két kérés között
- A valós életben előfordulnak olyan helyzetek, amikor szükséges a kliens nyomon követése különböző kérések között
- Session: a kliens és a webszerver között egyedileg végbemenő műveletek sorozatai, amelyek hosszabb időn keresztül is fennállnak

Munkamenet - Bevezető

29

- Munkafolyamat lehet például valamilyen internetes tranzakció kezelés vagy elektronikus postafiók ellenőrzés, stb.
- A sütiktől eltérően, a munkafolyamat-ban az adatokat a szerveren tartjuk, a kliensnek biztosítunk egy egyedi kulcsot (**Session ID**), amely egyértelműen azonosítja azt és a hozzá tartozó szerveroldali adatokat is.

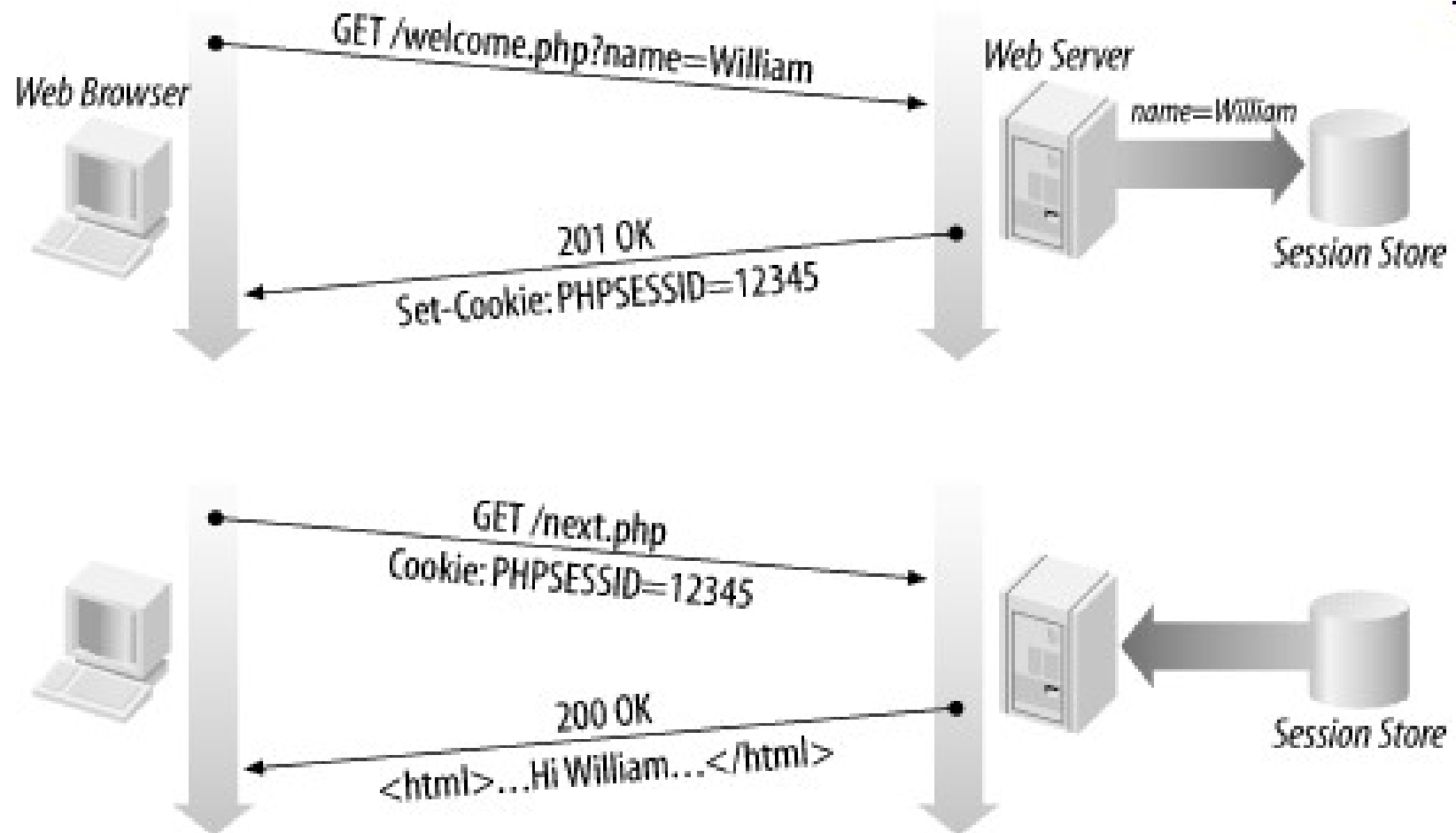
Hogyan működik?

30

- Amikor a látogató először érkezik oldalunkhoz, generálunk egy azonosítót, amit eljuttatunk a klienshez oly módon, hogy azt a kliens a látogató következő oldal lekérésekor eljuttassa a szerverre.
- Ezután egy ilyen azonosító (sessionid) egy adott felhasználó munkamenetét fogja jelképezni, és ezen azonosító alapján a szerveren állapotinformációkat, adatokat tárolhatunk
- Kliens oldalon ez az egyedi azonosító (session id) vagy sütiben vagy a böngésző címsorában(ez a ritkább) tárolódik

Hogyan működik?

31



Munkamenet kezelése PHP-ben

32

- Számos függvény létezik ezek kezelésére
- Munkamenet indítása: mielőtt információkat tárolnánk, el kell indítani a munkamenetet

```
<?php session_start(); ?>
<html>
  <body>
    </body>
</html>
```

- Az előbbi kód regisztrálja a session-t a szerveren, engedélyezi a felhasználói információk mentését, és a session azonosítót továbbítja a felhasználó számára

Munkamenet kezelése PHP-ben

33

- Adatok elhelyezése a munkamenetben: a `session_start()` parancsot követően létrejön a `$_SESSION` nevű tömb, ami egyrészt tartalmazza a munkamenet során már korábban elhelyezett adatokat, valamint újabbakat tehetünk bele

Munkamenet kezelése PHP-ben

34

□ Példa: egyszerű kattintás számláló

```
<?php
session_start();

if (isset($_SESSION['counter'])) {
    print $_SESSION['counter'];
    $_SESSION['counter']++;
    print "<a href=\"counter.php\">növel</a>";
}
else {
    $_SESSION['counter'] = 1;
    print $_SESSION['counter'];
    print "<a href=\"counter.php\"> növel</a>";
}

print "<br> A szesszióazonosító:".session_id();

?>
```

Kimenet

22[növel](#)
A szesszióazonosító:evtrcgitp8vl1fh8oq2hb8lmr5

Munkamenet kezelése PHP-ben

35

□ Példa: egyszerű kattintás számláló

Request Headers view parsed

```
GET /WebProg_Laborok/L8/sessions/counter.php HTTP/1.1
Host: localhost:7777
Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, e Gecko) Chrome/46.0.2490.86 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8,hu;q=0.6,und;q=0.4
Cookie: firstVisitTime=1447257360; PHPSESSID=a3bdm71f1s78bb46mi51v8d9n5
```

Response Headers view parsed

```
HTTP/1.1 200 OK
Date: Tue, 17 Nov 2015 09:30:49 GMT
Server: Apache/2.4.16 (Win32) OpenSSL/1.0.1p PHP/5.6.12
X-Powered-By: PHP/5.6.12
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, Pragma: no-cache
Content-Length: 87
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```

Munkamenet kezelése PHP-ben

36

□ Példa: session típusú számláló

```
<?php
session_start();
if (isset($_SESSION['view'])) {
    $_SESSION['view'] = $_SESSION['view'] + 1;
} else {
    $_SESSION['view'] = 1;
}
echo "Page Views Count On This Session:" . $_SESSION['view'];
?>
```

Munkamenet törlése

37

- A session adatok törlése az *unset()* vagy a *session_destroy()* függvényekkel történik.
 - ▣ Az *unset()* függvényt a session változó törlésére használjuk
 - ▣ A session teljes törlése a *session_destroy()* függvénnyel lehetséges

```
<?php
    unset($_SESSION['counter']);
?>
```

Példa: adat továbbvitele másik oldalra

38

```
<?php
session_start();
?>
<html>
  <head>
    <title>sessionok gyakorlása</title>
  </head>
  <body>
    <form action="" method="post">
      Név: <input type="text" name="nev" />
      <input type="submit" value="mehet" name="mehet" />
    </form>
  </body>
</html>
<?php
if (isset($_POST['mehet']))
{
    $_SESSION['nev'] = $_POST['nev'];
    print "<a href=\"kiir.php\">Tovább</a>";
}
?>
```

Példa: adat továbbvitele másik oldalra

39

kiir.php

```
<?php
session_start();
?>
<html>
  <head>
    <title>Session_gyakorlás</title>
  </head>
  <body>
    <?php
      print "<h1>Üdvözöllek ".$_SESSION['nev']."</h1>";
    ?>
  </body>
</html>
```

Hitelesítés munkamenetekkel

40

Login.php

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

<html>
    <!--<meta http-equiv="refresh" content="6">-->
    <form name="form1" method="post">
        <table>
            <tr>
                <td>Username</td>
                <td><input type="text" name="text"></td>
            </tr>
            <tr>
                <td>Password</td>
                <td><input type="password" name="pwd"></td>
            </tr>
            <tr>
                <td><input type="submit" value="SignIn" name="submit1"></td>
            </tr>
        </table>
    </form>
</html>
```

```
<?php
    if (isset($_POST['submit1'])) {
        $v1 = "admin";
        $v2 = "admin";
        $v3 = $_POST['text'];
        $v4 = $_POST['pwd'];
        if ($v1 == $v3 && $v2 == $v4) {
            $_SESSION['luser'] = $v1;
            $_SESSION['start'] = time(); // Taking now logged in time.
            // Ending a session in 30 minutes from the starting time.
            $_SESSION['expire'] = $_SESSION['start'] + 10;
            header('Location: HomePage.php');
        } else {
            echo "Please enter the username or password again!";
        }
    }
?>
```

Username	<input type="text"/>
Password	<input type="password"/>
<input type="submit" value="SignIn"/>	

Hitelesítés munkamenetekkel

41

HomePage.php

```
<?php
    session_start();

    if (!isset($_SESSION['luser'])) {
        echo "Please Login again";
        echo "<a href='Login.php'>Click Here to Login</a>";
    }
    else {
        $now = time(); // Checking the time now when home page starts.

        if ($now > $_SESSION['expire']) {
            session_destroy();
            echo "Your session has expired! <a href='Login.php'>Login here</a>";
        }
        else { //Starting this else one [else1]
            <!-- From here all HTML coding can be done -->
            <html>
                Welcome
                <?php
                    echo $_SESSION['luser'];
                    echo "<a href='Logout.php'>Log out</a>";
                ?>
            </html>
        }
    }
?>
```

Welcome admin [Log out](#)

Hitelesítés munkamenetekkel

42

LogOut.php

```
<?php
    session_start();
    session_destroy();
    header('Location: Login.php');
?>
```

Welcome admin [Log out](#)



Your session has expired! [Login here](#)



Username	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="SignIn"/>	

WordPress projekttema javaslatok

43

□ **Webshop bővítmények:**

- WP e-Commerce Plugin
- WP Online Store
- Woocommerce Plugin
- eShop Plugin
- Cart66 Lite
- Jigoshop WordPress e-commerce Plugin
- Quick Shop Plugin
- YAK shopping cart Plugin
- Zingiri WebShop v 2.5.9 magyar fordítás

□ **Leírás magyarul a fenti bővítményekről:**

<http://efrud.hu/a-nagy-wordpress-webaruhaz-bovitmeny-szemle/>

WordPress projekttéma javaslatok

44

□ Étterem sablon:

- Confit (<http://theme.wordpress.com/themes/confit/>)
- Easy Restaurant Menu Manager
- WPPizza
- ReDi Restaurant Reservation

A témával kapcsolatosan itt lehet még olvasni:

<http://www.moeseo.com/10-best-wordpress-plugins-for-a-restaurant-website/>

□ Online rendelés/kölcsönzés:

- Checkfront Online Booking System
- Rezgo Online Booking

A témával kapcsolatosan itt lehet még olvasni:

<http://www.wpmayor.com/plugin-reviews/best-wordpress-booking-plugins/>