

Tampereen ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma
Digimedia
Riku Kestilä

Opinnäytetyö

Akateemisen konferenssin WWW-sivuston toteuttaminen PHP- ja PostgreSQL-tekniikoilla

Työn ohjaaja lehtori Petri Heliniemi

Työn tilaaja PRIMA2007 ja ITE 2007 konferenssien organisointikomitea,
lehtori Esa Kujansuu, professori Rakakoti Sadananda

Tampere 05/2009

Tekijä	Riku Kestilä
Työ	Akateemisen konferenssin WWW-sivuston toteuttaminen PHP- ja PostgreSQL-tekniikoilla
Sivumäärä	42
Valmistumisaika	toukokuu 2009
Työn ohjaaja	Lehtori Petri Heliniemi
Työn tilaaja	PRIMA2007 ja ITE2007 konferenssien organisointikomiteat, lehtori Esa Kujansuu, TAMK ja professori Ramakoti Sadananda, Rangsitin yliopisto

TIIVISTELMÄ

Marraskuussa 2007 Thaimaan Bangkokissa järjestettiin kaksi IT-alan kansainvälistä akateemista konferenssia, ITE2007 ja PRIMA2007. Niitä edeltäneen puolen vuoden aikana suoritin työharjoitteluni paikan päällä molempien tapahtumien järjestelyjä ja valmisteluja tehden. Yksi harjoittelujaksoni tärkeimmistä tehtävistä oli kehittää konferensseille monipuoliset web-sivut, jotka toimivat sekä tapahtumien käyntikortteina että tärkeinä työkaluina kaikille konferenssien organisaatioihin kuuluville.

Opinnäytetyö käsittelee pääsääntöisesti niitä toimintoja, joiden toteuttaminen WWW-sivuille tiettyihin aikarajoihin mennessä oli molempien konferenssien onnistumisen kannalta hyvin tärkeää. Näitä toimintoja olivat mm. sähköpostilista konferensseista tiedottamiselle, sisällönhallintajärjestelmä sivujen sisällön vaivattomaan muokkaamiseen ja luottokorttimaksumahdollisuus osallistumismaksujen maksamiselle.

Työn tuloksena syntyi kaksi WWW-sivua www.ite2007.org ja www.prima2007.org. Molemmat sivut toteutettiin lähes samalla tavalla, käyttämällä PHP4-ohjelmointikieltä tiedon käsittelemiseen palvelimella ja selaimessa sekä PostgreSQL tietokannanhallintajärjestelmää varastoimaan tietoa relaatiotietokannan tauluihin.

Koska minulla ei juuri ollut aikaisempaa PHP- ja tietokantakokemusta, täytyi minun opiskella paljon asioita itsenäisesti. Työharjoittelujakso oli erittäin opettavainen ja olen enemmän kuin tyytyväinen lopputulokseen. Sivustojen kehittäminen itsenäisesti alusta loppuun antoi minulle arvokasta kokemusta dynaamisten WWW-sivujen tekemisestä ja tietokantaohjelmoinnista, samoin kuin työharjoittelujakso antoi arvokasta näkemystä vieraan maan tapoihin ja kulttuuriin.

Writer	Riku Kestilä
Thesis	Developing Web Pages for an Academical Conference with PHP and PostgreSQL
Pages	42
Graduation time	May 2009
Thesis Supervisor	Lecturer Petri Heliniemi
Co-operating Company	PRIMA2007 and ITE2007 Conference Organization Committees, Lecturer Esa Kujansuu, TAMK and Professor Ramakoti Sadananda, University of Rangsit

Abstract

In November 2007, two international academic conferences, ITE2007 and PRIMA2007 were held in Bangkok, Thailand. During the six months prior to the conferences I did my job studies on location in Thailand, handling the arrangements and preparations of both conferences. During that period, one of my most important tasks was to develop fully functional web pages for both conferences, to act as calling cards and useful tools for all the people involved.

This thesis mainly focuses on those functionalities that were necessary to develop and implement before certain deadlines to make the conferences successful. Some of these functionalities were a mailing list application, for promoting the conferences, a content management system, for effortless updating of the content and the possibility to pay the entry fees in advance with a credit card.

As a result, two web pages, www.ite2007.org and www.prima2007.org were born. Both pages were produced by almost identical means, using PHP4-programming language for handling data at the server and on the browser and PostgreSQL database management system for storing the data in relational database tables.

Because I had very little prior experience and knowledge of PHP and databases, I had to study many things independently. Because of that, the job study period was a really good learning experience for me and I am more than happy with the results. Developing the pages from the start to the finish without much help from anyone else gave me valuable experience and knowledge of dynamic web page programming and databases, as well as the time spent abroad gave me valuable insight into the customs and culture of foreign country.

Keywords: conference, WWW-pages, PostgreSQL, PHP

Sisällysluettelo

Sanasto.....	5
1 Johdanto.....	7
2 Toimeksiantajat	9
2.1 PRIMA2007	9
2.2 ITE2007	9
3 Käytetyt menetelmät.....	11
4 Sivustojen toteuttaminen	13
4.1 Sivustoille asetetut vaatimukset.....	13
4.1.1 Toiminnalliset vaatimukset	13
4.1.2 Ei-toiminnalliset vaatimukset.....	14
5 Toteutetut toiminnot	16
5.1 Tietokantayhteys	16
5.2 Käyttäjähallinta	18
5.3 Uloskirjautuminen.....	23
5.4 Sähköpostilista	23
5.5 Content Management System, sisällönhallintajärjestelmä	26
5.6 Konferenssiin ilmoittautuminen ja rekisteröityminen	30
5.6.1 Luottokorttimaksu	32
5.7 Ilmoittautumisten ja osallistumismaksujen seuraaminen.....	33
5.8 Kirjanpitytyökalu	35
Lähteet	40
Liitteet.....	41

Sanasto

PHP - Hypertext Parser Protocol on avoimen lähdekoodin ohjelmointikieli, jonka avulla voidaan toteuttaa WWW-sivuille toiminnallisuuksia ja dynaamista sisältöä. Skriptit ajetaan palvelimella, jossa PHP-ohjelmisto tulkkaa sen ja palauttaa lopputuloksen selaimelle.

URL - uniform resource locator on WWW-sivun osoite, joka näkyy selaimen osoitekentässä.

GET method - metodi jolla HTML-lomakkeeseen kirjoitettu dataa voidaan lähettää eteenpäin lisäämällä se lomakkeen prosessoivaa skriptiä kutsuvaan URL:ään. Sen hyötyinä on yksinkertaisuus, haittapuolena rajoitettu määrä dataa voidaan lähettää ja lähetetty data näkyy selaimessa, mikä voi olla tietoturvariski.

POST method - metodi, jolla data lähetetään erillisenä pakettina lomakkeen prosessoivan skriptin kanssa. Sen hyötyjä ovat rajoittamaton datansiirtomahdollisuus ja tietoturva. Haittapuolena siinä on sen hitaus GET-metodiin verrattuna.

Sessio - joukko tilapäisiä muuttujia, jotka ovat olemassa, kunnes selain suljetaan tai sessio tuhoetaan.

SQL - Structured Query Language on relaatiotietokantojen käsittelyyn tarkoitettu kysely-kieli.

Relaatiotietokanta - tietokantamalli, jossa tieto tallennetaan toisiinsa viittaaviin tauluihin.

PostgreSQL - avoimen lähdekoodin tietokannanhoidon järjestelmä.

SSH - Secure Shell Protocol on suojattu etäyhteys, jolla voidaan ajaa komentokäyttöliittymästä esim. SQL-kyselyjä.

FTP - File Transfer Protocol on suojattu tiedonsiirtoyhteys, jolla voidaan siirtää esimerkiksi html-dokumentti kehittäjän koneelta ylläpitävälle palvelimelle.

1 Johdanto

Useimmat asiakaslähtöiset ja toimeksiantona tehdyt WWW-sivut edellyttävät toimintoja, joihin XHTML ei enää kertakaikkiaan riitä. Yhä useammin käytetään niin kutsuttuja palvelinpuolen ohjelmointikieliä, kuten esim. PHP:tä yhdessä jonkin tietokannan, kuten MySQL tai PostgreSQL, kanssa. Tällaisia toimintoja ovat esimerkiksi käyttäjähallinta, sivuille rekisteröityminen ja luottokorttimaksumahdollisuus, joiden toteuttaminen vaatii käyttäjän antaman datan muokkaamista, käsittelemistä ja varastointia.

Vuoden 2007 marraskuussa järjestettiin Thaimaan Bangkokissa lähes peräkkäisinä päivinä kaksi informaatioteknologia-alan akateemista konferenssia, ITE2007 ja PRIMA2007. PRIMA2007 oli järjestyksessään jo kymmenes, kun taas ITE2007 järjestettiin ensimmäistä kertaa. Minulle ja opiskelijatoverilleni Jaakko Lautarinteelle tarjoutui mahdollisuus päästä suorittamaan opintoihimme kuuluva työharjoittelujakso Thaimaassa molempien tapahtumien järjestelyjä ja valmisteluja tehden.

Yksi harjoittelujaksomme tärkeimmistä tehtävistä oli kehittää konferensseille monipuoliset WWW-sivut, jotka toimisivat sekä tapahtumien käyntikortteina, että tärkeinä työkaluina kaikille konferenssien organisaatioihin kuuluville. Erityisen tärkeää oli, että sivuja olisi helppo käyttää ja muokata myös sellaisten henkilöiden, joilla ei ole kokemusta ohjelmoinnista tai WWW-sivunkuvauskielistä.

Lukuun ottamatta joitakin toimintoja, kuten sisällönhallintajärjestelmää, sähköpostilistaa ja konferenssiin ilmoittautumista, tarkkoja määrittelyjä sivujen toiminnallisuudelle ei aloittaessamme ollut. Käytännössä saimme siis vapaat kädet toteuttaa sivut haluamallamme tavalla, huomioiden kuitenkin tekijänoikeudet, mahdolliset kehityksen aikana ilmenevät lisävaatimukset, sekä tekniset rajoitukset. Rajoituksista merkittävimmät olivat Tampereen ammattikorkeakoulun tyh.tpu.fi-palvelimen käyttämä PHP versio 4 sekä tietokannanhallintajärjestelmä PostgreSQL.

Hyvin pian aloitettuamme työskentelyn projektin parissa yhdessä opiskelijatoverini Jaakko Lautarinteen kanssa päädyimme työnjakoon, jossa Jaakko vastasi sivustojen graafisen ilmeen ja rakenteen suunnittelusta ja toteutuksesta, ja minä kaikesta tietokanta- ja palvelinpuolen ohjelmoinnista. Työn aloittaminen oli erittäin haasteellista, johtuen vähäisestä ohjelmointikokemuksestani.

Tavoitteiden saavuttamiseksi oli tärkeää opiskella itsenäisesti tarvittavia tietoja saatavilla olleista kirjoista. Käytössäni ei ollut kovin paljoa opiskelumateriaalia, koska Thaimaassa ei ollut saatavilla kovinkaan paljoa englanninkielistä ohjelmointikirjallisuutta. Kirjat, joita käytin apunani työn suunnittelussa ja toteuttamisessa eivät sellaisenaan sopineet suoraan hyödynnettäviksi, koska ne olivat osittain vanhentuneita ja käsittelivät MySQL-tietokantaa, käytössäni olleen PostgreSQL-tietokannan sijaan. Hyödyllisin yksittäinen lähde oli. *Migrating from MySQL to PostgreSQL in PHP* (Zervaas 2005), jonka avulla pystyin opettelemaan asiat ensiksi MySQL-tietokantaa käyttäen ja sitten PostgreSQL-kantaan kääntäen. Vaikka lähdemateriaalin hankkimiseen ennen Thaimaahan lähtöä ei ollut paljoa aikaa, eikä lentokoneeseen suurta määrää kirjoja kannattanut ottaakaan, oli käytössäni ollut materiaali hyödyllistä ja itseopiskelumateriaalina asiansa ajavaa. Työharjoittelujaksoon kuului myös jonkin verran käytännön valmisteluja, joiden kattava käsittely ei ole oleellista tämän raportin puitteissa.

Puolen vuoden työn tuloksena syntyivät WWW-sivut www.ite2007.org ja www.prima2007.org. Keskityn opinnäytetyön kirjallisessa raportissa kertomaan omasta vastuualueestani projektissa ja kuvaamaan niitä tekniikoita, työvälineitä ja toimintoja, joita tarvittiin projektin lopputuloksena syntyneiden sivustojen toteuttamiseen.

2 Toimeksiantajat

Työharjoittelupaikkani ja tämä opinnäytteen toimeksiantajina toimivat kaksi eri akateemista konferenssia organisaatioineen. Seuraavassa on lyhyt esittely molemmista.

2.1 PRIMA2007

PRIMA (Pacific Rim International Workshop on Multiagents) on Tyynenmeren alueella vuotuisasti järjestettävä konferenssi ja työpaja, jonka aiheita ovat pääasiassa agentti- ja multiagenttijärjestelmät. Vuoden 2007 PRIMA oli järjestyksessään jo kymmenes, ja pyrki kokoamaan yhteen arvostetuimmat ja osaavimmat tutkijat ympäri maailman esittelemään tutkimustöitään agenttijärjestelmiin liittyvistä aiheista. Konferenssin päätekijänä toimi erittäin arvostettu intialainen tietojenkäsittelytieteen professori Ramakoti Sadananda, joka työskenteli konferenssit isännöinneellä Rangsitin yliopistolla tutkimusosaston johtajana.

2.2 ITE2007

Ite (IT towards empowerment) pyrkii saattamaan yhteen tutkijoita, ammattilaisia ja osaajia eri IT-alan osa-alueilta keskustelemaan siitä, miten nykyaikaisen informaatioteknologian hyödyt saataisiin mahdollisimman laajasti käyttöön ilman, että esimerkiksi eri elintasoisten tai -ikäisten ihmisten välille syntyy eriarvoisuutta (Sadananda & Kujansuu 2007). Käytännössä tämä tarkoittaa mm. edullisten tietokoneiden saatavuuden parantamista kehitysmaihin, ja ohjelmointi- sekä muiden tietoteknisten taitojen kehittämistä ikään, sukupuoleen tai sosiaaliseen statukseen katsomatta. Kantavina teemoina vuonna 2007 ensimmäistä kertaa järjestetyssä konferenssissa olivat tietotekniikan tarjoamat mahdollisuudet, keinot ja teknologiat tasa-arvoisemman yhteiskunnan edistäjänä. Konferenssin pääasiallinen vetovastuu oli Tampereen ammattikorkeakoulussa opettavalla Esa Kujansuulla, joka osana EU-rahoitteista Codewitz-Asialink-projektia koordinoi eri tahojen välistä yhteistyötä. Kujansuu toimi myös erit-

täin aktiivisesti mukana järjestämässä hieman ITE:ä ennen järjestettyä PRIMA2007-konferenssia yhdessä professori Sadanandan kanssa.

Molemmat konferenssit järjestettiin Bangkokin keskustassa sijaitsevan Rangsitin yliopiston city-kampuksen tiloissa, mistä johtuen Rangsitin yliopisto eli RSU oli tärkeässä roolissa molempien konferenssien onnistumisen kannalta.

Olimme lähes päivittäisessä puhelin- ja sähköpostiyhteydessä professori Sadanandaan ja kävimme lisäksi useasti tapaamassa häntä. Paikallisena yhteyshenkilönä toimi professori Sadanandan lisäksi Rangsitin yliopiston kv-toimiston työntekijät sekä yliopiston talousosastolta hra. Pitawasi.

Pidimme myös jatkuvasti yhteyttä Tampereen ammattikorkeakoulussa opettavaan Esa Kujansuuhun, joka antoi tasaisin väliajoin lisäohjeistusta projektiin, ja jonka puoleen saatoimme kääntyä mm. ohjelmointiin liittyvissä ongelmissa. Kujansuu myös vieraili luonamme Thaimaassa kaksi kertaa harjoittelujakson aikana, alkuvaiheessa ja hieman ennen loppua, antamassa palautetta, neuvoja ja väliaikatietoja siitä, miten konferenssien organisointi edistyy.

Kävimme lisäksi jatkuvaa sähköpostikirjeenvaihtoa Australiassa sijaitsevan Darwinin yliopiston edustajan Aditya Ghosen kanssa, sekä italialaisen professori Guido Governatoren kanssa. Kaikenkaikkiaan molempia konferensseja organisoivat erilliset ohjelmakomiteat ja organisointikomiteat, joista jälkimmäiseen meidän vapaaehtoistyötä tekevinä opiskelijoina molempien konferenssien osalta luettiin. Pääasiassa toimeksiantajinamme toimivat kuitenkin Sadananda ja Kujansuu, joille raportoimme edistymisestämme säännöllisesti.

3 Käytetyt menetelmät

Sivut toteutettiin TAMKIn tyh-nimiselle Debian GNU/Linux-palvelimelle, johon on asennettu PHP versio 4 ja PostgreSQL tietokannanohjausjärjestelmä. Projektia varten palvelimelle oli luotu yksi ITE-niminen tietokanta, johon molempien konferenssien tiedot oli tarkoitus tallentaa.

Projektia aloitettaessa osasin ainoastaan perusteet PHP-ohjelmoinnista ja SQL-ky-selykielestä, joten muiden harjoittelujaksoon kuuluneiden järjestelytehtävien lisäksi aikaa kului paljon asioiden opiskelemiseen itsenäisesti. Käytössäni ollut opiskelu- ja lähdemateriaali pysyi samana alusta loppuun, koska lisää kirjallisuutta oli vaikeata saada vieraskielisessä maassa. Itseopiskelu oli pääsääntöisesti kirjallisuuden tutkimista, internetin hakupalveluiden ja keskustelupalstojen käyttämistä, sekä koodin kirjoittamista ja testaamista.

Varsinainen ohjelmointi tehtiin tietotaidon puutteesta ja PHP:n 4-version rajoituksista johtuen pääasiassa proseduraalisesti, eli ohjelmakoodi suoritetaan ylhäältä alas siinä järjestyksessä, kuin se tiedostossa esiintyy. Olio-ohjelmointi, eli funktioiden sijoittaminen luokkiin ja niistä uudelleenkäytettävien olioiden johtaminen, on viime vuosina syrjäyttänyt proseduraalisen ohjelmoinnin suosiossa, johtuen sen uudelleen käytettävyydestä, ja sovellettavuudesta suuriin, usean ohjelmoijan projekteihin (Naramore, Gerner, Le Scouarnec, Stolz & Glass 2005, 79-82). Katsoin kuitenkin parhaaksi opetella ensin PHP-ohjelmoinnin perinteisellä tavalla, koska olio-ohjelmointi olisi ollut hieman liian haastavaa tuolloisella osaamisellani.

Ohjelmakoodi kehitettiin ja testattiin omalle kannettavalle tietokoneelleni asentamalla XAMPP Apache/PHP-palvelimella, johon asensin lisäksi PostgreSQL-palvelimen harjoittelujakson alussa. Kun ohjelman toimivuus oli varmistettu, se siirrettiin Filezilla-nimisellä FTP-yhteysohjelmalla Suomessa sijaitsevalle tyh-palvelimelle, ja testattiin vielä uudelleen ennen varsinaiseen tuotantokäyttöön ottamista. Samalla tavalla myös

tietokantataulut kehitettiin ensiksi testiympäristössä ja ajettiin sitten tauluiksi Putty-nimisellä ssh-ohjelmalla tyh.tpu.fi-palvelimelle.

Konferenssien kansainvälisestä luonteesta johtuen sivuista oli tehtävä englanninkieliset, joten samalla oli luonnollista tehdä ohjelmakoodin muuttujat ja kommentoinnit englanniksi.

4 Sivustojen toteuttaminen

Sivustojen toteuttaminen aloitettiin määrittelemällä niiden käyttötarkoitus sekä onnistumisen kannalta kriittiset vaatimukset, joista lisää seuraavassa.

4.1 Sivustoille asetetut vaatimukset

Sivustojen tarkoituksena oli toimia työkaluna ja apuvälineenä konferenssiorganisaation jäsenille, sekä tietopakettina konferensseista kiinnostuneille. Alusta asti oli selvää, että yksi tärkeimmistä toiminnoista tulisi olemaan mahdollisuus ilmoittautua konferenssiin verkkosivujen kautta, sekä maksaa konferenssin osallistumismaksu luottokortilla, niin ikään verkkosivujen kautta. Tärkeää oli myös se, että täysin HTML-kieltä taitamaton käyttäjä voisi päivittää sivujen tekstisisältöä. Tämä siksi, koska työharjoittelujaksomme päättyi puoli vuotta ennen konferensseja, emmekä olisi siten enää olleet tekemässä päivityksiä sivuille. Projektin edetessä Kujansuu ja Sadananda määrittelivät uusia toimintoja, joita sivuille tulisi implementoida ja joitakin alussa määriteltyjä toimintoja jätettiin samalla pois lopullisesta tuotteesta.

Ensimmäisessä tapaamisessa ennen lähtöämme Thaimaahan, kävimme Kujansuun kanssa läpi niitä toiminnallisia ja ei-toiminnallisia vaatimuksia, joita heti varhaisessa vaiheessa tiedettiin tällaisen akateemisen konferenssin WWW-sivustolta edellytettävän. Tuon tapaamisen pohjalta laadimme yksinkertainen vaatimusmäärittelyn, joka käydään läpi seuraavaksi.

4.1.1 Toiminnalliset vaatimukset

Toiminnallisuudella tarkoitetaan tässä tapauksessa niitä asioita, joita käyttäjän odotetaan voivan sivustolla tehdä (Kolehmainen 2006, 455-475). Ohessa on listattuna tärkeimmät toiminnalliset vaatimukset:

- ✓ Käyttäjähallinta.
- ✓ Sisäänkirjautuminen.
- ✓ Uloskirjautuminen.
- ✓ Sähköpostilista.
- ✓ Vieraaksi rekisteröityminen.
- ✓ Luottokortilla maksaminen.
- ✓ Sivujen sisällön muokkaaminen HTML-lomakkeilla (Content Management System).
- ✓ Kirjanpitytyökalu.

4.1.2 Ei-toiminnalliset vaatimukset

Ei-toiminnalliset vaatimukset rajaavat toteutettavan sovelluksen reunaehdot ja määrittelevät käytettävyyteen liittyviä asioita yleisellä tasolla:

- ✓ Oltava käytettävissä yleisimmillä WWW-selaimilla.
- ✓ Käytettävä avoimen lähdekoodin tekniikoita (PHP).
- ✓ Tietokantana PostgreSQL.
- ✓ Englanninkielisyys.
- ✓ Tietoturvallisuus.
- ✓ Toimintavarmuus.

Toiminnalliset vaatimukset on lueteltu kiireellisyysjärjestyksessä, eli käyttäjähallinta oli saatava ensimmäisenä kuntoon ja sen jälkeen sähköpostilista, jotta professori Sadananda saattoi alkaa lähettämään sähköposteja konferensseista kiinnostuneille.

Sähköpostilistan jälkeen kiireisimmät toiminnot olivat rekisteröityminen ja luottokorttimaksu, jotka oli saatava toimimaan mahdollisimman aikaisessa vaiheessa, jotta ennakoilmoittautuminen olisi mahdollista jo puoli vuotta ennen varsinaisia konferensseja.

5 Toteutetut toiminnot

Projektin edetessä ja konferenssien lähestyessä vaatimusmäärittelyt muuttuivat muutama kertaan ja lopulliseen versioon sivustoista päätyivät seuraavaksi esittelyssä olevat toiminnot.

5.1 Tietokantayhteys

Kaiken perusta oli muodostaa yhteys Tampereen ammattikorkeakoulun palvelimella sijaitsevaan SQL-tietokantaan. Tietokanta on ehdoton edellytys kaikelle sellaiselle web-ohjelmoinnille, jossa käyttäjän antamaa tietoa täytyy tallentaa ja verrata jo tallennettuun dataan. Tässä projektissa käytettiin PostgreSQL versiota 8, johon yhteys muodostettiin koulun palvelimen käyttämän PHP versio 4:n avulla. Taulut luotiin kantaan ottamalla SSH-yhteys Putty-nimisellä ilmaisohjelmalla, kirjautumalla palvelimelle ja kirjoittamalla komentokehoitteen taulun luomiseen vaadittavat SQL-komennot (Liite 1).

Käytännössä web-sovelluksen yhteys toimi siten, että yhteyden vaatimat asetustiedot oli tallennettu tietoturvasyistä erilliseen config.php-tiedostoon. Näin meneteltäessä tärkeät tietokantatiedot eivät ole missään vaiheessa näkyvissä itse koodissa ja riski sille, että tietokantaan pääsevät käsiksi asiaankuulumattomat henkilöt, pienenee merkittävästi (Koodiesimerkki 1).

```
<?php
```

```
$dbuser = "*****";  
$dbpassword = "*****";  
$dbdatabase = "ite2007";  
$dbhost="tyh.tpu.fi";  
$admin_email="info.ite2007@gmail.com";  
$config_basedir="http://www.ite2007.org/";  
$config_dir2="http://www.ite2007.org/test2/submit_paper.php";  
$config_conference_name= "ITE2007";  
?>
```

Koodiesimerkki 1 Config.php-tiedoston sisältö

Config1.php sisältää siis tietokantaan yhdistämisen kannalta välttämättömät tiedot, kuten käyttäjätunnuksen, salasanan ja tietokannan nimen (korvattu esimerkissä tähdillä turvallisuussyistä) muuttujiksi tallennettuina. Asetustiedostoon voidaan tallentaa muuttujina myös muuta dataa, johon jossakin vaiheessa mahdollisesti halutaan viitata käyttämällä samaa muuttujaa. Edellä kuvattu ITE:n asetustiedosto sisältää mm. sivun hallinnoijan sähköpostiosoitteen sekä sekä sivuston etusivun URL-osoitteen.

Config.php-tiedosto sisällytetään halutun skriptin alkuun require-funktiolla ennen tietokantayhteyden ottamista, kirjoittamalla require('config1.php');

Varsinainen yhteys otetaan PostgreSQL:lle ominaisella pg_connect-funktiolla, joka vaatii parametreikseen tietokannan palvelimen nimen ("host"), tietokannan nimen ("dbname"), auktorisoidun käyttäjätunnuksen ("user") ja salasanan ("password") (Koodiesimerkki 2).

```
$conn = pg_connect("
host=$dbhost
dbname = $dbdatabase
user = $dbuser
password = $dbpassword")
or die('Could not connect to PSQL database. ' . pg_last_error());
```

Koodiesimerkki 2 Tietokantayhteyden muodostaminen pg_connect-funktiolla

Tässä tapauksessa pg_connect-funktio ottaa parametreikseen config1.php tiedostoissa alustetut muuttujat joiden todelliset arvot eivät näy koodissa. Sen jälkeen, kun tietokantayhteys selaimelta palvelimelle on avattu, voidaan kantaan alkaa tehdä kyselyjä. Pg_last_error()-funktion antaa virheilmoituksen, mikäli yhteyden muodostaminen annetuilla parametreilla ei onnistu, jolloin kehitysvaiheessa voidaan paremmin todeta, mistä mahdollinen virhetilanne aiheutuu.

5.2 Käyttäjähallinta

Järjestelmän kehittäminen aloitettiin määrittämällä kaikki ne tapaukset, joissa sitä tulisi käyttää. Ensimmäiseksi oli määriteltävä eri käyttäjäroolit ja niiden saamat oikeudet.

Lopputuloksena syntyi käyttäjärooleihin perustuva järjestelmä, jossa eri käyttäjillä on erilaiset oikeudet ja näkymä sivujen sisältöön. Kaikkein korkeimpana käyttäjäroolina administrator eli hallinnoija, jolla on täydet oikeudet ja pääsy kaikkeen sisältöön. Hallinnoijan rooli sisältää myös alemmat roolit eli hallinnoija pystyy näkemään muiden käyttäjien tiedot. (Naramore ym. 2005, 370-375.)

Käyttötapauksessa käyttäjä kirjautuu sisään sivuille kirjoittamalla hänelle annetun käyttäjätunnuksen ja salasanan, minkä jälkeen hänelle avautuu hänen roolinsa mukainen näkymä ja valintamahdollisuudet. Käyttäjä voi kirjautua ulos järjestelmästä erillisellä painikkeella.

Sivuja selaavalla satunnaisella käyttäjällä ei ole oikeuksia, vaan ainoastaan mahdollisuus liittyä sähköpostilistalle ja ilmoittautua konferenssiin joko vieraaksi tai paperin esittelijäksi.

Sivujen hallinnoijalla eli administratorilla, on täydet oikeudet, sekä käyttäjätunnus ja salasana, joilla kirjautua sivuille.


Alun perin suunnitelmissa oli luoda useita eri käyttäjätasoja administratorin lisäksi, kuten sivuille kirjautuva tutkimuspaperin kirjoittaja sekä sivuille kirjautuva tutkimuspaperin arvostelija. Näistä käyttäjätasosta luovuttiin kuitenkin myöhemmin, koska tutkimuspapereiden lähettäminen ja arvosteleminen päätettiin siirtää ulkoiseen Easychair-järjestelmään (www.easychair.org).

Käyttäjätiedot sisältävään users-tauluun (Taulukko 1) tallennetaan käyttäjän nimitiedot, sähköpostiosoite, käyttäjätunnus sekä salasana. Huomattavaa on, että taulusta löytyy user_role-niminen rivi, mikä mahdollistaa eri käyttäjätasojen lisäämisen järjestelmään, mikäli tarvetta tulisi.

Taulukko 1 Käyttäjien tiedot sisältävä taulu

ite_users		
	Tietotyyppi	Selitys
id	serial	yksilöivä perusavain, juokseva numerotunnus
username	character, varying	käyttäjätunnus
password	character, varying	salasana
user_role	character, varying	käyttäjärooli
first_name	character, varying	käyttäjän etunimi
last_name	character, varying	käyttäjän sukunimi
email	character, varying	käyttäjän sähköpostiosoite

Käyttäjähallinnan perusidea on sisäänkirjautumisessa, joka alkaa login.php-tiedoston tuottamasta HTML-lomakkeesta, johon käyttäjä syöttää käyttäjätunnuksensa ja salasansa (Kuva 1).



IT TOWARDS EMPOWERMENT, INTERNATIONAL CONFERENCE

Username:

Password:

Login

Kuva 1 Sisäänkirjautuminen

Käyttäjän painaessa login-painiketta lomakekenttiin kirjoitetut tiedot välittyvät POST-metodin välityksellä globaaleina muuttujina eteenpäin.

Admin_login2.php-tiedostoon on sisällytetty require_once-funktiolla functions.php-tiedosto, joka sisältää lukuisia WWW-sivuja varten kirjoittamiani PHP-funktioita. Erillisen funktio-tiedoston käyttäminen vähentää koodin toistumista eri sivuilla ja tekee muutosten ja korjausten tekemisestä vähemmän työlästä, koska muutokset täytyy tehdä ainoastaan yhteen paikkaan (Zandstra 2000, 175-176).

Login-funktiossa haetaan aluksi käyttäjän syöttämä käyttäjätunnus \$_REQUEST-funktiolla ja tallennetaan se \$username-nimiseen muuttujaan. Seuraavaksi tehdään kysely tietokantaan ja haetaan users-tilusta vastaava käyttäjätunnus, salasana ja käyttäjärooli. Mikäli käyttäjän antamaa käyttäjätunnusta vastaava tunnus löytyy myös users-tilun username-rivien joukosta, luetaan kyseiseen käyttäjätunnukseen liittyvä password-rivi ja tallennetaan se \$salasana-nimisenä muuttujana.

Tämän jälkeen alkaa ehtolause, jossa vertaillaan kahta salasanaa toisiinsa:

```
if ((strcmp($_REQUEST['password'],$salasana)) == 0) {
```

Mikäli käyttäjän antama salasana ja tietokannan tilusta löytyvä salasana vastaavat toisiaan, luodaan kolme sessiota:

```
session_register("user_logged");
session_register("user_password");
session_register("user_role");
```

Seuraavaksi asetetaan sessiot voimaan ja tallennetaan sessioihin tässä vaiheessa autentikoidusta käyttäjästä käyttäjätunnus, salasana ja käyttäjärooli:

```
$_SESSION['user_logged'] = $row['username'];
$_SESSION['user_password'] = $row['password'];
$_SESSION['user_role'] = $row['user_role'];
```

Jos sisäänkirjautuvan käyttäjän user_role users-tilussa on admin, selaimen näkymä ohjautuu sivun hallinnoijalle tarkoitetulle logged_admin.php-sivulle:

```
if ($_SESSION['user_role'] == "admin") {
    header("Location: logged_admin.php");
}
```

Haluttaessa käyttäjä voidaan ohjata toiseen, eri käyttäjäoikeuksiseen näkymään, kuten esimerkiksi paperin kirjoittajan sivulle:

```
else if ($_SESSION['user_role'] == "paper_author") {
    header("Location: logged_author.php");
}
//ohjaus paper author-sivulle
}
```

Tätä mahdollisuutta ei kuitenkaan hyödynnetä valmiilla sivustolla, koska paperin kirjoittajat käyttivät sivuston ulkopuolista easychair-järjestelmää tutkimuspapereidensa lähettämiseen.

Lopuksi, jos salasanat eivät vastaa toisiaan, käyttäjä ohjataan takaisin sisäänkirjautumissivulle, virheilmoituksen kera:

```
else {
header("Location: " . $config_basedir . "index.php?page=login_error");
}
}
```

Tässä vaiheessa ehtolauseessa tarkastetaan GET-muuttujaan tallennettu arvo, ja mikäli se on 'error', tulostetaan teksti ”Incorrect login!”:

```
<?php
if($_GET['error']) {
    echo "Incorrect login!";
}
?>
```

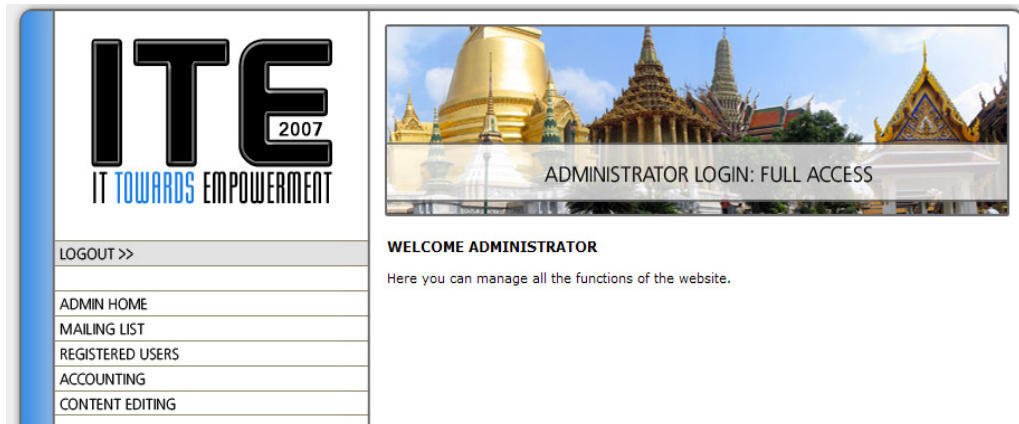
Ehtolauseen päättyessä käyttäjä on joko ohjautunut hänen käyttäjäoikeuksiaan vastaavalle sivulle (Kuva 2) tai takaisin sisäänkirjautumisnäkyymään. Sivun, jolle käyttäjä ohjautuu autentikoinnin jälkeen, sisältää auth.admin.inc-nimisen tiedoston, joka tarkistaa heti sivua ladattaessa palvelimelta, onko asianmukaiset PHP-sessiot asetettu:

```
if ((isset($_SESSION['user_logged']) &&
$_SESSION['user_logged'] == TRUE &&
$_SESSION['user_role'] == "admin" ||
```

```
(isset($_SESSION['user_password']) &&
$_SESSION['user_password'] == TRUE) &&
$_SESSION['user_role'] == "admin") {
//do nothing!! }
```

Ehtolauseessa tarkastetaan, onko sessio 'user_logged' asetettu, ja onko 'user_role'-session arvo 'admin', minkä lisäksi tarkastetaan, että user_password- session on asetettu. Mikäli oikeat sessiot löytyvät, skripti ei tee mitään, vaan antaa sivun latautua. Muussa tapauksessa sivua ei lähetetä selaimelle, vaan käyttäjä uudelleenohjataan sisäänkirjautumissivulle header()-funktioilla:

```
else {
$redirect = $_SERVER['PHP_SELF'];
header("Refresh: 3; URL=index.php?page=login");
echo "You are currently not logged in, we are redirecting you, " .
"be patient!<br>";
echo "(If your browser doesn't support this, " .
"<a href='\"index.php?page=login\"'>click here</a>";
die();
}
```



Kuva 2 Onnistunut sisäänkirjautuminen

Käytännössä siis auth.admin.inc-tiedostolla varmistetaan se, että kukaan ei voi päästä suoraan vaikkapa hallinnointisivulle kirjoittamalla sen selaimen URL-osoitteeksi.

5.3 Uloskirjautuminen

Uloskirjautuminen tapahtuu käyttäjän painaessa logout-linkkiä, joka ohjaa selaimen logout.php tiedoston kautta header()-funktiota käyttäen takaisin etusivulle ja poistaa samalla asetetut sessiot session_destroy()-funktiota käyttäen. Sen jälkeen kun sessio on tuhottu, sivuston suojatulle alueelle ei pääse ilman uutta sisäänkirjautumista:

```
<?php
session_start(); session_destroy();
require ("config1.php");
header("Location: " . $config_basedir); ?>
```

5.4 Sähköpostilista

Sähköpostilista on toiminto, jolla kuka tahansa sivuja selaava voi ilmaista kiinnostuksensa konferensseja kohtaan, liittyä sähköpostilistalle ja vastaanottaa sivuja hallinnoivan henkilön sähköpostiviesteissä uutisia ja tietoa konferensseista. Administrator näkee omassa näkymässään listan kaikista sähköpostilistalle liittyneistä ja voi lähettää kaikille heille viestejä samanaikaisesti sekä poistaa ja lisätä henkilöitä (Naramore ym. 2005, 481-482).

Käyttötapauksessa käyttäjä kirjoittaa etunimensä, sukunimensä ja sähköpostiosoitteensa sivulta löytyviin lomakekenttiin (Kuva 3) ja lähettää tiedot eteenpäin. Hetken kuluttua käyttäjä vastaanottaa varmistusviestin ilmoittamaansa sähköpostiosoitteeseen ja hän voi joko vahvistaa tai perua ilmoittautumisensa listalle seuraamalla sähköpostiviestissä olevaa linkkiä.

JOIN THE MAILING LIST

To join the mailing list, please provide your:

E-Mail Address:

First Name:

Last Name:

Kuva 3 Sähköpostilistalle liittyminen

Mailinglist.php-sivulta löytyvä lomake käyttää user_transact.php-tiedostoa välittääkseen sähköpostilistalle liittyvän henkilön tiedot tietokannan ite_ml_users-tauluun (Taulukko 1) ja lisää samalla ite_ml_subscriptions-tauluun, (Taulukko 2), tilaajan id:n ja pending tiedoksi luvun 0, mikä tarkoittaa, että listalle liittymistä ei ole vielä vahvistettu. User_transact.php-tiedosto lähettää tilaajan sähköpostiosoitteeseen mail()-funktiolla sähköpostiviestin, jossa pyydetään vahvistamaan tai perumaan tilaus (Koodiesimerkki 3).

Taulukko 1 Sähköpostilistalle liittyneet

ite_ml_users		
	Tietotyyppi	Selitys
user_id	serial	perusavain, postilistan jäsenen id
firstname	character, varying	postilistan jäsenen etunimi
lastname	character, varying	postilistan jäsenen sukunimi
email	character, varying	postilistan jäsenen sähköpostiosoite

Taulukko 2 Sähköpostilistan tilaukset

ite_ml_subscriptions		
	Tietotyyppi	Selitys
ml_id	serial	vierasavain, joka viittaa ite_ml_lists-tauluun
user_id	serial	vierasavain, joka viittaa ite_ml_users-tauluun
pending	integer	Arvo 1 tai 0, jonka mukaan tilaus on joka aktivoitu tai aktivoimaton

```
$subject = 'Mailing list confirmation';
$body = "Hello " . $_POST['firstname'] . "\n" .
"Our records indicate that you have subscribed to " . "the " . $listname . " mailing list.\n\n" .
```


"If you did not, please accept our apologies. You will not be receiving any further e-mails if you do not visit the confirmation URL.\n\n".

"If you subscribed, please confirm this by visiting "

"the following URL:\n" . \$url;

\$mailmsg = new SimpleMail();

\$mailmsg->send(\$_POST['email'],\$subject,\$body,\$headers);

\$redirect = "index.php?page=thanks";

Koodiesimerkki 3 Sähköpostiviestin lähettäminen Simplemail-luokkaa käyttäen

Itseopiskelumateriaaliani (Naramore ym. 2005) soveltaen käyttöön otettu Simplemail-luokka kokoaa sähköpostiviestin \$subject-, \$body- ja \$headers-muuttujista ja lähettää sen POST-parametrin mukaiseen sähköpostiosoitteeseen vastaanottajalle.

Käyttäjän vahvistettua liittymisensä, hänen nimensä ja sähköpostiosoitteensa näkyy sivun hallinnoijalle, sille varatussa osiossa (Kuva 4), jossa kaikki sähköpostiosoitteen ilmoittaneet ja tilauksen vahvistaneet henkilöt listautuvat. Tämä lista toimii myös suunta-antavasti mittarina siitä, kuinka montaa ihmistä konferenssin aihealueet kiinnostavat. Hallinnoija voi kirjoittaa haluamansa viestin Send a quick message -linkin takaa löytyvälle lomakkeelle (Kuva 5) ja painaa Send message -painiketta lähettääksensä viestin kaikille postilistalla oleville.

MAILING LIST

[Instructions](#) | [Browse users](#) | [Send a quick message](#)

Below is a list of users:

 Lastname:

Firstname:

E-Mail: cme_thailand@yahoo.com

 Lastname: Azad

Firstname: Abul Kalam

E-Mail: kalam11@gmail.com

Kuva 4 Sähköpostilistan hallinnointinäkökulma

MAILING LIST[Instructions](#) | [Browse users](#) | [Send a quick message](#)

Compose Message:

Subject:

Message:

Kuva 5 Viestin lähettäminen postilistalle oleville

5.5 Content Management System, sisällönhallintajärjestelmä

Perinteinen tapa WWW-sivujen päivittämiseen on ottaa FTP-yhteys palvelimelle, jolla sivuston sisältävät HTML-dokumentit sijaitsevat ja ladata haluttu dokumentti omalle työasemalle. Lataamisen jälkeen dokumenttiin voidaan tehdä halutut muutokset ja tallentaa se takaisin palvelimelle, minkä jälkeen sivulle tehdyt päivitykset tulevat voimaan. Tämä tapa edellyttää käyttäjältä kuitenkin yleensä luku- ja kirjoitusoikeuksia palvelimelle, jolla sivut sijaitsevat. Lisäksi käyttäjän täytyy osata käyttää FTP-yhteysohjelmaa sekä lukea ja ymmärtää kyseessä olevan WWW-sivun kirjoittamisessa käytettyä sivunkuvauskieltä, jotta siihen voidaan tehdä tarvittavat muutokset.

Sisällönhallintajärjestelmällä tarkoitetaan WWW-sivuja, joita on mahdollista muokata ilman tietämystä sivunkuvauskielistä. Käytännössä tämä toimii usein siten, että riittäväillä käyttäjäoikeuksilla varustettu käyttäjä kykenee muokkaamaan esimerkiksi sivujen tekstisisältöä lomakenttien avulla, kirjoittamalla tekstikenttään halutun tekstin, joka tallentamisen ja selainikkunan päivittämisen jälkeen näkyy sivulla juuri samassa muodossa. (Naramore ym. 2005)

Sisällönhallintajärjestelmän toteuttaminen on hieman työläämpää kuin tavallisen WWW-sivun toteuttaminen ja vaatii aina relaatiotietokannan ja siihen liittyvän ohjelmointikielen, kuten PHP, käyttämistä toteutusvaiheessa. Tämän projektin osalta kävi jo varhaisessa vaiheessa selväksi, että päivityksiä tultaisiin tekemään lähes päivittäin myös senkin jälkeen kun oma harjoittelujaksomme olisi päättynyt. Erityistä huomiota oli kiinnitettävä siihen, että WWW-sivulta löytyvä päiväkohtainen aikataulu olisi välttämättä muokattavissa (Kuva 6).



The screenshot shows the ITE 2007 website. On the left is a navigation menu with links: LOGIN >>, HOME, IMPORTANT DATES, INVITED SPEAKERS, PROGRAMME, JOIN THE MAILING LIST, PAPER & POSTER SUBMISSION, REGISTRATION, LOCATION, ACCOMMODATION, TOURS, and CONTACTS. The main content area features a banner image of a golden pagoda with the text 'ITE TOWARDS EMPOWERMENT, INTERNATIONAL CONFERENCE'. Below the banner is a 'PROGRAMME' section for Monday, Nov 26, Tuesday, Nov 27, and Wednesday, Nov 28. An 'Edit' button is visible. The schedule for Monday, November 26, 2007, is as follows:

Monday, November 26, 2007	
8.30-9.00	Registration
9.00-9.30	Opening Session Remarks - R.Sadananda, Co-Chair Welcome - Chonawat Srisa-an, Rangsit University
9.30-10.15	Coffee Break
10.15-11.30	Invited Speaker - Dr. Essig <i>The Quality of Information Challenge; Requirements for new Business Integration Technologies</i>

Kuva 6 Aikataulun muokkaaminen

Esimerkkitapauksessa itsekirjoitettu PHP-funktio `edit_ite_mon` hakee tietokannan `ite_mon`-taulusta (Taulukko 4) sinne tallennetun tekstisisällön ja tulostaa sen HTML-lomakkeeseen. Tässä tapauksessa jokainen `<td>`-tagin väliin tulostettava, muokattava sisältö on omana rivinä `ite_mon`-taulussa (Koodiesimerkki 4).


```

function save_ite_mon()
{
$content1 = $_GET['1'];
$sql = "UPDATE ite_home_text SET content = ".$content1." WHERE id = 1";
pg_query($sql);
header("Location: " . "http://www.ite2007.org/index.php?page=home");
}

```

Koodiesimerkki 5 Käyttäjän maanantain aikatauluun tekemät muutokset tallentava save_ite_mon-funktio

Tämän jälkeen näkymä siirtyy takaisin WWW-sivulle, jossa tehdyt muutokset ovat välittömästi nähtävissä (Kuva 8 ja Koodiesimerkki 6).

The screenshot shows the ITE 2007 website interface. On the left is a navigation menu with links: LOGIN >>, HOME, IMPORTANT DATES, INVITED SPEAKERS, PROGRAMME, JOIN THE MAILING LIST, PAPER & POSTER SUBMISSION, REGISTRATION, LOCATION, ACCOMMODATION, TOURS, and CONTACTS. The main content area features a banner image of a golden stupa with the text "IT TOWARDS EMPOWERMENT, INTERNATIONAL CONFERENCE". Below the banner is a "PROGRAMME" section for "Monday, Nov 26 | Tuesday, Nov 27 | Wednesday, Nov 28". An "Edit" button is visible. The schedule for Monday, November 26, 2007, is as follows:

Monday, November 26, 2007	
8.30-9.00	Registration editoitua sisältoä.... !!!
9.00-9.30	Opening Session Remarks - R.Sadananda, Co-Chair Welcome - Chonawat Srisa-an, Rangsit University
9.30-10.15	Coffee Break
10.15-11.30	Invited Speaker - Dr. Essig <i>The Quality of Information Challenge; Requirements for new Business Integration Technologies</i>

Kuva 8 Muokattu sisältö näkyy sivulla käyttäjän painettua save-painiketta

```

function show_ite_mon()
{
if (strcmp($_SESSION['user_role'], "admin") == 0){
print("<p>
<form action=\"programme.php\" method=\"GET\">
<input type=\"hidden\" name=\"action\" value=\"edit_ite_mon\">
<input type=\"submit\" value=\"Edit\">
</form></p>");
}
}

```

```

require_once("config1.php");
print("<table>");
print("<tr>");
print("<td class='list' colspan=2>");

$sql_string = "SELECT * FROM ite_mon WHERE id = 1 ";
$result = pg_query($sql_string);
$row = pg_fetch_array($result);

print("<p class='listheader'\>" . $row['content'] . "</p>");
print("</td>");
print("</tr>");
print("<tr>");
print("<td class='list2'\>");

```

Koodiesimerkki 6 Show_ite_mon-funktio näyttää selaimessa tietokantaan tallennetun, muutetun aikataulun

5.6 Konferenssiin ilmoittautuminen ja rekisteröityminen

Tärkeä osa konferenssien järjestelemistä oli saada mahdollisimman pian mahdollisimman paljon sitovia ilmoittautumisia molempiin konferensseihin, jotta saataisiin arvio käytettävissä olevasta budjetista ja kävijämäärästä. Tätä varten sivuille piti toteuttaa toiminto (Kuva 9), jolla konferenssivieraat pystyivät rekisteröitymään konferenssiin osallistujiksi ja maksamaan osallistumismaksun turvallisesti verkossa luottokortilla, tilisiirrolla tai käteismaksulla paikanpäällä konferenssissa. Aikaisessa vaiheessa rekisteröityneille vieraille tarjottiin myös alennusta osallistumismaksusta. Tärkeintä oli saada luottokorttimaksu toimimaan hyvissä ajoin, jotta konferenssiin ennakkoon ilmoittautuneilta saatiin osallistumismaksu.

IT TOWARDS EMPOWERMENT, INTERNATIONAL CONFERENCE

REGISTRATION

To register on the ITE2007 , fill in the form below.

There are three payment options: credit card & bank transfer & cash.
 By choosing credit card you will be redirected to the secure Thai Epay payment service and there you are instructed how to proceed further.
 By choosing bank transfer option you will get the account information by email and you should make the payment to the account as soon as possible.
 By choosing cash, you will pay the conference fee upon arrival.

Notice: The registration is binding.

Firstname
 Lastname
 Organization
 Country
 Email
 Your status:

I'm attending both Prima 2007 & ITE 2007 and paying with credit card:
 I am registering as a student

Payment method:
 Credit Card
 Bank Transfer
 Cash

Accepted credit cards: Visa, Amex, MC

Kuva 9 Rekisteröintilomake

Käyttötapauksessa konferenssiin ilmoittautuva käyttäjä kirjoittaa lomakenttiin etunimensä, sukunimensä, sähköpostiosoitteensa, edustamansa oppilaitoksen tai organisaation nimen sekä valitsee alasvetovalikosta kotimaansa. Lisäksi käyttäjän on valittava osallistujastatuksensa, joko osallistuja (presenter), tutkimuspaperin kirjoittaja (paper presenter) tai lyhyemmän nk. poster-esityksen pitäjä (poster presenter), minkä jälkeen käyttäjä vielä valitsee maksutavan ja valinnastaan siirtyen joko ohjautuu suojattuun ThaiEpay-luottokorttimaksujärjestelmään ja sieltä takaisin konferenssisivulle tai saa sähköpostiinsa varmistuksen siitä, että hänen ilmoittautumisensa on rekisteröity ja sitova.

5.6.1 Luottokorttimaksu

Luottokorttimaksujärjestelmänä sivustot käyttävät thaimaalaisista ThaiEpay-järjestelmää. Mikäli käyttäjä on valinnut rekisteröinnin yhteydessä maksutavaksi luottokorttimaksun, ohjataan hänet konferenssivustolta suojatussa yhteydessä ThaiEpay:n verkkosivuille, johon välittyvät myös HTML-lomakkeen piilotettuina kenttinä palvelun vaatimat parametrit (Taulukko 5).

Taulukko 5 ThaiEpay-järjestelmän parametrit

Parametri	Tietotyyppi	Selitys
merchantid	integer	asiakasnumero ThaiEPAY-järjestelmään
refno	integer	referenssinumero
customeremail	character	asiakkaan sähköpostiosoite
productdetail	character	tuotteen kuvaus
total	integer	kokonaissumma
postbackurl	character	URL, johon järjestelmä palauttaa parametrejä haluttaessa
returnurl	character	URL, johon järjestelmä ohjaa maksutapahtuman jälkeen

Onnistuneen luottokorttimaksutapahtuman jälkeen ThaiEpay palauttaa käyttäjän takaisin konferenssin verkkosivulle returnurl-parametrin mukaisesti. Samaan aikaan confirmed.php-tiedostossa oleva ehtolause tarkastaa postbackurl-parametrissa palaavan luvun, ja päivittää sitä vastaavan konferenssivieraan maksustatuksen tilaan confirmed, varmistettu (Koodiesimerkki 7).

```
if($_GET['conf']){
    $luku = $_GET['conf'];
    $query_confirm = "UPDATE ite_guests SET payment_status = 'CONFIRMED' WHERE id =
    $luku";
    pg_query($query_confirm);
```

Koodiesimerkki 7 ThaiEpay-palauttaa rekisteröityvän vieraan id:n, joka kaapataan \$_GET-muuttujan avulla ja verrataan tietokannassa olevaan

Tämän jälkeen käyttäjä saa sähköpostiinsa tiedon sekä onnistuneesta rekisteröitymisestä, että onnistuneesta luottokorttimaksusta.

5.7 Ilmoittautumisten ja osallistumismaksujen seuraaminen

Konferenssien kirjanpidon ja käytännön järjestelyjen, kuten aterioinnin järjestämisen vuoksi, on tärkeä kyetä seuraamaan ilmoittautumisten ja osallistumismaksujen määrää. Administratorilla on omassa näkymässään mahdollisuus listata kaikki ilmoittautuneet maksutavan mukaan sekä nähdä maksun status, joka oli joko varmistunut tai ei-varmistunut. Järjestelmä oli toteutettava siten, että luottokorttimaksut varmistuisivat automaattisesti luottokorttimaksujärjestelmästä onnistuneen maksun jälkeen saapuvan palautteen myötä ja vieraiden tekemät käteis- ja tilisiirtomaksut varmistettaisiin administratorin toimesta. Tämä toiminto toimi kätevänä työkaluna itse konferensseissa, mahdollistaen vieraiden statuksen tarkistamisen ja päivittämisen sitä mukaan, kun he saapuivat konferenssipaikalle. Käytännössä hallinnoijan ollessa `browse_registered_users.php`-sivulla, ohjelmakoodi hakee rekisteröityneiden vieraiden tiedot tietokannan taulusta (Taulukko 6 ja Koodiesimerkki 8).

Taulukko 6 Rekisteröityneet konferenssivieraat

ite_guests	Tietotyyppi	Selitys
id	serial	perusavain, juokseva numerotunnus
first_name	character, varying	konferenssivieraan etunimi
last_name	character, varying	konferenssivieraan sukunimi
email	character, varying	konferenssivieraan sähköpostiosoite
organization	character, varying	konferenssivieraan edustama organisaatio
country	character, varying	konferenssivieraan kotimaa
payment_method	character, varying	konferenssivieraan valitsema maksutapa
payment_status	character, varying	osallistumismaksun tila
guest_status	character, varying	konferenssivieraan status, esim. onko opiskelija vai luennoitsija

```
$query = "SELECT id, first_name, last_name, email, organization, country, guest_status,
payment_method, payment_status FROM ite_guests " .
"ORDER BY last_name";
$result = pg_query($query)
or die(pg_last_error());
```

Koodiesimerkki 8 Rekisteröityneiden vieraiden tiedot haetaan tietokannasta

Tämän jälkeen tiedot tulostetaan HTML-taulukkoon echo-komennolla (Kuva 10) while-silmukassa, joka tulostaa niin monta taulukkoriviä, kuin `ite_guests`-taulusta löytyy (Koodiesimerkki 9).

```

while ($row = pg_fetch_array($result)) {

echo "<table class='text' cellpadding=5>";
echo "<tr><td>Lastname:</td><td>" . $row['last_name'] . "</td>";
echo "<tr><td>Firstname:</td><td>" . $row['first_name'] . "</td>";
echo "<tr><td>E-Mail:</td><td>" . $row['email'] . "</td>";
echo "<tr><td>Organization:</td><td>" . $row['organization'] . "</td>";
echo "<tr><td>Country:</td><td>" . $row['country'] . "</td>";
echo "<tr><td>Guest Status:</td><td>" . $row['guest_status'] . "</td>";
echo "<tr><td>Payment Method:</td><td>" . $row['payment_method'] . "</td>";
echo "<tr><td>Payment Status:</td><td>" . $row['payment_status'] . "</td>";
echo "</table>";
?>

```

Koodiesimerkki 9 Echo-komento tulostaa tietokannasta poimitut rivit HTML-muotoon

Hallinnoija voi myös varmistaa vieraan maksustatuksen, esimerkiksi käteismaksutilanteessa, painamalla Confirm payment-painiketta (Koodiesimerkki 10).

```

<form>
<input type="button" onClick="parent.location='confirm_payment.php?id=?php echo $row['id'];?>" value="Confirm Payment" >

```

Koodiesimerkki 10 Confirm-painikkeen koodi

Hallinnoija voi lisäksi poistaa vieraan kokonaan vierastaulusta painamalla delete user-painiketta, mikäli vieras vaikkapa peruuttaa osallistumisensa (Koodiesimerkki 11).

```

<input type="button" onClick="parent.location='delete_user.php?id=?php echo $row['id'];?>" value="Delete User" >
</form>

```

Koodiesimerkki 11 Delete user-painikkeen koodi

IT TOWARDS EMPOWERMENT

LOGOUT >>

ADMIN HOME
MAILING LIST
REGISTERED USERS
ACCOUNTING
CONTENT EDITING

ADMINISTRATOR LOGIN: FULL ACCESS

REGISTERED USERS

[Instructions](#) | [Add a registered user](#) | [Browse registered users](#)
[Show Registered users ordered by the payment method and status](#)

Below is a list of registered users:

Lastname: Alam
 Firstname: Md. Jobair Bin
 E-Mail: jobair@ce.buet.ac.bd
 Organization: BUET
 Country: Bangladesh
 Guest Status: paper_presenter
 Payment Method: cash
 Payment Status: UNCONFIRMED

Kuva 10 Rekisteröityneiden käyttäjien seurantanäkymä

5.8 Kirjanpityökalu

Kirjanpityökalu oli tämän projektin puitteissa melko yksinkertainen toiminto, jolla kaikki administrator-tason oikeudet omaavat pääsivät kirjaamaan konferenssi organisaation tuloja sekä menoja. Työkalu oli tarkoitettu apuvälineeksi kaikille käytännön järjestelyjä tekeville konferenssiorganisaation jäsenille sekä yleiskatsaukseksi siitä, miten konferenssin käytössä olevia varoja käytettiin.

Käyttötapauksessa käyttäjä kykenee hakemaan tapahtumia päivämäärän mukaan ja lisäämään tai poistamaan tuloja tai menoja lomakekentän välityksellä (Kuva 11).

ACCOUNTING

[Instructions](#) | [Add accounting information](#) | [Browse accounting information](#)

Entry	Income	Expense	Ex-rate	Explanation	Year	Month	Day	
	0	0	0		2007	1	1	<input type="button" value="Add"/>

Kuva 11 HTML-lomake, jolla käyttäjä syöttää uuden budjettitapahtuman

Projektin graafisesta ulkoasusta vastanneen Lautarinteen suunnittelema layout valitettavasti asetti rajoitteita sille, kuinka leveitä tekstikenttiä kirjanpityökalun listauksissa voitiin näyttää. Tästä syystä mm. Entry- ja Explanation-kentät ovat melko ahtaita, eikä niihin voi kirjoittaa sisältöä kuin lyhyesti.

Käyttäjän ei tarvitse erikseen kirjoittaa omaa nimeänsä, vaan ohjelma tarkistaa sisäänkirjautuneen käyttäjän etu- ja sukunimen käynnissä olevasta sessiosta ja tallentaa ne jokaisen käyttäjän kirjaaman tulon tai menon yhteyteen:

```
$user = $_SESSION['user_logged'];
$usr_query ="SELECT first_name, last_name FROM ite_users WHERE username = '$user' ";
```

Käyttäjä voi selata tapahtumia haluamallaan aikavälillä (Kuva 12). Ohjelmakoodi hakee budjetin senhetkisen summan ja budjettitapahtumat tietokannasta (Taulukot 7 ja 8). Listattuna näkyy tapahtuman nimi, tulon tai menon suuruus, vaihtokurssi tapahtumahetkellä, tapahtuman päivämäärä, selitys tapahtumalle sekä tapahtuman tekijän nimi. Punaisesta rastista käyttäjä voi poistaa tapahtuman, jolloin myöskin tapahtuman tulo tai meno vähennetään tai lisätään kokonaisbudjettiin takaisin.

ACCOUNTING

[Instructions](#) | [Add accounting information](#) | [Browse accounting information](#)

The balance is: 1000 USD

Show events from the period:

2007-01-01 - 2008-03-01

Entry	Income	Expense	Ex-rate	Date	Explanation	Author	
Entry fee	1000	0	0	2007-04-01	Entry fees	Professor Sadananda	X

Kuva 12 Budjettitapahtumien seurantanäkymä


```
        . "" . $explanation . "" ,"  
        . "" . $usr_name . "" ,"  
        . "" . $currency . "" )";  
pg_query($sql);
```

```
$query = "UPDATE ite_balance SET balance = balance + " . $income . " ";  
pg_query($query);
```

```
$query = "UPDATE ite_balance SET balance = balance - " . $expense . " ";
```

```
pg_query($query);  
}
```

Koodiesimerkki 12 addEvent-funktion sisältämä koodi

6 Lopputulokset

Projekti saatiin verkkosivujen osalta päätökseen Juhannuksena 2007, minkä jälkeen palasimme Suomeen. Meistä riippumattomista syistä sivustojen liittäminen ThaiEpay-luottokorttimaksujärjestelmään uhkasi jäädä toteutumatta ja jouduin vielä Suomesta käsin hiomaan joitakin yksityiskohtia kuntoon. Juhannuksesta seuraavan puolen vuoden ajan sivut olivat aktiivisessa käytössä sekä konferenssiorganisaation jäsenillä että konferenssivierailta ja marraskuussa 2007 molemmat konferenssit järjestettiin onnistuneesti Bangkokin keskustassa sijaitsevissa Rangstin yliopiston tiloissa.

Ottaen huomioon, että aikaisempi kokemukseni PHP:sta ja relaatiotietokannoista WWW-sovelluksissa oli hyvin vähäistä, onnistun asetetuissa tavoitteissa erittäin hyvin. Silti jälkikäteen tarkasteltuna monta asiaa olisi voinut tehdä toisella tavalla ja paremmin. Sivut eivät ole kovinkaan tietoturvalliset, sillä käyttäjien antamaa syötettä ei tarkisteta juurikaan ollenkaan, mikä mahdollistaa esimerkiksi SQL-injektio-hyökkäyksen, jossa pahanaikeinen käyttäjä pääsee lomakekenttään SQL-komentoja kirjoittamalla vaikuttamaan sivun taustalla pyörivään tietokantaan (Bacon 2006, 84). Lisäksi sivut toistavat paikoittain samaa koodia useaan kertaan eri tiedostossa ja tietokantayhteys avataan toistuvasti turhaan monta kertaa sulkematta sitä välillä, mikä tekee sivuista palvelimelle raskaammat kuin olisi tarve.

Tästä huolimatta sivut toimivat suurimmaksi osaksi niin kuin pitikin, ilman suurempia teknisiä ongelmia ja olivat osaltaan saattamassa konferensseja menestyksekkääseen päätökseen marraskuussa 2007, aina siihen asti kun luottokortilla osallistumismaksunsa maksaneille konferenssivieraille tulostettiin kuitteja konferenssien päätöspäivänä 28.11.2007.

Lähteet

Kirjat

Bacon, Jono. Practical PHP & MySQL. 2006. Prentice Hall.

Kolehmainen, Kauko. 2006. PHP & MySQL Teoriasta käytäntöön. Readme.fi.

Naramore Elizabeth, Gerner Jason, Le Scouarnec Yann, Stolz Jeremy & Glass Michael K. 2005. Beginning PHP5, Apache, and MySQL Web Development. Wiley Publishing.

Sadananda Ramakoti & Kujansuu Esa. 2007. ITE2007 proceedings.

Zandstra, Matt. 2000. Teach yourself PHP 4 in 24 Hours. SAMS.

WWW

Zervaas, Quentin. 2005. Migrating from MySQL to PostgreSQL in PHP

Muuta kirjallisuutta

The PostgreSQL Global Development Group.2006. PostgreSQL 8.2.0 Documentation

Matthew Neil & Stones Richard. 2005. Beginning Databases with PostgreSQL: From Novice to Professional, Second Edition. Apress.

Liitteet

Liite 1: SQL-komennot taulujen luomiseen ite-konferenssille:

```
CREATE TABLE ite_users (
  id serial NOT NULL,
  username character varying(50),
  "password" character varying(255),
  user_role character varying(50),
  first_name character varying(50),
  last_name character varying(50),
  email character varying(50)
CONSTRAINT ite_users_pkey PRIMARY KEY (id)
WITHOUT OIDS;
```

```
CREATE TABLE ite_ml_users (
  user_id serial NOT NULL,
  firstname character varying(255) NOT NULL,
  lastname character varying(255) NOT NULL,
  email character varying(255) NOT NULL
);
CREATE TABLE ite_ml_lists (
  ml_id serial NOT NULL,
  listname character varying(255) DEFAULT ""::character varying NOT NULL
) WITHOUT OIDS;
```

```
CREATE TABLE ite_ml_subscriptions (
  ml_id serial NOT NULL,
  user_id serial NOT NULL,
  pending integer DEFAULT 1 NOT NULL
) WITHOUT OIDS;
```

```
CREATE TABLE ite_guests (
  id serial NOT NULL,
  first_name character varying(50),
  last_name character varying(50),
  email character varying(50),
  organization character varying(50),
  country character varying(100),
  payment_method character varying(30),
  payment_status character varying(30),
  guest_status character varying(50)
) WITHOUT OIDS;
```

```
CREATE TABLE ite_balance (
  id serial NOT NULL,
  balance numeric
) WITHOUT OIDS;
```

```
CREATE TABLE ite_account_events (  
  id serial NOT NULL,  
  entry character varying(255),  
  income numeric,  
  expense numeric,  
  exchange_rate numeric,  
  date date,  
  explanation character varying(255),  
  author character varying(100),  
  currency character varying(10)  
) WITHOUT OIDS;
```

```
CREATE TABLE home_text (  
  id serial NOT NULL,  
  content text  
) WITHOUT OIDS;
```