

## SZOFTVERLELTÁR (GYAKORLÁS) FELADAT MEGOLDÁSA

Ma már a legtöbb intézményben természetes tartozék a számítógép, majdnem minden irodában megtalálható. A telepített hardver- és szoftvereszközökről általában automatikusan generált nyilvántartás készül. Az adatbázis egy oktatással foglalkozó cég nyilvántartásának szoftverekre vonatkozó részéből tartalmaz kivonatot.

### Az adatbázis és szerkezetének kialakítása

```
CREATE DATABASE IF NOT EXISTS szoftverleltar
DEFAULT CHARACTER SET utf8mb4 (1)
COLLATE utf8mb4_hungarian_ci;

USE szoftverleltar;

CREATE TABLE szoftverleltar.gep ( (2)
id INT(2) NOT NULL PRIMARY KEY AUTO_INCREMENT , (3)
hely VARCHAR(4) NOT NULL , (4)
tipus VARCHAR(8) NOT NULL , (5)
ipcim VARCHAR(12) NOT NULL (6)
) ENGINE = InnoDB CHARSET=utf8mb4 COLLATE utf8mb4_hungarian_ci; (7)

CREATE TABLE szoftverleltar.szoftver ( (8)
id INT(3) NOT NULL PRIMARY KEY , (9)
nev VARCHAR(33) NOT NULL , (10)
kategoria VARCHAR(26) NOT NULL (11)
) ENGINE = InnoDB CHARSET=utf8mb4 COLLATE utf8mb4_hungarian_ci; (12)

CREATE TABLE szoftverleltar.telepites ( (13)
id INT(4) NOT NULL PRIMARY KEY AUTO_INCREMENT , (14)
gepid INT(2) NOT NULL , (15)
szoftverid INT(3) NOT NULL , (16)
verzio VARCHAR(17) NULL , (17)
datum DATE NULL , (18)
INDEX gepid_index (gepid) , (19)
INDEX szoftverid_index (szoftverid) , (20)
CONSTRAINT gep_megszoritas FOREIGN KEY (gepid) REFERENCES gep (id) (21)
ON DELETE CASCADE,
CONSTRAINT szoftver_megszoritas FOREIGN KEY (szoftverid) REFERENCES szoftver (id) (22)
ON DELETE CASCADE
) ENGINE = InnoDB CHARSET=utf8mb4 COLLATE utf8mb4_hungarian_ci; (23)
```

### Programozás

Először az adatbázishoz való kapcsolódás és a karakterkódolás beállítása az első lépés. Az adatbázis nevét megadjuk egy változóban és annak tábláit pedig egy tömbben. Erre azért van szükség, mivel az adatbázis műveletekhez egy univerzális megoldást szeretnénk használni: a tömb elemeire hivatkozva, majd azon belül azok mezőit bejárva végezzük el az adatmanipulációs eljárásokat valamint lekérdezve a rekordokat, ezek segítségével helyezzük el azok mezőit a weboldalon létrejövő táblázatokban:

```

<body>
<?php
    $adatbazis="szoftverleltar";
    $kapcs=mysqli_connect("localhost","root","","$adatbazis"); 24
    mysqli_set_charset($kapcs,"UTF8"); 25
    mysqli_select_db($kapcs,$adatbazis); 26
    $tablak=array("gep","szoftver","telepites");

```

Ezek után, ha nem került elküldésre az egyik 'submit' típusú gomb sem, azaz nem lett elküldve semmilyen paraméter, nincs beállítva a \$\_GET szuperglobális tömb egy eleme sem, az URL a php fájl nevére végződik, akkor alaphoz az űrlap kerüljön megjelenítésre, a megfelelő méretekben, mezőhosszokkal, alapértelmezett értékekkel, szükséges/opcionális, csak olvasható vagy írható megkötésekkel:

```

if (!isset($_GET['megnez']) && !isset($_GET['felvesz']) && !isset($_GET['torol']))
{
    // echo " <form name=urlap action='szoftverleltar.php' method=GET novalidate>\n";
    // Teszteléskor megadhatjuk a novalidate paramétert, hogy ne ellenőrizze az űrlap értékeit.
    echo " <form name=urlap action='szoftverleltar.php' method=GET novalidate>\n"; 27
    echo " <h1>Szoftverleltár</h1>\n"; 28
    echo " <fieldset>\n";
    echo " <legend align=right>Szoftverleltár</legend>\n"; 30
    echo " <fieldset>\n";
    echo " <legend>Gép</legend>\n"; 32
    echo " <p><label for=gepid>Id: </label>\n";
    echo " <input type=text name=gepid value=NULL size=4 readonly> 33
    Értéke automatikusan növekvő szám, így NULL érték kerül elküldésre.</p>\n";
    echo " <p><label for=hely>Hely: </label>\n";
    echo " <input type=text name=hely maxlength=4 size=4 required></p>\n"; 34
    echo " <p><label for=tipus>Típus: </label>\n";
    echo " <input type=text name=tipus maxlength=8 size=8 required></p>\n"; 35
    echo " <p><label for=ipcim>IP-cím: </label>\n";
    echo " <input type=text name=ipcim maxlength=12 size=12 required></p>\n"; 36
    echo " </fieldset>\n"; 37
    echo " <fieldset>\n"; 38
    echo " <legend>Szoftver</legend>\n"; 38
    echo " <p><label for=szoftverid>Id: </label>\n";
    echo " <input type=number name=szoftverid min=1 max=999 required></p>\n"; 39
    echo " <p><label for=nev>Név: </label>\n";
    echo " <input type=text name=nev maxlength=33 size=33 required></p>\n"; 40
    echo " <p><label for=kategoria>Kategória: </label>\n";
    echo " <input type=text name=kategoria maxlength=26 size=26 required></p>\n"; 41
    echo " </fieldset>\n"; 42
    echo " <fieldset>\n"; 43
    echo " <legend>Telepítés</legend>\n"; 43
    echo " <p><label for=telepitesid>Id: </label>\n";
    echo " <input type=text name=telepitesid value=NULL size=4 readonly> 44
    Értéke automatikusan növekvő szám, így NULL érték kerül elküldésre.</p>\n";
    echo " <p><label for=kulso_gepid>Gép id: </label>\n";
    echo " <input type=number name=kulso_gepid min=1 max=99 required></p>\n"; 45
    echo " <p><label for=kulso_szoftverid>Szoftver id: </label>\n";
    echo " <input type=number name=kulso_szoftverid min=1 max=999 required></p>\n"; 46
    echo " <p><label for=verzio>Verzió: </label>\n";
    echo " <input type=text name=verzio maxlength=17 size=17></p>\n"; 47
    echo " <p><label for=datum>Dátum: </label>\n"; 48
    echo " <input type=date name=datum></p>\n";
    echo " </fieldset>\n";
    echo " <input type=submit name=megnez value='Megnéz'>\n";
    echo " <input type=submit name=felvesz value='Felvesz'>\n"; 49
    echo " <input type=submit name=torol value='Töröl'>\n";
    echo " </fieldset>\n";
    echo " </form>\n";
    echo "\n";
}

```

Ha a 'megnéz' gomb lett elküldve, akkor listázásra kerülnek az adatrekordok HTML táblázat formájában. De mivel ezt az összes gombra kattintás esetében meg kell tenni, ezért ezt egy külön eljárásban érdemes megírni és azt meghívni mindhárom esetben:

```

else if (isset($_GET['megnevez']))
    listaz();

```

```

function listaz() (50)
{
    // Egy kétdimenziós tömbben megkapjuk a táblák, azon belül mezők
    $tomb=tombbe(); // paraméterként elküldött értékeit
    foreach ($tomb as $tabla => $altomb) // Ciklussal végighaladok a tömbben, melynek
    { // elemei a táblák nevei, alelemei pedig szintén tömbök (a mezőneveikkel)
(51) echo " <details>\n"; // Az egyes táblákat ezzel összecsuksukhatóvá/lenyithatóvá
        echo " <summary>$tabla tábla tartalma</summary>\n"; // tesszük.
        ${$tabla}=mysqli_query($GLOBALS["kapcs"],"SELECT * FROM $tabla"); (52)
        $lekerdezes=${$tabla}; // Változó nevű változók: a lekérdezések neve -
        echo " <table border=1>\n"; // $gep, $szoftver, $telepites lesznek.
        echo " <tr>\n"; (53) // A mezők neveihez az altömbön is végigmegyünk
        foreach ($altomb as $mezo => $ertek) // egy ciklussal és mezőfejlécekként
            echo " <th>$mezo</th>\n"; // iratjuk ki azokat.
        echo " </tr>\n"; (53)
        mysqli_data_seek($lekerdezes,0); // A lekérdezett tábla
        while ($rekord=mysqli_fetch_array($lekerdezes)) // első rekordjára pozicionálunk.
(54) { // Ciklus segítségével kiolvastatjuk az adatrekordokat.
            echo " <tr>\n"; // A sorokon belül
            for ($k=0; $k<mysqli_num_fields($lekerdezes); $k++) // ciklussal végigpásztázzuk
            { // az egyes mezőket, hogy azok
                $mezonev=$lekerdezes->fetch_field_direct($k)->name; // sorszám szerinti nevei
                $ertek=$rekord[$mezonev]; // és a rekordban lévő értékei
                echo " <td>$ertek</td>\n"; // a megfelelő cellába kerüljön a táblázatban.
            }
            echo " </tr>\n"; (54)
        }
        echo " </table>\n";
        echo " </details>\n";
        echo " <br><br>\n";
        mysqli_free_result($lekerdezes);
    }
}

```

Figyeljük meg, hogy a 'listaz' eljárás első sorában egy 'tombbe' nevű függvény kerül meghívásra, ami visszatérési értéként a \$\_GET szuperglobális tömb paramétereit és azok értékeit szortírozza szét az adatbázis tábla elnevezései valamint azok mezőnevei szerinti többdimenziós, asszociatív tömbbe:

C:\Users\Tanar\Dropbox\enyim\www\tanfolyam\szoftverleltar.php:142:

```

array (size=3)
    'gep' =>
        array (size=4)
            'id' => string 'NULL' (length=4)
            'hely' => string 'T104' (length=4)
            'tipus' => string 'asztali' (length=7)
            'ipcim' => string '192.168.1.1' (length=11)
    'szoftver' =>
        array (size=3)
            'id' => string '107' (length=3)
            'nev' => string 'Adobe Photoshop' (length=15)
            'kategoria' => string 'grafikus' (length=8)
    'telepites' =>
        array (size=5)
            'id' => string 'NULL' (length=4)
            'gepid' => string '1' (length=1)
            'szoftverid' => string '2' (length=1)
            'verzio' => string 'ver19.1' (length=7)
            'datum' => string '2018-03-14' (length=10)

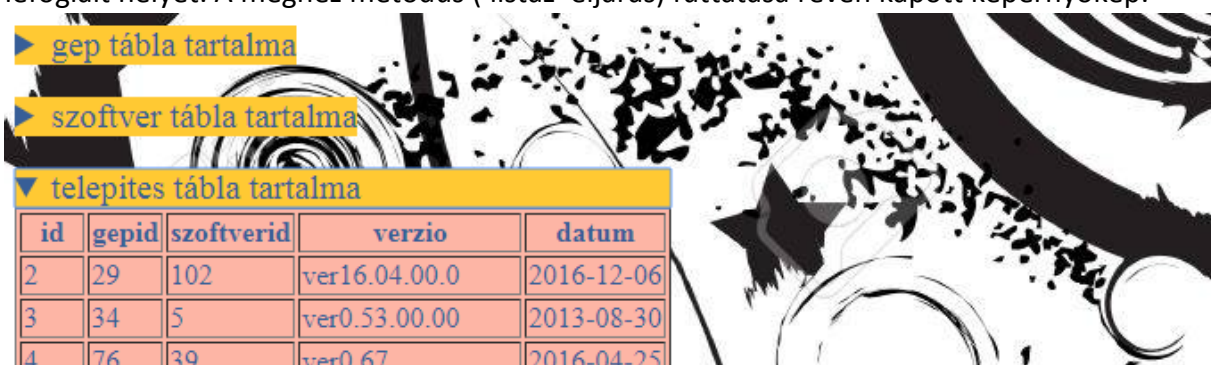
```

De hogyan jön létre ez a tömb? A már említett 'tombbe' nevű függvénnyel:

```
function tombbe()
{
    $i=0;
    $parameterek=array();
    foreach ($_GET as $urlapelem_nev => $ertekek) // A küldött paraméterek
        $parameterek[$i++]=$ertekek; // összegyűjtése egy indexelhető tömbben.
    array_pop($parameterek); // Az utolsó elem, az elküldött gomb törlése.
    $tablak=$GLOBALS["tablak"]; // Globális változóvá tesszük, mivel a függvényen kívüli
    $n=0;
    foreach ($tablak as $index => $tabla) // Az adatbázis tábláinak bejárása
    {
        ${$tablak[$index]}=mysql_query($GLOBALS["kapcs"],"SELECT * FROM $tabla");
        $lekerdezes=${$tablak[$index]}; // Változó nevű változók: a lekérdezések neve -
        // $gep, $szoftver, $telepites lesznek.
        $mezo_darab=$lekerdezes->field_count; // Így kinyerhettük azok meződarabszámait
        $mezoinfo=mysql_fetch_field($lekerdezes); // Az adott tábla egyes mezőinek
        do { // a tulajdonságaira vagyunk kíváncsiak, ezek közül is a mezők neveire.
            $tomb[$tabla][$mezoinfo->name]=$parameterek[$n];
            $n++;
        } while ($mezoinfo=mysql_fetch_field($lekerdezes)); // Ciklusban végigvesszük.
    }
    //echo var_dump($tomb); // Kiirathatjuk, hogy milyen tömböt ad vissza.
    //echo $tomb["gep"]["tipus"]; // Így lehet rá például hivatkozni.
    return $tomb;
}
```

A \$\_GET asszociatív tömb – amely a küldött űrlap elem neveit és azok megadott értékeit tartalmazza – egy indexelhető, 'parameterek' nevű tömbbe helyezzük át, mintegy konvertálva azt. Az utolsó elemét töröljük az 'array\_pop' függvénnyel, ami az elküldött gomb, mivel erre nem lesz szükségünk. A php forráskódunk elején megadott táblák tömböt bejárjuk és sorban lekérdezzük a tartalmukat, amiket változó nevű lekérdezés változókba helyezünk el. Mindezt azért, hogy hozzájussunk az egyes táblák mezőinek darabszámaihoz, illetve egy hátultesztelős ciklust használva egyéb mezőinfókhoz. A ciklusmagban felépítjük a tömböt, melynek „tábla-adik” és „mezőnevedik” pozíciójába az indexelhető tömb n-edik paraméterértéke kerül. Az így feltöltött tömb kerül átadásra, mint visszatérési érték.

Visszatérve a 'listaz' eljárásra az így megkapott kétdimenziós tömböt 'foreach' ciklussal végigpásztázzuk, amelynek első dimenziójában az adatbázis táblái helyezkednek el. Ezeknek kérdezzük le egyenként az összes rekordját változó nevű lekérdezés változókba. Majd elkezdjük a HTML tábla felépítését a kapott sorokból. Közben egy belső 'foreach' ciklussal az adott tábla mezőin lépegetünk végig, mivel a mezőnevekre szükségünk van a táblázatunk mezőneveinek, mezőfejléceinek megjelenítéséhez. A rekordok HTML táblázatban történő rögzítéséhez egy előltesztelős ciklussal végighaladunk a lekérdezés eredményeként megkapott sorokon (előtte nem baj, ha az elsőre pozícionálunk). Egy ezen belüli ciklussal a kapott rekordok mezőit vesszük sorra, hogy a HTML táblázatunk celláiba el tudjuk őket helyezni. Majd zárjuk a táblázatot, a 'details' elemet és felszabadítjuk a lekérdezés által lefoglalt helyet. A megnéz metódus ('listaz' eljárás) futtatása révén kapott képernyőkép:



gep tábla tartalma

id	gepid	szoftverid	verzio	datum
2	29	102	ver16.04.00.0	2016-12-06
3	34	5	ver0.53.00.00	2013-08-30
4	76	39	ver0.67	2016-04-25



Ha a 'felvesz' gomb lett elküldve, akkor a következő szelekciós ág hajtódik végre:

```
else if (isset($_GET['felvesz']))
{
    // Egy kétdimenziós tömbben megkapjuk a táblák, azon belül mezők
    $tomb=tombbe(); // paraméterként elküldött értékeit
    foreach ($tomb as $tabla => $altomb) // Ciklussal végighaladok a tömbben, melynek
    { // elemei a táblák nevei, alegelei pedig szintén tömbök (a mezőneveikkel)
55 echo " <span><b>$tabla</b> tábla tartalma a következő rekorddal bővült:</span>\n";
    echo " <table border=1>\n"; // Tájékoztatás, hogy most épp melyik tábla fog bővülni
    echo " <tr>\n";
    $beszur_str="INSERT INTO $tabla VALUES ("; // A query string létrehozása
    foreach ($altomb as $mezo => $ertek) // Az altomb mezőelemeinek értékein is végigmegy
    {
        if ($ertek!="NULL") // Ha nem növekményes mező (auto_increment),
        $beszur_str .= "'" . $ertek . "',"; // akkor az aposztrófok
        else // szükségesek.
        $beszur_str .= $ertek . ","; // A query string bővítése
        echo " <td>$ertek</td>\n"; // Tájékoztatás, hogy épp milyen mezőértékkel
    } // fog bővülni az adott tábla
    $beszur_str=rtrim($beszur_str,""); // Az utolsó vessző törlése
    $beszur_str .= ")"; // A query string lezárása
    echo " </tr>\n";
    echo " </table>\n";
    echo " <br>\n";
    if (!mysqli_query($kapcs,$beszur_str)) // A lekérdezés (beszúrás) futtatása
    echo "MySQLi hiba (" . mysqli_errno($kapcs) . "): " . mysqli_error($kapcs) . "<br>\n";
    }
    listaz();
}
```

Itt is szeretnék megkapni az asszociatív tömböt táblákba, mezőikbe besorolva az elküldött paraméterértékeket. Majd az eddigiekben is használt egymásba ágyazott 'foreach' ciklusokkal végig megyünk a táblákon és azok mezőin miközben két dolgot teszünk: létrehozuk a feladat-specifikációban kért egysoros táblázatokat (amelyek az új rekordokat jelölik) valamint generáljuk a query sztringet, amely jelen esetben az adattáblákba történő beszúrást végző INSERT parancsok lesznek. Mivel a VALUE-ban megadott, vesszővel elválasztott, új mezőértékek utolsó eleme után már nem kell a ciklusban még utoljára kiadott utolsó vessző, így azt egy jobb „nyírással” (rtrim) töröljük. Végül a táblázat sorának és magának a táblázatnak a lezárása után futtatjuk a kapott karakterláncsal a 'mysqli\_query'-t, elvégezve az adott rekord, adott táblába történő beszúrását. Mindhárom tábla adatmanipulációjának elvégzése után listázzuk a tartalmukat, az eredmény nyomkövethetősége végett. A felvesz metódus kimenete:

gep tábla tartalma a következő rekorddal bővült:

NULL	T104	asztali	192.168.1.1
------	------	---------	-------------

szoftver tábla tartalma a következő rekorddal bővült:

107	Adobe Photoshop	grafikus
-----	-----------------	----------

telepites tábla tartalma a következő rekorddal bővült:

NULL	1	2	ver19.1	2018-03-14
------	---	---	---------	------------

▶ gép tábla tartalma

▼ szoftver tábla tartalma

id	nev	kategoria
1	AIR	plug-in
2	FastStone Image Viewer	képekezelés
5	InfraRecorder	DVD-írás
7	PDF Split And Merge Basic	pdf-manipulálás

A 'torol' gomb lenyomása esetén is hasonló a metódus:

```
else if (isset($_GET['torol']))
{
    // Egy kétdimenziós tömbben megkapjuk a táblák, azon belül mezők
    $tomb=tombbe(); // paraméterként elküldött értékeit
    foreach ($tomb as $tabla => $altomb) // Ciklussal végighaladok a tömbön, melynek
    { // elemei a táblák nevei, alegelemei pedig szintén tömbök (a mezőneveikkel)
        57 echo " <span><b>$tabla</b> táblából a következő mezőkre érvényes feltételek
            együttes megléte esetén törölődnek rekordok: "; // Tájékoztatás a törlendőkről
        58 $torol_str="DELETE FROM $tabla WHERE ("; // A query string létrehozása
        foreach ($altomb as $mezo => $sertek) // Az altomb mezőelemeinek értékein is végigmegy
        if ($sertek!="NULL") // Ha nem növekményes mező (auto_increment),
        {
            58 $torol_str .= "$mezo='$sertek' AND "; // akkor része a törlési feltételnek
            57 echo "<i>$mezo</i>='$sertek' és "; // és a tájékoztató szövegnek.
        }
        $torol_str=rtrim($torol_str," AND "); // Az utolsó vessző törlése
        $torol_str .= ")"; // A query string lezárása
        echo "más nem.</span><br><br>\n";
        if (!mysqli_query($kapcs,$torol_str)) 58 // A lekérdezés (törlés) futtatása
            echo "MySqli hiba (" . mysqli_errno($kapcs) . "): " . mysqli_error($kapcs) . "<br>\n";
    }
    listaz(); 59
}
```

A kapott tömb tábláit és mezőit bejárjuk azért, hogy szintén létrehozzunk egy query sztringet, de most egy törlésre alkalmasat. Közben még egy tájékoztató karakterláncot is generálunk, amely a megadott mezőket és a hozzá tartozó feltételeket közli velünk, amelyek, ha „összezsengenek” bizonyos rekordokkal, akkor azok törölődnek az adatbázisból. Ez a szelekció is a 'listaz' eljárás meghívásával zárul. A törlés metódus eredménye:

gép táblából a következő mezőkre érvényes feltételek együttes megléte esetén törölődnek rekordok: hely='T306' és típus='asztali' és ipcim='192.168.2.5' és más nem.

szoftver táblából a következő mezőkre érvényes feltételek együttes megléte esetén törölődnek rekordok: id='12' és nev='Acrobat Reader' és kategoria='pdf-olvasás' és más nem.

telepites táblából a következő mezőkre érvényes feltételek együttes megléte esetén törölődnek rekordok: gepid='3' és szofverid='34' és verzio='ver6.0.450' és datum='2016-12-07' és más nem.

	id	hely	típus	ipcim
1	T104	Monitor		192.168.1.1
3	T302	asztali		192.168.2.3
4	T108	notebook		192.168.1.1
5	T301	asztali		192.168.2.4
7	T209	notebook		192.168.4.1
8	T208	notebook		192.168.4.2
9	T110	notebook		192.168.1.2
10	T310	asztali		192.168.2.6

Legvégül lezárjuk a kapcsolatot:

```
mysqli_close($kapcs); 60
?>
</body>
```

A feladat-specifikációba lefektetett formázási beállítások háttérképpel, háttérszínnel, betűszínnel, szélességek megadásával, keretek- és betűkre vonatkozó beállításokkal, külső- és belső margókkal, lekerekített elemekkel, árnyékolásokkal, kurzor és szövegdekorációval, stb. a következőkben követhetjük nyomon:

```

body {
    color: #375E97; (61)
    background-image: url(szoftverleltar-hatter.jpg); (62)
}

form {
    background-color: #fdb5a5; (63)
    width: 800px; (64)
}

form h1 {
    text-align: center; (65)
    margin: 5px 20px 20px 20px;
    font: small-caps 50pt Impact;
} (66) (67) (68)

fieldset {
    background-color: #ffc933; (69)
    border: 1px solid #375E97; (61)
} (70) (71)

form, fieldset {
    margin: 20px auto; (96)
    padding: 20px; (96)
    border-radius: 10px; (72)
    (73) box-shadow: 5px 5px 15px 0px #375E97; (61)
}

legend {
    color: white; (74)
    background-color: #3F861C; (75)
    border-radius: 10px 20px 30px 40px; (76)
    padding: 2px 16px; (96)
    letter-spacing: 4px; (77)
}

details, table {
    background-color: #fdb5a5; (63)
    display: inline-block;
    width: auto;
}

summary, span {
    background-color: #ffc933; (69)
    font-size: larger; (78)
}

label {
    font-weight: bold; (79)
}

```

```

input:not([type=submit]) {
    padding-left: 5px; (80)
    border: 1px solid #662210;
    border-radius: 5px; (84)
    box-shadow: inset 0 -10px 20px #eeccc0;
} (81) (82) (83)

input[type=submit] {
    background: #3F861C;
    background-image: linear-gradient(to bottom, #3F861C, #375E97); (85) (86)
    border: solid #fdb5a5 3px;
    border-radius: 13px; (87)
    color: #FFFFFF; (88)
    font: 200 12pt Open Sans; (89) (90) (91)
    padding: 11px; (96)
    box-shadow: 1px 1px 20px 0px #000000; (92)
    text-shadow: 5px 3px 6px #000000; (93)
    text-decoration: none;
    display: inline-block;
    cursor: pointer; (94)
    margin-right: 15px; (96)
}

input[type=submit]:hover {
    background: #375E97;
    background-image: linear-gradient(to bottom, #375E97, #3F861C); (95)
    text-decoration: none;
}

```