

Andmete muutmine, laulude lehestik

Lisamise ja kustutamisega saab küllalt palju ühekordseid registreerimisi ja vahetamisi korda ajada. Põhjalikumate rakenduste juures aga käivad sama andmerea eri tulbad eri kasutajate juurest läbi ning siis tuleb ka rea andmete muutmisele mõelda nii, et osa väärtusi säilib, mõned uuenevad. Siin näitena koostame laulude lehestiku, kus haldur saab neid tabelisse lisada, kasutajad lauludele punkte ja kommentaare jagada ning halduril pärast võimalik määrata, millised laulud parajasti välja paistavad ja millised mitte. Ning kõik see lehestik ehitatakse ühe andmetabeli peale, kus siis eri veerugude väärtusi saab sobivalt kasutada.

Sellise lehestiku baasipooleks piisab, kui laulude andmed on andmebaasis, tabelis nimega laulud(id, pealkiri, punktid, lisamisaeg, kommentaarid, avalik)

SQL-lause tabeli loomiseks:

```
CREATE TABLE laulud(  
  id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  pealkiri VARCHAR(50),  
  punktid INT DEFAULT 0,  
  lisamisaeg DATETIME,  
  kommentaarid TEXT,  
  avalik INT DEFAULT 1  
);
```

Sõna DEFAULT tulba taga määrab vaikimisi väärtuse - ehk kui laul luuakse, siis pole tal veel ühtki punkti, aga samas on avalik. Lisamisaja andmetüüp DATETIME näitab, et üheaegselt hoitakse meeles kuupäev ja kellaeg.

Lehestiku koostamise saab jagada suuremateks alamülesanneteks:

* Looge veebileht laulude lisamiseks andmebaasi.

Sisestada on vaja vaid pealkiri, tulemust näeb vaid baasist

* Looge veebileht lauludele plusspunkti andmiseks.

Iga laulu taga näeb talle antud punkte.

* Looge veebileht, mis näitaks vaid avalikke laule

Selle juures aitab SQL-i poolest päring laulu nr 1 näitel

```
SELECT pealkiri FROM laulud WHERE avalik=1
```

* Muutke andmebaasi käsklusega mõni laul peidetuks

Tarvilik SQL-käsklus sealjuures

```
UPDATE laulud SET avalik=0 WHERE id=1
```

Veenduge, et seda laulu avalike laulude lehele ei nähe

* Koosta haldusleht, mille abil on võimalik laule peita ja taas avalikuks muuta.

Uue laulu lisamine

Lisamiseks on vaja lisamisvormi ja salvestuskohta. Kui ülalpoolses näites kipus mõnevõrra segadust tekitama, et lisamisvorm avanes vaid vastaval viitel vajutades, siis siin püütud lisamine võimalikult lihtsaks teha. Lisamislahter on kohe lehe avamisel olemas. Ning samal lehel püütakse pealkiri kinni ja lisatakse andmed tabelisse. Lisamisaja väärtuseks NOW() annab serverikella praeguse aja. Kommentaariks lisatakse algul tühi tekst, pärast saab sinna kasutajate juttu juurde paigutada.

```
<?php
    $yhendus=new mysqli("localhost", "juku", "kala", "jukubaas1");
    if(!empty($_REQUEST["ueepealkiri"])){
        $kask=$yhendus->prepare(
            "INSERT INTO laulud(pealkiri, lisamisaeg, kommentaarid) VALUES(?, NOW(),
            ' ')" );
        $kask->bind_param("s", $_REQUEST["ueepealkiri"]);
        $kask->execute();
        echo $yhendus->error;
        header("Location: $_SERVER[PHP_SELF]");
        $yhendus->close();
        exit();
    }
?>
<!doctype html>
<html>
    <head>
        <title>Laulud</title>
    </head>
    <body>
        <h1>Laulud</h1>
        <form action=""?>
            Uue laulu pealkiri:
            <input type="text" name="ueepealkiri" />
            <input type="submit" value="Lisa laul" />
        </form>
    </body>
</html>
<?php
    $yhendus->close();
?>
```

Kõigepealt ilmub pealkirja lisamise lahter

Laulud

Uue laulu pealkiri:

Sinna võib kirjutada uue laulu pealkirja, vajutada lisamisnuppu ning ongi see laul tabelis kirjas.

Laulud

Uue laulu pealkiri:

Ülesandeid

- Pane näide tööle.
- Koosta tabel koolipeole kutsutava ansambli valimiseks ja hääletamiseks. Tabel ansamblid(id, ansamblinimi, punktid, kommentaarid, avalik, otsus). Kommentaarid tüübist TEXT, otsus VARCHAR(255).
- Koosta leht ansambli nime lisamiseks tabelisse. Kontrolli lehe tööd.

Lauludele punktide lisamine

Punktide lisamis leht koosneb kahest osast. Kõigepealt kuvatakse loetelu kõigis tabelis olevatest lauludest. Ning kui kasutaja ühele neist vajutab, siis lisatakse laulule punkt. Punkti lisamiseks on viide

```
<a href=?healaulu_id=$id>Lisa punkt</a>
```

ehk siis küsimärgiga viide viib kasutaja samale failile. Kaasa antakse parameeter nimega healaulu_id, väärtuseks selle laulu id, mille nimele parajasti vajutatakse.

Serveris lehe uuel avamisel kontrollitakse, kas parameeter healaulu_id on olemas. Kui jah, siis pannakse sellele laulule üks punkt juurde, ehk suurendatakse vastava välja väärtust.

Väärtuse suurendamiseks ühe võrra sobib SQL-lause

```
UPDATE laulud SET punktid=punktid+1 WHERE id=?
```

kus siis küsimärgi kohale tuleb pärast käsu ettevalmistust vastava laulu id.

Punktide jagamise kood tervikuna

```
<?php
$yhendus=new mysqli("localhost", "juku", "kala", "jukubaas1");
if(isset($_REQUEST["healaulu_id"])){
    $skask=$yhendus->prepare("UPDATE laulud SET punktid=punktid+1 WHERE id=?");
    $skask->bind_param("i", $_REQUEST["healaulu_id"]);
```

```

        $kask->execute();
    }
?>
<!doctype html>
<html>
    <head>
        <title>Laulud</title>
    </head>
    <body>
        <h1>Laulud</h1>
        <table>
            <?php
                $kask=$yhendus->prepare("SELECT id, pealkiri, punktid FROM laulud");
                $kask->bind_result($id, $pealkiri, $punktid);
                $kask->execute();
                while($kask->fetch()){
                    $pealkiri=htmlspecialchars($pealkiri);
                    echo "<tr>
                        <td>$pealkiri</td>
                        <td>$punktid</td>
                        <td><a href='?healaulu_id=$id'>Lisa punkt</a></td>
                    </tr>";
                }
            ?>
        </table>
    </body>
</html>
<?php
    $yhendus->close();
?>

```

Tulemusena ilmuvad olemasolevad laulud silma ette.

Laulud

Mutionu 9 [Lisa punkt](#)
Mutionu pidu 8 [Lisa punkt](#)
Muumioru lood 4 [Lisa punkt](#)
Rongisõit 0 [Lisa punkt](#)

Millisele reale vajutati, selle laulu punktide arv suureneb ühe võrra.



Laulud

Mutionu 9 [Lisa punkt](#)
Mutionu pidu 8 [Lisa punkt](#)
Muumioru lood 4 [Lisa punkt](#)
Rongisõit 1 [Lisa punkt](#)

Ülesandeid

- Tee näide läbi
- Loo/otsi eelneva ülesande ansamblite tabel.
- Loo võimalus ka ansamblitele punktide jagamiseks.
- Tekita juurde tulbad, kus on võimalik punkte ka kahe- ja kolmekaupa jagada.

Laulude peitmine ja avalikustamine

"Päris" rakenduste juures enamasti ei kustutata midagi lihtsalt ära, vaid lihtsalt märgitakse, et vastav kirje/rida on "arhiveeritud". Selliselt on võimalik ka veebiandmete kaudu toimunu ajalugu vaadata ning vajadusel mõningaid kohti siluda/taastada. Kui millalgi on vaja surnud ridade arvel andmebaasi mahtu vähendada, siis need enamasti korjatakse kusagile varundustabelisse või faili kokku ning lastakse seal veel mõnda aega olla, kuni loota võib, et sealt enam midagi tähtsamat vajalikuks ei osutu. Sarnase avalikustamise ja peitmise mängime läbi ka laulude puhul. Kas laul on avalik või mitte, seda saab lugeda vastavast tulbast: 0 - peidetud, 1- avalik. Lehel avalike laulude näitamiseks pannakse laulude näitamise juures päringule lihtsalt WHERE-tingimus avalik=1.

avalikudlaulud.php

```
<?php
    $yhendus=new mysqli("localhost", "juku", "kala", "jukubaas1");
?>
<!doctype html>
<html>
    <head>
        <title>Laulud</title>
    </head>
    <body>
        <h1>Laulud</h1>
        <table>
            <?php
                $kask=$yhendus->prepare(
                    "SELECT id, pealkiri, punktid FROM laulud WHERE avalik=1");
                $kask->bind_result($id, $pealkiri, $punktid);
                $kask->execute();
                while($kask->fetch()){
                    $pealkiri=htmlspecialchars($pealkiri);
                    echo "<tr>
                        <td>$pealkiri</td>
                        <td>$punktid</td>
                    </tr>";
                }
            ?>
        </table>
    </body>
</html>
<?php
    $yhendus->close();
?>
```

Laulud esiota illusasti avalikult näha.

Laulud

Mutionu	9
Mutionu pidu	8
Muumioru lood	4
Rongisõit	1

Laulude peitmiseks saab nende nimed ette kuvada ning taga oleva viite kaudu siis loo peidetuks muuta. Jällegi tuleb kaasa anda loo id ning lehe ülaosas uuel laadimisel sellele vastavalt reageerida. Vastavalt saadetud peitmise_id-le käivitatakse SQL-lause

```
UPDATE laulud SET avalik=0 WHERE id=?
```

mille tulemusena vajutatud viitega laulu tulpale avalik antakse väärtuseks 0 ning laulu enam avalike laulude seas ei kuvata.

haldus3.php

```
<?php
    $yhendus=new mysqli("localhost", "juku", "kala", "jukubaas1");
    if(isset($_REQUEST["peitmise_id"])){
        $kask=$yhendus->prepare("UPDATE laulud SET avalik=0 WHERE id=?");
        $kask->bind_param("i", $_REQUEST["peitmise_id"]);
        $kask->execute();
    }
?>
<!doctype html>
<html>
    <head>
        <title>Laulud</title>
    </head>
    <body>
        <h1>Laulud</h1>
        <table>
            <?php
                $kask=$yhendus->prepare("SELECT id, pealkiri, avalik FROM laulud");
                $kask->bind_result($id, $pealkiri, $avalik);
                $kask->execute();
                while($kask->fetch()){
                    $pealkiri=htmlspecialchars($pealkiri);
                    echo "<tr>
                        <td>$pealkiri</td>
                        <td>$avalik</td>
                        <td><a href='?peitmise_id=$id'>Peida</a></td>
                    </tr>";
                }
            ?>
        </table>
    </body>
```

```
</html>
<?php
    $yhendus->close();
?>
```

Algul kõik lood avalikud



Laulud

Mutionu 1 [Peida](#)
Mutionu pidu 1 [Peida](#)
Muumioru lood 1 [Peida](#)
Rongisõit 1 [Peida](#)

localhost:8080/~jaagup/if12/1/laluhaldus/haldus3.php?peitmise_id=3

Pärast vajutust läks Muumioru lugude avalik-tulp nulliks.



Laulud

Mutionu 1 [Peida](#)
Mutionu pidu 1 [Peida](#)
Muumioru lood 0 [Peida](#)
Rongisõit 1 [Peida](#)

Tulemusena seda lugu avalike laulude all ei kuvata.



Laulud

Mutionu 9
Mutionu pidu 8
Rongisõit 1

Lihtsamal juhul piirduvadki lehe oskused vaid peitmisega. Näiteks kui vaja roppe veebikommentaare varju panna, siis tavalisel halduril võib täiesti piisata peitmise-viitest. Erandkorras tagasipaneku võib kasvõi eraldi väikese lehena ehitada. Kui aga soov mõlemas suunas

määramised samale lehele panna, siis ka see võimalik ning nii siinses näites ka tehakse. Üheks mooduseks oleks teha lehele eraldi tulp peitmise, eraldi avalikustamise tarbeks. Mõngase sättimise tulemusena pääseb aga ühe tulgaga - lihtsalt tuleb viiteid ja sõnu nõnda kohendada, et vajutuse peale olemasolevas seisus muutus tekiks.

Lehe päises on parameetrite kohta kaks valikut. Kui tuleb peitmise_id, siis vastav laul peidetakse. Kui tuleb avamise_id, siis selle id-ga rida muudetakse nähtavaks.

Õige teksti ja viite näitamiseks sobib lõik

```
$avamistekst="Ava";  
$avamisparam="avamise_id";  
$avamisseisund="Peidetud";
```

Ehk siis algul eeldatakse, et lugu pole avalik, parameetri nimeks saab avamise_id ning kasutajale nähtav sõna on "Peidetud". Alloleval real trükitakse muutujate väärtused nõnda ka lehele.

```
<td><a href='?$avamisparam=$id'>$avamistekst</a></td>
```

Kui aga päringust selgub, et laul siiski on avalik, siis pööratakse muutujate väärtused ümber ning väljatrüki tulemusena tekib oluord, kus vajutuse tulemusena pannakse lugu peitu.

```
if($avalik==1){  
    $avamistekst="Peida";  
    $avamisparam="peitmise_id";  
    $avamisseisund="Avatud";  
}
```

Nii ongi võimalik samal kohal korduvalt klõpsides laulu seisundit avalikust peidetuks ja tagasi muuta.

haldus4.php

```
<?php  
$yhendus=new mysqli("localhost", "juku", "kala", "jukubaas1");  
if(isset($_REQUEST["peitmise_id"])){  
    $kask=$yhendus->prepare("UPDATE laulud SET avalik=0 WHERE id=?");  
    $kask->bind_param("i", $_REQUEST["peitmise_id"]);  
    $kask->execute();  
}  
if(isset($_REQUEST["avamise_id"])){  
    $kask=$yhendus->prepare("UPDATE laulud SET avalik=1 WHERE id=?");  
    $kask->bind_param("i", $_REQUEST["avamise_id"]);  
    $kask->execute();  
}  
?>  
<!doctype html>  
<html>  
    <head>  
        <title>Laulud</title>  
    </head>  
    <body>  
        <h1>Laulud</h1>  
        <table>  
            <?php  
                $kask=$yhendus->prepare("SELECT id, pealkiri, avalik FROM laulud");  
                $kask->bind_result($id, $pealkiri, $avalik);  
                $kask->execute();
```



```

while($kask->fetch()){
    $pealkiri=htmlspecialchars($pealkiri);
    $savamistekst="Ava";
    $savamisparam="avamise_id";
    $savamisseisund="Peidetud";
    if($avalik==1){
        $savamistekst="Peida";
        $savamisparam="peitmise_id";
        $savamisseisund="Avatud";
    }
    echo "<tr>
        <td>$pealkiri</td>
        <td>$savamisseisund</td>
        <td><a href='?$savamisparam=$id'>$savamistekst</a></td>
    </tr>";
}
?>
</table>
</body>
</html>
<?php
    $yhendus->close();
?>

```

Alustuseks näha et eelnevalt peidetud Muumioru lood on endiselt peidus.

localhost:8080/~jaagup/if12/1/lauluhaldus/haldus4.php

Laulud

Mutionu Avatud [Peida](#)
Mutionu pidu Avatud [Peida](#)
Muumioru lood Peidetud [Ava](#)
Rongisõit Avatud [Peida](#)

localhost:8080/~jaagup/if12/1/lauluhaldus/haldus4.php?avamise_id=3

Avamisviite peale tuleb lugu avalikuks

localhost:8080/~jaagup/if12/1/lauluhaldus/haldus4.php?avamise_id=3

Laulud

Mutionu Avatud [Peida](#)
Mutionu pidu Avatud [Peida](#)
Muumioru lood Avatud [Peida](#)
Rongisõit Avatud [Peida](#)

Samuti tekib ta siis laulude üldisesse loendisse



Laulud

Mutionu	9
Mutionu pidu	8
Muumioru lood	4
Rongisõit	1

Ülesandeid

- Tee näide läbi
- Lisa ansamblite lehele avalikustamise ja peitmise võimalus.
- Võimalda eraldi peita ja näidata korraga kõiki neid ansambleid, kel pole veel ühtegi punkti.

Kommenteerimine

Veebilehtedele kirjutatakse kommentaare ja täiendusi päris mitmel puhul. Ajalehtedes kommenteeritakse uudiseid, tehnikud märgivad tehtud töid, siin püüame kokku koguda laulude kohta tehtavad kommentaarid. Keerukamal juhul tasub teha kommentaaride jaoks eraldi andmetabel, siis on võimalik neid mugavalt näiteks kirjutaja või loomisaja järgi järjestada. Lihtsamal juhul aga piisab iga laulu kohta lihtsalt ühest kommentaariväljast vastavas tulbas, kuhu inimeste kirjutatud kommentaarid üksteise otsa lisatakse.

Lisamise puhul tuleb kuidagi kindlaks määrata, et millise laulu juurde vastav kommentaar kirjutatakse. Siin näites tehakse iga laulu juures olevasse lahtrisse eraldi vorm. Sinna sisse pannakse varjatud väli laulu kohta, millele uus kommentaar kirjutatakse. Edasi lisatakse tekstiväli ning sisestusnupp. Korraga saadetakse veebilehitsejast serverisse vaid ühe vormi andmed - just selle omad, kus sisestusnuppu vajutati. Sellise trikiga saabki hoolitseda, et soovitud laulu id läheb koos kommentaaritekstiga kaasa.

```
<td>
  <form action='?'>
    <input type='hidden' name='uue_kommentaari_id' value='$id' />
    <input type='text' name='uus_kommentaar' />
    <input type='submit' value='Lisa kommentaar' />
  </form>
</td>
```

Lehe päises vaadatakse, kas saabus uue kommentaari id. Kui jah, siis lisatakse saabunud kommentaari teksti selle laulu kommentaarilahtri teksti lõppu ja pannakse reavahetus vahele. Lehe sisu avanemisel saab nõnda juba uut kommentaari näha. Käsk htmlspecialchars asendab erisümbolid, nl2br asendab tekstis olevad reavahetused HTML-i
-käskudega.

haldus5.php

```
<?php
    $yhendus=new mysqli("localhost", "juku", "kala", "jukubaas1");
    if(isset($_REQUEST["uue_kommentaari_id"])){
        $kask=$yhendus->prepare(
            "UPDATE laulud SET kommentaarid=CONCAT(kommentaariid, ?) WHERE id=?");
        $kommentaari_lisa="\n".$_REQUEST["uus_kommentaari"]."\n";
        $kask->bind_param("si", $kommentaari_lisa,
$_REQUEST["uue_kommentaari_id"]);
        $kask->execute();
    }
?>
<!doctype html>
<html>
    <head>
        <title>Laulud</title>
    </head>
    <body>
        <h1>Laulud</h1>
        <table>
            <?php
                $kask=$yhendus->prepare(
                    "SELECT id, pealkiri, kommentaarid FROM laulud");
                $kask->bind_result($id, $pealkiri, $kommentaariid);
                $kask->execute();
                while($kask->fetch()){
                    $pealkiri=htmlspecialchars($pealkiri);
                    $kommentaariid=nl2br(htmlspecialchars($kommentaariid));
                    echo "<tr>
                        <td>$pealkiri</td>
                        <td>$kommentaariid</td>
                        <td>
                            <form action='?'>
                                <input type='hidden' name='uue_kommentaari_id'
value='$id' />
                                <input type='text' name='uus_kommentaari' />
                                <input type='submit' value='Lisa kommentaar' />
                            </form>
                        </td>
                    </tr>";
                }
            ?>
        </table>
    </body>
</html>
<?php
    $yhendus->close();
?>
```

Tulemusena siis tekitab iga laulu nime järel kommentaari lisamise lahtri. Vahepealses tulbas näha laulule eelnevalt lisatud kommentaarid. Lahtrisse võib lisada uue kommentaari.

Laulud

	Hea lõõtsalugu		
	Ühe oktaavi piires		
Mutionu	Lastelaul	<input type="text"/>	Lisa kommentaar
Mutionu pidu		<input type="text"/>	Lisa kommentaar
Muumioru lood	Kõlab instrumentaalis paremini	<input type="text"/>	Lisa kommentaar
Rongisõit		Piilupardi lugu	Lisa kommentaar

Pärast lisamisnupule vajutamist näeb seda kommentaari juba laulu nime järel.

Laulud

	Hea lõõtsalugu		
	Ühe oktaavi piires		
Mutionu	Lastelaul	<input type="text"/>	Lisa kommentaar
Mutionu pidu		<input type="text"/>	Lisa kommentaar
Muumioru lood	Kõlab instrumentaalis paremini	<input type="text"/>	Lisa kommentaar
Rongisõit		Piilupardi lugu	Lisa kommentaar

Ülesandeid

- Tee näide läbi
- Lisa kommenteerimisvõimalus ansamblivaliku lehele
- Lisa kommentaarile automaatselt ka aeg PHP date-käskluse abil

Haldus laulude kaupa

Tähelepanekud õppijate juures kipuvad näitama, et kui ühe toiminguga lehest eraldi aru saadud ning mõistetakse selle põhjal ka sarnaseid lehti teisel teemal koostada - sellest veel ei pruugi piisata, et mitme toimetuse ühele lehele kokkupanek sama hõlpsasti käiks. Seetõttu ka siin näide, kuidas punktide haldus ning kommentaaride lisamine võimalik samale lehele kokku tõsta. Lihtsamal kujul käiks see nõnda, et laulu taga on kaks tulpa - üks punktide määramiseks ning teine kommentaari lisamiseks. Ning üleval oleks vastavad andmetabeli muutmise plokid järjestikku, if-lausega saab kontrollida, et kumma toiminguga parajasti tegemist.

Siin aga on keerukamalt ette võetud ning tehtud nõnda, et laulude loetelu on eraldi ning sealt valitud laulu andmed näidatakse lehe ülaosas. Kuna korraga aktiivne vaid üks laul, siis jagub lehel rohkem ruumi temaga seotud ettevõtmiste tarbeks.

Laulude loetellu küsitakse kasutaja lehele välja avalikud laulud. Iga laulu pealkiri muutub väljatrükil viiteks, mis näitab samale veebilehele, kuid kuhu antakse kaasa valitud laulu id (`href='?id=$id'`).

```
<table>
  <?php
    $kask=$yhendus->prepare(
      "SELECT id, pealkiri, punktid FROM laulud WHERE avalik=1");
    $kask->bind_result($id, $pealkiri, $punktid);
    $kask->execute();
    while($kask->fetch()){
      $pealkiri=htmlspecialchars($pealkiri);
      echo "<tr>
        <td><a href='?id=$id'>$pealkiri</a></td>
        <td>$punktid</td>
      </tr>";
    }
  ?>
</table>
```

Kui laulu id valiti, sellisel juhul küsitakse lehel pärast body algust välja päring vaid selle laulu andmete kätte saamiseks.

```
if(isset($_REQUEST["id"])){
    $kask=$yhendus->prepare("SELECT id, pealkiri, kommentaarid, punktid,
    lisamisaeg FROM laulud WHERE id=?");
```

Sealt edasi võib siis ka laulu kommentaari sisestada või punkti lisada.

Andmete tegelik baasi kirjutamine toimub lehe uuel avanemisel kui vastavad parameetrid kaasas. See osa näha kohe uue väljatrüki alguses.

laululeht.php

```
<?php
$yhendus=new mysqli("localhost", "juku", "kala", "jukubaas1");
if(isSet($_REQUEST["healaulu_id"])){
    $kask=$yhendus->prepare("UPDATE laulud SET punktid=punktid+1 WHERE id=?");
    $kask->bind_param("i", $_REQUEST["healaulu_id"]);
    $kask->execute();
}
if(isSet($_REQUEST["uue_kommentaari_id"])){
    $kask=$yhendus->prepare(
        "UPDATE laulud SET kommentaarid=CONCAT(kommentaariid, ?) WHERE id=?");
    $kommentaarilisa="\n".$_REQUEST["uus_kommentaari"]."\n";
    $kask->bind_param("si",
        $kommentaarilisa, $_REQUEST["uue_kommentaari_id"]);
    $kask->execute();
}
?>
<!doctype html>
<html>
    <head>
        <title>Laulud</title>
    </head>
    <body>
        <?php
            if(isSet($_REQUEST["id"])){
                $kask=$yhendus->prepare(
                    "SELECT id, pealkiri, kommentaarid, punktid, lisamisaeg
                     FROM laulud WHERE id=?");
                $kask->bind_param("i", $_REQUEST["id"]);
                $kask->bind_result(
                    $id, $pealkiri, $kommentaariid, $punktid, $lisamisaeg);
                $kask->execute();
                if($kask->fetch()){
                    $pealkiri=htmlspecialchars($pealkiri);
                    $kommentaariid=nl2br(htmlspecialchars($kommentaariid));
                    echo "
                        <h2>$pealkiri</h2>
                        <dl>
                            <dt>Punkte:</dt>
                            <dd>$punktid</dd>
                            <dt>Lisatud:</dt>
                            <dd>$lisamisaeg</dd>
                            <dt>Kommentaariid:</dt>
                            <dd>$kommentaariid</dd>
                        </dl>
                        <a href='?healaulu_id=$id'>Lisa punkt</a><br />
                        <form action='?'>
                            <input type='hidden'
                                name='uue_kommentaari_id' value='$id' />
                            <input type='text' name='uus_kommentaari' />
                            <input type='submit' value='Lisa kommentaar' />
                        </form>
                    ";
                $kask->close();
            }
        ?>
    <h1>Laulud</h1>
    <table>
```

```

<?php
    $kask=$yhendus->prepare(
        "SELECT id, pealkiri, punktid FROM laulud WHERE avalik=1");
    $kask->bind_result($id, $pealkiri, $punktid);
    $kask->execute();
    while($kask->fetch()){
        $pealkiri=htmlspecialchars($pealkiri);
        echo "<tr>
            <td><a href='?id=$id'>$pealkiri</a></td>
            <td>$punktid</td>
            </tr>";
    }
?>
</table>
</body>
</html>
<?php
    $yhendus->close();
?>

```

Algul avaneb lehel laulude loetelu nagu tavaliselt, igähele taha kirjutatud selle laulu punktide arv.



Laulud

Mutionu	9
Mutionu pidu	8
Muumioru lood	4
Rongisõit	1

Laulule vajutades saadetakse uuele lehepäringule kaasa selle laulu id (näha aadressiribal).

Saabunud id järgi küsitakse välja vastava laulu muud andmed ning näidatakse kasutajale. Samuti pannakse sinna siis viide punkti lisamiseks ning koht kommentaari sisestamiseks.

Mutionu pidu

Punkte:

8

Lisatud:

2012-10-10 13:30:30

Kommentaariid:

[Lisa punkt](#)

Lisa kommentaar

Laulud

[Mutionu](#) 9

[Mutionu pidu](#) 8

[Muumioru lood](#) 4

[Rongisõit](#) 1

Punktiviitele vajutamisel lisatakse punkt andmetabelisse ning seda näeb lehe järgmisel avamisel. Mutionu pidu on kaheksale punktile ühe juurde saanud.

Laulud

[Mutionu](#) 9

[Mutionu pidu](#) 9

[Muumioru lood](#) 4

[Rongisõit](#) 1

Ja kommentaariid jõuavad ka ilusti laulule külge.



Mutionu pidu

Punkte:

9

Lisatud:

2012-10-10 13:30:30

Kommentaariid:

[Lisa punkt](#)

Päris pikk laul

Lisa kommentaar

Laulud

[Mutionu](#) 9

[Mutionu pidu](#) 9

[Muumioru lood](#) 4

[Rongisõit](#) 1

Nõnda võib mõlema oskuse koos toimimist ühel lehel imetleda ning omale järgmiste rakenduste loomise juures alusnäidisena võtta.

Mutionu pidu

Punkte:

9

Lisatud:

2012-10-10 13:30:30

Kommentaariid:

Päris pikk laul

[Lisa punkt](#)

Lisa kommentaar

Laulud

[Mutionu](#) 9

[Mutionu pidu](#) 9

[Muumioru lood](#) 4

[Rongisõit](#) 1

Ülesandeid

- Tee näide läbi
- Pane ka ansamblite lehel kommenteerimine ja häälte andmine samale lehele
- Võimalda anda ka vastuhääli, st hääli vajutusega vähemaks võtta

Kohviautomaat

Andmetabeli kuju: (id, jooginimi, topsepakis, topsejuua)

Topside arv pakis näitab, mitu topsitait saab juua ühe täitepakendi sisestamise peale.

- Loo tabel SQL-lausega. Lisa joogina kohv. Täitepaki suuruseks 50 topsi jagu pulbrit, algul masin tühi, juua pole midagi. Loo SQL-lause juua olevate topside arvu suurendamiseks täitepaki jagu. Käivita.
- Automaadi käivitatav leht vähendab juua olevate topside arvu ühe võrra. Vaataja leht näitab seda arvu.
- Automaat saab hakkama mitme joogiga (kohv, tee, kakao). Lehel näidatakse vaid neid jooke, millel on vähemasti üks tops juua. Joomise tulemusena vähendatakse vastava joogi olemasolevate topside loendurit. Halduslehel saab joodavate topside arvu kogust suurendada täitepaki jagu.