

**TURUN YLIOPISTO**  
**Tietojenkäsittelytiede**  
Web ohjelmointi  
20.5.2008

Jyri Lehtonen (72039)

Periodi IV

# Harjoitustyö ”Fire Bird”

# Sisällysluettelo

<b>1. Tehtävän kuvaus</b> .....	<b>1</b>
<b>2. Web sivuston idea</b> .....	<b>1</b>
2.1 Ratkaisuperiaate .....	1
2.1 Rakennekaavio .....	2
<b>3. Kaskadiset tyyliohjeet</b> .....	<b>3</b>
<b>4. Tietokanta ja PHP</b> .....	<b>4</b>
<b>5. JavaScript toiminto</b> .....	<b>12</b>
<b>6. Web sivuston osien kuvaaminen</b> .....	<b>13</b>
6.1 Julkinen sivu ja kirjautuminen .....	13
6.2 Yksityinen sivusto.....	15
<b>7. Testausjärjestely</b> .....	<b>19</b>
<b>Liite 1: Ensimmäinen suunnitelma</b> .....	<b>22</b>
<b>Liite 2: Lopullinen suunnitelma</b> .....	<b>23</b>

## **1. Tehtävän kuvaus**

Tehtävänanto on luoda web-ohjelmointi tekniikoilla toimiva kokonaisuus, joko ryhmätyönä tai yksin. Sivustolla on ylläpidettävä johonkin aihepiiriin liittyvää tietoa ja mahdollistettava erilaisten käyttäjien näkymät ja toiminnot. Sivustossa on oltava ainakin seuraavat ominaisuudet: tiedot tietokannassa, ylläpitäjän näkymä, käyttäjän näkymä, autentikointi, sekä asiakaspuolen että palvelinpuolen ohjelmointia. Sivut eivät saa olla pelkästään staattisia. Valitsin toteuttaa sivuston ja sen ominaisuudet yksin, koska koin tällöin oppivani tekniikat paremmin.

## **2. Web sivuston idea**

Sivuston idea rakentuu kolmesta pilarista. Ensinnäkin sen on tarkoitus olla tietyn yhteisön kokoontumispaikka. Tämä yhteisö on tuntenut toisensa jo pitkän aikaa, ja tarvitsee oman pienen osan internetistä. Toiseksi sen on tarkoitus olla tietokanta, joka täyttää kyseisen yhteisön vaatimukset. Ominaisuuksina on oltava ainakin viestintämahdollisuus, uutisten ilmaisu, sekä erilaisten mediamuotojen jakaminen. Kolmanneksi sivusto ylläpitää yhteisön taiteellista puolta. Tietokannan ja sivuston avulla yhteisön on mahdollista keskustella töistä ja antaa niitä näytille. Kaikki mediamuodot ovat yhteisön itse tuottamia.

### **2.1 Ratkaisuperiaate**

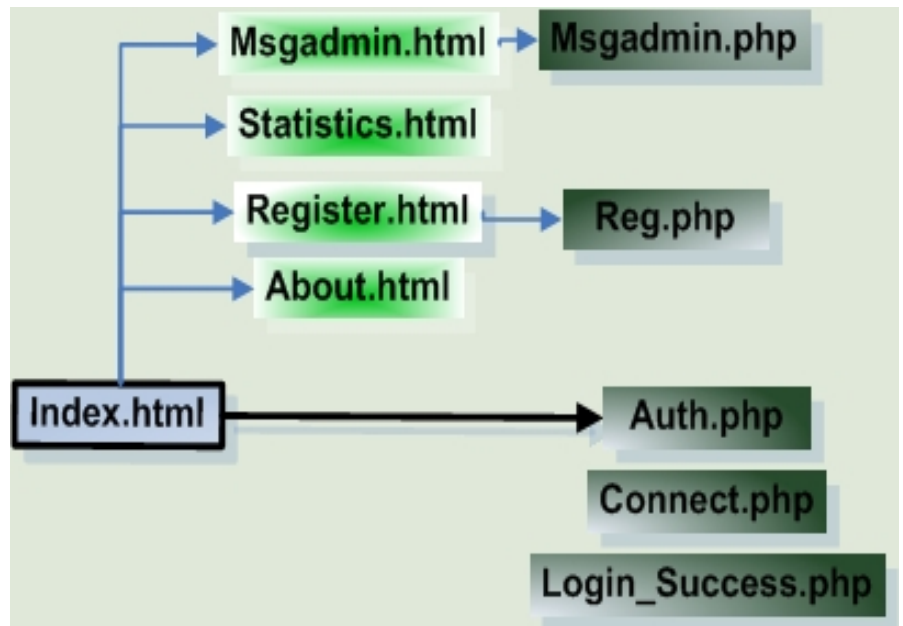
Sivusto on toteutettu käyttämällä sekä Adobe Dreamweaver CS3 että Microsoft FrontPage 2003 sovelluksilla. Pääsääntöisesti koko Hypertext Markup Language (HTML) koodi on kirjoitettu kuitenkin käsin, eikä web-ohjelmointi ohjelmien tehostetoimintoja ole käytetty. Syynä tähän on se, että koodia pystyy paremmin hallitsemaan, muokkaamaan ja jäsentämään itse. Sovellukset ovat olleet käytössä niiden tehokkaan esikatselunäkymän tähden.

Sivustoa varten on luotu oma Cascading Style Sheet (CSS). Tätä käyttämällä on HTML:n ulkoasua pystytty muokkaamaan halutun näköiseksi. CSS:n käyttö on kattavaa, sillä lähes jokainen elementti hyödyntää sitä. Koska sivustossa on lukuisia tietokantatoimintoja, on niitä varten käytetty Hypertext Preprocessor (PHP) kieltä. PHP mahdollistaa autentikoinnin, lukuisien tietokannan taulujen sisällön näyttämisen halutulla tavalla HTML sivustossa, laskurin, päivämäärän näyttämisen, yksinkertaisen viestintätoiminnon luomisen, tietokannan taulujen muokkaamisen sekä tiedostojen lataamisen palvelimelle. Tietokantana on Structured Query Language (SQL), tarkemmin määriteltynä mySQL. Tietokan-

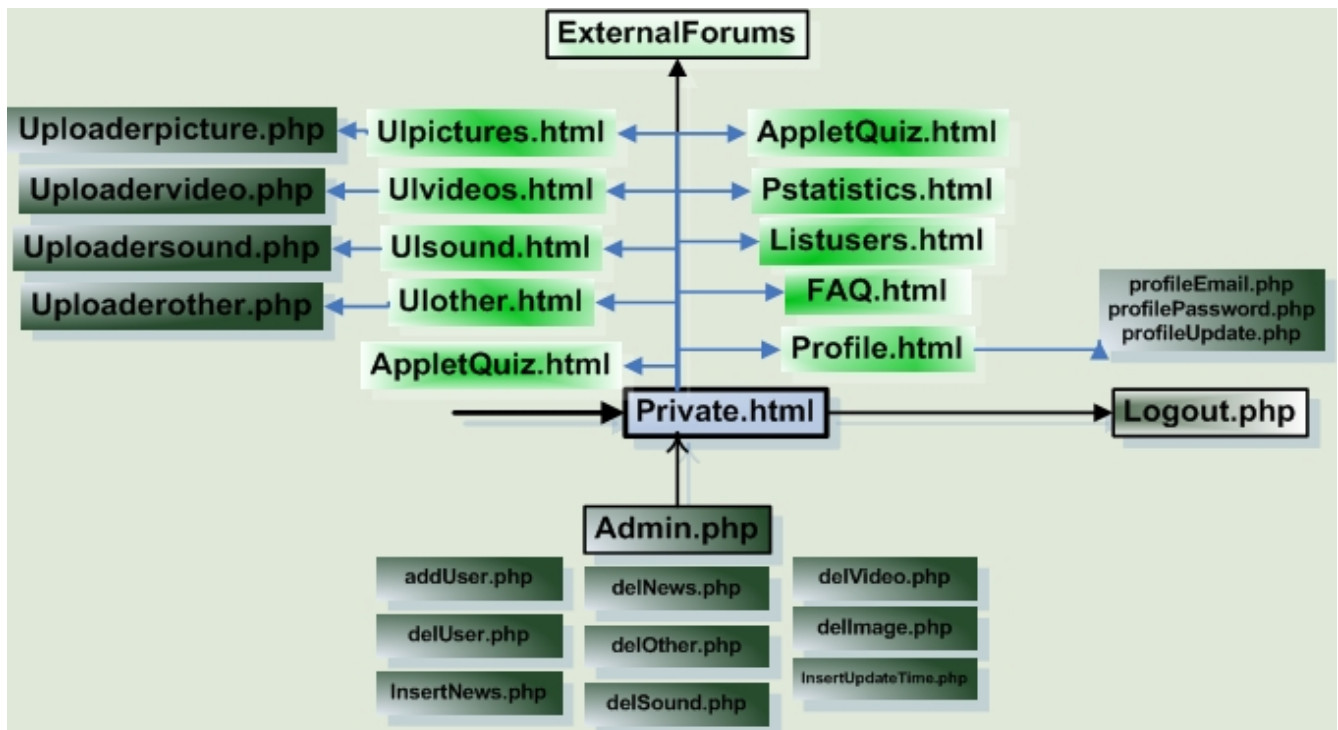
nan taulut mahdollistavat käyttäjän tunnistamisen, erilaisten käyttäjien oikeudet sekä tiedonvälityksen.

## 2.1 Rakennekaavio

Kuvassa 1 on esitetty julkisen puolen rakennekaavio. Index.html toimii pääsivuna, jonka kautta voi päästä ilman tunnuksia viestittämään ylläpitäjälle, näkemään julkiset tilastot, rekisteröitymään uudeksi käyttäjäksi, lukemaan sivuston merkityksen sekä kirjautumaan sisään suljetulle puolelle. Kuvassa 2 on mallinnettu suojattu puoli. Kirjautumisen jälkeen siirrytään Private.html:ään. Sivulta on mahdollista sen jälkeen siirtyä tai aktivoida kuvion mukaisesti toimintoja.



Kuva 1: Julkinen puoli



Kuva 2: Suojattu puoli

### 3. Kaskadisetyyliohjeet

Kaskadisetyyliohjeet ovat erityisesti WWW-dokumenteille kehitetty tyyliloheiden laji. CSS:n käyttö nopeuttaa tyyliden määrittämisen dokumenttiin. Se mahdollistaa selvän jäsennyksen tyyleihin, kun jokainen tyylilaji käy esimerkiksi CSS-tiedoston kautta, johon on kirjattuna mahdollisesti kaikki dokumentissa käytetyt tyylit.

Alla on otteita harjoitustyön CSS-tiedostosta selityksineen.

```
A:link {font-weight:bold; text-decoration: none; color:#c86400}
A:visited {text-decoration: none; color:#c85000;font-weight:bold;}
A:active {text-decoration: none}
A:hover {color:#FFFFFF; text-decoration: none}
```

A:\* määritykset antavat harjoitustyön linkeille yleisen toimintamäärittelyn. A:link määrittää minkälaisena linkit näkyvät ”lepotilassa” sivustossa. Linkit ovat lihavoituja sekä niillä on teemaan sopiva väri. A:visited muuttaa linkin ulkoasua määrittelyjen mukaan kun linkkiä on kerran painettu. Linkki pysyy edelleen lihavoituna, mutta väri tummenee. A:hover antaa tehosteen linkille kun vie hiiren sen päälle. Harjoitustyössä linkit muuttuvat valkoisiksi hiiren alla.

```
BODY {
    font-family: Verdana,Tahoma,Arial;
    color: white;
    font-size: 12pt;
    background-color : black;
}
```

BODY:ssä on määritelty dokumentin käyttämät standardit asetukset: fontti, tekstin väri, tekstin koko sekä taustan väri. Aina kun HTML dokumenttiin liitetään tämä CSS-tiedosto, nämä tyylit tulevat käyttöön ja näkyviin. Harjoitustyössä BODY on määritelty siten, että teksti on valkoista ja tausta on musta.

Alla on luettelomaisesti esitelty muut tyylit jota dokumentti käyttää. Ne on pyritty nimeämään mahdollisimman tyhjentävästi, jotta niiden käyttö sekä niiden vaikutukset olisi helppo tunnistaa.

.smallwhitetext {...}	.inputarea {...}	.tableheader {...}
.whitetext {...}	.btn {...}	.tablerow {...}
.bigwhitetext {...}	.errtext {...}	.tablefooter {...}
.inputtext {...}	.oktext {...}	
.pwtext {...}	.bgtable {...}	

## 4. Tietokanta ja PHP

Tietokantana toimii mySQL versio 5.0. HTML sivusto käyttää tietokantaa PHP:n avulla. Alla on lueteltu taulut ja niiden rakenne. Tietokannan rakenne on hyvin yksinkertainen. Taulujen välillä on hyvin vähän vuorovaikutusta. Ne ovat yksinkertaisesti tietovarastoja, joiden avulla HTML sivusto pystyy päivittämään ja rakentamaan itsensä dynaamisesti. Taulut ovat esitelty viereisen järjestyksen mukaisesti. Sivustossa on lukuisia pieniä PHP skriptejä, sekä yhteensä 27 ulkoista skriptiä. Tästä johtuen ovat vain oleellisimmat esiteltynä.

```
+-----+
| Tables_in_flamebird |
+-----+
| auth_counter        |
| counter             |
| images              |
| login               |
| msg                 |
| other               |
| publicnews          |
| site_update         |
| sounds              |
| users               |
| videos              |
+-----+
11 rows in set (0.00 sec)
```

Kuva 3: Kaikki tietokannan taulut

Auth\_counter taulun tarkoituksena on pitää yllä käyttäjien henkilökohtaisesta kirjautumisesta tietoa. Private.html sivuston oikeassa yläreunassa on kirjautujan nimen alapuolella ”You visited us last time ...” sekä ”You have entered... ..time(s)”. Seuraavan PHP koodin avulla HTML sivu hakee käyttäjän omat tiedot tietokannasta, ja näyttää ne sivulle. Tätä ennen on kuitenkin esiteltävä connect.php, tiedosto joka määrittelee tietokantayhteyden.

```
<?php
    $host="127.0.0.1:3306";      // Host name
    $username="root";          // Mysql username
    $password="*****";       // Mysql password
    $db_name="flamebird";      // Database name

    mysql_connect("$host", "$username", "$password")or die("cannot connect");
    mysql_select_db("$db_name")or die("cannot select DB");
?>
```

```

<?php
@include("connect.php");
$user=$_SESSION['myusername'];
$sql="SELECT last_login FROM users WHERE login='$user'";
$result=mysql_query($sql);

while($row=mysql_fetch_array($result)) {
    $printA=$row[0];

    if ($printA=="" or $printA==null) {
        echo "...this is your first time!";
    }

    else {
        echo $printA;
        echo ".";
    }

    $aika=date('j F Y');
    $addtime="UPDATE users SET last_login='$aika' WHERE login='$user'";

    mysql_query($addtime); }
?>

```

```

<?php
@include("connect.php");
$user=$_SESSION['myusername'];

$sql="SELECT login_count FROM users WHERE login='$user'";
$result=mysql_query($sql);

while($row=mysql_fetch_array($result)) {
    $printA=$row[0];
    $printA=$printA + 1;

    $sqlupdate="UPDATE users SET login_count='$printA' WHERE login='$user'";
    mysql_query($sqlupdate);

    echo $printA;
    echo " time(s).";
}
?>

```

```

mysql> describe auth_counter;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(10)| NO   | PRI | 0       |       |
| paiva | date   | YES  |     | NULL    |       |
| authed | int(10)| YES  |     | NULL    |       |
| realname | varchar(20) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

Kuva 4: Auth\_counter -taulun rakenne

Julkisella etusivulla (Index.html) on sivun alimmalla Frame:lla määritelty laskin, joka laskee kuinka monta kertaa sivu on ladattu. Tiedot säilytetään counter -taulukossa, ja laskin on hyvin yksinkertaisesti toteutettu.

```
<?php

    @include("connect.php");

    $sql="INSERT INTO counter VALUES(0,0)";
    mysql_query($sql);

    $check="SELECT MAX(number) FROM counter WHERE id=0";
    $isoin=mysql_query($check);

    while($row=mysql_fetch_array($isoin)) {
        $tulos=$row[0];
        echo "$tulos";
    }

    $poisto="DELETE FROM counter WHERE number < $tulos";
    mysql_query($poisto);

?>
```

```
mysql> describe counter;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(3) | NO   |     | 0       |      |
| number | int(255) | NO   | PRI | NULL    | auto_increment |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Kuva 5: Counter -taulun rakenne

Sivuston suojatulla puolella on mahdollista tallentaa mediamuotoja palvelimelle. Tästä johtuen taulut "images", "videos", "sounds", sekä "others" kuvaavat median tallentamista ja tietojen hakemista palvelimelta. Ne toimivat kaikki samalla periaatteella, josta johtuen on vain "images" taulu sekä sen PHP komennot esiteltyinä alla. Ensimmäinen skripti hakee tietokannasta kuvien tiedot ja päivittää HTML sivua näyttämään kaikki kuvat linkkeinä, jota painamalla kuva aukeaa. Toinen skripti toteuttaa kuvan siirtämisen palvelimelle.



```

<?php
    @include("connect.php");

    $sql="SELECT * FROM images";
    $result=mysql_query($sql);

    while($row=mysql_fetch_array($result)) {
        $nimi=$row[1];
        $paikka=$row[2];

        echo ("

```

```

mysql> describe images;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int(10)       | NO   | PRI | NULL    | auto_increment |
| name  | varchar(30)   | YES  |     | NULL    |                |
| url   | varchar(100)  | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

```

Kuva 6: Images -taulun rakenne

Login -taulu on tarkoitettu pelkästään "Admin" käyttäjien seurattavaksi. Tämän tarkoituksena on valvoa ketä kirjautuu järjestelmään ja mihin aikaan. Taulua kutsutaan "Admin" työkaluista automaattisesti sivun avautuessa.

```
<?php
    @include("connect.php");
    $sql="SELECT * FROM login";
    $result=mysql_query($sql);
        while($row=mysql_fetch_array($result)) {
            $tulosa=$row[0];
            $tulosa=$row[1];
            $tulosa=$row[2];
            $tulosa=$row[3];

            echo (" $tulosa $tulosa $tulosa $tulosa <br>");
        }
?>
```

```
mysql> describe login;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int(100)      | NO   | PRI | NULL    | auto_increment |
| date  | varchar(20)   | YES  |     | NULL    |                |
| time  | varchar(20)   | YES  |     | NULL    |                |
| user  | varchar(20)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.09 sec)
```

Kuva 7: Login -taulun rakenne

Suojatun sivuston etusivulla on pikaviestintätoiminto. Tämä toiminto toimii myös PHP:n ja tietokannan kautta. Jokainen näytettävä attribuutti on HTML sivulla omassa taulukossaan, josta johtuu se, että jokaiselle on olemassa oma PHP skripti. Skriptit toimivat samalla tavalla, josta johtuen vain yksi esitellään.

```

<?php
    @include("connect.php");

    $checkmsg="SELECT * FROM msg ORDER BY id desc";
    $tulosa=mysql_query($checkmsg);

        while($row=mysql_fetch_array($tulosa)) {
            $printC=$row[3];

            echo " $printC <br>";

        }
?>

```

```
mysql> describe msg;
```

Field	Type	Null	Key	Default	Extra
id	int(10)	NO	PRI	NULL	auto_increment
user	varchar(25)	NO		NULL	
msg	varchar(255)	YES		NULL	
date	varchar(20)	YES		NULL	
time	varchar(20)	YES		NULL	

```
5 rows in set (0.08 sec)
```

Kuva 8: Msg -taulun rakenne

Sivusto on toteutettu siten, että lähes kaikkea tietoa on mahdollista hallita sivun oman järjestelmänvalvojayökalujen kautta. Julkisella sivulla on keskimmaisella Framella oikeassa reunassa julkisia uutisia. Nämä uutiset ovat pääasiassa päivityksiä varten. Julkisia uutisia voi poistaa tai lisätä sivustosta seuraavasti. Samalla periaatteella toimivat suurin osa muistakin järjestelmänvalvojan toiminnoista, joten niitä ei ole esiteltyä enempää.

```

<?php

    @include("connect.php");

    $paiva=$_POST['paiva'];
    $uutinen=$_POST['uutinen'];

    $sql="INSERT INTO publicnews (pvm, data) VALUES('$paiva', '$uutinen)";
    mysql_query($sql);

?>

```

```
<?php
```

```
@include("connect.php");

$paiva=$_POST['paiva'];

$sql="DELETE FROM publicnews WHERE id='$paiva'";
mysql_query($sql);
```

```
?>
```

```
mysql> describe publicnews;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int(3)        | NO   | PRI | NULL    | auto_increment |
| pvm   | varchar(12)   | YES  |     | NULL    |                |
| data  | varchar(100)  | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.08 sec)
```

Kuva 9: Msg -taulun rakenne

Kirjautuminen tapahtuu tietenkin julkiselta sivulta. Teknisesti kirjautuminen tapahtuu tietojen tarkistamisen avulla tietokannasta. PHP skriptillä tehdään kysely tietokantaan.

```
<?php
```

```
$tbl_name="users"; // Table name

// Connect to server and select database.
@include("connect.php");

// username and password sent from form
$myusername=$_POST['myusername'];
$mypassword=$_POST['mypassword'];

// To protect against MySQL injection
$myusername = stripslashes($myusername);
$mypassword = stripslashes($mypassword);
$myusername = mysql_real_escape_string($myusername);
$mypassword = mysql_real_escape_string($mypassword);

$sql=sprintf("SELECT * FROM $tbl_name WHERE login='$myusername'
              and password='$mypassword'");
$result=mysql_query($sql);

$aika=date('j F Y');
$kello=date('H:i:s');

$whoitis="INSERT INTO login (date, time, user)
          VALUES('$aika', '$kello', '$myusername')";

mysql_query($whoitis);

$count=mysql_num_rows($result);
```

```

        if($count==1){
            session_start();

            $_SESSION['loggedin'] = 1;
            $_SESSION['myusername'] = $myusername;
            $_SESSION['mypassword'] = $mypassword;
        mysql_close($conn);

        header ("location:login_success.php");

    }

    else {echo "Wrong Username or Password";}

?>

Login_Success.php

<?php
header("Location: entrance.html", "true");
?>

```

```

mysql> describe users;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id             | int(3)        | NO   | PRI | NULL    | auto_increment |
| realname       | varchar(20)   | YES  |     | NULL    |                 |
| login          | varchar(20)   | YES  |     | NULL    |                 |
| password       | varchar(20)   | YES  |     | NULL    |                 |
| email          | varchar(20)   | YES  |     | NULL    |                 |
| type           | int(1)        | YES  |     | 0       |                 |
| login_count    | int(3)        | YES  |     | NULL    |                 |
| last_login     | varchar(20)   | YES  |     | NULL    |                 |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.08 sec)

```

Kuva 10: Users -taulu

## 5. JavaScript toiminto

Alustavassa suunnitelmassa oli tarkoituksena vain tehdä PHP:llä yksinkertainen ajannäyttö toiminto suojatun sivuston etusivulle. Suunnitelmat muuttuivat kun tutustuin JavaScriptiin. Tästä johtuen vaihdoin kellon JavaScriptin reaaliaikaisesti päivittyvään vaihtoehtoon. Alla oleva skripti toteuttaa kellon HTML sivulle. Skripti hakee tunnit, minuutit ja sekunnit sekä sisältää kolme metodia: kellon päivittäminen, aloittaminen sekä sammuttaminen.

```
<script language="JavaScript">
    var clockID = 0;

    function UpdateClock() {
        if(clockID) {
            clearTimeout(clockID);
            clockID = 0;
        }

        var tDate = new Date();

        document.theClock.theTime.value = ""
            + tDate.getHours() + ":"
            + tDate.getMinutes() + ":"
            + tDate.getSeconds();

        clockID = setTimeout("UpdateClock()", 1000);
    }

    function StartClock() {
        clockID = setTimeout("UpdateClock()", 500);
    }

    function KillClock() {
        if(clockID) {
            clearTimeout(clockID);
            clockID = 0;
        }
    }
</script>
```

Sivustossa on yksilöllinen CSS –tyylittely. Alla oleva HTML koodi asettaa kellon toimimaan lomakkeeseen. Lomakkeen tyyli muokataan teemaan sopivaksi ”class” attribuutilla.

```
<body onload="StartClock()" onunload="KillClock()">
<center><form name="theClock">
<input type="text" name="theTime" size=8 style="text-align:center"
class="inputarea">
</form></center>
```

## 6. Web sivuston osien kuvaaminen

### 6.1 Julkinen sivu ja kirjautuminen

Julkinen sivu (`Index.html`) on jaettu vanhanaikaisesti kolmeen Frame –osaan.

```
<frameset rows="40%,39%,11%" framespacing="0" border="0" frameborder="0">
  <frame name="Upper" src="IndexLogo.html" noresize scrolling="no">
  <frame name="Middle" src="IndexBody.html" scrolling="yes" >
  <frame name="Lower" src="IndexLower.html" noresize scrolling="no">
</frameset>
```

Ensimmäinen Frame (`IndexLogo.html`) sisältää vain sivuston logon sekä tervetulotekstin.

Toinen Frame (`IndexBody.html`) on etusivun tärkein osa. Sivun on jaettu kolmeen taulukonosaan, jotta sisältö olisi mahdollista asetella symmetrisesti. "Members only" kohdassa käyttäjät voivat kirjautua suojatulle puolelle aikaisemmin esitetyllä tavalla (kts. kappale 4). "Visitors" kohdassa on sivuston vierailijoille tietoa. "What is this all about?" staattinen sivu kertoo kolmen avainsanan kautta sivuston merkityksen ja funktion. "Register here" antaa uudelle käyttäjälle mahdollisuuden luoda tunnus, ja kirjautua sisälle. Tunnusten teko tapahtuu staattisen `register.html` sivun kautta, joka lähettää form tiedot `reg.php`:lle. Tämän toiminta on esitetty alla.

```
<?php
    $table="users";
    @include("connect.php");
        $realname=$_POST['realname'];
        $nickname=$_POST['nickname'];
        $password1=$_POST['password1'];
        $password2=$_POST['password2'];
        $email=$_POST['email'];

    if ($realname & $nickname & $email) {
    if($password1==$password2) {
    $sql="INSERT INTO $table (realname, login, password, email, type)
VALUES('$realname', '$nickname', '$password1', '$email', 0)";
        mysql_query($sql);
            }
    else echo "Your password fields are not equal!";}
    else echo "Please fill in all of the fields!";

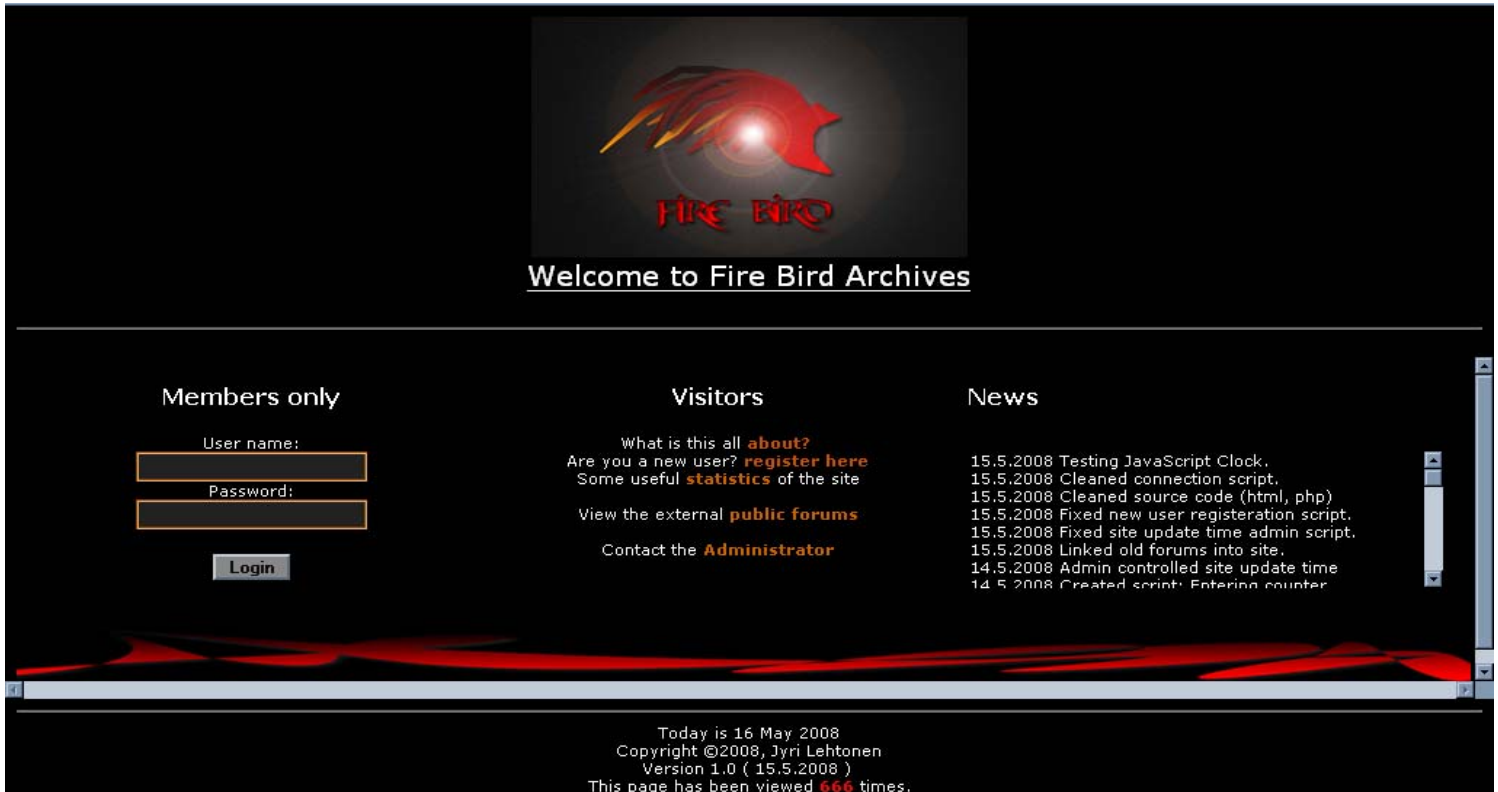
?>
```

"Some useful statistics of the site" sisältää joukon yksinkertaisia PHP komentoja, jotka kyselevät tietokannan taulukoista tietoa, laskevat rivit ja näyttävät ne sivun lukijalle. Sivustossa ei ole itsessään toteutettu keskustelupalstaa, vaan se on ulkoistettu erilliselle sivustolle. Vierailijat voivat myös lähettää viestin sivuston ylläpitäjälle etusivulta. msgadmin.html lähettää tiedot msgadmin.php:lle, jonka toiminta on esitetty alla:

```
<?php
$nimi=$_POST['nimi'];
$viesti=$_POST['viesti'];
mail("FB_Management@techie.com", "FBA Public - $nimi ", "$viesti");
?>
```

Kaikkein vasemmanpuolisessa taulukkopaikassa on varattu tilaa julkisille uutisille. Nämä uutiset ovat tietokannassa, ja niitä pystyy sivuston ylläpitäjä päivittämään suojatulla puolella olevista järjestelmänvalvonta työkaluista.

Kolmannessa Frame:ssa (IndexLower.html) on muutama yksinkertainen PHP komento. Ensimmäinen tarkistaa nykyisen päiväyksen ja näyttää sen HTML:n seassa. Toinen on sivuston uusimman päivityksen päiväys, jonka ylläpitäjä voi järjestelmänvalvoja työkaluista muuttaa. Kolmas on laskuri, joka tarkistaa kuinka monta kertaa sivua on ladattu.



Kuva 11: Julkinen sivu



## 6.2 Yksityinen sivusto

Julkisella sivulla kirjautumisen jälkeen avautuu suojattu sivusto. Suojattu sivusto on jaettu julkisen mukaan myös kolmeen erilliseen Frame:en..

```
<frameset rows="45%,35%,10%" framespacing="0" frameborder="0">
  <frame name="Main" src="private.html" noresize scrolling="no">
  <frame name="Body" src="PrivateBody.html" scrolling="yes">
  <frame name="Feet" src="privateDown.html" noresize scrolling="no">
</frameset>
```

Ensimmäinen Frame (`private.html`) sisältää hallintapuolen. Rakenne on kuten julkisen sivuston keskeisin Frame. Vasemmanpuolimmaisimmassa taulukonosassa on sivuston logo, sekä aikaisemmin käsitelty JavaScript (kts. kappale 5). Keskellä on lista linkkejä: usein kysytyt kysymykset, listaa kaikki käyttäjät, etsi sivustosta (ei ole toiminnassa vielä), näytä suojatun puolen tilastot, näytä aputiedosto. Listassa on mukana myös ulkoistettu keskustelupalsta sekä pieni JavaApplet sovellus. Oikeimmanpuolimmaisimmassa osassa on joukko pieniä PHP toimintoja. Ensimmäiseksi haetaan käyttäjän nimi, ja näytetään se HTML sivulla:

```
<?php
    print $_SESSION['myusername'];
?>
```

Toisena skriptinä on edellisen kirjautumiskerran näyttäminen. Kolmantena on jokaiselle käyttäjälle yksilöllinen kirjautumiskertojen laskuri.

```
<?php

@include("connect.php");

$user=$_SESSION['myusername'];

$sql="SELECT last_login FROM users WHERE login='$user'";
$result=mysql_query($sql);

while($row=mysql_fetch_array($result)) {
    $printA=$row[0];

    if ($printA=="" or $printA==null) {
        echo "...this is your first time!";
    }

    else {echo $printA;
        echo "."; }

    $aika=date('j F Y');
```

```

        $addtime="UPDATE users SET last_login='$aika' WHERE login='$user'";
mysql_query($addtime);

    }
?>
<?php

@include("connect.php");
$user=$_SESSION['myusername'];

$sql="SELECT login_count FROM users WHERE login='$user'";
$result=mysql_query($sql);

    while($row=mysql_fetch_array($result)) {
        $printA=$row[0];
        $printA=$printA + 1;

        $sqlupdate="UPDATE users SET login_count='$printA' WHERE login='$user'";
        mysql_query($sqlupdate);

        echo $printA;
        echo " time(s).";
    }
?>

```

Käyttäjä pystyy muuttamaan omia tietojaan painamalla oman nimensä päälle. Tämä avaa `profile.html:n`. Tiedonmuuttamiset sekä järjestelmänvalvonta työkalut käsitellään myöhemmin kappaleessa 7: Testausjärjestelyt.

Keskimmäinen Frame (`PrivateBody.html`) toimii tärkeimpänä näkymänä. Ylimmän Frame:n valikkoja painamalla saadaan uusia sivuja näkyviin keskimmäiseen Frame:en. Main –valinta avautuu ensimmäiseksi, ja siinä on pikaviestintätoiminta. Pikaviestintä on toteutettu PHP:n ja MySQL:n avulla. Viestintäkenttä on neljään osaan jaettu taulukko, joihin jokaiseen haetaan tietokannasta tiedot alla olevalla komennolla, mutta tulostetaan PHP:llä vain tietty sarake jokaiseen kohtaan.

```
$checkmsg="SELECT * FROM msg ORDER BY id desc";
```

Päävalikon mediavalinnat toimivat kaikki samalla tavalla. Valinnan jälkeen PHP skripti hakee tietokannasta mahdolliset median saatavuudet ja listaa ne sivulle linkeiksi. Linkkejä voidaan valita, jolloin media avautuu uuteen sivuun.

```

<?php
    @include("connect.php");
    $sql="SELECT * FROM images";
    $result=mysql_query($sql);

        while($row=mysql_fetch_array($result)) {
            $nimi=$row[1];
            $paikka=$row[2];

            echo ("<a href='images/$paikka' target='_blank'> $nimi </a><br>");
        }
?>

```

Medivalintojen vieressä on mahdollisuus ladata palvelimelle kyseistä mediamuotoa. Lataus on samanlainen mediamuodosta riippumatta. Latauskansio on jokaisella kuitenkin eri.

```

<?php
    $target_path = "images/";
    $target_path = $target_path . basename( $_FILES['uploadedfile']['name']);
    if(move_uploaded_file($_FILES['uploadedfile']['tmp_name'], $target_path)) {
    echo "The file ". basename( $_FILES['uploadedfile']['name']).
        " has been uploaded";

        @include("connect.php");

        $filename=$_POST['nimetys'];
        $url=$_POST['tiedostonimi'];
        $sql="INSERT INTO images (name, url) VALUES('$filename', '$url')";
            mysql_query($sql);

        }

        else{echo "There was an error uploading the file, please try again!";}
?>

```

Alimmainen Frame (PrivateDown.html) sisältää hyvin vähän tietoa. Frame esittää nykyisen päivämäärän, tekijän sekä version tietokannan kautta näkyviin.



Welcome to Fire Bird Private

16:6:23

## Information and Actions

- View the **FAQ**
- List all **users**
- **Search** the archives
- Display the **statistics**
- Read the **help file**
- View the external **forums**
- Play java **Applet Quiz**

Hi, **Jay** | [\[Logout\]](#)

You visited us last time 16 May 2008.  
You have entered 97 time(s).

You can edit your profile properties by clicking on your username above.

**Admin tools**

Main

Pictures | Upload a file

Videos | Upload a file

Sound | Upload a file

Other | Upload a file

### Instant messaging:

	<u>Date:</u>	<u>Time:</u>	<u>User:</u>	<u>Message:</u>
Date:	16 May 2008	12:18	cyberpunk	Moro
16 May 2008	16 May 2008	10:07	dude	test
Time:	16 May 2008	09:11	dude	hi
16:06	15 May 2008	12:44	Jay	test
Your name:	15 May 2008	09:46	Jay	Yaer.
Jay	15 May 2008	09:40	Jay	Tuned up
Your message:	14 May 2008	16:18	Test11	Nimenvaihto...
	14 May 2008	10:12	Jay	Kellon testausta.
	13 May 2008	16:25	Jay	Yeah.
	13 May 2008	16:23	Jay	Noh nythän näkyy kellokin.

Send

Today is 16 May 2008  
Copyright ©2008, Jyri Lehtonen  
Version 1.0 ( 15.5.2008 )

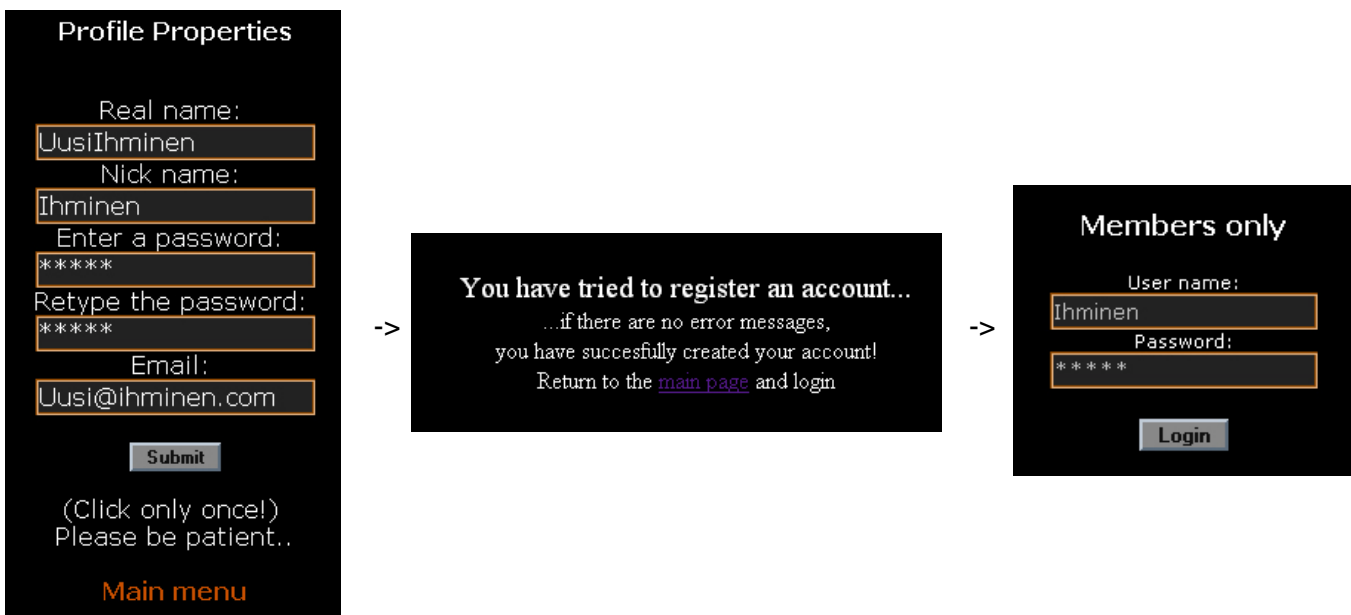
Kuva 12: Suojattu sivu

## 7. Testausjärjestely

Testaus toteutettiin seuraavasti:

1. Uuden käyttäjän luominen
2. Kirjautuminen suojatulle sivustolle
3. Oman profiilin muokkaaminen
4. Järjestelmänvalvojan työkalujen käyttö
5. Tiedoston lataaminen palvelimelle
6. Kirjautuminen ulos

Testi (1-2): Julkiselta sivulta -> "Are you a new user? Register here" -> Täytä kaavakkeet -> Kirjautu

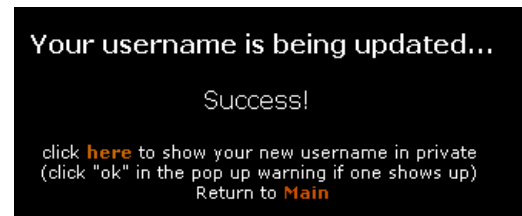


The diagram illustrates the registration process flow. It starts with the 'Profile Properties' form, which includes fields for Real name (UusiIhminen), Nick name (Ihminen), Password (\*\*\*\*), Retype the password (\*\*\*\*), and Email (Uusi@ihminen.com). A 'Submit' button is at the bottom. Below the form, it says '(Click only once!) Please be patient..' and 'Main menu'. An arrow points to a confirmation message: 'You have tried to register an account... ..if there are no error messages, you have successfully created your account! Return to the [main page](#) and login'. A second arrow points to the 'Members only' login page, which has fields for User name (Ihminen) and Password (\*\*\*\*), and a 'Login' button.

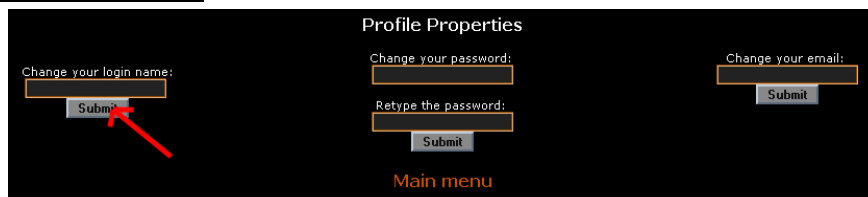
Testi (3): Suojatulta sivulta -> Valitse oma nimi ylhäältä -> Vaihda käyttäjän nimi



The screenshot shows a user profile page for 'Ihminen'. It includes a 'Logout' link, a message 'Hi, Ihminen | [Logout]', and a note: 'You visited ... last time ...this is your first time! You have entered 1 time(s).'. Below this, it says 'You can edit your profile properties by clicking on your username above.' and 'Admin tools'.

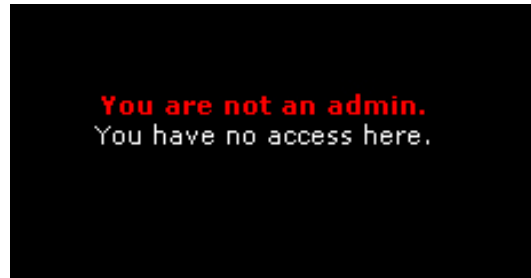
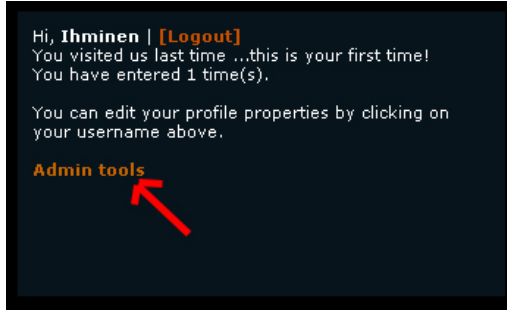


The screenshot shows a confirmation message: 'Your username is being updated... Success!'. It includes instructions: 'click **here** to show your new username in private (click "ok" in the pop up warning if one shows up) Return to **Main**'.

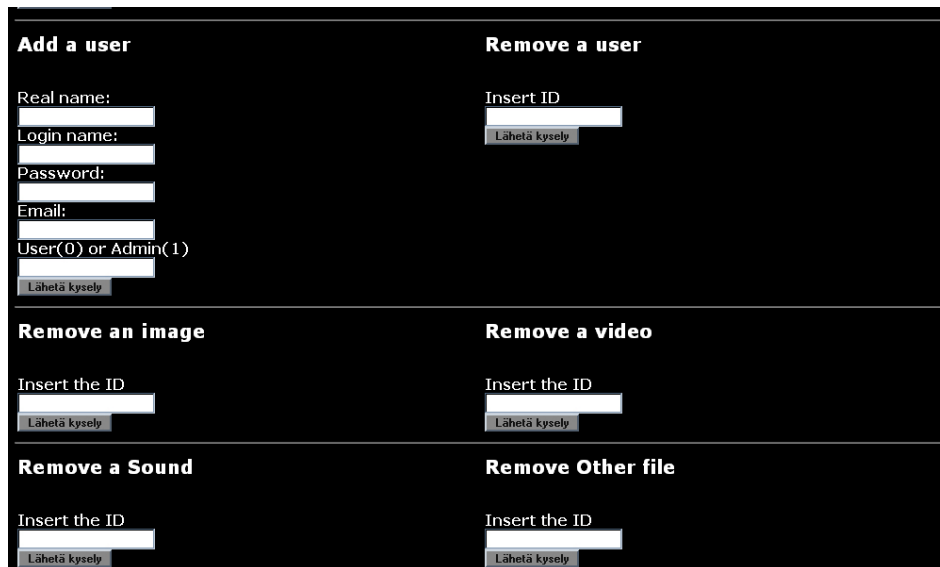


The screenshot shows the 'Profile Properties' form with three sections: 'Change your login name:' with a 'Submit' button, 'Change your password:' with a 'Submit' button, and 'Change your email:' with a 'Submit' button. A red arrow points to the 'Submit' button under the login name field. Below the form is the 'Main menu' link.

Testi (4): Suojatulta sivulta -> Valitse Admin tools ylhäältä



Annettaan käyttäjälle "Ihminen" testimielessä oikeudet järjestelmänvalvojaksi.



Testi (5): Valitse "Upload a file" jostain mediamuodosta -> Kirjaa tiedot -> Valitse "Upload File"



Choose a file to upload: (10mt limit)

hto\Työpöytä\test01.jpg

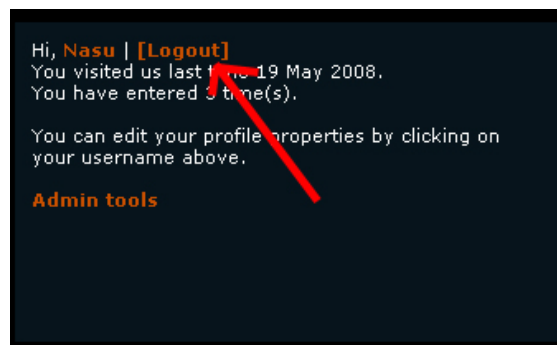
Name the picture (for example: My First Image)

Picture address name (for example: myfirstimage.jpg)

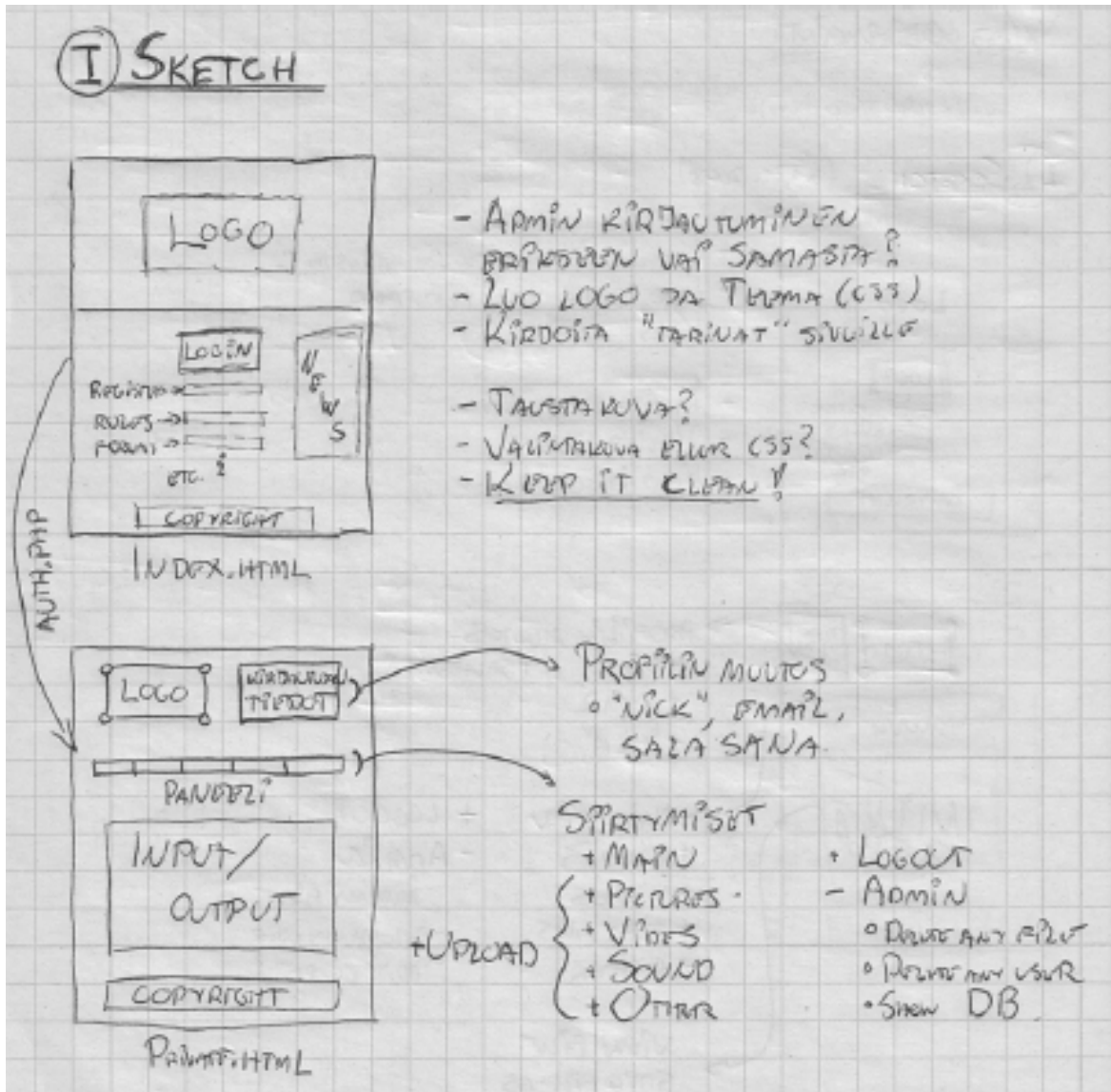
**The file test01.jpg has been uploaded**



Testi (6): Valitse "Logout"



# Liite 1: Ensimmäinen suunnitelma





## Liite 2: Lopullinen suunnitelma

