

Tampereen ammattikorkeakoulu  
Tietojenkäsittelyn koulutusohjelma  
Heli Edvik

Opinnäytetyö

## **Ainejärjestön internetsivuston toteutus**

Työn ohjaaja Petri Heliniemi  
Tampere 06/2009

Tampereen ammattikorkeakoulu  
Tietojenkäsittelyn koulutusohjelma

Tekijä	Heli Edvik
Työn nimi	Ainejärjestön internetsivuston toteutus
Sivumäärä	38
Valmistumisaika	kesäkuu 2009
Työn ohjaaja	Petri Heliniemi

---

## TIIVISTELMÄ

Opinnäytetyön sovellusosan tarkoituksena oli tehdä uudet internetsivut, joilla saataisiin ainejärjestön sivuille lisää kävijöitä mielenkiintoisemman ulkoasun ja sisällön avulla. Lisäksi tarkoitus oli helpottaa sivuston ylläpitoa, kun aiemmin sivustolla ei ollut ollenkaan erillistä ylläpitoa.

Sovelluksen käytännön toteutus tapahtui PHP-ohjelmointikielen sekä MySQL-tietokannan avulla. Sivuston ulkoasu toteutettiin XHTML- ja CSS-kielillä. Käytettävät tekniikat valittiin sovelluksen toteuttajan taitojen mukaan.

Lopputuloksena syntyi uudet internetsivut, joissa on toimiva ylläpito. Internetsivut vastasivat alussa asetettuja tavoitteita. Kävijöiden määrän lisääntymisestä ei ole kuitenkaan todisteita, koska sen mittaaminen tai minkäänlainen mielipidekysely ei kuulunut opinnäytetyöni piiriin.

Opinnäytetyön raporttiosa sisältää hieman teoriaa käytetyistä tekniikoista. Lisäksi kerrotaan, mihin ratkaisuihin päädyttiin sovelluksen toteutuksessa ja miten sovelluksen eri osat on toteutettu.

TAMK University of Applied Sciences  
Business Information Systems

Writer	Heli Edvik
Thesis	Creating a Student Association Internet Pages
Pages	38
Graduation time	June 2009
Thesis Supervisor	Petri Heliniemi

---

## ABSTRACT

The purpose of this final thesis work was to create new Internet pages for a student association. These new pages were going to bring more visitors to the site with their new graphic appearance and content. Another aim was to make it simpler to update the pages through their own administration, which did not exist in the prior version.

The actual implementation was created with the PHP-programming language and MySQL database. The appearance of the site was created with XHTML and CSS. The techniques used in the making of this application were chosen according to the skills of the person responsible.

As a result, the new pages were created and they have a functional administration. The Internet pages met the goals that were set in the beginning. However, there is no proof of the increase of the users on the pages, because this type of research was not part of my final thesis.

There is some theory of the technologies in question in this final thesis. It also contains the description of the creation of the different parts of the application and why the solutions and decisions were made as they were.

## Sisällysluettelo

1	Johdanto .....	5
2	Tausta .....	7
2.1	Historiaa .....	7
2.2	Toimintaperiaatteita .....	8
3	Käytetyt teknologiat .....	10
3.1	PHP .....	10
3.2	MySQL.....	11
3.3	XHTML ja CSS .....	12
4	Sovelluksen työvaiheet .....	14
4.1	Lähtökohdat .....	14
4.2	Sivuston ulkoasu ja rakenne.....	15
4.3	Tietokannan rakentaminen.....	17
4.4	Tietokannan toteutus .....	19
4.5	Sovelluksen toiminnallisuus .....	20
4.6	Ylläpito.....	24
5	Lopputulos .....	28
5.1	Ongelmat.....	28
5.2	Onnistumiset .....	29
6	Yhteenveto .....	30
	Lähdeluettelo.....	31
	Liitteet .....	32
	Liite 1: Sanasto.....	32
	Liite 2: Valikon XHTML- ja CSS-koodit.....	34

# 1 Johdanto

Opinnäytetyöni aiheena oli uudistettujen internetsivujen tekeminen eräälle ainejärjestölle. Heidän vanhat sivunsa olivat kirjaimellisesti vanhat, ja ne kaipasivat jotain uudistusta, jotta sivuilla saataisiin käymään enemmän ihmisiä ja erityisesti heidän koulunsa ja varsinkin oman alansa opiskelijoita. Sivujen kautta oli hoidettu tiedotus ainejärjestön tapahtumista sekä uutisista ja sivuilla olevalla lomakkeella pystyi liittymään ainejärjestön jäseneksi. Sivuilta löytyi myös tietoa ainejärjestön historiasta ja kuvia menneistä tapahtumista. Yksi ongelmista oli se, että ainejärjestön hallituksen jäsenillä ei ollut kokemusta internetsivujen tekemisestä, joten kukaan ei siis osannut lisätä mitään suoraan lähdekoodin joukkoon. Vanhoilta sivuilta puuttui täysin erillinen ylläpito-osio, joten sivujen päivittäminen kävi hieman hankalaksi. Tapahtumien lisääminen onnistui Googlen kalenteriin, joka oli upotettu etusivulle, mutta siinä olikin kaikki päivitys, mitä he pystyivät tekemään.

Lähdin liikkeelle opinnäytetyöni kanssa sillä periaatteella, että minulla oli toimeksiantaja. Myöhemmin jouduin kuitenkin luopumaan toimeksiantajasta aikatauluongelmien takia. Materiaalin toimitus viivästyi, ja kaikki venähti, joten kiinnostus loppui. Osa materiaalista on edelleen toimittamatta, ja vaikka minua vannotettiin pitämään kiinni aikatauluista, he eivät sitä itse tehneet. Sen vuoksi jouduin hieman muuttamaan alkuperäistä suunnitelmaa, ja tein opinnäytetyön loppuun ilman toimeksiantajaa. Pohjaan kuitenkin tekemääni käytännön työhön tässä opinnäytetyössä kuvaamani ratkaisut, vaikka sinänsä en käytä tätä toimeksiantotyönä. Tavoitteena työssäni oli se, että saisin rakennettua internetsivut, joissa olisi toimiva ylläpito, ja jotka olisivat mahdollisimman paljon nykystandardien mukaiset. Halusin kerrata oppimiani menetelmiä, koska niistä oli paljon jo unohtunut. Halusin myös samalla oppia uutta tuntemistani menetelmistä ja tekniikoista, jos vaikka joskus tulevaisuudessa tulisi tehtyä enemmänkin internetsivuja.

Tekniikat internetsivujen toteutukseen valitsin sen perusteella, mitä osasin jo ennestään. Käytin siis työssäni XHTML:ää, PHP:tä ja MySQL:ää. Lisäksi pieni koodinpätkä JavaScript-kieltä piti myös kirjoittaa, vaikka JS:ä en aiemmin ole käyttänytkään. Uusien tekniikoiden opetteleminen tämän työn puitteissa olisi ollut hieman liioiteltua varsinkin, kun vanhoissa oli runsaasti kertaamista. Tässä opinnäytetyöraportissani kerron taustaa

tekniikoista ja tekemistäni ratkaisuista. Liitteeseen 1 olen kerännyt käyttämiäni hieman ehkä vieraita sanoja, ja selittänyt niiden tarkoitukset.

Pyrin käyttämään työssäni mahdollisimman monipuolisia lähteitä. Varsinkin sovelluksen toteutusvaiheessa käytin apuna runsaasti internetiä, mutta kirjoitusvaiheessa lähinnä kirjoja. Yritin myös valita mahdollisimman luotettavat lähteet, jotta tieto olisi totuudenmukaista. Osaan teksteistä oli lähes mahdotonta löytää hyviä lähteitä, joten parissa kohtaa jouduin luottamaan Wikipediaan. Se ei kuulu luotettavimpien lähteideni kärkipäähän, mutta tiedot ainejärjestöistä ja muusta aiheeseen liittyvästä vaikuttivat minusta luotettavalta ja vuosiluvut vahvistuivat ainejärjestöjen omilta sivuilta, joten tulin siihen lopputulokseen, että lähde oli tarpeeksi luotettava tämän tiedon julkaisemiseen. Kirjojen käyttöä lähteinä hankaloitti se, että niistä suurin osa oli melko vanhoja, mutta toisaalta käyttämäni tekniikat ja niiden historia ei muuksi ole aikojen saatossa muuttunut, joten siinä mielessä vanhatkin kirjat ovat luotettavia. Lisäksi kirjoissa keskitytään yleensä yhden tekniikan erinomaisuuksien esittelyyn, muttei kerrota sen puutteista tai rajoitteista mitään. Poikkeuksellisesti vain eräässä CSS-kirjassa, joka löytyy lähdeluettelostakin, on kerrottu myös CSS:n huonoista puolista, mikä oli mielestäni todella piristävää.

## 2 Tausta

Tässä luvussa kerrotaan, mikä ylipäätään on ainejärjestö, ja mistä heidän toiminnassaan on kyse. Se kuuluu kuitenkin oleellisena osana oppinäytetyöhöni.

### 2.1 Historiaa

Ylioppilaskunta on julkisyhteisö, jollainen on jokaisessa Suomen yliopistossa. Kaikki yliopistossa opiskelevat perustutkinto-opiskelijat kuuluvat ylioppilaskuntaan, joka voi hyväksyä jäsenikseen myös muita yliopisto-opiskelijoita. Suomen ensimmäinen ylioppilaskunta perustettiin Aleksanterin yliopistoon vuonna 1868 nimellä Suomen ylioppilaskunta, ja se toimii nykyään nimellä Helsingin yliopiston ylioppilaskunta. Nimeä muutettiin, kun maahan alettiin perustaa muita yliopistoja. (Wikipedia, 2009a. )

Tällä hetkellä Helsingin yliopistossa on 11 tiedekuntaa, joissa kaikissa opiskelijat ovat muodostaneet erilaisia järjestöjä, jotka auttavat yhdistämään samaa alaa opiskelevia ylioppilaita. Useissa tiedekunnissa ainejärjestöjen yhteistyötä koordinoi tiedekuntajärjestö. Yliopistossa toimii myös tiedekuntarajat ylittäviä monitieteisiä ainejärjestöjä, joissa opiskelijat voivat siis olla erikoistuneita eri tieteenaloille. (Helsingin yliopiston oppilaskunta, 2008.)

Ainejärjestöt toimivat siis ylioppilaskuntien alaisuudessa, mutta hyvin itsenäisesti. Oppilaskuntien ja samalla ainejärjestöjenkin toiminnasta on määritetty laissa seuraavaa: ”Jokaisella oppilaitoksella, jossa järjestetään tässä laissa tarkoitettua koulutusta, on opiskelijoista muodostuva oppilaskunta. Oppilaskunnan tehtävänä on edistää opiskelijoiden yhteistoimintaa ja koulutyötä. Oppilaskunta käyttää opiskelijoiden puhevaltaa 36 §:ssä säädetyissä asioissa.” (Finlex, 1998.)

Ainejärjestöt toimivat yleensä yliopistojen tai korkeakoulujen piirissä. Niiden tarkoituksena on tukea opiskelua ja järjestää tapahtumia ja toimintaa jäsenilleen, sekä ajaa jäseniensä etua ja yhdistää opiskelijoita. Teknillisten yliopistojen toiminta on melko lailla samaa, mutta niiden vastaavia järjestöjä kutsutaan yleensä killoiksi. Vanhin ainejärjestö, Helsingin yliopistolla toimiva Teologian Ylioppilaiden Tiedekuntayhdistys eli TYT, on perustettu vuonna 1853. Tampereen yliopiston vanhin ainejärjestö, Hallat Ry, on perus-

tettu vuonna 1928. Itse asiassa ainejärjestöt yleistyivätkin varsinaisesti vasta toisen maailmansodan jälkeen. (Wikipedia, 2009b.)

Ainejärjestöjen koot vaihtelevat laidasta laitaan. Ne voivat olla todella suuria, koko tiedekunnan kattavia, tai sitten hyvin pieniä, muutaman kymmenen jäsenen yhdistyksiä. Helsingin yliopistolla on pienissä ja epäyhtenäisissä tiedekunnissa vain yksi ainejärjestö, kun taas suuremmissa ja monitieteellisimmissä tiedekunnissa on useita eri ainejärjestöjä, joiden yhteistyöelimenä toimii niin sanottu kattojärjestö. (Wikipedia, 2009b.)

Teekkareiden ainejärjestöille yleinen nimitys on kilta. Ensimmäinen perustettiin jo vuonna 1891 nykyisen Teknillisen korkeakoulun yhteyteen. Killoilla on suuri merkitys perinteisen teekkarikulttuurin edistämässä ja ylläpitämisessä. Toiminta muistuttaa suurimmalta osalta muidenkin ainejärjestöjen toimintaa; myös killat järjestävät toimintaa ja tapahtumia, sekä toimivat yhdyssiteenä koulutusohjelmiansa opiskelijoille. Fuksien, eli aloittaneiden opiskelijoiden, tutorointi on myös kiltojen vastuulla. Paikallisella tasolla kiltojen toiminta on lähes identtistä, mutta valtakunnallisella tasolla vaihtelu voi olla suurta. Saman alan kiltojen yhteistyö on todella merkittävää. Tunnusmerkkinä killoille toimivat katukuvastakin tutut haalarit, joiden väri vaihtelee killan mukaan. Myös muilla ainejärjestöillä on käytössään haalarit, joiden väri vaihtelee milloin koulun ja milloin alan mukaan. (Wikipedia, 2009c.)

## 2.2 Toimintaperiaatteita

Ainejärjestöille näkyvyys ja tunnettuus ovat erittäin tärkeitä ominaisuuksia. Jos ainejärjestö näkyy ja kuuluu, sen avulla järjestöön saadaan mukaan lisää jäseniä, millä taas saadaan lisättyä järjestön tunnettavuutta. Erinäisten yhteistyökumppaneiden avulla on helpompaa järjestää toimintaa ja tapahtumia kuin ilman yhteistyökumppaneita. Yhteistyökumppaneiden hankkiminen todella pienelle ainejärjestölle on varmasti vaikeampaa kuin suuremmalle ainejärjestölle. Jos yhteistyöstä ei ole hyötyä kummallekin osapuolelle, se ei ole kannattavaa. Tämän vuoksi on tärkeää, että ainejärjestöillä on mm. internet-sivustoja, joilla omaa sekä yhteistyökumppaneiden toimintaa voi mainostaa. Lisäksi omat lehdet ovat myös yksi tiedonvälityskanava. Harvoilla pienillä ainejärjestöillä on



kuitenkaan omaa lehteä, vaan lähinnä heidän kattojärjestönsä tai ylioppilaskuntansa lehdet ovat käytössä.

Kun ainejärjestö järjestää tapahtumia, kuten juhlat, on hyvä miettiä aina etukäteen, miksi kyseinen tapahtuma kannattaa järjestää. On hyvä erottua joukosta ja hankkia jokin syy juhlimiselle sen perinteisen ravintolaan lähtemisen sijaan. Jos idea on tarpeeksi hyvä, se erottuu joukosta ja saa jäsenet, sekä muutkin opiskelijat innostumaan. Juhlia voi ajatella vaikka jäsenetuna ja/tai mahdollisuutena kerätä rahaa järjestölle, jotta voidaan taata jatkossa paremmat edut. Aina kuitenkin pitää muistaa ne opiskelijat ja jäsenet, joita varten juhlat järjestetään, eikä ajaa vain järjestön hallituksen etua. (Laine & Vierelä 1998, 42)

Ainejärjestöille helppo kanava näkyä on internet, kun se nykypäivänä on vielä kovin suosittukin. On jo melkein harvinaista, jos jollain toimintaa järjestävällä taholla, oli se sitten mitä toimintaa tahansa, ei ole omia internetsivuja. Niiden olemassaolo ylipäättään ei kuitenkaan täysin riitä, vaan niiden ylläpitäminen on todella tärkeää. Jos sivuilla ei tapahdu mitään, ei kovinkaan moni jaksa niillä useampaan otteeseen vieraila. Tärkeintä on pitää yllä tiedotusta tapahtumista ja pitää muutenkin jäsenet ajan tasalla. Kuvat kiinnostavat usein myös monia, joten galleriat ja mahdollisesti myös erilaiset forumit ovat hyvä lisä sivuille. Tähän kaikkeen vaaditaan hallituksen paneutumista työlleen ja oikeasti halua ylläpitää järjestön toimintaa ja huomioida jäseniään, jotta toiminta olisi kannattavaa. Jäsenien sulkeminen ulkopuolella aiheuttaa yleensä jäsenmäärän vähenemistä, koska jäsenille tulee helposti olo, että mitään ei tapahdu ja jäsenyys on aivan turhaa.

### 3 Käytetyt teknologiat

Vaikka käytin työni sovellusosuudessa monien jo ennestään tuntemia tekniikoita, halusin silti kertoa niistä lyhyesti. Jos mietin, että ystäväni tai sukulaiseni lukisi työni, ei kovinkaan moni tietäisi, mistä näissä tekniikoissa on kyse. Minua itseäni myös kiinnosti tekniikoiden kehitys, koska en siihen aiemmin juurikaan ollut perehtynyt.

#### 3.1 PHP

PHP on avoin skriptauskieli ja se perustuu muihin ohjelmointikieliin, kuten Perl, ASP ja Java. Sen avulla pystytään toteuttamaan erilaisia dynaamisia online-ympäristöjä. Tarina juontaa juurensa vuoteen 1994, jolloin ensimmäisen version siitä julkaisi Rasmus Lerdorf. Se julkistettiin nimellä Personal Home Page Tools, koska kyseessä oli alun perin joukko makroja, jotka autoivat koodaajia kotisivujen ylläpidossa. Tästä siis nimi PHP. Nykyään PHP on laajentunut huomattavasti ja se tunnetaan virallisesti nimellä PHP: HyperText Preprocessor, joka on suomeksi hypertekstin esikäsittelijä. Sen ominaisuudet ovat huomattavasti lisääntyneet sitten julkaisun ja PHP:stä on tullut täysverinen ohjelmointikieli. (Zandstra 2001, 13 - 20)

Vuonna 1997 julkaistiin PHP3, joka syntyi erään koodausryhmän työn tuloksena. Siinä oli muun muassa uusi jäsentelijä, jonka olivat kehittäneet Zeev Suraski ja Andi Gutmans. Se erosi edellisestä versiosta syntaksiltaan ja sisälsi muutenkin uusia piirteitä. Tämä julkistus löi läpi ja PHP jatkoi kasvuaan. Marraskuussa 1999 PHP oli käytössä jo yli miljoonassa palvelimessa ja vuonna 2000 helmikuussa kyseinen luku oli noussut jo 1,4 miljoonaan. Suosiota lisäsi myös tuki Apachelle ja MySQL:lle, jotka nekin olivat ilmaisia ja avointa lähdekoodia. (Zandstra 2001, 20)

Kyseessä on siis palvelinpuolen skriptikieli, joka jäsennellään PHP-moduulin toimesta. Vaikka PHP kirjoitetaankin tavallisesti HTML:n yhteydessä, PHP-skriptiä ei kuitenkaan lähetetä suoraan asiakkaalle, kuten tavalliselle HTML-sivulle tehdään. Skriptissä olevat HTML-elementit jätetään käsittelemättä, mutta PHP tulkitaan sekä suoritetaan. PHP:llä pystytään toteuttamaan kyselyt tietokantaan, kuvien luominen, tiedostojen luku, tiedoston kirjoittaminen ja keskustelu etäpalvelimien kanssa. Kun PHP on tulkattu ja suori-

tettu, tulostus yhdistetään HTML:ään, josta syntyy se lopullinen tuotos, jonka käyttäjä näkee näytöllään selaimessa. (Zandstra 2001, 20)

PHP:n käyttöön löytyy monia syitä, mutta ehkä tärkeimpänä niistä on se, että kyseessä on open source eli avoin lähdekoodi, joka on täysin ilmaista. Kuka tahansa voi missä tahansa käyttää sitä oman sovelluksensa kehitykseen ilman, että joutuu maksamaan kaltaista rojalteja. PHP on nopeampi koodata, sekä myös nopeampi suorittaa kuin muut vastaavat kielet, koska PHP on tehty internetsivujen luomista varten ja koodi löytyy suoraan internetsivulta. Lisäksi se toimii eri palvelimissa ja käyttöjärjestelmissä, joka mahdollistaa skriptien siirtämisen palvelinalustalta toiselle ilman muutoksia. Suosionsa ansiosta sitä kehitetään jatkuvasti, ja se on myös helposti muokattavissa, joten kehitys on mahdollista. Osaavat koodaajat pystyvät halutessaan lisäämään sovelluksensa PHP-koodiin toiminnallisuutta ja ominaisuuksia. PHP:ssä on silti jo valmiiksi paljon funktioita, joten sen avulla pärjää ilman lisäyksiäkin. Se on tehty internetsivuja varten, ja se on helposti opittavissa. Tukikin on ilmainen, koska internetissä on todella paljon uutispalstoja ja muita, joista voi kysellä apua ja vastaus tulee yleensä aika nopeasti. Mahdollisuuksien ja kiinnostuksen mukaan on myös helppo opetella muita ohjelmointikieliä, jos PHP:n osaaminen on jo pohjalla. (Atkinson 1999, 5 - 6)

## 3.2 MySQL

MySQL kehitettiin alun perin verkkosovelluksiin. Tämän mahdollistivat sen nopeus, laajennettavuus ja ylläpidon helppous. Kyseessä on siis relaatiotietokantojen hallintajärjestelmä, jolla on jo useita miljoonia käyttäjiä. Käyttäjiin kuuluu niin yksityisiä henkilöitä kuin suuria yrityksiä. Jälkimmäisestä esimerkkinä on muun muassa Yahoo!. Relatiotietokannat ovat yksinkertaisesti selitettynä joukko tauluja, jotka koostuvat erilaisista sarakkeista ja riveistä. Nämä ovat suhteessa toisiinsa tiettyyn sarakkeeseen merkityllä arvolla. MySQL on relaatiotietokantojen hallintajärjestelmä, joka varastoi kaikki tietokannat, taulut, sarakkeet ja rivit datoiin ja käsittelee niitä yhtenä kokonaisuutena. Käyttäjät saavat eritasoisia käyttöoikeuksia, joita MySQL RDBMS (Relational Database Management System) hallitsee. MySQL pitää kirjaa näiden käyttäjien toimista ja vastaa käyttäjien kyselyihin. Kyselyt suoritetaan SQL-komennoilla. SQL, eli Structured Query Language, on rakenteinen kyselykieli, jota käytetään kommunikointiin relaatio-

tietokantojen kanssa. Se on ollut olemassa jo 1970-luvulta asti, jolloin se on luotu IBM:n tutkimuskeskuksessa. (Meloni 2003, 7 - 11)

MySQL on nopea ja helppokäyttöinen. Se on kymmeniä tuhansia euroja halvempi, kuin Oracle, sillä MySQL:stä on sekä kaupallisia että avoimen lähdekoodin versioita. Avoimen lähdekoodin hyöty on muun muassa siinä, että se on useimmiten lähes ilmainen. MySQL:n on siirrettävyydeltään erittäin hyvä ja se on yhteensopiva minkä tahansa ohjelmointikielen kanssa. MySQL on lisäksi niin sanotusti monisäikeinen palvelin, eli se aloittaa aina uuden palvelinprosessin, kun yhteys muodostuu. Tämä taas tarkoittaa sitä, että prosessin päättyessä äkillisesti tai palvelimen ylikuormittuessa, se ei kaadu, vaan sulkee sen yhden prosessin ja toiminta voi jatkua. Tämä yleensä lisää MySQL:n nopeutta. (Meloni 2003, 11 - 12)

### **3.3 XHTML ja CSS**

Lyhykäisyydessään sanottuna XHTML yhdistää HTML-sanaston ja XML-syntaksin parhaat puolet. XHTML luotiin, koska HTML ei enää täyttänyt vaatimuksia, joita nykyajan tarpeet asettavat. Se on taaksepäin yhteensopiva kaikkien selaimien kanssa, jopa aina versioon 2.0 asti. XHTML:n siirrettävyys on todella hyvä, sillä kaikki selaimet sekä internetiin kytkettävät pienlaitteet esittävät asiakirjat täysin samalla tavalla. Sitä voi myös hyvin laajentaa omilla tageilla määrittelemällä uuden nimiavaruuden. Nimiavaruus tarkoittaa tässä tapauksessa uutta tagijoukkoa, joka eroaa täysin kaikista muista joukoista. XHTML:n DTD:llä (Document Type Definition) määritellään nuo nimialueet, jotta asiakas osaisi tulkita niitä. (Boumphrey, Greer, Raggett, Raggett, Schnitzenbaumer & Wugofski 2001, 33 - 35)

CSS, eli Cascading Style Sheets, on merkintäjärjestelmä, joka antaa ohjeita web-sivujen ja muidenkin dokumenttien ulkoasusta. Yhtä ehdotusten kokonaisuutta sanotaan tyyliohjeksi eli tyylisäännöksi. Sen ulkoasuehdotukset ovat erillisenä kokonaisuutena omassa tiedostossaan, joka mahdollistaa eri tyyliä esimerkiksi tulostukselle ja kuvaruudulle. Selain voi halutessaan huomioida tyyliasetukset tai jättää ne myös huomioimatta. Tämä on mahdollista silloin, kun käyttäjällä on erikoisselain, joka käyttää aina esimerkiksi vain yhtä kirjasinlajia joko teknisistä syistä tai käyttäjän itse asettamana. Tällöin

sivu esitetään kuin tyyliohjetta ei olisikaan. On myös mahdollista antaa useamman tyyliohjeen vaikuttaa samaan aikaan internetsivun ulkoasuun. (Korpela 2008, XII - 5)

CSS on monipuolisempi kuin HTML-koodin omat sisäiset ulkoasu-komennot. Se on suunniteltu pelkästään ulkoasun muotoiluun, ja siinä on useita eri tapoja esittää tietyn ominaisuuden arvo. Esimerkiksi HTML:ssä käytetty lihavoinnin `<b>`-tagi on vain lihavointi, ilman sen kummempia määrittämiä. CSS:n avulla voi muokata viivan paksuutta määrittelemällä CSS-tiedostoon tekstit `font-weight`, jolla on yhdeksän erilaista arvoa, eli samalla myös yhdeksän eri paksuutta tekstin koolle. CSS:ssä on paljon uusia ominaisuuksia ja sillä saa enemmän vaihtoehtoja tyylimäärittelyyn. CSS:n tyylimääritteet pysyvät tekemään samaan tiedostoon kuin XHTML-koodin, mutta se ei ole suositeltava tapa, koska se hankaloittaa tyylien muokkaamista ja ylläpitoa. Kannattaa aina tehdä CSS:stä oma tiedostonsa, sillä se mahdollistaa tyyliominaisuuksien käytön useassa eri paikassa. Tämä mahdollistaa myös omien tyyliominaisuuksien luomisen, joista voi tehdä valmiita pohjia jatkoa ajatellen. Sivujen ulkoasun muutto on todella helppoa, kun muokattavana on vain yksi tiedosto usean sijaan. Kun CSS on omassa tiedostossaan, usean eri tyylin näyttämisen on mahdollista samaan aikaan. Tätä ominaisuutta selaimet eivät tosin valitettavasti vielä tue kovinkaan hyvin. (Korpela 2008, 25 - 26)

Ongelmiakin toki esiintyy. CSS on hankalaa oppia pelkällä kokeilulla ja kopioinnilla. Komennot on helppo ymmärtää väärin, ja kaikki ei aina ole aivan täysin loogista, eikä sen mukaista, mihin aiemmin on tottunut. Siksi CSS vaatii hieman enemmän opettelua, mutta ulkoa ei kaikkea kannata opetella. Ongelmana on myös se, että selaimet tulkitsevat usein väärin CSS-tyylimäärittelyt, eikä lopputulos olekaan halutunlainen. Tämä ongelma on kuitenkin nykyään parantunut, mutta CSS:n käyttö vaatii vielä hieman varovaisuutta. (Korpela 2008, XII - 2)

## 4 Sovelluksen työvaiheet

Työn toteutus lähti perinteisesti liikkeelle siitä, että ensin oli jokin tarve jollakin, jota sitten lähdettiin toteuttamaan. Aiemmin luvussa 3 kerrottiin hieman teoriaa käytetyistä menetelmistä ja tässä kuvataan, mihin ratkaisuihin päädyttiin internetsivujen toteuttamisessa.

### 4.1 Lähtökohdat

Vaatumuksena opinnäytetyössä kuvattavassa sovellusosuudessa oli se, että ainejärjestön näkyvyys paranisi ja sivuilla kävisi enemmän käyttäjiä kuin aiemmin. Lisäksi yhtenä osana oli täysin puuttuvan ylläpidon lisääminen sivustoon. Tämä oli tärkeää, koska harva kuitenkin loppujen lopuksi pystyy lisäämään tekstiä ja kuvia XHTML-koodin sekaan varsinkaan, jos ei koskaan aiemmin ole ollut tekemisissä internetsivujen koodaamisen kanssa. Ainejärjestön hallituksen kanssa oli jo sitä paitsi todettu, että he eivät pysty päivittämään internetsivujaan lisäämällä haluttuja asioista, kuten kuvia tai tekstiä XHTML-koodin sekaan. Ulkoasusta toivottiin myös pirteämpää kuin vanhasta, joka oli aika synkkä.

Sivuston toteutus lähti käyntiin sillä, että sovimme tapaamisen ainejärjestön hallituksen kanssa. Palaverissa käytiin läpi, mitä uutta sivuille halutaan, miten sivujen värimaailma toteutetaan, ja mitkä asiat ainakin pitäisi sivuille saada. Palaverin aikana ideoita sateli todella paljon, mutta läheskään kaikki eivät olleet toteutuskelpoisia. Tämä johtui siitä, että suurin osa ideoista oli lähinnä huumoria, kuten esimerkiksi sivuilla juokseva possu, jonka käyttäjä voisi metsästää hiirellä napsauttamalla ja saada siitä jonkin palkinnon. Sain melko vapaat kädet sivujen tekemiseen, mutta tietyissä rajoissa. Toivoin, että hallitukselta olisi tullut hieman enemmän osviittaa siitä, mitä he oikeasti haluavat, koska myöhemmässä vaiheessa osoittautui, että kaikki ei sitten ihan toiminutkaan niin kuin he olisivat halunneet. Aloituvaiheessa ei kuitenkaan ollut ongelmia.

## 4.2 Sivuston ulkoasu ja rakenne

Aloitin sivuston tekemisen ulkoasun suunnittelusta. Toiveena oli, että se olisi näyttänyt suurin piirtein samalta kuin erään toisen ainejärjestön internetsivut, jotka näkyvät kuvassa 1. Värimaailman toivottiin olevan pinkki ja huomiota herättävä, ja sen piti pohjautua heidän yhdistyksensä haalarimerkkiin, joka on kuvassa 2. Logo oli muuten sama kuin vanhoillakin sivuilla, mutta siihen piti lisätä skannattu kuva kyseisestä haalarimerkistä (kuva 2). Kuvan laatu oli kuitenkin sen verran huono ja pikselöitynyt, että sen käytöstä luovuttiin ainakin toistaiseksi. Kuva oli skannattu joskus aikoja sitten eikä ainejärjestön internetsivuista vastannut henkilö ollut löytää sitä edes palvelimelta. Lähetin ensimmäisen version ulkoasusta (logo, alavetovalikko ja rakenne noin suurin piirtein) ainejärjestölle arvioitavaksi ennen suurempaa koodausta, jotta sain suunnan, mille pohjalle alkaa rakentaa.

The screenshot shows the website 'Finanssi ry' for 'Oulun yliopiston kauppatieteiden opiskelijat'. The page has a purple header with navigation links: Etusivu, Ajankohtaista, Finanssi ry, Opinasiat, Vapaa-aika, Yhteistyö, SEFE, Yhteystiedot, and In English. The main content area is divided into several sections:

- Palvelut**: Valmennuskurssi, Intranet, Tenttiarkisto, Kuvagalleria, Linkkejä, Palaute, Etsi, Sivukartta.
- Kesän tapahtumia**: Tulossa:
  - 28.5. Kesäkauden avajaiset & School's Out bileet @ Onnela
  - 4.7. Beach volley -turnaus
  - 25.8. Fuksiristeily
  - 15.9. Fuksisuunnistus
  - 17.-18.10. Vuosijuhla
- Hyödyllistä katsottavaa**: Oulu@VV -videe
- Hyödyllistä luettavaa**:
  - >> EGO-nomi 2/2009
  - >> Sefen palkkasuosittukset 2009
- Uudet pienryhmäohjaajat vuodelle 2009**:
 

Johtaminen	Kv. liiketoiminta	Kansantaloustiede
Mika Harju	Jini Majanen	Antti Nuoto
Kauko Keskisärkkä	Joni Vajus-Anttila	Antti Savikuja
Ari Kolivisto		Ville Suvanto
- Laskentatoimi**: Anne Harjulahti, Hanna Hauta, Mikko Huotari, Martti Kultalahti
- Markkinointi**: Teemu Kaasinen, Outi Rajaniemi, Veli-Pekka Rönty, Saara Satokangas
- Rahoitus**: Hanna Halonen, Niko Huusko

Sidebars include:

- Yhteistyössä**: se\$fe logo
- Viikon kysymys**: Joko kesäloma alkoi? (radio buttons for Yes, No, and a question mark)
- Muuta mukavaa**:
  - Oulun yliopisto
  - TaTK
  - Webmail
  - WebOodi
  - Optima
  - Kirasto (Oula / Nelli)
  - Unirestan ruokalistat
  - Oulun kaupunki
  - Linjakas-reittipalvelu
  - Sääennuste
  - Kyltit.net

Kuva 1: Ulkoasun malli



Kuva 2: Haalarimerkki

Sivu on jaettu siten, että koko sisältö on yhden div-elementin sisällä. Annoin kyseiselle elementille nimeksi wrapper. Sen sisään tulivat divit header, menu, content, leftcol ja rightcol. Header-divin sisällä on ainoastaan logo ja sen taustalla oleva kuva. Suunnitelma oli ensin, että siihen laitettaisiin vain pelkästään logo, mutta se osoittautui hieman tylsäksi vaihtoehdoksi. Sain käyttää ainejärjestön kuvagalleriaa vapaasti hyödykseni, joten pyysin hallitusta valitsemaan mieleisensä kuvan, josta muokkasin PhotoShopilla sivuston muuhun värimaailmaan sopivan taustan logolle. Menu-divin sisältä löytyy alasvetovalikko. Kyseinen valikko aiheutti jonkin verran ongelmia ja CSS:ään tutustumista. Halusin tehdä samanlaisen alasvetovalikon kuin hallituksen esimerkkinä käyttämän ainejärjestön internetsivuilla oli. Heidän sivustonsa lähdekoodia tutkittuani huomasin, että kyseinen valikkorakenne oli toteutettu JavaScriptillä. En ole aiemmin ohjelmoinut kyseisellä kielellä, ja tulin siihen tulokseen, että sen opettelu olisi liian työlästä ja aikaa vievää tätä tarkoitusta varten. Valmiita koodeja tällaisen valikon toteuttamiseen olisi löytynyt internetistä, mutta niiden sisällön ymmärtäminen ei ollut helppoa. Oli vaikeaa tulkita, mikä kohta koodia tarkoitti mitään kohtaa alasvetovalikossa. Olisin ehkä ollut toista mieltä, jos olisin tiennyt, kuinka hankalaa moinen valikko oli toteuttaa CSS:llä.

Kuvassa 3 näkyy, miten valikko toimii. Kun hiiren vie laatikon eli linkin päälle, laatikko vaihtaa väriä ja siitä avautuu alavalikko. Tämä ei kuitenkaan tapahdu ihan joka kohdassa, sillä kaikille ei ollut alavalikkoa. Valikot on jaoteltu mahdollisimman loogisesti, jotta käyttäjä pystyy päättämään, mitä minkin linkin alta löytyy. Opiskeluun on kerätty siihen kuuluvat asiat, kuten sähköposti, Moodle ja ynnä muut sellaiset. Elämän alta löytyy muun muassa ruokalista ja hyödyllisiä linkkejä. Tämä valikkorakenne on toteutettu kokonaan CSS:llä, ja se toimi yllättäen useissa eri selaimissa. Tällä valikkoratkai-



sulla XHTML-koodiin tuli monta riviä tekstiä lisää ja CSS-tiedosto sai myös kiitettävän määrän pituutta. Liitteestä 2 löytyy sekä XHTML-koodi että CSS-tiedoston osa, joilla tämä valikko on rakennettu.

Etusivu	Opiskelu	Elämä	Yhdistys	Yhteistyö	Yhteystiedot	Vuosijuhlat
	Koulu					
	Kurssit ja tentit	Tenttiarkisto				
	Kirjasto					
	Sähköposti					
	Webboodi					
	Moodle					
	TuKY					
	SEFE					

Kuva 3: Valikkorakenne

Divit content, leftcol ja rightcol muodostavat loppuosan sivusta, ja kuten nimistä voi päätellä, myös tavallaan sen oleellisimman osan. Sivun vasemmasta reunasta eli divistä leftcol löytyvät yhteistyökumppaneiden logot. Oikeasta reunasta eli divistä rightcol löytyy säätiedot sekä mahdollisesti mainoksia tai muuta mukavaa, joka halutaan etusivulle. Keskelle eli content-diviin tulee siis kaikki varsinainen asia. Toteutin pienen koodinpätkän avulla sen, että jokaisesta linkistä napsautettaessa, paitsi niistä, jotka lähtevät niin sanotusti ulos, ladataan sisältö content-diviin, eikä ladata koko sivua aina uusiksi.

### 4.3 Tietokannan rakentaminen

Tietokannan suunnittelussa käytin ratkaisua yksinkertainen on kaunista. Kannan koko oli todella pieni, joten sen toteuttamisesta selvisi kahdella taululla. Ensimmäinen taulu on uutisia ja toinen tapahtumia varten. Sekä tapahtumat että uutiset olisi voinut laittaa samaan tauluun, mutta se olisi hankaloittanut ylläpitoa. Tietokannan laajentaminen tulevaisuudessa olisi myös hankaloitunut ja viimeistään silloin näille kahdelle olisi pitänyt rakentaa erilliset taulut.

Kuvassa 4 on nähtävissä Uutiset-työkalun rakenne. Työkalussa on uutisen lisäämiseen välttämättä tarvittavat kentät. Kenttien nimet ovat englanniksi sen vuoksi, että nimissä ei voi olla

”ääkkösiä”, joten niitä on helpompi tulkita englanniksi. Kukaan uutinen saa yksilöivän id:n, joka luodaan automaattisesti. Tämän on tarkoitus helpottaa hakuja ja muokkausta. Kuvan 4 head tarkoittaa otsikko-kenttää ja written on uutisen kirjoittaja. Päivämäärään on tehty erikseen kentät päivälle, kuukaudelle ja vuodelle (kuvassa 4 day, month ja year), koska MySQL:n tarjoama valmis DATE -arvomäärittely esittää tulostettaessa päivämäärän muodossa xx-xx-xxxx, eli pisteiden sijaan numeroita erottavat viivat eikä sitä pysty muuttamaan. Tein PHP:llä hyvin yksinkertaisen koodinpätkän, joka näyttää päivämäärän muodossa xx.xx.xxxx:

```
<? echo $rows['day']; ?>.<? echo $rows['month']; ?>.<? echo $rows['year']; ?> <? echo $rows['time']; ?>
```

Viimeinen story-kenttä on varattu itse uutiselle.

Field	Type	Null	Key	Default	Extra
id	int(5)	NO	PRI	NULL	auto_increment
head	varchar(50)	NO			
written	varchar(50)	YES		NULL	
day	int(2)	NO			
month	int(2)	NO			
year	int(4)	YES		NULL	
story	varchar(3000)	YES		NULL	

Kuva 4: Uutiset-taulu (tietokannassa nimellä news)

Tietyissä kentissä arvon on mahdollista olla myös NULL, joka tarkoittaa sitä, että niihin ei ole pakko syöttää mitään arvoa. Id tulee automaattisesti ja on pakollinen, koska se on myös avaimena. Avaimen avulla tauluja pystytään liittämään toisiinsa erilaisilla liittok-silla. Myös otsikko (head), päivä (day) ja month (kuukausi) ovat pakollisia. Päädyin ratkaisuun siksi, että jos varsinaista uutista ei sinänsä ole, mutta haluaa esimerkiksi otsi-kolla vain tiedottaa jotain, näinkin voi tehdä. Myöskään ei välttämättä aina ole tiedossa uutisen kirjoittajaa tai tapahtuman paikkaa, tai sitten jotain tietoja ei välttämättä haluta julkaista.

Tapahtumat-taulun (kuva 5) rakenne on lähes sama kuin Uutiset-taulunkin rakenne. Jokaisella tapahtumalla on oma id. Tapahtumalle annetaan nimi ja aika. Päivämäärä on samoin kuin uutisissa eli päivä, kuukausi ja vuosi ovat erikseen. Lisäksi tapahtumalle on annettu vielä aika-kenttä, johon voi määrittää joko pelkän alkamisajan, tai sitten kokonaiskeston, eli alkamis- ja päättymisajan. Paikka on myös tapahtumasta oleellinen tieto, ja lopuksi on vielä mahdollisuus antaa kuvaus tapahtumasta, jos on jotain erityistä, esimerkiksi jokin tietty teema tai muuta sellaista.

Field	Type	Null	Key	Default	Extra
id	int(5)	NO	PRI	NULL	auto_increment
name	varchar(50)	NO			
day	int(2)	NO			
month	int(2)	NO			
year	int(4)	YES		NULL	
time	varchar(20)	YES		NULL	
place	varchar(50)	YES		NULL	
description	varchar(3000)	YES		NULL	

Kuva 5: Tapahtumat-taulu (tietokannassa nimellä happening)

Tapahtumien tulostus on hieman erilainen kuin uutisten. Uutiset tulostuvat kokonaisuudessaan etusivulle, mutta tapahtumista tulostuu vain nimi ja päivämäärä. Tämä siitä syystä, että oleellisinta on nähdä koska ja mitä. Halutessaan käyttäjä voi sitten napsauttaa tapahtuman nimeä, josta näkee lisätiedot, kuten paikan, ajan ja muita yksityiskohtia.

#### 4.4 Tietokannan toteutus

Kaikki koodaus tehtiin aluksi TAMK:n palvelimelle, ja vasta valmis versio siirrettiin ainejärjestön palvelimelle. Myös tältä palvelimelta löytyy tuki MySQL:lle sekä PHP:lle, joten siirtäminen oli helppoa. Palvelimella on myös helppokäyttöinen hallintapaneeli, josta löytyy muun muassa PHPMyAdmin ja MySQL-tietokantojen hallinta. Tietokannan olisi pystynyt luomaan graafisessa käyttöliittymässä, mutta syötin sen mieluummin SQL-komentona, koska silloin kaikki ainakin varmasti meni halutusti.

Tietokannan luominen oli sen pienen koon vuoksi helppoa. Tein SQL-komennot tietokannan luomista varten. Alla on esimerkkinä Uutiset-taulun luomiskomento.

```
CREATE TABLE news (
  id INT(5) AUTO_INCREMENT PRIMARY KEY,
  head VARCHAR(50) NOT NULL,
  written VARCHAR(50),
  day INT(2) NOT NULL,
  month INT(2) NOT NULL,
  year INT(4),
  story VARCHAR(3000)
);
```

Tauluun on määritelty `PRIMARY KEY`, joka on tiedon yksilöivä tunniste. Ja se myös luodaan automaattisesti, jotta käyttäjän ei itse sitä tarvitse syöttää. Siitä johtuu tuo `AUTO INCREMENT`. `INT` ja `VARCHAR` tarkoittavat kentän tyyppiä, jotka tarkoittavat siis kokonaislukua (lyhenne sanasta integer) ja merkkijonoa. Sulkeissa oleva numero määrää, montako merkkiä kentässä enimmillään saa olla. `VARCHAR` on tyyppinä toimivampi kuin `CHAR`. `CHAR`-määritteeseen kenttään tulee myös merkkijono, mutta merkkien määrä on rajattu kentän pituuteen. `VARCHAR`-määritteeseen kenttään pystyy kirjoittamaan merkkejä haluamansa määrän kentän pituudesta huolimatta. Tätä määritelystä kannattaa käyttää silloin, kun kentässä olevien merkkien määrä saattaa vaihdella, kuten tässäkin tapauksessa. `NOT NULL` tarkoittaa sitä, että kenttää ei voi jättää tyhjäksi, vaan siihen on aina laitettava jokin arvo. Tapahtumat-taulu meni lähes samalla kaavalla, mutta kenttien nimet olivat tietysti erilaisia. Nekin tosin olivat englanniksi.

## 4.5 Sovelluksen toiminnallisuus

Kun kaikki muu oli suurin piirtein kunnossa, aloin toteuttaa sovellukseen PHP-koodia. Tarvitsin koodia etusivulla, jäseneksi liittymisessä ja tapahtuma-sivulla. Etusivulle oli tarkoituksena tulostaa kannasta uutiset kokonaisuudessaan ja tapahtumista otsikot sekä päivämäärät. Tapahtuman otsikkoa napsautettaessa, aukeaisi uusi sivu, jolla on tarkemmat tiedot kyseisestä tapahtumasta. Liittymissivulla tarvittiin koodinpätkä, joka tarkis-

taisi, että tiedot ovat oikeanlaisia, kuten sähköpostin muoto ja muut tällaiset seikat. Lisäksi tiedot piti lähettää ainejärjestön sähköpostiin.

Erillisistä config.php-, dbopen.php- ja dbclose.php-tiedostoista löytyy määrittelyjä yhteydestä tietokantaan. Kaksi ensimmäistä vaadittiin käyttöön aina koodin alussa ja viimeinen oli aina PHP-koodin lopussa require\_once-komennolla. Kyseessä ovat siis tiedostot, joissa on yhteyden avaaminen (dbopen.php) ja sulkeminen (dbclose.php) tietokantaan, sekä määrittely (config.php), joka kertoo mistä tietokannasta on kyse. Sisällöt ovat seuraavanlaiset:

config.php:

```
<?php
$dbuser = 'user';
$dbpass = 'pass';
$dbhost = 'host';
$dbname = 'name';
$dbtype = 'type';
?>
```

Heittomerkkien välissä ovat kyseessä olevan kannan tiedot, eli järjestyksessä käyttäjä, salasana, host eli palvelin, kannan nimi ja kannan tyyppi. Oikeita tietoja kantaan kirjautumisesta ei tähän ymmärrettävistä syistä voinut laittaa.

dbopen.php:

```
<?php
$conn = mysql_connect($dbhost, $dbuser, $dbpass)
    or die('Ei voida yhdistää: ' . mysql_error());
mysql_select_db($dbname, $conn);
?>
```

Tässä otetaan yhteys määritettyyn tietokantaan, joka löytyy määritetyltä palvelimelta, käyttäjätunnuksen ja salasanan avulla (määrittelyt tiedostossa config.php). Jos tämä ei onnistu, yhteys katkaistaan ja annetaan virheilmoitus. Jos yhdistys onnistuu, palvelimelta valitaan oikea tietokanta ja yhdistetään siihen.

dbclose.php:

```
<?php  
    mysql_close($conn);  
?>
```

Tällä skriptillä suljetaan tietokantayhteys.

Kuvassa 6 on pätkä etusivun koodista. Kyseisellä pätkällä hoidetaan tapahtumien otsikoiden ja päivämäärän tulostus etusivulle. Ensin otetaan yhteys kantaan ja sen jälkeen tehdään SQL-haku, jossa haetaan kannasta tapahtuman id, nimi, päivä ja kuukausi. Nämä tiedot haetaan taulusta nimeltä happening. Tapahtumat järjestetään etusivulle ensin kuukauden mukaan nousevasti ja sen jälkeen päivän mukaan nousevasti. Nämä tiedot järjestetään taulukkoon, jolloin tapahtuman nimestä tehdään linkki, jota napsauttamalla kannasta haetaan oikean id:n mukaan loputkin tiedot ja avataan ne content-diviin. Päivämäärä tulostetaan muotoon päivä.kuukausi, mutta sitä ei ole linkissä mukana, vaan se vain näkyy etusivulla. Tapahtumat\_katso.php-tiedostoon on koottu kaikki loputkin tiedot tapahtumasta, jotka on syötetty tietokantaan ja ne tulostetaan käyttäjälle. Lisäksi sivulla on Takaisin-linkki, josta käyttäjä pääsee takaisin tapahtumat sivulle, jonka tulostus on sama kuin etusivunkin tapahtumien tulostus, eli nimet ja päivämäärät. Uutiset tulostetaan kokonaisuudessaan etusivulle, eli siihen haetaan vain tietokannasta kaikki tieto ja tulostetaan se. Ainoastaan id jätetään tulostamatta. Samalla tavalla siis kuin tuolla tapahtumat\_katso.php-sivulla on tehty tapahtumien tietojen tulostus.

```

<table>
<?php
    require_once('conf/config.php');
    require_once('conf/dbopen.php');

    $sql="SELECT id, name, day, month
        FROM happening
        ORDER BY month ASC, day ASC";

    $result=mysql_query($sql);

    while($rows=mysql_fetch_array($result)){
?>
<tr>
    <td>
        <a href="#" onclick="getdata('tapahtumat_katso.php?id=
            <? echo $rows['id']; ?>', 'content');" >
            <? echo $rows['name']; ?>
        </a>
    </td>
    <td align="center">
        <? echo $rows['day']; ?>.<? echo $rows['month']; ?>.
    </td>
</tr>

<?php
// Exit looping
}
?>
</table>

```

Kuva 6: Tapahtumien linkit etusivulla

Jäseneksi liittyminen tapahtuu lomakkeelta, jossa on viisi kenttää ja kaikki tiedot ovat pakollisia täyttää. Kentät ovat nimi, sähköpostiosoite, kotiosoite, puhelinnumero ja syntymäaika. Näille kenttien sisällöille tehdään tarkistus ennen tietojen lähettämistä ainejärjestön sähköpostiin. Tiedot tarkistetaan seuraavanlaisella koodilla:

```

$errors=0;
$error="The following errors occurred while processing your form input.<ul>";
pt_register('POST','Nimi');
pt_register('POST','Sähköposti');
pt_register('POST','Kotiosoite');

```

```

pt_register('POST','Puhelinnumero');
pt_register('POST','Syntymaika');
if($Nimi=="" || $Sahkoposti=="" || $Kotiosoite=="" || $Puhelinnumero=="" || $Syntymaika=="") {
$errors=1;
$error.="<li>You did not enter one or more of the required fields. Please go back and try again.";
}
if(!eregi("^([a-z0-9]+([\_\.\-][a-z0-9]+)*" . "@" . "([a-z0-9]+([\_\.\-][a-z0-9]+)*)+" . "\.[a-z]{2,}" . "$",$Sahkoposti)) {
$error.="<li>Invalid email address entered";
$errors=1;
}
if($errors==1) echo $error;

```

Ensin varmistetaan if-lausekkeella, että kaikissa kentissä on tekstiä ja tehdään myös tarkistus, että sähköpostiosoite on oikeanlainen. Sähköpostiosoitteen oikeanlaisen merkinnän tarkistus tapahtuu siten, että if-lausekkeen sisään on määritetty kirjaimet ja numerot, joita kentässä voi esiintyä. ”Ääkköset” on suljettu pois. Tämän jälkeen pitää tulla @-merkki ja taas samat kirjain- ja numeromäärytykset. Väliin tulee piste ja pisteen jälkeen enintään kaksi kirjainta. Ainejärjestön jäseneksi voivat liittyä vain kyseisen aineen opiskelijat ja heidän sähköpostiosoitteensa ovat muotoa etunimi.sukunimi@koulu.fi. Kaikilla on .fi-pääte, joten tarkistuksessa viimeisen pisteen jälkeen riittää kaksi kirjainta. Jos jommassakummassa esiintyy virheitä tai puutteita, käyttäjälle tulee virheviesti. Kun kaikki on kunnossa, käyttäjä ohjautuu kiitos-sivulle, jossa lukevat hänen syöttämänsä tiedot ja kiitos jäseneksi liittymisestä.

## 4.6 Ylläpito

Ylläpidon sisältö on selkeä. Siellä pystyy lisäämään, muokkaamaan ja poistamaan niin uutisia kuin tapahtumiakin. Sisään kirjautuessa käyttäjälle tulee ohje, jossa kerrotaan, mitä mistäkin linkistä tapahtuu. Tapahtumat-linkin alta löytyy kolme uutta linkkiä; Lisää-, Muokkaa- ja Poista-linkit.



Tapahtumaa lisättäessä kuvassa 7 olevalle lomakkeelle kirjoitetaan kenttiin tiedot tapahtumasta. Koodissa on määritelty, että otsikko, päivä ja kuukausi ovat pakollisia kenttiä, joten niille tehdään myös tarkistus, että kyseiset kohdat on varmasti täytetty. Käyttäjälle asia on kerrottu pienessä info-tekstissä, joka lukee sivun yläreunassa. Tallenna-painikkeesta tiedot tallentuvat kantaan ja nimi sekä päivämäärä tulostuvat varsinaiselle julkiselle etusivulle. Tyhjennä-painikkeesta lomakkeen kentät tyhjäntyvät ja homman voi aloittaa puhtaalta pöydältä.

## Lisää uusi tapahtuma

Tapahtuma:

Päivä:

Kuukausi:

Vuosi:

Kellonaika:

Paikka:

Kuvaus:

Kuva 7: Tapahtuman lisäämislomake

Tapahtumien muokkaus (kuva 8) tapahtuu siten, että käyttäjälle tulostetaan kaikkien kantaan lisättyjen tapahtumien otsikot, jotka kaikki ovat linkkejä. Haluttua otsikkoa napsautettaessa, tiedot tulostuvat samanlaiselle lomakkeelle (kuva 9) kuin kuvassa 7, mutta siinä on jo tosiaan käyttäjän aiemmin lisäämät tiedot. Käyttäjä voi muuttaa mitä tahansa haluamansa kentän tietoa, ja painettuaan Tallenna-painiketta, ne päivitetään kantaan UPDATE-komennolla.

## Tapahtumien muokkaus

Valitse tapahtuma:

[Pupujen tutustumisbileet - Muokkaa](#)

[Harkkasitsit - Muokkaa](#)

[Yhteistyökumppaneitten luentopäivä - Muokkaa](#)

Kuva 8: Tapahtumien muokkauksen linkkilista

## Tapahtumien muokkaus

Valitse tapahtuma:

Nimi:

Päivä:

Kuukausi:

Vuosi:

Aika:

Paikka:

```
<b>Pupujen tutustumisbileet</b>
<p>
Uusien opiskelijoiden kouluvuosi lähtee käyntiin
perinteisillä tutustumisbileillä.</p>
```

Kuvaus:

Kuva 9: Muokkaus-lomake

Tapahtumat poistetaan (kuva 10) siten, että käyttäjälle tulostuu alasvetovalikkoon lista, jossa lukee kaikkien lisättyjen tapahtumien otsikot. Listalta valitaan yksi otsikko, eli se tapahtuma, joka halutaan tietysti poistaa. Sen jälkeen painetaan Poista-painiketta ja tapahtuma poistetaan DELETE-komennolla.

## Tapahtuman poisto

Tapahtumat:  
 ▾

Kuva 10: Tapahtuman poistaminen

Ylläpidon salaamiseen muilta käyttäjiltä oli oikeastaan kaksi vaihtoehtoa. Ensimmäinen ajatus oli toteuttaa sessionhallinnan ja tietokannan avulla kunnan kirjautuminen sivuille. Kuitenkin tarve salasanojen määrälle oli todella pieni, se oli nimittäin tasan yksi. Sivuja päivitetään sen verran harvoin ja aina maksimissaan yhden henkilön toimesta kerralla, että tarvetta millekään massiiviselle salasanajärjestelylle ei lopulta ollut. Päätin siis toteuttaa salaamisen htaccessin avulla. Alla on esimerkki .htaccess-tiedoston sisällöstä.

```
AuthType Basic
AuthName "Restricted Files"
AuthUserFile /usr/local/apache/passwd/passwords
Require user rbowen
```

AuthType kertoo palvelimelle menetelmän, jolla käyttäjä tunnistetaan. Vaihtoehtoina ovat ”Basic” ja ”Digest”. Jälkimmäinen on turvallisempi, koska silloin salasana lähetetään salattuna eli kryptattuna. AuthName kertoo sen osion nimen, joka on salattu. AuthUserFile on polku salasanatiedostoon ja Require kertoo, kuka käyttäjistä pääsee tälle alueelle. (Apache Software Foundation, 2009)

Ainejärjestön hallituksen jäsenille ei ollut erillistä käyttökoulutusta ylläpitoon, sillä sisään kirjautumisen jälkeen käyttäjälle tulee ohjeistus ylläpidon käytöstä. Ylläpitoon kirjaututaan oman linkin kautta, jota ei ole varsinaisella sivustolla näkyvissä. En laittanut ylläpito-osioon linkkiä, koska se vähentää normaalikäyttäjien yrityksiä päästä kirjautumaan ylläpitoon, kun he eivät tiedä sen olemassaolosta. Ylläpitoa testasin ensin TAMKin palvelimella lisäämällä ja poistamalla sekä muokkaamalla useita uutisia sekä tapahtumia. Testasin jonkin verran vielä uudestaan ainejärjestön omalla palvelimella, jotta ylläpidon toimivuus varmistui.

## 5 Lopputulos

Internetsivujen tekeminen ja opinnäytetyöraportin kirjoittaminen tuntuivat todella pitkältä prosessilta, mutta toisaalta opinnäytetyön tekemiseen varattu tuntimääräkin oli melko suuri. Kaikki ei sujunut ihan suunnitelmien mukaan, mutta harvemmin niin silti yleensä meneekään.

### 5.1 Ongelmat

Sovelluksen kanssa oli loppujen lopuksi paljon enemmän ongelmia, kuin mihin olin aluksi varautunut. Oli yllättävää huomata, miten paljon tekniikoista oli unohtunut, kun niitä ei ollut vähään aikaan käyttänyt. Tyylitiedostojen kanssa esiintyi huomattava määrä ongelmia eikä mikään toiminut suunnitelmien mukaan. PHP-koodin kanssa esiintyi myös jonkin verran ongelmia, koska kyseisestä kielestä oli jo lähes kaikki unohtunut. Ainoa, mistä koodaamisen aloittaessani muistin jotain, oli XHTML ja MySQL. Onneksi oli sentään jotain, mutta vaikeaa oli silti alussa. Lopputulokseen en kuitenkaan ole aivan täysin tyytyväinen, koska aika loppui kesken sekä yhteisymmärrys ja materiaalin toimitus ainejärjestön hallituksen kanssa. Huomattavan paljon asioita hankaloitti myös sekin seikka, että kävin kokopäivätoissa samanaikaisesti jo tammikuusta asti. Viikolla ei juuri ollut aikaa paneutua opinnäytetyöhön ja myönnettäköön, että viikonloppuisin sitä kiinnostustakaan ei niin kovin paljon ollut.

Muita ongelmia aiheutti muun muassa se, että kolme vuotta vanha kannettava tietokoneeni oli vihdoinkin tullut tiensä päähän, joten sen kanssa työskentely osoittautui mahdottomaksi. Sain lainaksi pöytäkoneen, mutta sitten eteen tuli uusi ongelma; internet. Olin juuri muuttanut, ja internetyhteyteni ei ollutkaan auennut sovitulla aikataululla, koska palveluntarjoajan päässä ilmeni yllättäviä ongelmia. Lopulta kävi niin, että olin kuukauden ilman internetyhteyttä, joka vähensi jälleen mahdollisuuksiani tehdä lopputyötäni. On ollut mielenkiintoista huomata, miten moni seikka lopulta tällaiseenkin asiaan vaikuttaa, ja miten kaikkea ei vain mitenkään pysty ottamaan huomioon aikatauluja tehdessä.

## 5.2 Onnistumiset

Sivusto ajaa mielestäni asiansa hyvin. Se kertoo, mistä ainejärjestöstä on kyse, ja mikä sivuston tarkoitus on. Navigointi on helppoa ja kaikki käyttäjälle tarpeellinen on otettu huomioon. Vanha ulkoasu oli todella raskas, ja uusi on nyt huomattavasti pirteämpi. Sivusto on helppokäyttöisempi valikkorakenteensa ansiosta ja etusivulta löytyy heti kaikki oleellinen. Ylläpito oli ehkä hieman alkeellinen, mutta joka tapauksessa siitä tuli parempi kuin vanhasta, jossa ylläpitoa ei ollut ollenkaan. Sinänsä ei ollut kovin suurta rimaa ylitettävänä. Olen silti ylpeä saavutuksestani. Vaikka ylläpidosta tulikin helppo ja yksinkertainen, kukaan ei ole sinne vielääkään lisännyt mitään. Uutisia ja tapahtumia ei ole yhtään. Tapahtumia oli tasan kaksi kappaletta, ne jotka itse lisäsin testausvaiheessa. Nämä olivat silti ihan virallisia tapahtumia, koska testasin silloin heidän palvelimellaan sivujen toimintaa. TAMKin palvelimella testatessani kehittelin päästäni tapahtumia, mutta onneksi kanta tuli luotua uudestaan heidän palvelimelleen, jotta sinne ei jäänyt mitään epämääräistä.

Tätä sivustoa tehdessäni opin todella paljon uutta ja sain kerrattua paljon vanhaa. Nyt pitäisi vain ylläpitää taitoja ja koodailla vapaa-ajallakin, ettei kaikki pääsisi taas ajan kuluessa unohtumaan. Kärsivällisyyskin jossain määrin kasvoi, kun kaiken kanssa piti vain vääntää ja vääntää. Toki oli hetkiä, jolloin luovuttaminen ei ollut kaukana, mutta nyt kun on valmista, niin on todella hyvä mieli.

## 6 Yhteenveto

Koko prosessiin meni yllättävän kauan aikaa, vaikka sitä ei alussa ihan täysin ajatellut. Sovelluksen tekeminen oli aikaa vievää, kuten myös tämän raportin kirjoittaminen. Oli vaikeaa alussa miettiä, mistä tämän raportin oikeastaan edes tekisi, varsinkin, kun ainejärjestön hallituksen kanssa alkoi tulla ongelmia ja aika loppua. Sain kuin sainkin sitten molemmat lopulta tehtyä.

Sovelluksen teossa sivuista ei tullut ulkonäöllisesti juuri sitä, mitä olisin halunnut. Sivusta olisi voinut tehdä tyylikkäämmät ainakin jossain määrin, mutta sivut tehtiin kuitenkin toimeksiantajan toiveiden mukaan. Alussa, kun opinnäytetyötä suunniteltiin, ei ainejärjestön hallituksella tosin ollut mielipiteitä, ja sain vapaat kädet. Kun jotain oli valmiina jo, alkoikin mielipiteitä tulla ja muutospyyntöjä sadella. Opin tässä samalla myös paljon ainejärjestöjen toiminnasta, tai ainakin tämän yhden, ja lisäksi, siitä miten toimia, kun on toimeksiantaja, jonka kanssa asiat eivät mene alkuperäisten suunnitelmien mukaan.

Asioiden kertaaminen ja uudelleen opetteleminen vei todella paljon aikaa. Edellisestä kurssista TAMKissa, jossa näitä aiheita käsiteltiin, oli jo jonkin verran aikaa, ja yllättävän paljon oli päässyt unohtumaan, kun tekniikoita ei ollut tullut käytettyä. Nyt ne tulivat kaikki kertaheitolla kerrattua, eivätkä toivottavasti heti unohdu. Tarkoituksena olisi nyt ylläpitää taitoja vähän paremmin, jotta seuraavan kerran, kun suostun johonkin tämän tyyliin projektiin, ei aikaa asioiden uudelleen opetteluun mene ihan näin paljon, kuin nyt.

## Lähdeluettelo

- Apache Software Foundation 2009. Authentication, Authorization and Access Control. [online] [viitattu 29.5.2009]  
<http://httpd.apache.org/docs/2.0/howto/auth.html>
- Atkinson, Leon 1999. Core PHP Programming: using PHP to Build Dynamic Web Sites. Upper Saddle River: Prentice Hall PTR.
- Boumphrey, Frank, Greer, Cassandra, Raggett, Dave, Raggett, Jenny, Schnitzenbaumer, Sebastian & Wugofski, Ted 2001. XHTML Ohjelmoijan käsikirja (suomennos). Kauko Kolehmainen. IT Press.
- Finlex 2008. Laki ammatillisesta koulutuksesta 21.8.1998/630 [online] [viitattu 4.5.2009]  
[http://www.finlex.fi/fi/laki/ajantasa/1998/19980630?search\[type\]=pika&search\[pika\]=ammatillisesta%20koulutuksesta](http://www.finlex.fi/fi/laki/ajantasa/1998/19980630?search[type]=pika&search[pika]=ammatillisesta%20koulutuksesta)
- Helsingin yliopiston oppilaskunta 2008. HYY - Jäsenelle / Järjestötoiminta / Tiedekunta ja ainejärjestöt [online][viitattu 4.5.2009]  
<http://www.hyy.helsinki.fi/suomi/101/>
- Korpela, Jukka K. 2008. CSS verkkosivujen muotoilussa. Porvoo: WS Bookwell.
- Laine, Hanna & Vierelä, Mika 1998. Opiskelijayhdistystoiminnan opas. Helsinki: Haka paino Oy.
- Meloni, Julie C. 2003. MySQL Trainer Kit (suomennos). Riitta Santala-Köykkä. IT Press.
- Wikipedia 2009a  
[online] [viitattu 4.5.2009]  
<http://fi.wikipedia.org/wiki/Ylioppilaskunta>
- Wikipedia 2009b  
[online] [viitattu 27.4.2009]  
<http://fi.wikipedia.org/wiki/Ainej%C3%A4rjest%C3%B6>
- Wikipedia 2009c  
[online] [viitattu 27.4.2009]  
[http://fi.wikipedia.org/wiki/Kilta\\_\(ainej%C3%A4rjest%C3%B6\)](http://fi.wikipedia.org/wiki/Kilta_(ainej%C3%A4rjest%C3%B6))
- Zandstra, Matt 2001. PHP Trainer Kit (suomennos). Tmi S. Kolehmainen. IT Press.

## Liitteet

### Liite 1: Sanasto

Apache	Avoimeen lähdekoodiin perustuva HTTP-palvelinohjelma. Apache Software Foundationin tunnetuin tuote.
Div	Määrittää tietyn osan tai kohdan HTML-koodissa. Käytetään usein ryhmittämään kappale-elementtejä, jotta ne voidaan muotoilla tyyleillä.
Forum	Keskustelupalsta internetissä, jossa voi luoda omia keskusteluja tai kommentoida toisten aiheita.
htaccess	Hakemistokohtainen asetustiedosto. Tiedostolla voidaan asettaa omia virhesivuja, suojata sivuja salasanalla, uudelleenohjata käyttäjiä tai estää vaikka kuvien linkitys ulkopuolisilta sivustoilta.
HTML	Hypertext Markup Language. Internetsivujen rakennuskieli, joka on avoimesti standardoitu.
Nimiavaruus	XML-nimiavaruus on yksikäsitteinen URL-merkkijono. Käsitteellisesti nimiavaruus niputtaa eri sovellusten sanastot niin sanotusti omiin lokeroihinsa, minkä avulla eri sovellusten samannimiset sanat eivät mene sekaisin.
Oracle	Tunnettu relaatiotietokanta sekä sen hallintaohjelmistoa valmistavan yhtiön nimi. Normaaliassa kielenkäytössä samaa nimeä käytetään sekä kannasta että ohjelmistosta.
PhotoShop	Adobe Systemsin kehittämä kuvankäsittelyohjelma.



PHPMysqlAdmin	Selaimen kautta käytettävä MySQL-tietokannan hallintatyökalu.
Skripti	Komentosarja.
Tagi	HTML-komento jonka avulla ohjataan sitä, miten tieto internet-sivulle tulostuu.
XML	Extensible Markup Language. Rakenteellinen kuvauskieli, joka auttaa jäsentämään laajoja tietomassoja selkeämmin.

## Liite 2: Valikon XHTML- ja CSS-koodit

### XHTML

```

<table id="head">
<tr>
<td>
<div id="header">

</div>
</td>
<tr>
<td>
<ul id="menu">
<li><a href="index.php"><b>Etusivu</b></a></li>
<li class="current"><a class="sub"><b>Opiskelu</b>
<!--[if gte IE 7]><!--</a><!--<![endif]-->
<!--[if lte IE 6]><table><tr><td><!--[endif]-->
<ul>
<li><a href="http://www.tse.fi/FI/yksikot/erillislaitokset/pori/Pages/default.aspx"
href="#url">Koulu</a></li>
<li><a class="fly" href="#url">Kurssit ja tentit
<!--[if gte IE 7]><!--</a><!--<![endif]-->
<!--[if lte IE 6]><table><tr><td><!--[endif]-->
<ul>
<li><a href="http://www.tuky.fi/tenttiarkisto/porin-yksikon-tentit/"
href="#url">Tenttiarkisto</a></li>
</ul>
<!--[if lte IE 6]></td></tr></table></a><!--[endif]-->
</li>
<li><a
href="http://www.tut.fi/public/index.cfm?MainSel=5661&Sel=5661&Show=4590&Site
id=89" href="#url">Kirjasto</a></li>
<li><a href="http://mail.tse.fi/" href="#url">Sähköposti</a></li>
<li><a href="https://oodi-www.it.helsinki.fi/tukkk/" href="#url">Weboodi</a></li>
<li><a href="https://moodle.utu.fi/" href="#url">Moodle</a></li>
<li><a href="http://www.tuky.fi/" href="#url">TuKY</a></li>
<li><a class="fly" href="#url">SEFE
<!--[if gte IE 7]><!--</a><!--<![endif]-->
<!--[if lte IE 6]><table><tr><td><!--[endif]-->
<ul>
<li><a href="#" onclick="getdata('sefe.php','content');" href="#url">SEFE</a></li>
<li><a href="#" onclick="getdata('etupa.php','content');" href="#url">Edut ja palve-
lut</a></li>
<li><a href="#" onclick="getdata('faq.php','content');" href="#url">FAQ</a></li>
<li><a href="#" onclick="getdata('valmistuminen.php','content');"
href="#url">Valmistuminen</a></li>
</ul>
<!--[if lte IE 6]></td></tr></table></a><!--[endif]-->
</li>

```

```

</ul>
<!--[if lte IE 6]></td></tr></table></a><![endif]-->
</li>
<li class="current"><a class="sub"><b>Elämä</b>
<!--[if gte IE 7]><!--></a><!--<![endif]-->
<!--[if lte IE 6]><table><tr><td><![endif]-->
<ul>
<li><a href="#" onclick="getdata('tapahtuu.php','content');"
href="#url">Tapahtumat</a></li>
<li><a href="http://www.tuky.fi/palvelut/kyliste" href="#url">Kyliste</a></li>
<li><a href="http://porinkylterit.1g.fi" href="#url">Galleria</a></li>
<li><a href="http://www.pori.tut.fi/ajankohtaista/sofia/" href="#url">Sofia</a></li>
<li><a href="#" onclick="getdata('linkit.php','content');" href="#url">Linkit</a></li>
</ul>
<!--[if lte IE 6]></td></tr></table></a><![endif]-->
</li>
<li class="current"><a class="sub"><b>Yhdistys</b>
<!--[if gte IE 7]><!--></a><!--<![endif]-->
<!--[if lte IE 6]><table><tr><td><![endif]-->
<ul>
<li><a href="#" onclick="getdata('porky.php','content');" href="#url">PorKy
Ry</a></li>
<li><a href="#" onclick="getdata('hallitus.php','content');"
href="#url">Hallitus</a></li>
<li><a href="#" onclick="getdata('poytakirjat.php','content');" href="#url">Pöytäkirjat
yms.</a></li>
<li><a href="#" onclick="getdata('liittyminen.php','content');" href="#url">Liity jäse-
neksi</a></li>
<li><a href="http://www.porinkylterit.fi/hallitus/index.html" href="#url">Hallitus-
sivut</a></li>
</ul>
<!--[if lte IE 6]></td></tr></table></a><![endif]-->
</li>
<li id="contact"><a href="#" on-
click="getdata('yhteistyö.php','content');"><b>Yhteistyö</b></a></li>
<li id="contact"><a href="#" on-
click="getdata('yhteystiedot.php','content');"><b>Yhteystiedot</b></a></li>
<li id="contact"><a
href="http://www.porinkylterit.fi/vuosijuhlat/"><b>Vuosijuhlat</b></a></li>
</ul>
</td>
</tr>
</table>

```

## CSS

```
/* Valikon säätö alkaa */
#menu {
padding-top: 10px;
margin:0;
list-style:none;
height:40px;
position:relative;
left: 70px;
z-index:500;
font-family: verdana, arial, sans-serif;
}

#menu li {
float:left;
margin-right: 1px;
}

#menu li a {
display:block;
float:left;
height:40px;
line-height:40px;
background:#FFE3E3;
color:#656565;
text-decoration:none;
font-size:11px;
font-weight:bold;
padding:0 30px 0 20px;
}

#menu table {
border-collapse:collapse;
width:0;
height:0;
position:absolute;
top:0;
left:0;
}

#menu li a:hover {
z-index:200;
position:relative;
color:#656565;
background-color:#FFC0CB;
}

#menu li:hover {
position:relative;
```

```
z-index:200;
}

#menu li:hover > a {
color:#656565;
}

#menu li:hover > a.sub {
color:#656565;
background-color:#FFC0CB;
}

#menu li a.sub {
background: #FFE3E3;
}

#menu li.current a.sub {
color:#656565;
}

#menu :hover ul {
left:0;
top:40px;
width:120px;
background:#444;
}

#menu ul,
#menu :hover ul ul {
position:absolute;
left:-9999px;
top:-9999px;
width:0;
height:0;
margin:0;
padding:0;
list-style:none;
}

#menu :hover ul :hover ul {
left: 120px;
top: -1px;
background: #FFE3E3;
white-space:nowrap;
width: 120px;
z-index: 200;
height: auto;
}
```

```
#menu :hover ul li {
  margin:0;
  border-top:1px solid #666;}
#menu :hover ul li a {
  width: 120px;
  padding: 0;
  text-indent: 10px;
  background: #FFE3E3;
  color: #656565;
  height: 30px;
  line-height: 30px;
}

#menu :hover ul li a.fly {
  background:#FFE3E3
}

#menu :hover ul :hover {
  background-color:#FFC0CB;
  color:#656565;
}

#menu :hover ul :hover a.fly {
  background-color:#FFC0CB;
  color:#656565;
}

#menu :hover ul li.currentsub a {
  color:#656565;
}

#menu :hover ul li.currentsub a.fly {
  color:#656565;
}

#menu :hover ul :hover ul li a {
  width: 120px;
  padding: 0;
  text-indent: 10px;
  background: #FFE3E3;
  color: #656565;
}

#menu :hover ul :hover ul :hover {
  background-color:#FFC0CB;
  color:#656565;
}

/* Valikon säätö loppuu */
```