

LIKUNTAPÄIVÄKIRJA

Pajulahden urheiluopisto – Koulutuskeskus Salpaus

LAHDEN AMMATTIKORKEAKOULU

Tekniikan ala

Tietotekniikka

Ohjelmistotekniikka

Opinnäytetyö

Kevät 2011

Henri Hellsten

Lahden ammattikorkeakoulu
Tietotekniikka

HELLSTEN, HENRI:

Liikuntapäiväkirja
Pajulahden urheiluopisto –
Koulutuskeskus Salpaus

Tietotekniikan opinnäytetyö, 39 sivua

Kevät 2011

TIIVISTELMÄ

Tässä opinnäytetyössä toteutetaan web-pohjainen liikuntapäiväkirjasovellus Pajulahden urheiluopiston sekä koulutuskeskus Salpauksen käyttöön. Sovelluksen on tarkoitus toimia etätyökaluna opiskelijoille ja opettajille liikuntakurssien suorituksessa. Valmentajina toimivien käyttäjien on tarkoitus rakentaa kuntoilijoilleen viikoittaiset liikuntaohjelmat, ja kuntoilijoiden on taas tarkoitus täyttää suorittamiaan liikuntasuorituksia päiväkirjaansa. Opettajat voivat seurata näiden kahden ryhmän toimintaa ja sen pohjalta antaa heille kurssin lopuksi arvosanat.

Kommunikointi muiden käyttäjien kanssa on hoidettu tekemällä kurssikohtaiset keskustelualueet. Sivustolta on myös mahdollista saada ulos tulostettava viikoittainen suoritusten yhteenveto, koska valmentajat tarvitsevat paperisena viikoittaiset raportit tekemistään ohjelmista ja siitä, miten kuntoilijat ovat niitä suorittaneet. Kuntoilijoiden viikoittaista etenemistä voi seurata reaaliajassa päivittyvästä liikuntapiirakasta.

Sivusto on toteutettu kokonaan dynaamiseksi käyttäen JavaScriptin jQuery-kirjastoa ja sen tarjoamia Ajax-toimintoja. Palvelimenpään ohjelmointi on hoidettua käyttäen PHP:tä ja tietokantana sivustolla on MySQL. Versionhallintajärjestelmänä projektissa toimi Git.

Avainsanat: liikuntapäiväkirja, jQuery, Ajax, PHP, MySQL, HTML, CSS, web-ohjelmointi

Lahti University of Applied Sciences
Degree Programme in Information Technology

HELLSTEN, HENRI:

Sports diary
Pajulahti Sports Institute –
Koulutuskeskus Salpaus

Bachelor's Thesis in Information Technology, 39 pages

Spring 2011

ABSTRACT

The objective of this thesis was to create a web-based exercising diary application for the use of Pajulahti Sports Institute and koulutuskeskus Salpaus. The purpose of the application is to serve as a remote tool for students and teachers attending sports classes. Students that acts as trainers are meant to build weekly exercising schedules for their trainees. Trainees are meant to make a note of the exercises they have done into their own diaries. Teachers are able to follow all the activities done by these two groups and in the end of the course give these students their grades based on the activities.

Communication between all the users is done trough course based forums. It is also possible to print out a summary of the exercises done on a weekly basis. This is because trainers needs to get a paper version of each week's schedule they have made that also shows how their own trainee followed it. It is also possible to follow the weekly progress of each trainee in real time from a visual exercising pie.

The web page was created fully dynamic using jQuery JavaScript library and Ajax functions offered by the library. Server-side programming was done by using PHP and the database in the web page is MySQL. Git served as a version control system in this project.

Key words: sports diary, jQuery, Ajax, PHP, MySQL, HTML, CSS, web-programming

SISÄLLYS

1	JOHDANTO	1
2	ASIAKASVAATIMUKSET	2
2.1	Liikuntakurssi	2
2.2	Liikuntapäiväkirja	3
2.3	Kommunikointi	5
2.4	Muuta	6
3	JQUERY	7
3.1	Yleisesti	7
3.2	Käyttöönotto ja käyttö	8
3.3	DOM-manipulointi	9
3.4	Kommunikointi serverin kanssa	13
3.5	Graafikolle	16
4	LIIKUNTAPÄIVÄKIRJA	17
4.1	Yleistä	17
4.2	Suunnittelu	17
4.3	Dynaamisuus	18
4.4	Sivusto	21
4.4.1	Käyttäjryhmät	21
4.4.2	Rekisteröityminen ja sisäänkirjautuminen	22
4.4.3	Kurssit	24
4.4.4	Päiväkirja	25
4.4.5	Keskustelut	30
4.4.6	Linkit	31
4.4.7	Käyttäjät	32
4.4.8	Asetukset	33
4.4.9	Sivuston ohje	34
4.5	Yhteensopivuus ja testaus	35
5	YHTEENVETO	37
	LÄHTEET	38

1 JOHDANTO

Tässä opinnäytetyössä luotiin liikuntapäiväkirjasivusto Pajulahden urheiluopistolle sekä koulutuskeskus Salpaukselle. Pajulahdessa sijaitsevassa urheiluopistossa järjestetään liikunnan ammatillista koulutusta sekä nuorille että aikuisille. Koulutuskeskus Salpaus tarjoaa paljon erilaisia koulutusmahdollisuuksia nuorille sekä aikuisille niin ammatillisella kuin lukiopuolellakin.

Liikuntapäiväkirjasivuston on tavoitteena toimia etätyökaluna liikuntakursseja suoritettaville opiskelijoille ja opettajille. Pajulahden urheiluopiston opiskelijat toimivat sivustolla valmentajien roolissa ja laativat omille kuntoilijoilleen viikoittaisia kuntoilu aikatauluja. Koulutuskeskus Salpauksen opiskelijat taas toimivat kuntoilijoiden roolissa, jotka yrittävät pysytellä laadituissa aikatauluissa ja merkata tekemisiään päiväkirjaansa. Opettajien tulee pystyä näkemään kaikki opiskelijoiden välillä tapahtuva kommunikointi, aikataulut ja päiväkirjamerkinnät. Näiden tietojen avulla opettajat antavat arvosanat opiskelijoilleen kurssien lopussa. Opettajien on myös pystyttävä jättämään sivustolle ilmoituksia kurssiin liittyvistä asioista tai tulevista tapaamisista.

Liikuntapäiväkirjan käyttäjät tulevat olemaan eri paikkakunnilla kurssin suorituksen aikana, joten työ päädyttiin toteuttamaan web-pohjaiseksi. Tämä mahdollistaa liikuntapäiväkirjan käytön miltä tahansa tietokoneelta, jossa on internetselain sekä internetyhteys. Tästä johtuen ongelmaksi tulee sivuston toimivuus eri internet-selaimilla ja niiden eri versioilla.

Tässä työssä luvussa kaksi käydään läpi liikuntapäiväkirjaa koskevat asiakkaan vaatimukset ja tarpeet. Luvussa kolme perehdytään jQueryn käyttöön ja sen yleisimpiin metodeihin, joita työssä on käytetty. Luku neljä sisältää tarkemman kuvauksen liikuntapäiväkirjasivuston rakenteesta ja toiminnasta.

2 ASIAKASVAATIMUKSET

2.1 Liikuntakurssi

Liikuntakurssien alussa kaikki siihen osallistuvat opiskelijat ja opettajat tapaavat toisensa Pajulahdessa järjestettävässä liikuntapäivässä. Tämän liikuntapäivän aikana käydään läpi kuntoiluun liittyvää asiaa sekä tutustutaan erilaisiin liikuntamuotoihin ja mahdollisuuksiin. Opiskelijoilla on myös mahdollisuus tutustua Pajulahden urheiluopiston järjestämiin palveluihin.

Teoriaosuuden jälkeen koulutuskeskus Salpauksen sekä Pajulahden urheiluopiston opiskelijoista muodostetaan parit. Jokainen Salpauksen opiskelija saa itselleen yhden henkilökohtaisen kuntovalmentajan, joka on urheiluopiston liikunnanohjaajaksi opiskeleva opiskelijasta. Tämä pari tulee pysymään samana koko kurssin suorituksen ajan.

Koulutuskeskus Salpauksen opiskelijat suorittavat liikuntapäivän aikana ensimmäisen kuntotestin, jonka urheiluopiston liikunnanohjaajat ovat laatineet. Tämän kuntotestin sekä liikunnanohjaajan ja kuntoilijan välisen henkilökohtaisen keskustelun pohjalta liikunnanohjaajat pystyvät laatimaan omille kuntoilijoilleen heidän senhetkiseen kuntoonsa sopivat viikoittaiset kuntoilu-aikataulut. Nämä aikataulut tulee saada syötettyä liikuntapäiväkirjaan, koska liikuntakurssi suoritetaan etänä.

Itse liikuntakurssi kestää kahdesta kolmeen kuukautta, jonka aikana opiskelijat ja opettajat kommunikoivat toistensa kanssa käyttäen liikuntapäiväkirjaa. Kurssin päätteeksi järjestetään uusi liikuntapäivä Pajulahdessa. Tämän liikuntapäivän aikana Salpauksen opiskelijat suorittavat saman kuntotestin kuin kurssin alussakin. Näiden kahden kuntotestin tuloksia vertaillaan toisiinsa ja selvitetään, mitä kuntoilijoiden kunnoille on kurssin aikana käynyt. Liikunnanohjaajat neuvovat myös tarvittaessa, miten kuntoilijoiden tulisi jatkaa siitä eteenpäin.

2.2 Liikuntapäiväkirja

Valmentajien on tarkoitus laatia kuntoilijoilleen viikoittainen kuntoilu aikataulu sivuston päiväkirja-osioon. Päiväkirjaan tulee pystyä lisäämään suorituksia viikon jokaiselle päivälle. Kuntoilijoiden on pystyttävä näkemään valmentajien lisäämät aikataulut sekä merkitsemään, miten he ovat laadittuja aikatauluja suorittaneet ja miltä suoritukset ovat tuntuneet. Kuntoilijoiden on myös pystyttävä lisäämään itse suorituksia päiväkirjaan, jos ovat tehneet myös jotain muuta, kuin mitä valmentaja on laatinut.

Valmentajat laativat viikoittaiset kuntoilu aikataulut UKK-Instituutin liikuntapiirakan pohjalta. Kuviossa 1 on UKK-Instituutin laatima virallinen liikuntapiirakka. Liikuntapiirakan ulompi ympyrä kuvaa kestävyyskuntoliikunnan määrää viikoittaisella tasolla ja sisempi ympyrä lihaskuntoa ja liikehallintaa.

Kestävyyskunto on jaettu kahteen eri osaan: reippaasti ja rasittavasti. Kumpikin osa sisältää erilaisia liikuntalajeja, joista rasittavan alla on luonnollisesti hieman rasittavampia liikuntalajeja. Saadakseen viikoittaisen liikuntasuosituksen täyteen tulee kuntoilijan suorittaa kestävyyskuntoa reippaasti alta löytyviä liikuntalajeja yhteensä kaksi ja puoli tuntia viikossa tai vaihtoehtoisesti kestävyyskuntoa rasittavasti alta löytyviä liikuntalajeja yksi tunti ja 15 minuuttia viikossa. Lihaskuntoa ja liikehallintaa tulee suorittaa kaksi kertaa viikossa. Näihin kumpaankaan ei ole mitään tarkkaa aikarajaa, mutta valmentajien tulisi tietää, mitä ja kuinka paljon kuntoilijoiden tulisi milloinkin tehdä. Aikataulut laaditaan jokaiselle kuntoilijalle henkilökohtaisesti heidän senhetkistä kuntoaan vastaavaksi.



KUVIO 1. UKK-Instituutin laatima liikuntapiirakka (UKK-Instituutti 2011)

Liikuntapäiväkirjaa täytettäessä kuntoilijoiden on myös merkittävä, miltä minkäkin suorituksen rasitus heistä itsestään tuntui. Tähän käytetään apuna Borgin asteikkoa. Kuviossa 2 on UKK-Instituutin laatima Borgin asteikko. Asteikossa on numeraaliset arvot erilaisille rasitustasoille ja kutakin rasitustasoa vastaava selvitys. Kuntoilijoiden tulee merkitä ”Miltä rasitus tuntuu?” –sarakkeen alta löytyvä teksti jokaiseen suoritukseensa. Borgin asteikossa on selvennetty selkokielellä, miltä hengityksen tulisi tuntua milläkin rasitustasolla. Selvennys löytyy taulukon sarakkeesta ”Havainnoi hengitystä”.

Miltä rasitus tuntuu nyt?*

	Miltä rasitus tuntuu?	Sopivuus	Havainnoi hengitystä
6			
7	erittäin kevyt		normaali hengitysrytmi
8			
9	hyvin kevyt		
10		sopii kaikille	hengitys kiihtyy (puhuminen sujuu)
11	kevyt		
12		sopii terveys- ja kuntoliikujille	hengästyttää (puhuminen vaikeutuu)
13	hieman rasittava		
14			
15	rasittava		
16		sopii kovalle kuntoilijoille ja urheilijoille silloin tällöin	puuskuttaa (puhuminen mahdotonta)
17	hyvin rasittava		
18			
19	erittäin rasittava		
20			

*ns. Borgin asteikko UKK-instituutti

KUVIO 2. UKK-Instituutin laatima Borgin asteikko (UKK-Instituutti 2010)

Liikuntapäiväkirjassa tulee pystyä valitsemaan jokin liikuntapiirakan liikuntamuodoista sekä kyseisen liikuntamuodon alta löytyvä liikuntalaji. Kuntoilijoiden on myös pystyttävä lisäämään kyseisen liikuntalajin suorituksen aikana tuntunut rasitustaso. Valmentajien on pystyttävä merkitsemään, kuinka kauan suorituksen tulisi kestää minuutteina sekä kuntoilijoiden, kuinka kauan heillä siihen aikaa lopullisesti meni. Valmentajien ja kuntoilijoiden on myös pystyttävä jättämään vapaamuotoinen kommentti jokaista liikuntasuoritusta koskien. Opettajat pystyvät näkemään kaikki liikuntapäiväkirjaan tehdyt merkinnät.

2.3 Kommunikointi

Opettajien on pystyttävä ilmoittamaan kaikille kurssin opiskelijoille kurssiin liittyvistä asioista tai tulevista tapaamisista. Myös opiskelijoiden on tarvittaessa pystyttävä jättämään viestiä kurssiin liittyvistä asioista opettajille tai kaikille muille opiskelijoille.

2.4 Muuta

Käyttäjien tulisi pystyä jättämään omia tarpeelliseksi katsomiaan linkkejä sivustolle muiden käyttäjien nähtäväksi. Linkkien pitäisi näkyä listana, ja jokaisella linkillä on jokin sitä kuvaava nimi. Sivuston käytöstä on myös tehtävä ohje käyttäjille. Eri käyttäjäryhmät voivat tehdä eri asioista sivustolla, joten jokaisen käyttäjäryhmän ohje tulisi näyttää vain heille tarkoitettut asiat.

3 JQUERY

jQuery on JavaScript-kirjasto, jonka tarkoitus on yksinkertaistaa ja nopeuttaa JavaScriptin kirjoitusta ja käyttöä. Tämän takia jQueryn motto onkin ”write less, do more” eli kirjoita vähemmän, tee enemmän. Yksi hyödyllisimmistä ominaisuuksista on internet-selaimen tunnistus ja sen myötä JavaScriptin käyttö siten, että se toimii samalla lailla kaikilla selaimilla. jQuery sisältää kaikki yleisimmät JavaScript-kirjaston ominaisuudet DOM-manipuloinnista animaatioihin ja Ajax:n käyttöön. jQuery on myöskin yksi tunnetuimmista ja käytetyimmistä JavaScript-kirjastoista maailmalla. Sitä käyttää monet isot yritykset, kuten Microsoft ja Nokia. (The jQuery Project 2010.)

3.1 Yleisesti

jQuery on John Resigin käsialaa ja se syntyi New Yorkissa järjestetyssä BarCamp konferenssissa vuonna 2006. Tämän jälkeen se on kehittynyt yhdeksi käytetyimmistä JavaScript-kirjastoista. Sen tarkoitus on siis vähentää kehittäjien koodin kirjoittamisen tarvetta ja täten myös nopeuttaa projektien valmistumista. Vaikka jQuery vähentää ja lyhentää koodia, on sen syntaksi silti helppo ja yksinkertainen lukea sekä kirjoittaa. (Wikipedia 2011a.)

jQuery-kirjasto on yksittäinen tiedosto. Sen saa joko pakkaamattomana kommenttien kera n. 207kt (ns. kehittämismoodi) tai pakattuna kommentit ja kaikki turhat merkit poistettuna n. 29kt. Pakattu tiedosto on se, jonka käyttäjä haluaa laittaa lopulliseen tuotukseensa, koska se latautuu nopeammin pienen kokonsa takia. (Wikipedia 2011a.)

Helpon syntaksin ansiosta jQueryn käytön opettelemiseen ei mene kuin muutamia päiviä. JavaScriptin tuntemus auttaa asiaa huomattavasti. jQuery sisältää erittäin suuren määrän toimintoja, ja niiden kaikkien opettelemiseen voi tosin mennä kuukausia.

jQuery on ilmainen JavaScript-kirjasto, jota voi kuka vain käyttää vapaasti. Kuten JavaScriptiä, voi myös jQueryä käyttää lähes minkä tahansa web-ohjelmointikielen kanssa. Yksi jQueryn parhaista puolista on ominaisuus, joka testaa miten käyttäjän selain suorittaa kutakin toimintoa. Jos selain ei osaa suorittaa toimintoa niin kuin pitäisi, niin jQuery yrittää kiertää sen ja saada toimimaan käyttämällä toisenlaista menetelmää. Tämän ansiosta jQueryllä tehtyjen koodien pitäisi toimia jokaisella selaimella ja vieläpä lähestulkoon samalla tapaa.

3.2 Käyttöönotto ja käyttö

jQueryn käyttöönotto on tehty helpoksi, käyttäjän täytyy vain linkittää jQuery-kirjasto sivulle, jossa sitä haluaa käyttää. Kuviossa 3 on esimerkki jQueryn linkittämisestä. Esimerkki olettaa, että jquery.js tiedosto on samassa hakemistossa kuin sivu, joka sitä käyttää. jQuery-tiedoston voi myös linkittää suoraan ulkoisesta palvelusta esim. Microsoftilta tai Googlelta. Kuviossa 4 on esimerkki tiedoston linkittämisestä Googlen palvelun kautta.

```

1 <head>
2 <script type="text/javascript" src="jquery.js"></script>
3 </head>

```

KUVIO 3. jQueryn käyttöönotto paikallisesta tiedostosta

```

1 <head>
2 <script
3   type="text/javascript"
4   src="https://ajax.googleapis.com/ajax/libs/jquery/1.5.1/jquery.min.js">
5 </script>
6 </head>

```

KUVIO 4. jQueryn käyttöönotto Googlen palvelusta

Yleisin tapa esitellä jQuery-kirjasto on ready() –funktion käyttö. jQueryä voi käyttää kahdella eri tapaa. Suoraan \$ –funktion kautta käytettynä metodit ovat aina jQueryn objekteja ja täten myös niihin voi ketjuttaa muita metodeja. \$. etuliite funktiot ovat työkalufunktioita, jotka eivät toimi jQueryn objekteina. Kuviossa 5 on esimerkki ready() –funktion käytöstä. (Wikipedia 2011a.)

Kuvion 5 jQuery esimerkki suorittaa `ready()` –funktion sisällä olevan koodin heti, kun sivuston DOM on latautunut kokonaan. Koodi hakee DOM:sta elementin tai elementit, joiden `id`-attribuutiksi on määritelty ”`div1`”. Elementit löydettyään se suorittaa jQueryn `css()` –metodin, joka vaihtaa tai asettaa kyseisten elementtien CSS attribuutin ”`color`” arvoksi ”`#f00`”. Eli elementtien sisällä olevan tekstin väri vaihtuu punaiseksi. Tämä vaikuttaa myös kaikkiin sisarelementteihin, joka tosin on CSS:n vakio-ominaisuus.

```

1 <script type="text/javascript">
2 $(document).ready(function(){
3   $('#div1').css('color', '#f00');
4 });
5 </script>

```

KUVIO 5. Elementin fontin värin vaihto jQueryn `css()` –metodilla

3.3 DOM-manipulointi

DOM eli Document Object Model on puumainen tapa esittää tietoa internet-sivustoilla. Se ei ole kielisidonnainen, ja se toimii kaikilla alustoilla. DOM:n tarkoitus on esittää ja tehdä HTML, XHTML ja XML dokumenttien objekteista vuorovaikuttaisia. DOM objektit sisältävät elementtejä, joihin voidaan viitata ja niitä voidaan manipuloida käytetyn ohjelmointikielen syntakseilla. (Wikipedia 2011b.)

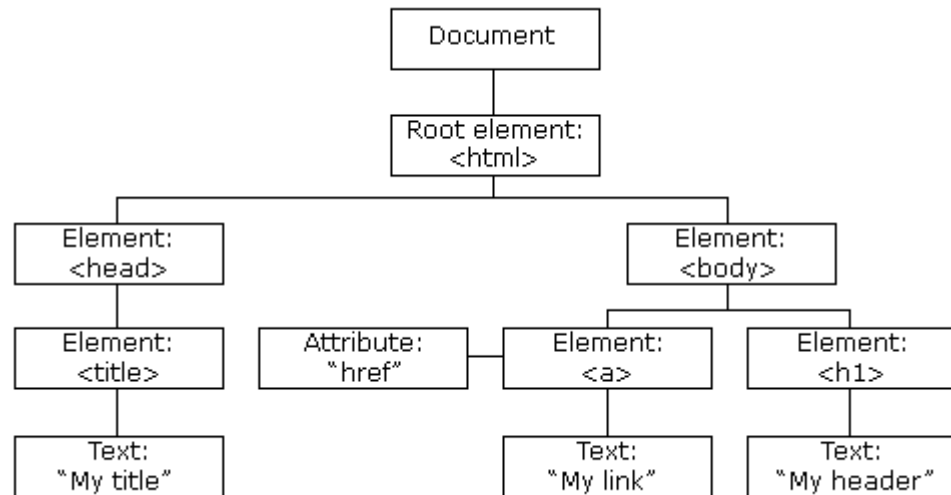
Kuviossa 6 on esimerkki HTML-lähdekoodista ja kuviossa 7 on esimerkki lähdekoodista muodostuvasta DOM-puusta. Kuvion 6 ensimmäisen ja yhdeksännen rivien on määritelty DOM-puun document-objektin juuri-elementin eli `html`-elementin sisältö. DOM-puussa `html`-elementin jälkeen tulevat `head`- ja `body`-elementit. Kuvion 6 kolmannella rivillä oleva `title`-elementti tulee puurakenteessa vasta `head`-elementin jälkeen ja `title`-elementin sisällä oleva teksti heti `title`-elementin jälkeen. Kuviossa 6 kuudennella rivillä oleva `a`-elementti sisältää attribuutin ”`href`” sekä tekstiä. Attribuutti ja teksti tulevat molemmat puurakenteessa suoraan `a`-elementistä jälkeen.

```

1 <html>
2   <head>
3     <title>My title</title>
4   </head>
5   <body>
6     <a href="">My link</a>
7     <h1>My header</h1>
8   </body>
9 </html>

```

KUVIO 6. Esimerkki HTML-sivun rakenteesta



KUVIO 7. Kuvion 7 HTML-lähdekoodista muodostuvasta DOM-puusta (W3Schools 2011)

jQueryssä voidaan valita elementtejä samalla syntaksilla kuin CSS-tyylissä. `$(')` –funktion sisään sijoitetaan hakukriteeri tai elementti, joka halutaan valita. Kuviossa 5 sivulla 9 on yksi esimerkki elementin valinnasta CSS:n syntaksilla. Kun elementti on valittu voi siihen alkaa ketjuttamaan muita jQueryn metodeja perään. (Weyl 2008.)

jQueryssä korvataan JavaScriptin `innerHTML` määreen omalla `html()` –metodilla. Tällä metodilla pystyy sekä hakemaan elementin sisällön, että asettamaan siihen uutta sisältöä. JavaScriptin `textNodes` määre on taas korvattu `text()` –metodilla, jolla myöskin voi sekä hakea että asettaa elementin sisällön. (Weyl 2008.)

Kuviossa 8 ensimmäisellä rivillä oleva koodi käyttää jQueryn `html()` –metodia ja hakee sen avulla elementistä, jonka `id` on `"mainContent"`, kaiken sisällön `html-`

tagit mukaan lukien ja asettaa ne JavaScript muuttujaan nimeltä ”myHTMLContent”. Kuvion toisella rivillä käytetään samaa metodia, mutta tällä kertaa sen avulla asetetaan dokumentin ensimmäiseen ”h2” elementtiin html() –metodin sisällä oleva teksti. Kuvion kolmannella ja neljännellä rivillä käytetty text() –metodi toimii täysin samalla tavalla kuin html() –metodi poikkeuksena kuitenkin se, että sillä ei voi asettaa eikä palauttaa html-tageja. (Weyl 2008.)

```

1 var myHTMLContent = $("#mainContent").html();
2 $("h2:first").html('This is new <em>header</em> content');
3 var myTextContent = $("#mainContent").text();
4 $("h2:first").text('Ugly <em>header</em>');

```

KUVIO 8. jQueryn html() ja text() –metodit (Weyl 2008)

Elementtien lisääminen jQueryllä tapahtuu before() ja after() –metodeilla. Before asettaa halutun sisällön ennen valittua elementtiä, kun taas after asettaa sisällön valitun elementin jälkeen. Kuviossa 9 on esimerkki metodien käytöstä.

Ensimmäinen rivi asettaa metodin sisällä olevan tekstin valitun elementin eteen, ja toinen rivi asettaa sisällön valitun elementin jälkeen. (Weyl 2008.)

```

1 $("#leftNav ul").before("<h2>Left Navigation</h2>");
2 $("#leftNav ul").after("<h4>New Header</h4>");

```

KUVIO 9. jQueryn before() ja after() –metodit (Weyl 2008)

Elementtien poisto ja tyhjennys jQueryn avulla tapahtuu metodeilla remove() ja empty(). Remove poistaa valitun elementin kokonaan dokumentista, kun taas empty poistaa valitun elementin koko sisällön mutta jättää itse elementin dokumenttiin. Kuvion 10 ensimmäisellä rivillä on esimerkki remove metodista. Se poistaa valitun elementin sisältä ensimmäisen ”li” –elementin. Kuvion toisella rivillä oleva koodi taas poistaa jokaisen ”#leftNav” –elementin sisällä olevan ”li” –elementin. Kuvion kolmannella rivillä näkee, että remove metodi pystyy ottamaan myös parametreja toisin kuin empty metodi. Kolmas rivi poistaa ”#leftNav” –elementin sisältä kaikki ”li” –elementit, jotka sisältävät tekstin ”Second”. Empty metodi toimii samalla tavalla kuin remove. Kuvion neljännellä rivillä tyhjennetään ”#mainContent” –elementin ensimmäinen ”h2” –elementti ja viidennellä rivillä tyhjennetään ”#mainContent” –elementin kaikki ”h2” –elementit, jotka sisältävät tekstin ”Second”. (Weyl 2008.)

```

1 $("#leftNav li:first").remove();
2 $("#leftNav li").remove();
3 $("#leftNav li").remove(":contains('Second')");
4 $("#mainContent h2:first").empty();
5 $("#mainContent h2:contains('second')).empty();

```

KUVIO 10. jQueryn remove() ja empty() –metodit (Weyl 2008)

jQuery mahdollistaa helpon navigoinnin DOM-elementtien isäntiin käyttämällä parent() tai parents() –metodeja. Parent() –metodi liikkuu DOM-puussa yhden elementin taaksepäin ja valitsee tämän elementin tehden siitä jQueryn objektin. Parents() –metodi valitsee jokaisen isäntäelementin DOM-puusta html-elementtiin asti ja tekee näistä elementeistä taulukkomaisen jQuery objektin. Parents() –metodi ottaa myös sisäänsä attribuutin, jolla voi CSS-tyylisesti määritellä, mitä elementtejä isännistä valitaan.

Kuviossa 11 on esimerkki HTML sivuston rakenteesta, ja kuviossa 12 on esitellään, kuinka parent() ja parents() –metodit toimivat. Kuvion 12 esimerkissä lähdetään aina liikkeelle kuvion 11 span-elementistä ja kommentoidut kohdat kuviossa 12 kertovat, minkälaisen objektin kyseinen kutsu palauttaa. Kuvion 12 ensimmäisellä rivillä oleva koodi palauttaa siis jQueryn objektin, joka sisältää sen isäntäelementin eli p-elementin. Kuvion toisella rivillä käytetty metodi parents() valitsee tässä esimerkissä kaikki kuvion 11 elementit pois lukien span-elementin. Kuvion 12 kolmannella rivillä käytetään parents() –metodin attribuuttia, joka tässä tapauksessa valitsee kaikki div-elementit, jotka ovat kuvion 11 span-elementin isäntiä. Neljännellä rivillä kuviossa 12 käytetään jQueryn ”eq()” valitsinta, joka valitsee tässä tapauksessa objektin ensimmäisen arvon. Tämä objekti siis palauttaa ”div.two” –elementin, koska se on span-elementistä taaksepäin lähtiessä ensimmäinen isäntäelementti, ”div.one” on taas kyseisen elementin seuraava isäntä, ja täten sitä kutsuttaisiin valitsimella ”eq(1)”.


```

1 <html>
2 <body>
3
4 <div class="one">
5   <div class="two">
6     <p>
7       <span>Some text</span>
8     </p>
9   </div>
10 </div>
11
12 </body>
13 </html>

```

KUVIO 11. Esimerkki HTML-sivun rakenteesta (Jdsharp 2008)

```

1 $('span').parent(); // [p]
2 $('span').parents(); // [p, div.two, div.one, body, html]
3 $('span').parents('div'); // [div.two, div.one]
4 $('span').parents('div:eq(0)'); // [div.two]

```

KUVIO 12. jQueryn parent() ja parents() –metodit ja niiden palautusarvot kuvion 11 esimerkissä (Jdsharp 2008)

jQueryssä on myös mahdollista hakea DOM-puusta elementtejä käyttämällä find() –metodia. Metodien funktioon pitää määritellä elementti, jonka sisältä find() –metodi hakee siihen määritellyjä elementtejä. Kuvion 13 esimerkissä on lähtö elementiksi määritelty ”body” –elementti. Kommentoidut kohdat kuvion 13 esimerkissä ovat find() –metodin palauttavat objektit, jotka se on löytänyt kyseisillä arvoilla. Find() –metodin parametriksi määritellään CSS-syntaksin mukainen elementtien haku.

```

1 $('body').find('span'); // [span]
2 $('body').find('div'); // [div.one, div.two]
3 $('body').find('div.one'); // [div.one]

```

KUVIO 13. jQueryn find() –metodin käyttö ja sen palautusarvot

3.4 Kommunikointi serverin kanssa

jQueryssä kommunikointi serverin kanssa tapahtuu Ajaxilla. Ajax eli Asynchronous JavaScript and XML on tapa välittää ja vastaanottaa tietoa serveriltä päivittämättä koko sivua uudestaan. jQuery sisältää erityisen hyvät työkalut Ajaxin käyttöön.

Ajax toimii siten, että se lähettää joko POST tai GET menetelmällä tietoa halutulle sivulle. Tässä voi siten myös hyödyntää esim. PHP:tä ja sen `$_POST` ja `$_GET` –funktioita tiedon vastaanottamiseen ja prosessoimiseen serverin päässä. PHP:n käyttö mahdollistaa myös tietokannan kuten MySQL:n käytön dynaamisissa sovelluksissa.

jQueryssä on useita tapoja käyttää Ajax-kutsuja. Näiden kutsujen välinen eroavaisuus voi aluksi olla hankala hahmottaa. Yleisimmät tavat käyttää Ajaxia ovat jQueryn metodit `load()`, `post()` ja `get()`. Näistä metodeista `post()` ja `get()` toimivat täysin samalla tavalla, poikkeuksena on vain menetelmä, jolla tieto lähetetään eteenpäin. `Post()` –metodi välittää tiedon POST menetelmällä kun taas `get()` –metodi käyttää GET menetelmää. (Yishi 2009.)

Kuviossa 14 on esimerkki `load()` –metodin käytöstä. Esimerkissä metodi lataa aina halutun sivun vastauksen suoraan elementtiin, jonka id on ”result”. `Load()` –metodin ensimmäinen parametri on sivusto, jonka tiedot halutaan hakea, tässä tapauksessa `data.php`. Ensimmäinen rivi lataa siis kaiken sen tiedon, mitä `data.php` tiedosto vain palauttaa vakiona. Toisella rivillä on käytetty `load()` –metodia välittämään `data.php` tiedostoon GET menetelmällä parametri ”id”, jonka arvo on ”1”. Toinen rivi siis lataa `data.php` tiedoston palauttaman tiedon, joka sille lähetetyllä parametrilla tulee. Kuvion kolmannella rivillä välitetään jälleen parametri ”id”, jonka arvo on ”1”, mutta tällä kertaa se välitetään POST menetelmällä. Neljännellä rivillä välitetään `data.php` tiedostoon sama tieto kuin kolmannella rivilläkin, mutta tässä käytetään vielä `load()` –metodin takaisinkutsufunktiota. Takaisinkutsufunktio jää odottamaan Ajax-kutsun onnistumista ja sen onnistuttua suorittaa itsensä. Tässä tapauksessa heti kutsun onnistuttua näytetään vastaanotettu tieto JavaScriptin `alert` –ikkunassa.

```

1 $('#result').load('data.php');
2 $('#result').load('data.php?id=1');
3 $('#result').load('data.php', { id: 1 });
4 $('#result').load('data.php', { id: 1 }, function(data){
5   alert('Response data:\n'+ data);
6 });

```

KUVIO 14. jQueryn `load()` –metodin käyttö

Kuten jo aiemmin mainittu jQueryn `post()` ja `get()` –metodit ovat toiminnaltaan täysin samat, niissä tiedon välitykseen käytetään vain eri menetelmää. Kuviossa 15 olevat esimerkit toimivat siis molemmilla metodeilla samalla tavalla.

Parametrit, jotka `post()` ja `get()` –metodit ottavat sisäänsä ovat täysin samat kuin `load()` –metodissa. Ensimmäinen parametri määrittelee tiedoston, toiseen tulee objektina lähetettävä tieto ja kolmanteen takaisinkutsufunktio. Eroavaisuus näiden kahden metodin ja `load()` –metodin välillä näkyy kuvion 15 kolmannesta viidenteen riviin. Jos `get()` –metodia halutaan käyttää `load()` –metodin tavoin on se tehtävä vasta `get()` –metodin takaisinkutsufunktiossa määrittämällä manuaalisesti, mihin saatu tieto laitetaan.

```

1 $.get('data.php');
2 $.get('data.php', { name: 'john' });
3 $.get('data.php', { name: 'john' }, function(data){
4   $('#result').html(data);
5 });

```

KUVIO 15. jQueryn `get()` –metodin käyttö

jQueryssä on myös mahdollista käyttää sen alemman tason `ajax()` –metodia, joka mahdollistaa tarvittaessa `ajax`-kutsun purkamisen pienempiin osiin. Tämän metodin käyttö on hieman monimutkaisempaa verrattuna esimerkiksi `load()` ja `get()` –metodeihin. `Ajax()` –metodi ottaa sisäänsä yhden parametrin, joka on aina objekti niistä arvoista, joilla `ajax`-kutsu halutaan suorittaa.

Kuviossa 16 on esimerkki `ajax()` –metodin käytöstä ja sen rankenteesta. Metodiin voi määritellä paljon erilaisia arvoja, mutta tässä esimerkissä sitä käytetään samalla lailla kuin kuvion 14 neljännellä rivillä olevassa esimerkissä. Aluksi `ajax()` –metodiin määritellään menetelmä, jolla tieto lähetetään eteenpäin (rivi 2). Sen jälkeen määritellään tiedosto, johon tieto lähetetään (rivi 3) ja kolmanneksi määritellään lähetettävä tieto (rivi 4). Kuvion viidennellä rivillä on takaisinkutsufunktio, joka tässä esimerkissä ajetaan vain, jos `ajax`-kutsu on onnistunut. Takaisinkutsufunktio näyttää JavaScriptin `alert` –ikkunassa vastaanotetun tiedon.

```
1 $.ajax({
2   type: "POST",
3   url: "data.php",
4   data: "name=John",
5   success: function(data){
6     alert('Response data:\n'+ data);
7   }
8 });
```

KUVIO 16. jQueryn ajax() –metodi

3.5 Graafikolle

Normaalisti graafikoiden tehtävänkuvaan ei ole kuulunut millään lailla JavaScriptin tai minkään muun ohjelmointikielen käyttö. Nykypäivänä asiat alkavat tosin olemaan toisin kun sivustojen ulkoasut monimutkaistuvat eikä niiden teko ei enää onnistu pelkästään HTML:n ja CSS:n avulla. Vaikka monet graafikot vierastavat ajatusta jonkin ohjelmointikielen käytöstä, on jQuery tehnyt JavaScriptin käytöstä melko helppoa ja yksinkertaista. jQueryn CSS-tyylinen elementtien valinta ja yksinkertainen syntaksi sekä metodien nimet tekevät sen hyväksi vaihtoehdoksi graafikolle. (Lewis 2010.)

jQueryn dokumentaatio on muihin JavaScript-kirjastoihin verrattuna selkeä. Sitä ei ole rakennettu pelkästään kirjaston funktioiden ja metodien ympärille. Sieltä löytyy selkokielellä ja hyvin järjestetysti kaikki tarvittavat asiat, joita jQueryn käyttöön tarvitsee. Hyvänä esimerkkinä on ”usein kysytyt kysymykset”, jonka alta löytyy suoraan selkeällä englannin kielellä selitettynä erilaisia jQueryyn liittyviä asioita. (Lewis 2010.)

4 LIIKUNTAPÄIVÄKIRJA

4.1 Yleistä

Liikuntapäiväkirjan käyttö alkaa palveluun rekisteröitymällä. Sivustolle pystyy rekisteröitymään kuka vain. Rekisteröityessä käyttäjän tarvitsee syöttää erinäisiä tietoja itsestään, jotka sitten tallennetaan järjestelmään. Rekisteröitymisen jälkeen käyttäjä voi kirjautua sisään palveluun ja ilmoittautua liikuntakurssilleen. Tässä vaiheessa käyttäjä voi käyttää vain oman kurssinsa keskustelualuetta sekä linkit-osiota. Päiväkirja-osio ei aukea ennen kuin opettajat ovat muodostaneet kuntoilijoista ja valmentajista parit.

Parien muodostuksen jälkeen valmentajien päiväkirja-osioon ilmestyy lista heidän pareistaan. Parin valitsemalla pääsee kyseisen kuntoilijan viikoittaiseen päiväkirjanäkymään. Kuntoilijoilla päiväkirja-osioon taas ilmestyy suoraan oma viikoittainen päiväkirjanäkymä. Valmentajat tulevat laatimaan viikoittaiset kuntoilu-aikataulut kuntoilijoilleen päiväkirja-osiossa ja kuntoilijat taas merkkäavat ja antavat palautetta tehdyistä suorituksistaan päiväkirjaansa.

4.2 Suunnittelu

Ulkoasun toteutukseen käytettiin HTML- ja CSS -tekniikoita. Sivuston elementit ja rakenne luodaan HTML:lla ja sen jälkeen käyttäen CSS:ää ne asetellaan haluttuihin paikkoihin ja muotoillaan halutun näköiseksi. Tämän osuuden projektista hoitivat pääsääntöisesti visualistit.

Projektin tietokannaksi valittiin MySQL, joka on alustariippumaton tietokantaratkaisu ja myös yleisesti käytetty tietokanta web-sovelluksissa. MySQL tekee tiedon tallentamisesta ja myöhemmin sen hakemisesta nopeaa ja helppoa. Tarvittavat tiedot käyttäjistä ja heidän tekemisistään tällä sivustolla tallennetaan MySQL-tietokantaan käyttäen PHP:tä.

Eri käyttäjäryhmät ja niiden omat näkymät eri osioissa sivustolla toi myös haastetta ja mielenkiintoa projektiin. Tätä ei voitu toteuttaa pelkän JavaScriptin avulla turvallisuussyistä. Käyttäjät pystyvät pääsemään käsiksi kaikkeen siihen, mitä JavaScript tekee, ja täten he voisivat myös asettaa itsellensä eri käyttäjäryhmän. Käyttäjäryhmien asetuksen ja tarkistuksen toteutukseen käytettiin PHP:tä ja sen session-muuttujaa, joka tallentuu pelkästään palvelimelle. Käyttäjät eivät siis itse pääse millään tapaa käsiksi session-muuttujiin.

Dynaamisella sivustolla tarkoitetaan sivustoa, jossa navigoidessa ei ladata koko sivua joka kerta uusiksi, vaan pelkästään tarvittava sisältö ladataan. Tämä säästää serverin tehoa sekä kaistaa ja tekee sivuston toiminnasta nopeampaa ja sulavampaa. Dynaamisuuden luomiseksi tarvitaan avuksi JavaScriptin Ajaxia. Projektiin päätettiin kuitenkin ottaa käyttöön JavaScriptille tehty jQuery-kirjasto, joka helpottaa ja nopeuttaa JavaScriptin ja etenkin Ajaxin käyttöä huomattavasti. jQuery tuo myös mukanaan hienoja ja helposti luotavia visuaalisia efektejä sivustolle.

4.3 Dynaamisuus

Tarvittavan sisällön lataaminen sivuston elementteihin tapahtuu pääsääntöisesti jQueryn Ajax-metodeilla. Ajax kutsuu tarvittavaa PHP-tiedostoa ja asettaa sen palautusarvona saadun sisällön haluttuun elementtiin. Koska käyttäjäryhmän tarkastus ja oikean sisällön näyttämisen valinta tapahtuu suoraan PHP:ssä, niin ei siitä tarvitse huolehtia missään vaiheessa JavaScript-koodia. Tästä syystä JavaScript-koodi tulee aina olemaa täysin sama kaikille käyttäjille, ainut ero eri käyttäjäryhmien välillä on Ajax-kutsujen saamat palautusarvot.

Kuviossa 17 näkyy sivuston ulkoasu ja rakenne heti sisäänkirjautumisen jälkeen. Dynaamisuus tällä sivustolla on toteutettu siten, että sivuston kaikkien eri osioiden – kurssit, päiväkirja, keskustelut, linkit ja käyttäjät – rakenne ladataan aina HTML-elementtiin, jonka id attribuutti on ”area”. Tämä elementti on kuviossa 17 näkyvä punainen juoksurata-alue.



KUVIO 17. Sivuston rakenne ja ulkoasu

Joka kerta osiota vaihdettaessa ladataan punaiselle alueelle uuden osion rakenne. Kuviossa 18 on esimerkki käyttäjät-osioon siirryttäessä sen rakenteen lataamisesta sivustolle dynaamisesti. Kuvion ensimmäisellä rivillä haetaan Ajaxilla users.php tiedoston sisältö. Tämä tiedosto sisältää pelkästään HTML-rakenteen käyttäjät-osioon. Ajax-kutsun onnistuttua ajetaan takaisinkutsufunktiossa määritely loadUsers() –funktio, joka taas lataa kaiken lopun tarvittun sisällön käyttäjät-osioon. Tällaisen menetelmän käyttö mahdollisti sen, että visualistit pystyivät tekemään osioiden ulkoasut valmiiksi, eikä koodarien tarvinnut koskea niihin. Koodarien tehtävä oli vain hakea ja ladata tarvittu tieto haluttuihin elementteihin eri osioissa.

```

1 $('#area').load('views/users.php', function(){
2   loadUsers();
3 });

```

KUVIO 18. Sisällön lataaminen dynaamisesti jQueryllä

Sivustolla on myös käytetty dynaamisia ikkunoita, jotka aukeavat suoraan sivuston päälle. Käytännössä nämä ”ikkunat” ovat aivan tavallisia div-elementtejä, jotka asetetaan näkymään kaikkien muiden elementtien päälle käyttäen CSS:n z-index attribuuttia. Sisältö elementteihin ladataan myös käyttäen Ajax-tekniikkaa.

Kuviossa 19 on esimerkki dynaamisen elementin kutsumiseen tehdystä funktiosta. Tämä funktio näyttää elementin käyttäen jQuery:n fade-efektiä. Elementti keskitetään aina selain-ikkunaan käyttäen JavaScriptiä, joka hakee selain-ikkunan sekä näytettävän elementin koot ja laskee elementin keskityksen. Kuvion 19

toisella rivillä määritellään tiedosto, jonka sisältö elementtiin halutaan ladata. Kolmannella rivillä määritellään kaksiulotteiseen taulukkoon tarvittaessa POST – menetelmällä tiedostoon lähetettävä tieto. Kuvion neljännellä ja viidennellä rivillä määritellään elementin korkeus ja pituus, ja kuudennella rivillä on tarvittaessa takaisinkutsufunktio.

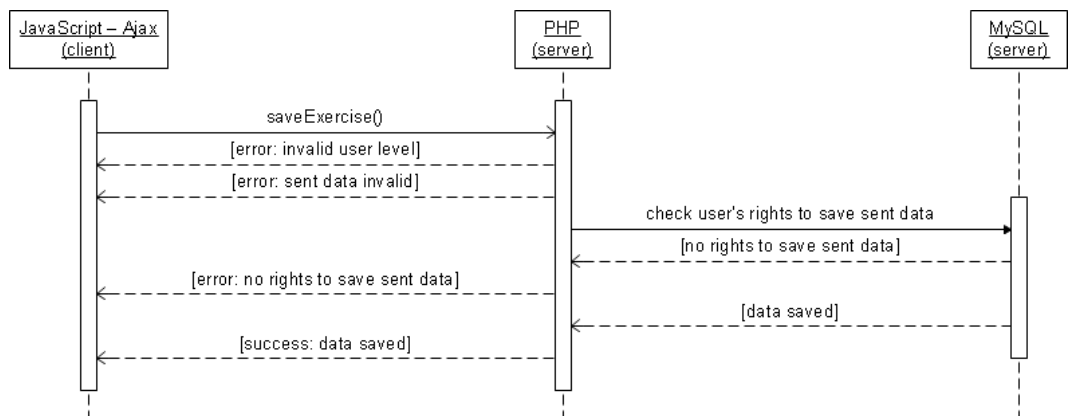
```

1 overlapBoxToggle(
2   'lib/diaryGetExercises.php',
3   [['date',date],['mod',mod]],
4   '520px',
5   '800px',
6   function(){
7     // callback
8   }
9 );

```

KUVIO 19. Dynaamisen elementin näyttämiseen käytetty funktio

Sivuston dynaamisuuden vuoksi tiedon tallennus tapahtuu Ajaxilla, joka välittää tallennettavat tiedot suoraan PHP-tiedostoon käyttäen POST menetelmää. PHP-tiedosta taas avaa yhteyden MySQL-tietokantaan ja tallentaa välitetyt tiedot. Koska tiedon tallennus tapahtuu Ajaxilla, joka on täysin JavaScript-pohjainen menetelmä, on käyttäjien mahdollista muuttaa PHP-tiedostoon välitettävää tietoa. Tämä on kuitenkin otettu suunnittelussa huomioon. PHP tarkistaa ennen tietokantayhteyden avaamista onko sisäänkirjautuneella käyttäjällä oikeasti oikeus tallentaa tietoa, jonka se on välittänyt. PHP tarkistaa ensin session-muuttujasta onko käyttäjäryhmä oikea sekä jossain tapauksissa relaatiohauilla tietokannasta esim. sen, onko kyseinen käyttäjä kyseisellä kurssilla.



KUVIO 20. Sekvenssikaavio tiedon tallennuksesta

Kuviossa 20 on sekvenssikaavio työssä käytetystä menetelmästä tallentaa käyttäjän luoma kuntoilusuoritus tietokantaan. Aluksi käyttäjä täyttää selaimessa annetut lomakkeen kentät, minkä jälkeen koko lomake lähetetään PHP-tiedostoon JavaScriptin avulla käyttäen Ajax-toimintoa. PHP-tiedostossa tarkastetaan onko käyttäjätaso oikea, jos se ei ole suoritus loppuu saman tien ja JavaScript saa paluuarvona nollan ja näyttää virheilmoituksen käyttäjälle. PHP-tiedosto tarkistaa myös onko lähetetty tieto oikeanlaista ja palauttaa nollan JavaScriptille, jos ei ole. Kun tieto on tarkistettu ja oikean laista, niin tarkistetaan vielä tietokannasta, että käyttäjä kuuluu kyseiseen pariin sekä kurssiin. Jos jompikumpi aiemmista ei täsmää niin PHP lopettaa taas suorituksen ja ilmoittaa JavaScriptille paluuarvona virheen. Kun kaikki tieto täsmää ja tallennus tietokantaan on onnistunut, niin PHP palauttaa JavaScriptille paluuarvona numeron yksi, joka näytetään käyttäjälle onnistuneena tallennuksena.

4.4 Sivusto

4.4.1 Käyttäjryhmät

Sivustolla on siis kolme eri käyttäjäryhmää, opettajat, valmentajat ja kuntoilijat, joista kaikki näkevät sivuston hieman eri tavalla. Opettajat näkevät sivustolla kaiken sisällön. He pystyvät myös tarvittaessa muokkaamaan ja poistamaan valmentajien ja kuntoilijoiden tietoja, keskustelualueen viestejä sekä urheilusuorituksia. Valmentajilla ja kuntoilijoilla näkymä on taas rajoitettu niihin asioihin, joita heidän on tarkoitus pystyä sivustolla tekemään.

Tällaisen ratkaisun toteuttamiseksi tässä projektissa käytettiin PHP:n session-muuttujaa. Idea on, että PHP hakee tietokannasta sisäänkirjautuvan käyttäjän käyttäjäryhmän – opettaja, valmentaja tai kuntoilija – ja tallentaa sen numeraalisena arvona PHP:n session-muuttujaan. Näin käyttäjät eivät itse voi millään tavalla päästä väliin muuttamaan heidän omaa käyttäjäryhmäänsä. Kun PHP tietää mihin käyttäjäryhmään kyseinen käyttäjä kuuluu, niin se osaa näyttää vain ne asiat, jotka käyttäjän kuuluu nähdä.

Kuviossa 21 on esimerkki siitä, miten voidaan näyttää tietty sisältö vain opettajille. Opettajien numeerinen arvo käyttäjäryhmänä on yksi. Kuvion toisella rivillä oleva session-muuttuja asetetaan jo sisäänkirjautumisen yhteydessä ja tässä esimerkissä toisella rivillä tarkistetaan onko muuttujan arvo vähemmän tai yhtä paljon kuin yksi. Sisältö, joka opettajille haluttaisiin näyttää, pitäisi asettaa riveillä kaksi ja neljä olevien aaltosulkujen väliin. Kyseinen sisältö näkyisi siis vain, jos session-muuttujaan tallennettu arvo on vähemmän kuin kaksi, muussa tapauksessa kuvion 19 koodi palauttaisi tyhjän sivun.

```

1 <?php
2   if($_SESSION['userLevel'] <= 1){
3     // content here
4   }
5 ?>

```

KUVIO 21. PHP ja käyttäjäryhmän session-muuttuja

Sivustolle tehtiin myös admin-käyttäjä, jonka käyttäjäryhmä on nolla. Admin-käyttäjä pystyy tekemään sivustolla samat asiat kuin opettajatkin. Poikkeuksena se, että admin näkee kaikkien kurssien kaiken sisällön suoraan, kun taas opettajien pitää ensin liittyä kurssille nähdäkseen ja muokatakseen sen sisältöä.

Valmentajan ja kuntoilijan käyttäjän pystyy tekemään sivustolle suoraan rekisteröitymisvaiheessa. Rekisteröitymislomakkeessa kysytään käyttäjältä, onko hän tulossa käyttämään sivustoa valmentajan vai kuntoilijan roolista. Opettajien tunnuksia pystyy luomaan vain toiset opettajat tai admin. Opettajan tunnuksen luonti tapahtuu sivuston käyttäjät-osiossa, josta löytyy linkki tunnuksen luontiin. Tunnuksen luonnin yhteydessä pyydetään syöttämään luotavan opettajan oikea nimi, sähköpostiosoite, käyttäjätunnus sekä salasana. Tunnus on heti luonnin jälkeen käyttövalmis.

4.4.2 Rekisteröityminen ja sisäänkirjautuminen

Rekisteröitymis- ja sisäänkirjautumissivu on ensimmäinen asia, jonka käyttäjät näkevät sivustolle tullessa. Kuvioista 22 näkee sisäänkirjautumissivun ulkoasun. Tämä sivu on ulkoasultaan täysin erilainen kuin sivuston muut osat. Kuten

kuvioista 22 huomaa, vasemmalla puolella on rekisteröitymislomake ja oikealla puolella itse sisäänkirjautuminen.

KUVIO 22. Rekisteröityminen ja sisäänkirjautuminen sivustolle

Rekisteröitymisen yhteydessä käyttäjältä kysytään seuraavat asiat: nimi, sähköpostiosoite, syntymäpäivä, paino, käyttäjätunnus sekä salasana kahdesti. Rekisteröitymislomakkeessa on käytetty jQueryä tarkistamaan käyttäjän kenttiin syöttämät tiedot regular expressionia apuna käyttäen. Sähköpostiosoitteesta tarkistetaan, että siinä on @ -merkki sekä viimeisen pisteen yhteydessä oleva verkkotunnus on oikean pituinen, eli kahdesta neljään merkkiä. Syntymäpäivästä tarkistetaan että päivä ja kuukausi ovat numeraalisia ja yhden tai kahden merkin pituisia, sekä vuoden pituus neljä merkkiä. Käyttäjätunnuksessa tarkistetaan lomakkeessa vain sen pituus, joka on oltava vähintään neljä merkkiä. Käyttäjätunnuksen saatavuus tarkistetaan vasta lomakkeen lähetyksen jälkeen. Jos käyttäjätunnus on jo käytössä, käyttäjä saa siitä ilmoituksen ja voi sen jälkeen vaihtaa käyttäjätunnuksen ja kokeilla uusiksi. Salasanoissa tarkistetaan, että molemmat ovat samat, ja niiden pituuden on oltava vähintään neljä merkkiä.

Sisäänkirjautuminen tapahtuu normaalilla rutiinilla, eli käyttäjä syöttää käyttäjätunnuksen ja salasanan, minkä jälkeen lomakkeen tiedot lähetetään PHP-tiedostoon, joka taas avaa yhteyden MySQL-tietokantaan ja tarkistaa, onko kyseisen käyttäjän tunnus olemassa. Salasanat ovat tietokannassa MD5-salattuja, joten niitä ei selväkielisenä pysty mitenkään näkemään. Kirjautumisen yhteydessä

käyttäjän salasana muutetaan myös MD5-muotoiseksi ja verrataan 32-merkkistä hash-arvoa tietokannassa olevaan hash-arvoon. Jos käyttäjätunnus ja sen salasana oleva 32-merkkinen hash täsmäävät, niin käyttäjä siirretään liikuntapäiväkirjan päänäkömään, josta on esimerkki kuva kuviossa 17 sivulla 19.

4.4.3 Kurssit

Kurssit-osio on ensimmäinen paikka aloittaa sivuston käyttö. Jokaisen käyttäjän on oltava jollain kurssilla, jotta he näkevät sivustolla jotain muutakin kuin yleisen linkit osion. Kurssit-osiossa siis luodaan kurssit, ilmoitaututaan niihin sekä luodaan niihin liittyneistä käyttäjistä pareja. Luonnollisesti kurssien ja parien luonti onnistuu vain opettajilta, valmentajat ja kuntoilijat voivat vain liittyä kurseille sekä nähdä niiden infon tässä osiossa. Kuviossa 23 on kurssit-osion ulkoasu opettajan näkökulmasta.

Kurssit			
<i>Otto Opettaja</i>			
LISÄÄ KURSSI [1]			
Kurssin nimi	Käyttäjät		
Kurssi numero quatro	0	Ilmoitaudu	Info
Kurssi numero tres	5	Parit Käyttäjät	Info
Kurssi numero dos	4	Ilmoitaudu	Info
Kurssi numero uno	11	Parit Käyttäjät	Info

KUVIO 23. Kurssit-osion ulkoasu opettajan tunnuksella

Kurssin luonti on ensimmäinen asia, joka sivustoa käyttöönotettaessa täytyy tehdä. Se onnistuu painamalla ”lisää kurssi” -linkkiä, joka löytyy kuvion 23 oikeasta yläalaidasta. Linkkiä painamalla aukeaa kyseisen näkymän päälle uusi dynaaminen ikkuna, jossa kysytään kaikki kurssin luontiin tarvittavat tiedot.

Jokaisella kurssilla on oma kurssiavain, joka kurssin luonnin yhteydessä on kysytty. Kurssille ilmoittautuminen tapahtuu painamalla ”ilmoittaudu” -linkkiä, minkä jälkeen käyttäjää pyydetään syöttämään kurssiavain kyseiselle kurssille. Käyttäjän syötettyä oikean kurssiavaimen, on hän onnistuneesti ilmoittautunut kurssille. Tämän jälkeen hän näkee suoraan keskustelualue-osiossa kurssiin liittyvät keskustelut ja pystyy myös itse osallistumaan niihin. Käyttäjä ilmestyy myöskin opettajien näkemiin ”Parit” ja ”Käyttäjät” -linkkien alla oleviin listoihin.

Info-linkin alta löytyy kurssin tiedot sekä kuvaus kurssista. Kaikki käyttäjät näkevät kaikkien kurssien infot, vaikkeivät he olisi kurssille ilmoittautuneetkaan. Opettajat voivat vielä myöhemmin muokata kurssin luonnin yhteydessä kysytyjä tietoja info-osion alla.

Käyttäjät-linkin alta aukeaa lista kaikista kurssille liittyneistä käyttäjistä. Tässä osiossa opettajat voivat myös tarvittaessa poistaa käyttäjiä kursseilta yksinkertaisesti painamalla käyttäjän nimen kohdalla olevaa poista-linkkiä. Parit-linkin alta taas aukeaa lista kaikista kurssille luoduista pareista. Parien luonti onnistuu myös tässä osiossa. Parin luonti on hyvin yksinkertainen, kahdesta alavetovalikosta valitaan valmentaja ja kuntoilija, minkä jälkeen painetaan lisää-linkkiä. Tämä lisää parin ikkunassa olevaan listaan. Parien poisto tapahtuu täysin samalla lailla kuin käyttäjienkin: painetaan vain parin kohdalla olevaa ”poista” – linkkiä.

4.4.4 Päiväkirja

Päiväkirja-osio on tämän projektin tärkein asia. Päiväkirjaan valmentajat ja kuntoilijat täyttävät liikuntasuorituksiaan. Valmentajat asettavat viikoittaiset tavoitteet kuntoilijoille, ja kuntoilijat merkitsevät, mitä tekivät milloinkin. Opettajat taas voivat seurata kaikkien käyttäjien toimintaa ja arvioida oppilaita sen perusteella.

Kuntoilijoiden siirtyessä päiväkirjaan näkevät he saman tien kuvion 22 mukaisen näkymän. Valmentajat taas näkevät ensiksi listan kaikista pareistaan, joista heidän

tarvitsee valita yksi, jonka päiväkirjaan he haluavat päästä. Opettajilla taas aluksi näkyy lista kaikista kurssin pareista, joista heidänkin tarvitsee valita yksi päästäkseen näkemään päiväkirjan.

Kurssien ja parien vaihtaminen tapahtuu alavetovalikoista, jotka näkyvät kuvion 24 vasemmassa laidassa. Valikoiden alapuolella on ns. viikkonäkymä päiväkirjasta. Tälle sivulle tultaessa viikkonäkymässä näkyy aina meneillään oleva viikko. Päiviä painamalla aukeaa uusi ikkuna, jossa näkyy kyseisen päivän tavoitteet sekä tehdyt liikuntasuoritukset. Paksunnetut viikonpäivät tarkoittavat sitä, että joko valmentaja tai kuntoilija on merkannut kyseiselle päivälle jonkin liikuntasuorituksen. Tämä helpottaa kuntoilijoita, eikä heidän tarvitse käydä läpi jokaista päivää erikseen nähdäkseen, onko valmentaja merkannut niille mitään. Viikkojen välillä voi seilata painelemalla ”edellinen viikko” ja ”seuraava viikko”-linkkejä. Viikkonäkymän alapuolella on vielä laatikko, josta saa auki kyseisen viikon suorituksen yhteenvedon tai liikuntapiirakan.



KUVIO 24. Päiväkirja-osio

Kuviossa 25 näkyy, miltä yksittäisen päivän suoritukset näyttävät. Näkymän vasemmassa ylälaidassa on ”lisää harjoitus”-linkki, josta joko valmentaja tai kuntoilija voivat lisätä suorituksia kyseiselle päivälle. Taulukossa tummanvihreällä pohjalla olevalla rivillä on otsikot kuhunkin taulukon kohtaan. Vaaleampi sininen alue on aina valmentajan lisäämä, ja kuten tässä kuvassa

näky, niitä ei pysty muokkaamaan. Tämä johtuu siitä, että kuvan näkymä on kuntoilijan näkökulmasta otettu. Tummemmalla sinisellä pohjalla olevat asiat ovat aina kuntoilijan lisäämiä, niitä taas valmentajat eivät pysty muokkaamaan. Opettajat tosin pystyvät muokkaamaan sekä valmentajien että kuntoilijoiden lisäämiä kohtia tarvittaessa.

Liikuntasuoritusta lisätessä kuvion 25 näkymään ilmestyy kaksi uutta riviä, täysin samanlaisia kuin tummansininen alue. Käyttäjän tarvitsee syöttää kolme pakollista kohtaa – fyysisen kunnon osa-alueet, liikuntamuoto ja kesto – suorituksen lisätäkseen. Rasitustaso ja kommentti ovat vapaaehtoisia kenttiä, mutta silti suositeltavia täyttää. Kun kentät on täytetty painetaan ”luo” –linkkiä, joka on samanlainen kuin kuvion 25 ”tallenna” –linkki. Valmentajat ja kuntoilijat voivat poistaa vain omia suorituksiaan. Suorituksien poistaminen tapahtuu ”poista” –linkistä, joka näkyy ”tallenna” –linkin vasemmalla puolella. Kuvion 25 suoritus on valmentajan lisäämä, joten ”poista” –linkkiä ei kuvassa näy.

Kuvion 25 kolmessa alavetovalikossa on kussakin valmiiksi määriteltäviä arvoja. Fyysisen kunnon osa-alueet vaikuttavat kuntoilijan oman viikoittaisen liikuntapiirakan täyttämiseen. Fyysisen kunnon osa-alue määrittää, mitä kolmesta liikuntapiirakan lohkoista täytetään. Kesto taas määrittää sen, kuinka paljon kyseistä lohkoa täytetään.

LISÄÄ HARJOITUS **Tiistai 15.3.2011**

kestävyyskunto reippaasti = PPP = pitää pystyä puhumaan, 50-60% max syke
 kestävyyskunto rasittavasti = HH = hikoilee ja hengästyy, 60-70% max syke

Fyysisen kunnan osa-alueet	Liikuntamuoto	Rasitustaso ? (oma tuntemus)	Kesto (min)
Kestävyyskuntoa reippaasti	Sauvakävely		30
Voi kävellä kauemminkin!			
Kestävyyskuntoa reippaasti	Sauvakävely	Hieman rasittava	45
Oli niin hyvä keli, että tuli sauvakäveltyä 45 minuuttia!			

Tallenna

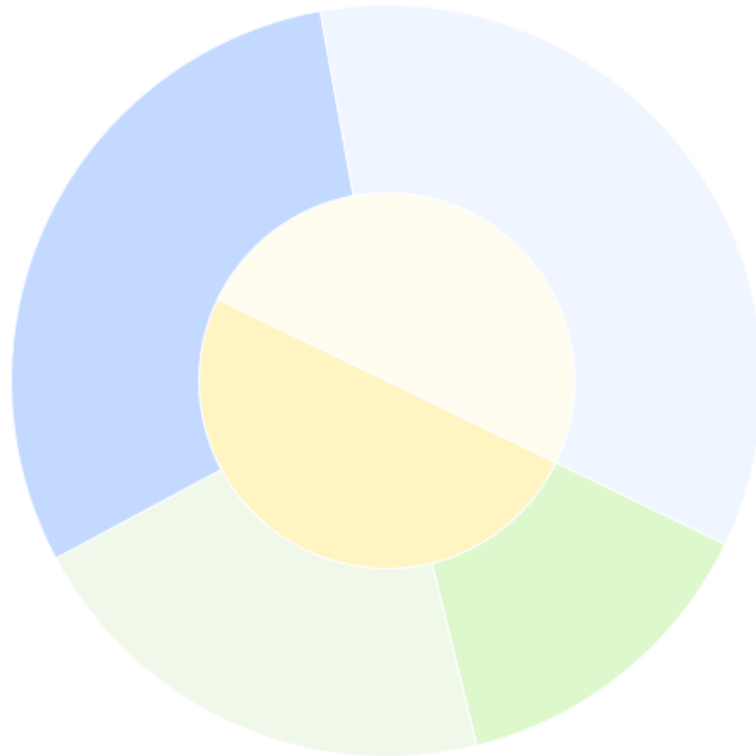
SULJE

KUVIO 25. Yksittäisen päivän tavoitteet ja suoritukset kuntoilijan näkökulmasta

Sivulla 26 kuvion 24 alimassa laatikossa olevasta yhteenvedosta saa listan kyseisen viikon valmentajan sekä kuntoilijan laatimista suorituksista. Tämä yhteenvedo on tehty tulostusta varten. Valmentajien on tulostettava viikoittain tekemänsä kuntoiluohjelmat sekä se, miten heidän kuntoilijansa ovat niitä noudattaneet. Yhteenvedo aukeaa selaimen uuteen ikkunaan, se on täysin mustavalkoinen ja sen asettelu on optimoitu tulostukseen.

Oma liikuntapiirakka, joka löytyy myöskin sivulta 26 kuvion 24 alimmasta laatikosta, näyttää kyseisen viikon suoritukset liikuntapiirakan tapaisesti. Kuviossa 26 on esimerkki viikoittaisesta liikuntapiirakasta. Sinisellä alueella on fyysisen kunnan osa-alue ”kestävyyskuntoa reippaasti”, vihreällä alueella taas on ”kestävyyskuntoa rasittavasti sekä keskellä on lihaskunto ja liikehallinta. Tummempi väri piirakassa tarkoittaa suoritettua osuutta viikoittaisesta suosituksesta, joka taas on vaaleampi alue. Liikuntapiirakan alla on vielä minuutteina ja prosentteina näytetty, kuinka paljon viikoittaista suositusta on suoritettu.

Liikuntapiirakka 14.03.2011 - 20.03.2011



Reippaasti: 70/150 min (47 %)

Rasittavasti: 30/75 min (40 %)

KUVIO 26. Viikoittainen oma liikuntapiirakka

Liikuntapiirakka on toteutettu Googlen Chart API:lla. Googlen Chart API mahdollistaa erilaisten kaavioiden ja diagrammien toteuttamisen annetuilla arvoilla. Arvot asetetaan suoraan Googlen URL-osoitteeseen, joka luo dynaamisesti niiden pohjalta saadun kuvion. Tätä URL-osoitetta voidaan käyttää suoraan HTML:n img-tagissa. Liikuntapiirakan saamiseksi tässä projektissa käytetään PHP:tä hakemaan viikoittaisten suoritusten kestot tietokannasta. Tämän jälkeen kestot erotellaan kolmeen eri liikuntapiirakan ryhmään ja lasketaan jokaisen ryhmän omat arvot yhteen. Yhteen lasketuista arvoista muodostetaan sitten URL-osoite, jolla saadaan liikuntapiirakan tapainen kuvio. Kuviossa 26 on esimerkki URL-osoitteesta, jolla kuvion 26 piirakka on piirretty.

```

1 http://chart.apis.google.com/chart
2 ?chf=bg,s,FFFFFF
3 &chs=600x500
4 &cht=pc
5 &chco=FFF4C2|FFFCF0|FFF4C2,DDF8CC|EFF8E9|C3D9FF|F0F6FF
6 &chd=t:1,1|14,21,30,35
7 &chp=0.45
8 &chtt=Liikuntapiirakka 14.03.2011 - 20.03.2011
9 &chts=676767,20

```

KUVIO 26. Liikuntapiirakan luomiseen käytetty Google Chart API:n URL-osoite

4.4.5 Keskustelut

Keskustelut-osiosta löytyy kurssikohtaiset keskustelalueet. Kuviossa 28 on esimerkki yhden kurssin sisällä olevista aiheista. Tämä osio näkyy ja toimii kaikilla käyttäjillä samalla lailla. Poikkeuksena tietenkin se, että kuntoilijat ja valmentajat voivat poistaa vain itse lisäämiään aiheita tai viestejä, kun taas opettajat voivat poistaa kenen tahansa aiheet ja viestit.



Kurssi numero uno				
Kurssit > Kurssi numero uno				Otto Opettaja
				LISÄÄ AIHE [1]
Aihe	Päivämäärä	Viestit	Viimeisin	
Jumpat	2.3.2011	#1	Maija Mallikas	Poista
Yleistä	18.1.2011	#3	Otto Opettaja	Poista
Tiedotteet	18.1.2011	#3	Maija Mallikas	Poista

KUVIO 28. Yhden kurssin aiheet kurssit-osiassa

Kurssin luonnin yhteydessä luodaan myös uusi keskustelalue kurssille. Keskustelualueeseen lisätään automaattisesti myös aihe ”Tiedotteet”, jonka sisään lisätään vielä yksi viesti kurssin luojaan tunnuksilla. Tämän viestin sisältö sisältää samat tiedot, kuin kurssin luonnin yhteydessä on annettu.

Aiheen lisääminen tapahtuu ”lisää aihe” –linkistä, joka sijaitsee kuviossa 28 oikeassa laidassa ylhäällä. Linkkiä painettaessa avautuu ikkuna, jossa kysytään aiheen nimi sekä viesti. Aiheeseen vastaaminen tapahtuu ”vastaa” –linkistä, joka vaihtuu ”lisää aihe” –linkin tilalle johonkin aiheeseen siirryttäessä.

Aiheiden ja viestien poistaminen tapahtuu ”poista” –linkistä, joka sijaitsee jokaisen aiheen tai viestin oikeassa reunassa. Viestien muokkaus tapahtuu ”muokkaa” –linkistä, joka on myös oikeassa laidassa. Muokatessa aukeaa uusi ikkuna, jossa on alkuperäisen viesti. Kun viesti on muokattu halutuksi painetaan vain ”tallenna” –linkkiä ja muutokset tallentuvat viestiin.

4.4.6 Linkit

Linkit-osiosta löytyy lista linkeistä, jotka muut käyttäjät ovat nähneet hyödyllisiksi. Kuviossa 29 on esimerkki linkit listasta. Listassa näkyy päivämäärä, jolloin linkki on listaan lisätty sekä linkin nimi. Kaikki sivuston käyttäjät voivat lisätä linkkejä tähän osioon, ja jälleen kerran valmentajat ja kuntoilijat voivat poistaa vain omia linkkejään. Opettajat taas voivat poistaa kaikkien käyttäjien linkkejä.



Linkit	
	<i>Lisää</i>
Pvm	Linkin nimi
1.3.2011	Energiankulutuslaskureita
1.3.2011	Näin testaat kuntosi

KUVIO 29. Linkit-osio

Linkin lisäys tapahtuu ”lisää” –linkistä. Tämä aukaisee uuden ikkunan, jossa kysytään linkin nimi ja URL-osoite. Linkki lisätään painamalla ”tallenna” –linkkiä, jonka jälkeen se näkyy heti kaikkien käyttäjien litkit listassa. Linkin poisto taas tapahtuu ”poista” –linkkiä painamalla, joka näkyisi rivin oikeassa laidassa.

4.4.7 Käyttäjät

Käyttäjät-osiossa näkyy lista kaikista käyttäjistä, jonka kanssa valmentaja tai kuntoilija on samalla kurssilla. Opettajat taas näkevät kaikki järjestelmään rekisteröityneet käyttäjät tässä listassa. Kuviossa 30 on esimerkki opettajan käyttäjät-osion näkymästä. Valmentajat ja kuntoilijat eivät näe ollenkaan ”lisää opettaja” –linkkiä eivätkä ”tiedot” ja ”sulje” –linkkejä.

Käyttäjät		Lisää opettaja	
Etsi käyttäjää		Etsi	
		[1] [2]	
Nimi	Rooli	Tiedot	Sulje
Maija Mallikas	Valmentaja	Tiedot	Sulje
Maija Meikäläinen	Kuntoilija	Tiedot	Sulje
Matti Mallikas	Kuntoilija	Tiedot	Sulje
Matti Meikäläinen	Kuntoilija	Tiedot	Sulje
Otto Opettaja	Opettaja	Tiedot	Sulje

KUVIO 30. Käyttäjät-osio

Listassa näkyy maksimissaan 10 käyttäjää yhtä aikaa, jos järjestelmässä käyttäjiä on enemmän kuin 10, niin ne ovat sitten omilla sivuillaan. Sivuja voi vaihtaa hakasuluissa olevista numeroista taulukon oikeassa ylälaidassa. Käyttäjät listataan aina aakkosjärjestyksessä. Etsi käyttäjää –toiminnolla voidaan rajata näytettävää käyttäjien määrää itse syöttämällä arvolla. Tämä toiminto etsii syötettyä arvoa käyttäjien sekä etu- että sukunimestä.

Opettajan lisäys järjestelmään tapahtuu käyttäjät-osiossa painamalla ”lisää opettaja” –linkkiä. Tämä aukaisee uuden ikkunan, jossa kysytään luotavan opettajan tiedot. Opettajista ei tarvitse tietää kuin nimi, sähköpostiosoite, käyttäjätunnus sekä salasana. Nämä syötettyään pitää painaa ”lisää” –linkkiä, joka luo uuden opettajakäyttäjän. Käyttäjätunnus on heti valmis käytettäväksi.

Järjestelmässä on myös mahdollisuus sulkea jokin käyttäjätunnus. Sulkemisella tarkoitetaan käyttäjätunnuksen ns. pois päältä kytkemistä, eli kyseisellä käyttäjätunnuksella ei enää pysty sen jälkeen kirjautumaan sisään. Tunnuksen sulkeminen ei poista kyseisen käyttäjän jo tekemiä asioita järjestelmään. Tunnuksen voi myös jälkeä päin vielä kytkeä päälle asettamalla sille uusi salasana ”tiedot” –linkin alta. Suljetut käyttäjät näkyvät aina käyttäjälistalla punaisella taustalla.

Opettajilla näkyy ”tiedot” –linkki, jonka alta näkee kyseisen käyttäjän kaikki henkilökohtaiset tiedot – sähköpostiosoite, syntymäpäivä, paino, opiskelijanumero, rooli sekä käyttäjätunnus. Tarvittaessa opettajat pystyvät myös vaihtamaan jonkin muun käyttäjän salasanan painamalla ”muuta salasana” –linkkiä. Salasanaa vaihdettaessa ei tarvitse tietää käyttäjän aiempaa salasanaa, jos käyttäjä on oikeasti unohtanut salasanansa.

4.4.8 Asetukset

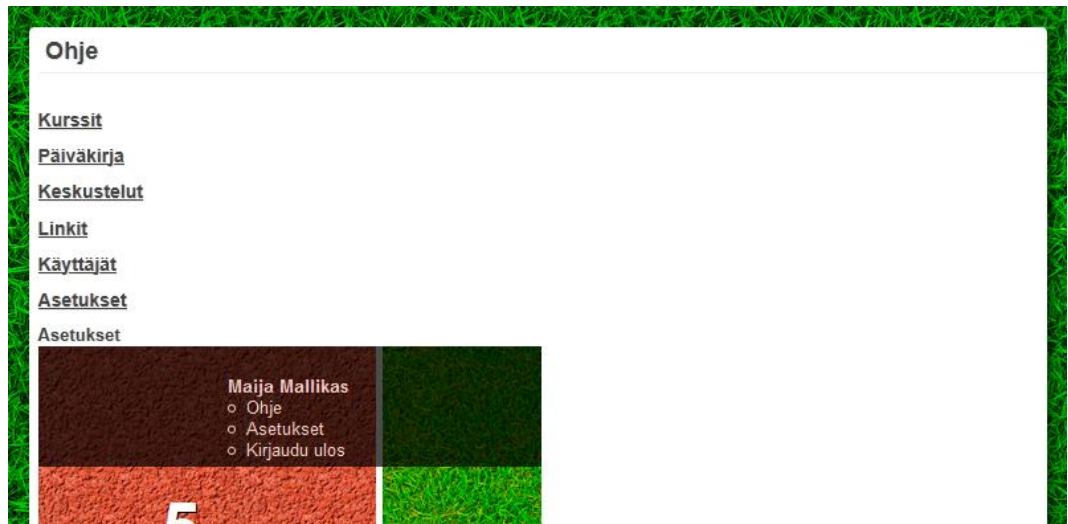
Asetukset-ikkunaan pääsee aina sivuston oikeasta ylälaidasta löytyvästä ”asetukset” –linkistä. Tässä ikkunassa pääsee katsomaan omia tietojaan sekä tarvittaessa muuttamaan niitä. Myös salasanan vaihto onnistuu tätä kautta. Omissa tiedoissa näkyy kaikki samat tiedot, jotka rekisteröitymisvaiheessa käyttäjältä kysytään. Nämä ovat: nimi, sähköpostiosoite, syntymäpäivä, paino, opiskelijanumero, rooli sekä käyttäjänimi. Käyttäjä ei tosin voi enää tässä vaiheessa vaihtaa käyttäjänimeä eikä roolia.

Omien tietojen muuttaminen tapahtuu painamalla ”muuta tietoja” –linkkiä, joka muuttaa kentät kirjoituskelpoisiksi. Kun tiedot on muutettu painetaan ”tallenna” –linkkiä, joka taas tallentaa tiedot ja tekee kentät pelkiksi tekstikentiksi. Oman salasanan vaihto onnistuu myös tästä osiosta. Se tapahtuu ”muuta salasana” –linkkiä painamalla, jonka jälkeen käyttäjältä kysytään vanha salasana sekä pyydetään syöttämään uusi salasana kahdesti. Kun salasanat on syötetty tallennetaan se painamalla ”tallenna salasana” –linkkiä.

4.4.9 Sivuston ohje

Sivuston ohjeeseen pääsee käsiksi myös aina sivuston oikeasta ylälaidasta löytyvästä ”ohje” –linkistä. Ohje on täysin erillinen osa sivustoa. Linkkiä painamalla aukeaa uusi selaimen ikkuna, jossa ohje näytetään. Ohje on tehty näyttämään käyttäjille vain ne asiat, joita he voivat sivustolla tehdä. Käytännössä siitä on siis kolme eri versiota.

Kuviossa 31 on näkyvissä ohjesivun ulkoasu. Se on tehty samalle pohjalle kuin sisäänkirjautumissivu. Ohjeeseen on lajiteltu kaikki sivuston eri osat linkeiksi, ja jokaista linkkiä painamalla aukeaa kyseisen osan ohje dynaamisesti linkin alle käyttäen jQueryn slideToggle() –metodia. Sisällön näytön erottelu tapahtuu käyttäen PHP:tä, joka tarkistaa käyttäjän roolin ja tulostaa vain ne osat, jotka kukin on sallittu näkemään.



KUVIO 31. Sivuston käyttöön tehty ohje

4.5 Yhteensopivuus ja testaus

Sivuston toimintaa on testattu kaikilla yleisimmillä selaimilla. Näihin kuuluvat mm. Firefox, Internet Explorer, Opera, Chrome sekä Safari. Firefox, Opera, Chrome ja Safari selaimista kaikista oli käytössä jossain vaiheessa vuotta 2010 tullut versio. Internet Explorerista taas oli käytössä versiot 7, 8 ja 9.

Sivusto toimi hyvin ja sulavasti Firefox, Chrome, Safari ja Internet Explorer 9 selaimilla. Nämä selaimet kaikki tukevat viimeisimpiä CSS:n ominaisuuksia, joita sivustolla on käytetty, ja täten sivuston ulkoasu oli myös täysin samanlainen kaikilla näistä selaimista.

Internet Explorer 8 ja Opera selaimet näyttävät sivun samalla lailla. Kumpikaan näistä selaimista ei tue CSS:n elementtien pyöristämistä eikä varjoja. Myöskin hieman hitaammalla tietokoneella huomaa, että Ajaxilla elementteihin ladattava tieto tulee välillä pienellä viiveellä.

Sivustoa on myös testattu kerran Internet Explorer 7 –versiolla. Sivusto kuitenkin näytti toimivan lähestulkoon samalla tapaa kuin Internet Explorer 8 –versiollakin. Ainoa asia, joka huomattiin, oli päiväkirjaosiossa oleva viikoittainen suoritusten yhteenvedon tulostus. Internet Explorer 7 –versiolla pienemmällä resoluutiolla

yhteenvetoa katsoessa menivät tekstien rivitykset täysin sekaisin. Asia kuitenkin korjaantui määrittämällä jokaisen rivin elementtiin tarpeeksi suuri leveys.

Sivustolla olevien eri käyttäjäryhmien takia Ajaxilla PHP-tiedostoihin lähetettävät pyynnöt ovat eri käyttäjillä erilaisia. Näiden pyyntöjen ja palautusarvojen erottelu tuotti paljon päänvaivaa ja sitäkin enemmän ongelmia. Kaikki pyynnöt piti suunnitella siten, että käyttäjät eivät pysty saamaan sellaista tietoa käsiinsä, mitä heidän ei pitäisi pystyä näkemään. Tietojen tallennuksessa on myös tehtävä joka kerta tarkistus, voiko kyseinen käyttäjä tallentaa kyseistä tietoa.

Kaikki ne PHP-tiedostot, joissa on jotain rajoituksia tai erilaisia palautusarvoja eri käyttäjille, on testattu. Testaus on tapahtunut kirjautumalla sivustolle eri käyttäjillä ja katsottu, saako kyseinen käyttäjä jonkin toisen käyttäjän tietoja mitenkään esille tai voiko se tallentaa tietoa, jota sen ei pitäisi pystyä tallentamaan.

5 YHTEENVETO

Liikuntapäiväkirja saatiin toteutettua asiakkaan vaatimuksia vastaavaksi, ja kaikki asiakasvaatimuksessa määritellyt asiat löytyvät sivustolta. Asiakas oli myös tyytyväinen sivuston lopputulokseen ja toimivuuteen. Sivustolle oli tarkoitus tehdä myös joitain lisäominaisuuksia, jotka tosin ajan puutteen vuoksi jäivät testaamatta tai hieman keskeneräisiksi eikä sen takia ole tällä hetkellä käytössä.

Sivustoa käyttää tällä hetkellä testiryhmä opiskelijoita, jotka suorittavat sivuston tarkoitusta vastaavaa liikuntakurssia etänä. Käyttäjäkokemukset ovat täten vielä aika vähäisiä tässä vaiheessa. Testiryhmän kokemusten perusteella sitten päätetään, aiotaanko projektia lähteä hankkeistamaan ja kehittämään eteenpäin.

Sivuston ulkoasussa olisi hieman paranneltavaa. Esimerkiksi kaikki teksti-linkit, jotka toimivat sivustolla nappeina voitaisiin korvata hieman huomattavammalla ratkaisulla. Sivuston värimaailmaa voisi myös katsoa läpi ja tehdä hieman yhdenmukaisemmaksi, ja muutenkin sivuston ulkoasusta voisi tehdä vielä selekämmän.

Näin jälkeinpäin ajatellen monia asioita olisi voinut projektissa tehdä toisin. Tosin projektin alussa ei tullut ajatelleeksi niin laajasti koko sivustoa ja sen toimintaa. Joitain asioita koodissa olisi voinut vielä yhdistää pienemmiksi kokonaisuuksiksi ja käyttää muissakin osioissa. Opin kuitenkin projektista taas paljon uutta jQuery:n ja PHP:n osalta ja niiden yhteiskäytöstä. Myöskin MySQL:n saralta tuli opittua taas muutamia uusia kikkoja.

LÄHTEET

Jdsharp 24.1.2008. jQuery parent() vs. parents() [viitattu 22.3.2011]. Saatavissa:

<http://jqueryminute.com/jquery-parent-vs-parents>

Lewis E. 19.7.2010. jQuery, A Designer's Perspective [viitattu 25.3.2011].

Saatavissa:

<http://msdn.microsoft.com/en-us/scriptjunkie/ff848255.aspx>

The jQuery Project. 2010. jQuery [viitattu 17.3.2011]. Saatavissa:

<http://jquery.com/>

UKK-Instituutti. 4.1.2011. Liikuntapiirakka [viitattu 13.3.2011]. Saatavissa:

<http://www.ukkinstituutti.fi/liikuntapiirakka>

UKK-Instituutti. 7.4.2010. Terveysliikuntaa ja kuntoliikuntaa [viitattu:

13.3.2011]. Saatavissa:

http://www.ukkinstituutti.fi/tietoa_terveysliikunnasta/aloittajan_liikuntaopas/terveysliikuntaa_ja_kuntoliikuntaa

Wikipedia 16.3.2011a. jQuery [viitattu 19.3.2011]. Saatavissa:

<http://en.wikipedia.org/wiki/Jquery>

Wikipedia 6.3.2011b. Document Object Model [viitattu 19.3.2011]. Saatavissa:

http://en.wikipedia.org/wiki/Document_Object_Model

W3Schools 2011. HTML DOM Tutorial [viitattu 14.4.2011]. Saatavissa:

<http://www.w3schools.com/html/dom/default.asp>

Weyl E. 15.11.2008. jQuery Tutorial: DOM Manipulation and Sorting [viitattu

21.3.2011]. Saatavissa:

<http://www.evotech.net/blog/2008/11/jquery-tutorial-dom-manipulation-and-sorting>

Yishi P. 18.8.2009. 5 Ways to Make Ajax Calls with jQuery [viitattu 25.3.2011].

Saatavissa:

<http://net.tutsplus.com/tutorials/javascript-ajax/5-ways-to-make-ajax-calls-with-jquery>