

KEMI-TORNION AMMATTIKORKEAKOULU

Kevyt, käyttäjäystävällinen ja turvallinen yritysverkkosivusto

Renne Kaipainen

Tietojenkäsittelyn koulutusohjelman opinnäytetyö
Web-asiantuntijan suuntautumisvaihtoehto
Tradenomi

TORNIO 2011

TIIVISTELMÄ

Kaipainen, Renne 2011. Kevyt, käyttäjäystävällinen ja turvallinen yritysverkkosivusto. Opinnäytetyö. Kemi-Tornion Ammattikorkeakoulu. Kaupan ja kulttuurin toimiala. Tornio. Sivuja 48. Liitteet 1-6.

Tämä opinnäytetyö on tehty toimeksiantona AP Nurminen Oy:lle. AP Nurminen Oy on suomalainen LVI-alan yritys Vantaalta. Yritys halusi laajentaa liiketoimintaansa verkkomarkkinoinnin puolelle ja tarvitsi uudet verkkosivut.

Tämän opinnäytetyön tavoitteena oli suunnitella ja toteuttaa uusi verkkosivusto AP Nurminen Oy:lle. Sivuston piti olla kevyt käyttää, käyttäjäystävällinen ja mahdollisimman turvallinen. Sivustolla piti olla yhteydenottolomake vierailijoita varten ja tilauslomake yhteistyökumppaneille ja sopimusasiakkaille. Tilauslomakkeen käyttöä varten piti suunnitella ja toteuttaa kirjautumistoiminto ja sille salasanaikäytäntö. Kaikki sivuston lomakkeet piti suojata roskapostirobottien varalta.

Tässä opinnäytetyössä käytettiin konstruktivistista tutkimusmetodia ja kvalitatiivista kyselytutkimusta haastatteleamalla toimeksiantajaa. Lisäksi työssä käytettiin kvalitatiivista ja kvantitatiivista ohjelmistotestausta. Verkkosivu itsessään toteutettiin evoluutiomallin mukaan HTML- ja CSS-kuvauskielillä sekä JavaScript- ja PHP-ohjelmointikielillä.

Työn lopputuloksena syntyi kevyt, käyttäjäystävällinen ja turvallinen verkkosivusto toimeksiantajalle.

Asiasanat: CSS, HTML, JavaScript, PHP, käytettävyys

ABSTRACT

Kaipainen, Renne 2011. A light, user friendly and safe company website. Bachelor's Thesis. Kemi-Tornio University of Applied Sciences. Business and Culture. Tornio. Pages 48. Appendices 1-6.

This thesis was commissioned by AP Nurminen Oy. AP Nurminen Oy is a Finnish private HVAC-engineering company from Vantaa. The company wanted to expand to Internet marketing and they needed a new company website.

The purpose of this thesis was to design and implement a website to AP Nurminen Oy. The new website had to be light to use, user friendly and relatively safe. The website had to have an email contact form for visitors and an order form for partners and contract customers. The order form needed login functionality and a password protocol to follow. All email forms had to be protected against spambots.

In this bachelor's thesis a constructive research method was used with a qualitative survey study by interviewing the commissioner. Also qualitative and quantitative software testing was used for the execution. The website itself was created with an evolution method by using HTML markup language, CSS style sheet language, JavaScript and PHP scripting language.

As a result of this bachelor's thesis, a light, user friendly and safe company website was created for the commissioner.

Keywords: CSS, HTML, JavaScript, PHP

SISÄLTÖ

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO	6
1.1	Toimeksiantaja	7
1.2	Tutkimusmenetelmät	7
1.3	Tiedonhankintamenetelmät	8
1.4	Työkalut ja työkaluohjelmat	8
1.5	Keskeiset käsitteet	9
2	VERKKOSIVUSTON SUUNNITTELU	11
2.1	Käyttäjystävällinen ulkoasu ja rakenne	12
2.2	Palvelut sivuston käyttäjille	13
3	VERKKOSIVUSTON RAKENTEELLINEN TOTEUTUS	14
3.1	HTML-rakenne	14
3.2	CSS-tyylimääritykset	17
3.3	JavaScript ja POPUP-ikkunat	18
4	VERKKOSIVUSTON PALVELUT JA PALVELINOHJELMOINTI	20
4.1	Verkkosivuston toiminta ja tietovirrat	20
4.2	Verkkosivuston HTML- lomakkeet	23
4.3	CAPTCHA- ja RECAPTCHA-tunnistukset	25
4.4	PHP-palvelinohjelmointi verkkosivustolla	26
4.5	CAPTCHA-tunnistuksen PHP-toteutus	27
4.6	PHP-posti- ja lomakekäsittelijä	28
4.7	Kirjautuminen tilauspalveluun	31
5	VERKKOSIVUSTON TESTAUS JA SUORITUSKYKY	33
5.1	Testaustyökalut ja -metodit	33
5.2	Havainnot ja evoluutiomalli	34

6	YHTEENVETO	35
	LÄHTEET	38
	LIITTEET	40

1 JOHDANTO

Internetissä julkaistuista verkkosivuista on tullut erittäin tärkeä, näkyvä ja tehokas osa yritysten markkinointia. Mainostaminen onkin nykypäivänä siirtynyt suurelta osin muista mediaformaateista Internetin puolelle, jossa markkinointikustannukset ovat alhaiset ja näkyvyys hyvä. Internetin laaja levinneisyys mahdollistaa myös erittäin suurten kohderyhmien tavoittamisen. Siksi yritykset ovat alkaneet mainostaa entistä enemmän verkossa ja uudistaneet sivustojaan nopeaan tahtiin. Liikkuvaa kuvaa ja animaatiota lisätään sivuille jatkuvasti, sisältöä kasvatetaan pienoissovelluksilla, kuten minipeleillä ja horoskoopeilla ja lopputuloksena saadaan aikaiseksi kaoottinen välkkyvä kokonaisuus, jota varsinkaan vanhemmat tietokoneet eivät kykene kunnolla käsittelemään.

Verkkosivujen houkuttelevuutta tavoiteltaessa liian suureksi kasvatetulla viihdesisällöllä, kirjavalla ulkoasulla tai monimutkaisella sivustorakenteella voi olla kuitenkin päinvastainenkin vaikutus. Mikäli sivustosta on vaikea löytää nopeasti etsimäänsä asiaa, tietokone ei jaksa pyörittää sitä kunnolla tai sitä on muuten vaan liian vaikea käyttää, informaatiohakuinen vierailija siirtyy nopeasti seuraavaan, itseään enemmän miellyttävään vaihtoehtoon. Tarjontaa on kyllä verkossa riittävästi.

Internetin suurimpina ongelmina ovat kuitenkin lisääntyvä roskapostitus ja tahaton tai tahallinen haittaohjelmien levittäminen. Roskapostirobotit eli automaattiset roskapostitusohjelmat selaavat verkkoa yrittäen löytää näkyviä sähköpostiosoitteita ja suojaamattomia yhteydenotto- ja tilauslomakkeita hyödyntääkseen niitä apuvälineinä roskamainonnalle. Kaikelta luvattomalta toiminnalta ei voi suojautua mutta sitä voi vaikeuttaa ja hidastaa huomattavasti.

Tämän opinnäytetyön tavoitteena oli suunnitella ja rakentaa käyttäjäystävällinen, toiminnaltaan kevyt, mahdollisimman tietoturvallinen ja helppokäyttöinen verkkosivusto markkinointi- ja yhteydenpitotarkoituksiin AP Nurminen Oy:n käyttöön niillä työkaluilla, mitä käytettävissä on. Vaikkakaan varsinainen asiasisällöntuotto verkkosivustolle ei alun perin kuulunut tämän opinnäytetyön aihealueeseen, tyylin ja värimaailman oli oltava visuaalisesti yhteensopivia yrityksen tuotteiden ja palveluiden markkinointiulkoasujen kanssa.

1.1 Toimeksiantaja

AP Nurminen Oy on LVI-alan yritys, joka työllistää viidestä kymmeneen henkilöä riippuen kausitilanteesta. Tällä hetkellä vakituisia työntekijöitä on seitsemän ja yritys tekee yhteistyötä mm. Tikkurilan ammattioppilaitoksen kanssa kouluttamalla harjoittelijoita työtehtäviin.

Aikaisemmin yritys on tuottanut erilaisia LVI-palveluita, mutta on nyt erikoistumassa maa-, ilma- ja vesilämpöpumppujen myyntiin, asennukseen sekä suunnitteluun. Tulevaisuudessa yrityksen tarkoituksena on luopua vähitellen kokonaan muusta LVI-toiminnasta ja ohjata markkinointi- ja myyntitoiminta pääosin verkkoon. Yrityksellä on edustus- ja yhteistyösopimuksia useiden LVI-alan kaluste- ja laitevalmistajien kanssa, esimerkkeinä: Thermia, Thermofloor, Oras, Hansgrohe, Upofloor ja Uponor. Monet talovalmistajat ja rakennusliikkeet käyttävät yrityksen palveluksia. Markkinointinäkyvyyttä yhteistyökumppaneiden kanssa halutaan myös tulevaisuudessa lisätä.

AP Nurminen Oy:llä on ollut aikaisemmin käytössään yksinkertainen, lähinnä yhteystiedot sisältävä verkkosivu. Yritys tarvitsi uudet markkinointiin soveltuvat verkkosivut vanhan mainossivun tilalle. Sivustosta piti löytyä yhteys- ja yritystietojen lisäksi yhteydenottopyyntölomake kaikille asiakkaille ja salasanasuojattu viestintä- ja tilausosio yhteistyökumppaneille.

1.2 Tutkimusmenetelmät

Opinnäytetyössä on käytetty tutkimusmetodina konstruktivistista menetelmää jolla kehitetään ratkaisuja tutkimusongelmiin. Metodia käytetään laajalti tietotekniikan sovellutusten suunnittelussa ja kehitystyössä. Konstruktivisessa tutkimuksessa tavoitteena on relevantin käytännön tai teoreettisen ongelman ratkaisu luomalla uusi konstruktio ratkaisuksi (Ryabov, 2009).

Tutkimustyö suoritettiin kahdessa vaiheessa. Ensimmäisessä vaiheessa määriteltiin verkkosivuston käyttötarkoitus ja toimeksiantajan siihen kohdistamat toiveet. Toinen vaihe koostui ensimmäisen vaiheen tulosten perusteella tehdyn prototyypimallin

testaamisesta ja arvioimisesta. Tutkimuksen tavoitteena oli löytää tarvittavat vastaukset keskeisiin tutkimusongelmiin.

Tutkimusongelmiksi määriteltiin kaksi keskeistä asiaa:

- Mitä yritys todellisuudessa verkkosivustoonsa tarvitsee eli mihin yritys tulee sitä käyttämään?
- Minkälainen verkkosivusto on rakenteeltaan ja toiminnoiltaan tarkoitukseensa sopivin?

1.3 Tiedonhankintamenetelmät

Kerätessä tietoa yrityksen tarpeista ja suunnitelmista verkkosivuston käytön suhteen tiedonhankintamenetelmänä käytettiin haastattelututkimusta, jossa haastateltavana oli toimeksiantaja. Kerätessä tietoa verkkosivuston rakenteellista ja toiminnallista kehitystyötä varten menetelmänä käytettiin ohjelmistotestausta. Haastattelututkimus oli kvalitatiivista eli laatuun perustuvaa ja ohjelmistotestaus yhdistelmä kvalitatiivista sekä kvantitatiivista eli määrään perustuvaa tutkimusta.

1.4 Työkalut ja työkaluohjelmat

Opinnäytetyötä tehdessä käytettiin kolmea tietokonetta. Yhteen koneista asennettiin Linux-pohjainen Ubuntu Server-käyttöjärjestelmä ja Apache-palvelinohjelma. Apache-http-palvelinohjelman avulla testattiin verkkosivuston toimivuutta kehitystyön edetessä. Myöhemmin testialustana käytettiin myös XAMPP-palvelinohjelmistoa ja Elisan PHP-tuella web-palvelinympäristöä. Kahdessa muussa tietokoneessa käytettiin Windows-pohjaisia käyttöjärjestelmiä, toisessa XP:tä ja toisessa Windows 7:ää. Molempia käytettiin verkkosivuston rakentamiseen ja toimivuuden varmistamiseen eri selainohjelmien kanssa. Windows tietokoneisiin oli asennettu Internet Explorer-selaimen lisäksi Mozilla Firefox- ja Google Chrome-selainohjelmat. Kaikissa koneissa oli myös ajanmukainen virustorjuntaohjelmisto, Ubuntu-koneessa ilmainen Avast Free Antivirus ja molemmissa Windows-koneissa F-Secure AVCS-tietoturvaohjelmisto sekä viimeisimmät ohjelmistopäivitykset. Työkaluohjelmina käytettiin Microsoftin Officea,

Notepadia sekä lisäksi verkosta saatavaa ilmaisen lähdekoodin GIMP-kuvankäsittelyohjelmaa.

1.5 Keskeiset käsitteet

Apache	On Avoimeen lähdekoodiin perustuva ilmainen http-verkkopalvelinohjelma, jossa on tuki PHP-ohjelmointikielelle. (The Apache Software Foundation 2010)
CAPTCHA	Completely Automated Public Turing Test To Tell Computers and Humans Apart. Ohjelma, jolla pyritään erottamaan tietokoneet ja ihmiset toisistaan. (CAPTCHA 2010)
CSS	Cascading style sheets, tyyliohjetiedosto HTML-sivujen näyttämiseen. (w3schools 2010a)
HTML	Hypertext Markup Language, verkkosivujen rakentamisessa käytetty kuvauskieli. (w3schools 2010b)
JavaScript	JavaScript-komentosarjakieli on kevyt ohjelmointikieli, jota käytetään yleensä HTML-sivuun upotettuna tuottamaan interaktiivista sisältöä. (w3schools 2010c)
LeechFTP	Ilmainen ja helppokäyttöinen tiedostojen siirto-ohjelma, jolla voidaan siirtää verkkosivujen sisältö palvelimelle. (LeechFTP 2010)
PHP	Hypertext Preprocessor, www-palvelimella suoritettava komento- ja ohjelmointikieli verkkosivujen dynaamista ja interaktiivista sisältöä varten. (w3schools 2010d)

Ubuntu	Avoimeen lähdekoodiin perustuva ilmainen käyttöjärjestelmä. (Ubuntu Suomi 2010)
XAMPP	Windows- ja Linux-ympäristöön tehty palvelinohjelmisto, joka sisältää Apache-http-palvelinsovelluksen. (ApacheFriends 2010)

2 VERKKOSIVUSTON SUUNNITTELU

Verkkosivujen suunnittelu aloitettiin analysoimalla toimeksiantajan toiveita sisällön ja ulkoasun suhteen sekä määrittelemällä sivuston nykyinen ja tuleva käyttötapa. Sisältötarpeita kartoitettaessa apuna käytettiin haastattelututkimusta. Tutkimuksessa haastateltiin toimeksiantajaa ja keskityttiin kartoittamaan yrityksen verkkosivujen tämänhetkinen ja tuleva käyttötarkoitus, yrityksessä lähitulevaisuudessa tapahtuvat mahdolliset, ennakoitavat ja suunnitellut muutokset sekä yritystoiminnan kannalta tärkeät toiminnot ja niiden päivitysmahdollisuudet. Tehtävää varten oli laadittu suuntaa-antava pienimuotoinen kyselykaavake (Liite 1).

Sivuston toimivuus ja hyvä käytettävyys pyrittiin varmistamaan seuraamalla Jakob Nielsenin (2010) heuristisia käytettävyysääntöjä siinä määrin kuin se oli mahdollista toimeksiantajan toiveita noudattaen. Nielsen määrittelee käytettävyydelle seuraavat ehdot:

- Sivusto on suunniteltava mahdollisimman kevyeksi poistamalla tarpeeton dialogi.
- Käyttäjien muistikuorma on minimoitava näyttämällä kaikki vaihtoehdot jatkuvasti.
- Käyttäjille on annettava palautetta kaikista toimista joko näkyvillä tapahtumilla tai ilmoituksilla.
- Käyttäjillä on oltava vapaus valita mitä he sivustossa tekevät, mitään ei saa avata, sulkea tai näyttää väkisin.
- Sivustossa on käytettävä lokalisoitua ”normaalimaailman” kieltä ja ilmaisia sisältöissä.
- Ohjeita ja apua on saatava tarvittaessa.
- Sivustossa liikkuminen on tehtävä käyttäjille vaivattomaksi ja tehokkaaksi. Myös pakotiet on sallittava.
- Käyttöliittymän on oltava yhdenmukainen. Samojen asioiden on tarkoitettava samaa joka paikassa.
- Kaikki virhetilanteet on pyrittävä ehkäisemään.
- Jos virhetilanne pääsee syntymään, siitä on annettava selväkielinen virheilmoitus eikä esimerkiksi epämääräistä virhekoodia.

Yritys oli aloittamassa edustus-, tilaus- ja markkinointiyhteistyön tavarantoimittajiensa kanssa ja siksi verkkosivuston ulkoasussa ja värimaailmassa piti huomioida myös yhteistyökumppaneiden näkyvyys ja sovittaa se tyyllillisesti ja hyväksyttävästi omaan markkinointiympäristöön. Sopimuksen mukaan yhteistyökumppaneiden piti hyväksyä verkkosivuston graafinen ulkoasu ennen kuin heidän kuvalinkejään, yrityslogojaan ja markkinointisisältöään saatiin käyttää. Tästä syystä johtuen suunnitteluvaihe painottui enemmän sivuston rakenteelliseen kuin tyyllilliseen ulkoasuun.

2.1 Käyttäjätystävällinen ulkoasu ja rakenne

Sivuston ulkoasua suunniteltaessa päätettiin varmistaa sen helppo käytettävyys pitämällä ulkoasu mahdollisimman yksinkertaisena ja muuttumattomana eli staattisena. Sivuston navigointivalikot pysyisivät kiinteästi samanlaisina ja samoissa paikoissa ja sisältöä päivitettäisiin vain yhteen osaan eli sivuston sisältöosaan. Toimeksiantajan kanssa etsittiin esimerkkejä rakenteeltaan sopivista sekä myös ulkonäöltään hyvältä näyttävistä sivumalleista muun muassa Internetiä selaamalla. Lopulta parhaimmaksi rakennemalliksi valittiin perinteinen otsikko - yläpalkki - vasen sivupalkki - sisältö - oikea sivupalkki - alapalkki -rakenne. Kyseisen suunnittelumallin todettiin antavan parhaat mahdollisuudet erillisten osien päivittämiseksi, laajentamiseksi ja muokkaamiseksi tulevaisuudessa kokonaisrakenteen siitä kärsimättä.

Ulkoasua suunniteltaessa päätettiin myös huomioida sivuston helppo katselu ja käyttö matalaresoluutioisilla näytöillä varustetuilla sekä hitaampia verkkoyhteysnopeuksia käyttävillä tietokoneilla ja kannettavilla laitteilla. Sivusto päätettiin avata selaimella rajatun kokoisena ja sivuston vieritys eli scrollaus rajoitettuna. Aukeavan ikkunan leveyttä ei kuitenkaan haluttu pienentää liikaa mobiililaitteita varten ulkonäön ja kirjasinkoon kustannuksella. Tekstin ja ikkunan kokosuhte päätettiin pitää kutakuinkin standardoidun pikseli per pikseli A4-koon mukaisena. Tekstuurit eli taustakuviointit, värit ja fontit eli kirjasinmallit päätettiin pitää mahdollisimman yksinkertaisina ja selvinä sekä kaikille yleisimmille selaimille yhteensopivina.

Sivuston rakennetta suunniteltaessa huomioitiin myös se tosiseikka, että suunnitteluvaiheessa valmiin kokonaiskuvan muodostaminen sisällöstä ja ulkoasusta ei ollut vielä mahdollista kaikilta osin. Markkinointisisältö ei ollut vielä kaikelta osin

valmiina ja lisäksi yhteistyökumppaneilla oli mahdollisuus esittää omat näkemyksensä ja muutosehdotuksensa ennen sivuston julkaisua.

2.2 Palvelut sivuston käyttäjille

Toimeksiantaja halusi tarjota tavanomaisen informaatio- ja markkinointisisällön lisäksi verkkosivuston palveluna yhteydenottomahdollisuuden kaikille vierailijoille suoraan sivuston kautta. Verkkosivustolle tarvittiin julkisen yhteydenottoväylän lisäksi rajattu ja turvallinen väylä yhteistyökumppaneiden ja sopimusasiakkaiden tilauksille ja tilausvahvistuksille. Tarkoituksia varten sivustolle päätettiin rakentaa lomakekäsittelijän valvoma lomake yhteydenottoihin ja käyttäjien tunnistautumista edellyttävä suojattu viestintä- ja tilausosa kirjautumistoimintoineen sekä sopivine salasanakäytäntöineen.

Yrityksen aiempi mainosverkkosivu oli ollut palvelimella, joka tuki vain HTML- ja kuvatiedostojen siirtoa selaimille ja yrityksen yhteystietojen yhteydessä näkyi myös suora linkitetty sähköpostiosoite yhteydenottoja varten. Yhteydenotot ja tilaukset suoritettiin suoraan asiakkaiden omien sähköpostiohjelmien kautta ja postia lähetettäessä osoite tallentui asiakkaiden sähköpostiosoitelista haittaohjelmien löydettäväksi. Roskapostista olikin tullut melko suuri haittatekijä yritykselle. Roskapostiroboteilta suojautuminen, näkyvien sähköpostiosoitteiden poistaminen ja tahattomien sekä tahallisten häiriöviestien estäminen asetettiin päätavoitteiksi palveluiden sisältöä ja toiminnallisuutta toteutettaessa.

Tarpeeksi tehokkaan suojauksen toteuttaminen edellyttäisi palvelinohjelmointia tukevan palvelinympäristön käyttöönottoa ja siihen sopivan ohjelmointikielen käyttöä. Siksi verkkosivuston sisällön sijoituspaikaksi päätettiin valita jonkun suuren ja luotettavan palveluntarjoajan yrityspalvelinympäristö, josta löytyisi käytettävien ohjelmointikielien tukien lisäksi oma sähköpostipalvelin. Suurten palveluntarjoajien palvelut ovat myös melko luotettavia ja hyvin suojattuja ja siksi suorista palvelintietomurroista aiheutuvat riskit voitiin käytännössä unohtaa. Suurilla palveluntarjoajilla on myös yleisesti ottaen parempi toipumiskyky virhetilanteista ja paremmat resurssit valvontaan kuin pienillä. Palvelinohjelmointi päätettiin toteuttaa PHP-ohjelmointikielillä.

3 VERKKOSIVUSTON RAKENTEELLINEN TOTEUTUS

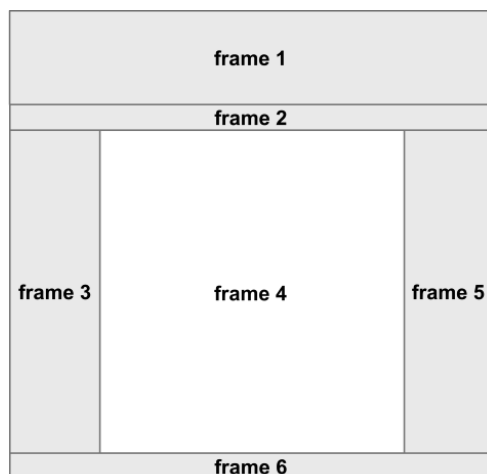
Verkkosivuston toteutuksessa käytettiin HTML- ja CSS-kuvauskieliä ulkoasun ja rakenteen muotoiluun. Lisäksi sivustolle lisättiin toiminnallista ohjelmointia, JavaScriptiä ja palvelimella suoritettavaa PHP-ohjelmointia. Kuvauskielten ja ohjelmointikoodien kirjoittamiseen käytettiin Windows-käyttöjärjestelmien mukana tulevaa kevyttä Notepad-tekstieditoria. Tehdyt Notepad-tekstitiedostot tallennettiin .txt-tiedostomuodon sijasta .html- , .css- ja .php-tiedostomuodoiksi, jolloin selain ja palvelimen PHP-tulkki osasivat lukea ja käsitellä niitä oikein.

3.1 HTML-rakenne

Matalimmat nykyisin yleisesti käytössä olevat näyttöresoluutiot ovat 1024 x 768 näyttöpistettä eli pikseliä. Joissakin kannettavissa näytön tarkkuus on korkeudeltaan vain 600 pistettä mutta leveys minimissään 1024 pistettä. Avattavan ikkuna päätettiin pitää 800-900 pikseliä leveänä. Toimeksiantaja itse käyttää telakoitavaa kannettavaa, jonka näyttöresoluutio on 1280 x 800 pikseliä ja telakoitaessa päänäytön koko on 1920 x 1200 pikseliä.

Sivusto aukeaa selainikkunan keskelle 870 pistettä leveänä ja 820 pistettä korkeana. Määrittelyssä käytettiin IFRAME-rakennetta. Vieritystä ei sallittu muuten kuin selaimen oman ikkunan pystysuunnassa. Rakenne toteutettiin HTML-kuvauskielillä (Liite 2.1). Sivuston koko määriteltiin kiinteäksi, mutta jos tilaa tarvitaan tulevaisuudessa lisää esimerkiksi mahdollisen verkkokaupan tuoteluetteloita varten, sen laajeneminen ylä-ala suunnassa sallitaan.

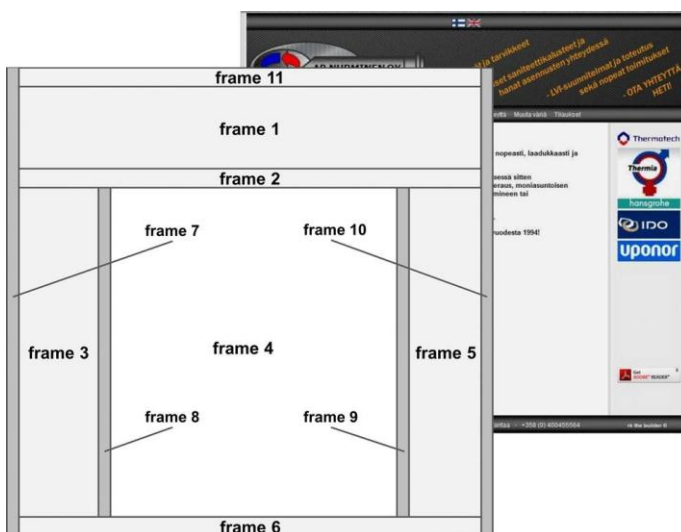
IFRAME-ikkunan sisälle aukeava sisältö määriteltiin suunnitelman mukaisella FRAMESET-rakenteella, jossa oli kuusi erillistä HTML-osaa eli periaatteessa jokainen FRAME-raami oli oma erikseen muotoiltava sisältösivunsa (Liite 2.2). FRAMESPACING- ja FRAMEBORDER-attribuuteilla määriteltiin raamien väliset tyhjät tilat ja rajat pois sekä NORESIZE-attribuutilla määrättiin koko kiinteäksi niin, että verkkosivun käyttäjä ei voi muuttaa raamin kokoa hiirellä vetämällä. Lopputulos oli alla olevan kuvan mukainen, raamien rajaukset on lisätty tarkastelua helpottamaan (Kuva 1).



Kuva 1. Ensimmäinen raamirakenne

Jokaiselle FRAME-osalle määrättiin oma käyttötarkoituksensa. ”frame 1” eli raami 1 oli HEADER-osaa eli otsikkoa varten, ”frame 2” oli ylävalikko linkeille, ”frame 3” oli vasen valikko linkeille, ”frame 4” oli tarkoitettu avattavalle varsinaiselle sisällölle, ”frame 5” oli oikea valikko yhteistyökumppaneiden logoista koostuville linkeille ja ”frame 6” oli alapalkki yhteystietoja, linkkejä ja muuta tarpeellista informaatiota varten.

Seuraavassa vaiheessa rakennetta muutettiin visuaalisista syistä monimutkaisemmaksi lisäämällä ylhäältä alas asti ulottuvat raamit alkuperäisen rakenteen molemmille puolille ja vasemman ja oikean valikkoraamin sisäpuolelle. Lisäksi yläreunaan otsikon yläpuolelle lisättiin uusi valikkopalkki (Kuva 2).

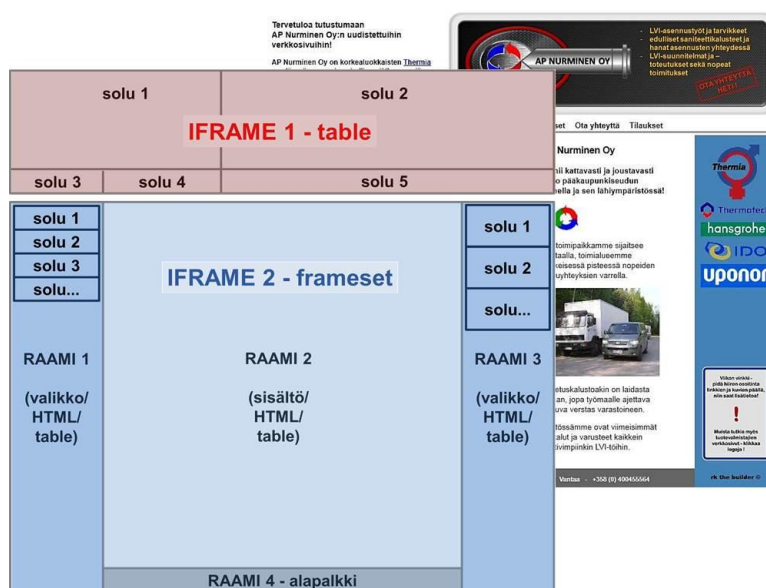


Kuva 2. Toisen version raamirakenne ja kuvatäytöt

Ylimääräiset raamit toimivat alustana tehosteina käytettäville kuvatiedostoille, joilla haluttiin antaa sivuille toimeksiantajan haluamaa kolmiulotteista ja teknistä ilmettä. Täyttönä käytettiin melko korkeatarkkuuksisia materiaalikuvia metallista ja hiilikuidusta. Rakenne oli kuitenkin raskas hidasta verkkonopeutta käytettäessä ja siksi sitä päätettiin keventää poistamalla mahdollisimman paljon sivustolle avautuvaa kuvamateriaalia. Asiasta kerrotaan lisää verkkosivuston testaus-osiossa.

Kolmannessa ja viimeisessä rakenneversiossa sivustoa muokattiin kevyemmäksi poistamalla mahdollisimman monta raamia käyttämällä apuna TABLE- taulukkorakennetta. Raamien käytössä oli myös ongelmansa muotoilujen suhteen käytettäessä CSS-tyylitiedostoja, koska FRAMESET-HTML-dokumenttityyppi ei tue tyylitiedostomuotoa oikein. IFRAME-rakenne säilytettiin käyttämällä kahta allekkain olevaa 860 pikseliä leveää IFRAME-osaa. Ylempi HEADER-otsikko-osalle ja alempi, varsinainen sisältö-osa, navigointipalkeille sekä varsinaiselle sisällölle (Liite 2.3).

Ikkunan vierittäminen sekä osien välit poistettiin ja rajat piilotettiin. HEADER-osan sisällölle sijoitettiin paikat TABLE- eli taulukko-rakenteella (Liite 2.4). Sisältö-osassa oli käytettävä edelleen jonkin verran raami-rakennetta, koska taulukkoon ei voitu määritellä avattavaksi uutta sisältösivua HTML-kuvauskielellä ja valikot haluttiin pitää muuttumattomina entisissä paikoissaan (Liite 2.5). Perusrakenne oli nyt valmis ja ulkoasun viimeistely alkoi (kuva 3).



Kuva 3. Valmis sivustorakenne

Tästä eteenpäin rakennetta ja ulkoasua käsiteltäessä keskitytään valmiiseen rakenneversioon.

3.2 CSS-tyylimääritykset

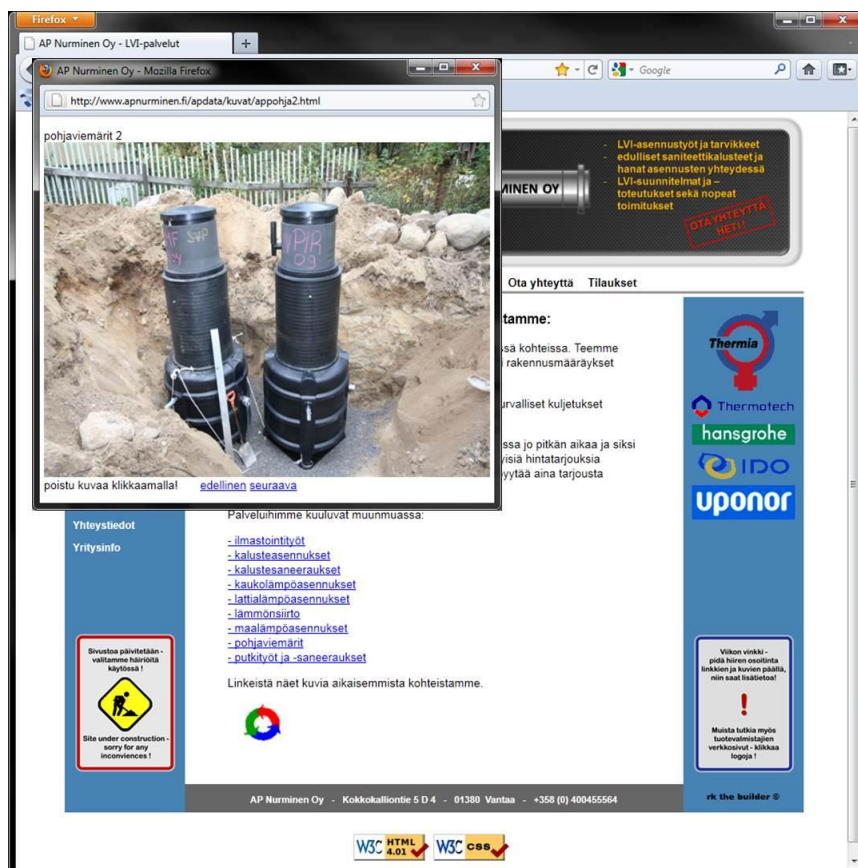
Sivuston ulkoasun määrittelyyn käytettiin CSS-tyylitiedostoja, joista verkkoselain hakee HTML-dokumenttien HEAD-elementtiin sijoitetun linkin perusteella tiedot, joita se käyttää tulostaakseen halutunlaisen tekstin, tehosteet ja väri- sekä tyylimaailman selaimen omien tyylimuotoilujen tilalle. CSS-tyylimääritykset voi myös sijoittaa eli upottaa suoraan HEAD-elementtiin ja molempia voidaan käyttää yhtä aikaa samassa dokumentissa. Sivuston HTML-dokumenteissa käytettiin upotettuja tyylimäärityksiä ja linkitettyjä tyylitiedostoja sekä jonkin verran molempien yhdistelmää.

Suurin osa muotoiluista toteutettiin erillisellä CSS-tyylitiedostolla. Upotettuja tyylimäärittelyjä käytettiin vain muutamassa sivuston osassa lähinnä taustavärien vaihtamiseen tai tekstin ja kuvien paikkojen ja kokojen hienosäätöön. Erillisen tyylitiedoston etuna upotettuihin määrittelyihin verrattuna on mahdollisuus tehdä vain yksi tiedosto, jolla voidaan määrätä koko verkkosivuston ulkoasu nimeämällä osat erikseen ja merkitsemällä ne HTML-dokumentteihin.

Sivuston HTML-dokumenttien HEAD-elementtiin laitettu linkki ohjaa selaimen hakemaan tyyli1-nimisen tyylitiedoston mukaiset ohjeet muotoiluun (Liite 3.1). Upotettu BODY-tyylimäärittely määrittelee taustavärien, sisällön marginaalin sivun ylälaitaan pikseleinä ja OVERFLOW-tribuutti piilottaa joissakin selaimissa scrollauskiellosta huolimatta näkyvän vierityspalkin ikkunan kokoa vetämällä muutettaessa. Valikkosivu esimerkiksi muotoiltiin CSS-tiedostolla, jossa BODY-osassa määritellään sivun ylämarginaali, tekstin väri, fontti ja tekstin koko pienemmäksi (Liite 3.2). Lisäksi tiedostolla määritellään kahden ”level1”- ja ”level2”-tunnisteilla merkityn linkin muotoilu samalla sivulla. Linkit voivat olla numeroituna joko saman tai eri HTML-dokumentin sisälle. Ensimmäisen linkin fontti on muutettu täysikokoiseksi, lihavoitu ja määritelty näyttämään tavalliselta tekstiltä, ei linkiltä. Toisessa on lihavointi poistettu ja teksti määritelty valkoiseksi. Viimeinen määrittely puolestaan muuttaa linkin värin hiiren osoittimen ollessa sen päällä.

3.3 JavaScript ja POPUP-ikkunat

Koska verkkosivuston aukeavan ikkunan kokoa rajoitettiin myös kannettaville tietokoneille sopivaksi ja muuttuva sisältö esitetään pääasiassa sivuston keskiosassa eli sisältö-osassa, kuvamateriaalin esittäminen vaikeutui tilanpuutteesta johtuen. Kuvien lukumäärä ei ole tällä hetkellä suuri, mutta tulevaisuudessa niiden määrä mitä todennäköisimmin moninkertaistuu ja niitä varten ei haluttu avata uutta välilehteä tai toista täysikokoista selainikkunaa. Lisäksi kuvalinkit haluttiin pitää koko ajan nähtävissä navigoinnin helpottamiseksi. Kuvien näyttäminen sivuston keskiosassa olisi vaatinut uuden linkkivalikon aukaisemista jossain muussa sivuston osassa ja sitä ei haluttu käytettävyyden kustannuksella tehdä. Ratkaisuksi keksittiin POPUP-ikkunoiden käyttö (Kuva 4).



Kuva 4. POPUP-kuvasivu Firefox-selainikkunassa

Sivusto päätettiin kolmiulotteistaa lisätilan saamiseksi normaalin selainikkunan päälle avautuvilla pienemmillä POPUP- eli ponnahdusikkunoilla. Ponnahdusikkunoihin ei aina

suhtauduta myötämielisesti Internetyhteisöissä johtuen niiden kyseenalaisesta maineesta pakkomainonnan apuvälineinä. Niinpä sivuston ikkunoiden käytössä päätettiin pysyä tarkoissa rajoissa ja Nielsenin sääntöjä mahdollisimman tarkasti soveltaen. Yhtään ikkunaa ei avata automaattisesti, niitä saa olla auki vain yksi kerrallaan ja niiden pitää myös sulkeutua itsestään ettei ylimääräisiä selainikkunoita jäisi päällekkäin auki.

Ratkaisu toteutettiin HTML-dokumentteihin upotetulla JavaScript-ohjelmointikielillä komentokehoitteilla. ”Kovalinkki”-sivun HTML-dokumentissa määriteltiin HEAD-elementtiin upotetulla FUNCTION POPUP -komentokehoitteella avattavaksi uusi ikkuna ja siinä avautuvan sivun linkki, ikkunan nimi, koko ja lisäksi sallittiin sen suurentaminen ikkunan nurkasta vetämällä (Liite 4.1). Kun ikkunan linkki jätetään määrittelemättä, ponnahtusikkuna aukeaa ilman etukäteen määrättyä sisältöä ja sivulla olevasta valikosta valittu kuvasivu avataan siinä. Napautettaessa haluttua kovalinkkiä ONCLICK-komento laukaisee FUNCTION POPUP -komentokehoitteen ja ”kuvasivu1” näytetään ”ruutu”-nimisessä ikkunassa, jonka koko on 530*450 pikseliä.

Jokainen kuva päätettiin esittää omana HTML-sivunaan eikä pelkkänä kuvatiedostona, koska ikkunan automaattinen sulkeutuminen voidaan määritellä HTML-dokumenttiin upotetulla JavaScript ONFOCUS/ONBLUR-tapahtumalla (Liite 4.2). Klikattaessa jotain muuta kohtaa näytöstä kuin avattua ikkunaa eli fokuoitaessa sivuston johonkin muuhun kohtaan, näytettävän sivun BODY-elementtiin upotettu ONBLUR-tapahtumakäsittelijä välittää tiedon selaimelle. Selain laukaisee SELF.CLOSE-komennon ja sulkee ponnahtusikkunan. Haluttaessa ikkuna voidaan myös sulkea kuvaa napauttamalla, jolloin kuvasivun kovalinkin ONCLICK-tapahtumakäsittelijä pyytää selainta sulkemaan ikkunan.

JavaScriptiä käytettiin aluksi myös sivuston lomakkeiden käsittelyssä, mutta siitä luovuttiin, koska jotkut, tosin harvinaisemmat, selaimet eivät tue kyseistä komentokehoitekieltä ja joissain tietokoneissa JavaScript-tuki on poistettu käytöstä joko tietoisesti tai tiedostamatta. Tästä syystä dynaamisen ohjelmasisällön suorittaminen siirrettiin palvelinkoneelle PHP-ympäristöön. POPUP-kuvaikkunoiden avaaminen oli sivustolla aina vapaaehtoista ja JavaScript-tuen puuttuminen ei häirinnyt sivuston muiden osien käyttöä.

4 VERKKOSIVUSTON PALVELUT JA PALVELINOHJELMOINTI

Verkkosivuston sisältö- ja palvelutarve määriteltiin haastattelemalla toimeksiantajaa ennen sisältö- ja rakennesuunnittelun alkua. Haastattelun aikana toimeksiantaja kertoi haluavansa sivuston valikkoon yhteydenottolinkin, jonka avulla kaikki sivustolla vierailevat voivat halutessaan lähettää palautetta ja yhteydenottopyynnön yrityksen sähköpostiosoitteeseen. Toivelistaan kuului myös mahdollisuus postin lajitteluun ennen sen aukaisua. Lisäksi sivustolle tarvittiin suojattu postiyhteys yhteistyökumppaneiden ja toimeksiantajan välille nopeita tilauksia ja tilausvahvistuksia varten.

Toimeksiantajan aikaisemmalla mainossivulla oli ollut kaikille näkyvä selväkielinen ja suoraan linkitetty sähköpostiosoite yhteydenottoja varten. Siksi yritykselle tuli jatkuvasti roskapostin lisäksi sekalaista postia asiakkailta ja muilta yrityksiltä. Lähetetty posti koostui yhteydenottopyynnöistä, tarjouspyynnöistä ja muiden yritysten myyntitarjouksista. Mukana oli myös yritykselle tahallisesti lähetettyä häirikköpostia. Pelkästään roskapostia tuli yrityksen sähköpostiosoitteeseen päivittäin enemmän kuin asiallista postia viikoittain. Siksi roskapostin ehkäisystä tuli yksi sivuston tärkeimmistä toiminnoista.

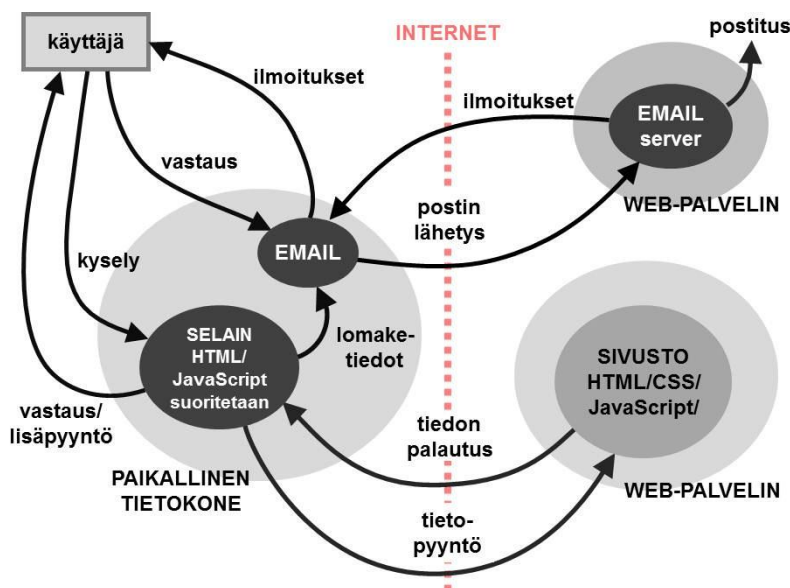
Verkkosivustoa suunniteltaessa pohdittiin erilaisia vaihtoehtoja sivuston avulla käyttäjiltä kerätyn tiedon lajitteluun ja roskapostiestoon. Melko nopeasti selvisi se tosiasia, että ilman erillisiä postitus- ja tilauslomakkeita ja palvelimella suoritettavaa dynaamista ohjelmointia ei toiminnallisia eikä turvallisuuteen liittyviä tavoitteita pystyttäisi täyttämään. Käyttämällä tarkoituksenmukaisiksi muokattuja lomakkeita ja käsittelemällä ne palvelimella suoritettavilla ohjelmilla sähköpostiosoite pystyttäisiin piilottamaan ja lomakkeiden sisältö tarkistamaan sekä lähetysoikeus varmentamaan eli autentikoimaan.

4.1 Verkkosivuston toiminta ja tietovirrat

Verkkosivustolle siirryttäessä eli sivustoa avattaessa asiakastietokoneen verkkoselain ottaa yhteyden käyttäjän antaman verkko- tai kotisivuosoitteen mukaiseen verkkopalvelimeen ja pyytää sieltä sivuston tarkasteluun tarvittavat tiedostot. Jos osoite on löydettävissä ja pyyntö on oikeutettu, palvelimen palvelinohjelmisto lähettää

tiedostot verkkoselaimelle. Selain tulostaa sivuston graafisen, jos sellainen on, käyttöliittymän näytölle ja kaikki tiedostot saatuaan katkaisee yhteyden palvelimeen. Toiminto on ainutkertainen eli aina, kun selaimen ikkunassa olevalta näkymältä halutaan vaihtaa toiselle tai sivuston muuttuvia osia päivitetään, tiedostot täytyy hakea verkkopalvelimelta uudestaan.

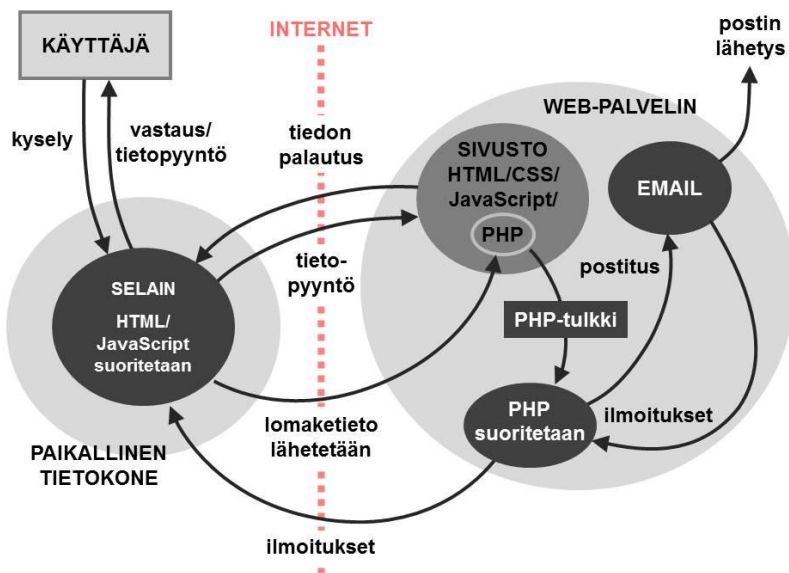
Nykyisin lähes jokaisella verkkosivustoilla on jonkinlainen yhteydenotto- tai palautemahdollisuus vierailijoille. Yleensä sivustoilla käytetään suoraa sähköpostiosoitetta tai -linkkiä, mutta paras ja tietoturvallisin ratkaisu olisi käyttää tarkoitukseen sopiviksi muokattuja sähköisiä lomakkeita. Lomaketiedostot pyydetään verkkopalvelimelta samalla tavalla kuin muutkin selaimella esitettävät tiedostot ja kun niihin täytetyt tiedot halutaan lähettää vastaanottajalle on olemassa kaksi vaihtoehtoa. Lomakkeen toimintamäärittelyssä sillä kerätyt tiedot voidaan määrätä lähetettäväksi joko selainkoneen sähköpostiohjelman tai verkkosivupalvelimen palvelinohjelmiston käsiteltäväksi. Mikäli käsitelijäksi valitaan paikallinen selaintietokone, lomakkeen tiedot ja lähetysosoite tallentuvat käytettävän sähköpostiohjelman kansiorakenteeseen haittaohjelmien ja luvattomien käyttäjien löydettäväksi. Sähköpostiosoitetta ei voi salata vaan se näkyy selainkoneella esteettä ja sitä voidaan käyttää myös tarkoituksellisesti väärin (Kuva 5).



Kuva 5. Lomakkeen lähetyks paikalliselta tietokoneelta

Lisäksi pitää huomioida se tosiasia, että selaintietokoneiden roskapostiesto- ja suojausohjelmien laatutaso on melko vaihteleva tai niitä ei ole asennettu lainkaan ja sen vuoksi riski sähköpostiosoitteen päätymiseen roskapostittajan postituslistalle on erittäin suuri. Mikäli verkkosivustolla oleva vaarantunut osoite vaihdetaan, sama toistuu todennäköisesti uudelleen. Lähetettäessä ja käsiteltäessä lomaketiedot palvelimella web-palvelinohjelmiston avulla sähköpostiosoite voidaan piilottaa selainkoneilta ja se voidaan myös tarvittaessa vaihtaa sivustolla huomaamattomasti. Verkkopalvelimien tietoturvaso on myös yleensä huomattavasti parempi kuin koti- tai yritystyöasemien.

Alla olevassa kuvassa esitetään tietovirtakaaviona yhteydenottopyynnön lähettäminen yritykselle web-palvelimen kautta (Kuva 6).



Kuva 6. Lomakkeiden lähetykset verkkosivupalvelimen avulla

Vierailijan halutessa lähettää yhteydenottopyynnön sivuston kautta yritykselle selain noutaa palvelimelta tiedoston tai tiedostot lomakesivua varten ja tulostaa sen näytölle. Lomake täytetään selainikkunassa ja sen tiedot lähetetään verkkopalvelimelle postinkäsittelyohjelman tarkastettavaksi. Mikäli postikäsittelijä hyväksyy saadut tiedot, se lähettää ne eteenpäin postipalvelinohjelmalle, joka taas lähettää ne sähköpostina annettuun osoitteeseen. Jos lomakkeelta saataviin tietoihin on määritelty ehtoja, jotka eivät täyty, postitus keskeytetään ja virheilmoitus lähetetään selaimelle.

4.2 Verkkosivuston HTML- lomakkeet

Verkkosivustolle rakennettiin yhteydenottopyyntöä, tilauspalveluun kirjautumista ja itse tilauspalvelua varten asiaankuuluvat lomakkeet. Kaikilla kolmella lomakkeella oli tiedonkeräyksessä oma tarkoituksensa ja siksi täyttöohjeistus ja tekstikenttien nimeäminen sekä lukumäärä vaihtelivat käyttötarpeen mukaan. Yhteydenotto- ja tilauslomakkeille otettiin myös käyttöön eri sähköpostiosoitteet postin lajittelun helpottamiseksi. Lomakkeet muotoiltiin ulkoasultaan mahdollisimman yksinkertaisiksi, käyttäjäystävällisiksi ja käyttäjiä opastaviksi. Ohjeistuksessa muistutettiin käyttäjiä myös siitä, että yritys tallensi aina kaikki viestit. Yhteydenottolomakkeessa varauduttiin lisäksi roskapostirobottien varalta CAPTCHA-tunnistuksella, jossa pyydettiin sivuston käyttäjää kirjoittamaan tarkistuskenttään vieressä olevan, satunnaisesti tulostetun kuvan merkit ennen viestin lähetystä. Tilauslomakkeessa ei ainakaan toistaiseksi katsottu tarpeelliseksi ottaa käyttöön CAPTCHA-tunnistusta, koska sen käyttö edellytti aina sisäänkirjautumista palveluun.

Kaikissa lomakesivuissa käytettiin samaa HTML-kuvauskielellä toteutettua rakennetta (Liite 5). Kirjoituskenttien ja tekstien sijoittelu lomakesivulle toteutettiin taulukko-rakenteella ja tyylimuotoilu CSS-tiedostolla. Lomaketiedoston HEAD-osaan laitettiin JavaScript-komentokehote, jonka tehtävänä oli tyhjentää koko lomake käyttäjän siirtyessä eteen tai taaksepäin sivustolla selaimen työkalurivin painikkeilla. Puhdistustoiminto määrättiin suoritettavaksi sekä lomakesivulta poistuttaessa että siihen palattaessa ONLOAD- ja ONUNLOAD-tapahtumakäsittelijöillä. Toimenpiteellä varmistettiin se, että täytettyä lomaketta ei voinut postittaa tahattomasti tai tahallisesti useaan kertaan yrityksen sähköpostiin ja CAPTCHA-tunnisteen merkkisarja päivitettiin joka kerta lomakesivua avattaessa. Kaikkien lomakkeiden käsittelymetodiksi määrättiin lähetävä POST ja toiminnaksi lähetettäessä suoritettava lomakekäsittelyohjelma, tässä tapauksessa palvelimella suoritettava ”postitus.php”-ohjelma, joka toimi postittamisen lisäksi lomaketarkastajana. Lomaketarkastaja varmisti kenttien täytölle määrättyt ehdot, kuten esimerkiksi pakollisten tietojen olemassaolon, ja jos ne täyttyivät, lomakkeen tiedot lähetettiin eteenpäin. Mikäli ehdot eivät täytyneet, lähetys keskeytettiin ja virheilmoitus annettiin.

Lomakesivulla käytettiin HTML-kuvauskielellä INPUT-tekstikenttiä ja -tekstialueita tiedon syöttämistä varten. Tiedostossa ei haluttu käyttää piilokenttiä sähköpostiosoitteita

ja muita käyttäjiltä salattuja tietoja varten, koska selaimen avulla lähdekoodin tutkiminen olisi paljastanut ne kuitenkin ja myös roskapostirobotit olisivat voineet löytää yrityksen sähköpostiosoitteet. Tekstikentille ja -alueiden koko määriteltiin SIZE-, MAXLENGTH-, ROWS-, ja COLS-attribuuteilla. Tekstikentät määriteltiin 30 merkkiä leveiksi ja kirjoitettavien merkkien enimmäismäärä oli 50. Poikkeuksina olivat CAPTCHA-tunnistukselle tehty kenttä, joka rajattiin samankokoiseksi kuin siihen kirjoitettavien merkkien määrä oli ja puhelinnumerolle varattu, vapaaehtoinen kenttä, johon sallittiin vain 20 merkin kirjoitus. Muut kentät määriteltiin pakollisiksi. Tekstialueelle ei määritelty merkkien enimmäismäärää, mutta alueen koko rajattiin kahdeksan merkkiä korkeaksi ja 55 leveäksi. Lomakkeelle laitettiin vielä kaksi INPUT-painiketta, RESET-painike lomakkeen tyhjennystä ja SUBMIT-painike sen lähettämistä varten. CAPTCHA-kuva tulostettiin linkityksen avulla lomakkeeseen palvelimelta ja sitä varten varattiin taulukkorakenteeseen oma paikkansa. Tilaus/tilausvahvistuslomakkeessa oli lisäkentät yritysnimelle, toimitusosoitteelle ja y-tunnukselle sekä painike palvelun keskeyttämistä ja uloskirjautumista varten. Tilauslomakkeessa oli myös muutama lisäkenttä yrityksen omaan käyttöön. CSS-tyylimäärittelyjen ja lomakemuotoilun jälkeen yhteydenottolomake näytti selaimella avattuna alla olevan kuvan mukaiselta (Kuva 7).

Yhteydenotto

Jos haluat, että otamme yhteyttä sinuun, täytä tämä lomake ja lähetä se! Voit myös antaa palautetta verkkosivuiltamme. Jos haluat asioida henkilökohtaisesti, yhteystiedot löytyvät [täältä](#). Muista, että viestisi ja yhteysosoitteesi tallentuvat järjestelmäämme!

Kirjoita viestisi (kaikki tähdellä merkityt tiedot ovat pakollisia!):

nimi*	sähköpostiosoite*
<input type="text"/>	<input type="text"/>
viestin aihe*	puhelinnumero
<input type="text"/>	<input type="text"/>

- kirjoita viestisi alla olevaan tekstikenttään*

- muista kirjoittaa myös puhelinnumerosi, jos haluat nopeaa yhteydenottoa



Kirjoita kuvan merkit tarkistuskenttään:

Lähetys ei onnistu ilman oikein täytettyä tarkistuskenttää!

Kuva 7. Verkkosivuston yhteydenottopyyntölomake

Lomakkeiden ulkoasut haluttiin pitää mahdollisimman yksinkertaisina, yhdenmukaisina ja tyyliältään samanlaisina muun sivuston osien kanssa ja ne määriteltiin avautumaan aina sivuston keski- eli sisältöosaan.

4.3 CAPTCHA- ja RECAPTCHA-tunnistukset

CAPTCHA-tunnistuksen tarkoituksena oli ihmisten ja Internetiä selaavien automaattisten roskapostitusohjelmien erottelu toisistaan ja estää näin verkkosivustojen kautta tapahtuva verkkopalvelinten käyttö massarokapostituksen työkaluina. CAPTCHAN toimintaperiaatteena oli se, että inhimillinen sivuston käyttäjä pystyi erottamaan satunnaisesti generoidun tai jostain sanaluettelosta sattumanvaraisesti valitun merkkijonon merkit epäselväksi vääristetystä kuvasta, mutta tietokone ei. CAPTCHA-ohjelma siis vertasi kuvan tekovaiheessa syntynyttä ja käyttäjän sivustolle syöttämää merkkijonoa toisiinsa ja joko hyväksyi tai hylkäsi ne. Jos merkkijonot olivat samanlaiset, käyttäjän haluama toimenpide hyväksyttiin.

CAPTCHA-tunnistus voitiin liittää verkkosivustoon useilla eri tavoilla. Helpoin ja melko yleinen tapa olisi ollut käyttää ilmaista, melko suosittua ja rekisteröitymistä edellyttävää, RECAPTCHA-tunnistusta, joka linkitetään sivustoille ja suoritetaan Google-palvelimella. Internetistä oli mahdollista löytää lisäksi useita muita ilmaisia, suoraan omalle sivustolle liitettäviä, Java- ja PHP-sovellutuksia. Sivustoa tehdessä kokeiltiin RECAPTCHA-tunnistuksen lisäksi yhtä Java- ja yhtä PHP-pohjaista tunnistusohjelmaa. Kaikkia kokeiltuja vaihtoehtoja vaivasi kuitenkin sama ja hankala ongelma: generoidut tunnistuskuvat olivat liian epäselviä ja aiheuttivat liian usein epäonnistumisia (Kuva 8).



Kuva 8. Icap-Java-, RECAPTCHA-link- ja Securimage-PHP-tunnistekuvat

Kaikki kokeillut valmiit versiot käyttivät kuvan käsittelyyn ja vääristykseen kuvapisteitä muokkaavia väritys- ja venytysalgoritmeja eli käskyjonoja käytäviä vääristysfilttereitä.

Samoja filttoreita käytetään myös kuvankäsittelyohjelmissa erikoisefektien luomiseen. Ongelmana filttoreiden käytössä oli se, että tunnistuksen murttamiseen suunnitellut erikoisohjelmat havaitsivat algoritmien käytön ja käyttivät niitä myös kuvan korjaamiseen tietokoneille luettavaan muotoon. Haluttaessa nostaa tietoturvasoa korkeammaksi algoritmit olisi voitu monimutkaistaa, mutta samalla CAPTCHA-kuvista olisi tullut entistä vaikeampia lukea. Valmiit ilmaisohjelmat hylättiin koska ne eivät palvelleet sivuston käyttäjien tarpeita tarpeeksi hyvin. Maksullisista versioista olisi löytynyt hyviäkin vaihtoehtoja, mutta toimeksiantaja ei halunnut käyttää niitä tässä vaiheessa. Linkitetty RECAPTCHA hylättiin myös siksi, että kaikki sivustoon liittyvä palvelinohjelmointi haluttiin pitää samalla verkkopalvelimella. Sivustolle päätettiin rakentaa omaan ideointiin perustuva, palvelimella suoritettava CAPTCHA-tunnistus.

4.4 PHP-palvelinohjelmointi verkkosivustolla

Verkkosivuston sisältö on yleensä samalla verkkopalvelimella, joka suorittaa PHP-ohjelman. Palvelimet ovat yleensä tarkemmin valvottuja ja laitteistot suorituskyvyiltään moninkertaisia työasemiin verrattuna ja siksi ratkaisu takaa normaalitilanteissa paremman tietoturvan sekä suorituskyvyn kuin työasemien omilla selaimilla suoritettujen ohjelmien. PHP-komentosarjat eli skriptit voidaan myös lähettää suorittavaksi ulkopuoliselle palvelimelle, jos omalla ei ole tukea kyseiselle ohjelmointikielelle. Pitää kuitenkin varautua siihen, että lähetettäessä tietoja oman palvelimen ulkopuolelle tietoturvan taso ja suorituskyky laskee. Tässä tapauksessa tuki kuitenkin löytyi omalta palvelimelta. Yrityksellä oli jo olemassa oleva laajakaistasopimus Elisan kanssa ja siksi kyseisen palveluntarjoajan PHP-ohjelmointituellinen yrityspalvelinympäristö valittiin myös luonnolliseksi verkkosivuston sijoituspaikaksi.

Lomakkeiden käsittely ja postitus toteutettiin kokonaisuudessaan PHP-ohjelmoinnilla. PHP valittiin lomakekäsittelijäksi JavaScriptin sijaan, koska PHP-komentosarjat suoritetaan aina verkkopalvelimella PHP-tulkin avulla. Sivuston käyttäjän selaimelle ei tulosteta suorittamisen aikana mitään, ellei sitä erikseen määritellä PHP-skriptissä. PHP-skriptejä ei voi myöskään nähdä sivuston lähdekoodia selaimen avulla tutkittaessa toisin kuin HTML-määrittelyjä. PHP-skriptit voidaan tehdä kuitenkin tahattomasti tai tahallisesti verkkopalvelinta kuormittaviksi, mikä saattaa hidastaa, tai pahimmassa

tapauksessa kaataa sen. Siksi on oltava tarkkana, että suoritettavaa PHP-ohjelmaa ei voi käyttää väärin tai se ei jää pyörimään ikuisesti ohjelmointivirheen vuoksi.

4.5 CAPTCHA-tunnistuksen PHP-toteutus

Oma CAPTCHA-toteutus perustui satunnaisten kuvien tulostamiseen lomakesivulle ja niiden tunnistamiseen ja nimeämiseen. Kuvien tulostus ja tunnistus päätettiin toteuttaa PHP-ohjelmoinnilla. Ensimmäisessä kokeiluversiossa tulostettiin kuvia erilaisista työkaluista ja pyydettiin käyttäjiä tunnistamaan ne. Lopullisessa versiossa tunnistekuvien tunnistustietona käytettiin numerosarjoja (Kuva 9).



Kuva 9. CAPTCHA-tunnistekuva

Kuvat CAPTCHA-tunnistusta varten tehtiin Microsoft PowerPoint-ohjelmalla ja ne käsiteltiin ja muotoiltiin ulkonäöltään ja kooltaan sopiviksi käyttötarkoitukseensa GIMP-kuvankäsittelyohjelmalla. Kuvat olivat nopeatekoisia ja niitä tehtiin useita, jottei sama kuva numerosarjoineen esiintyisi liian usein.

CAPTCHA-kuvien tulostusta varten tehtiin yksinkertainen PHP-ohjelma (Liite 6.1). Ohjelma suoritettiin palvelimella aina, kun verkkosivuston lomakesivu avattiin tai päivitettiin ja se arpoi RAND-funktiolla yhden numeron väliltä 1- n. N-merkki kuvasi kuvien enimmäislukumäärää. Käytännössä kuvia olisi voinut olla vain vaikka kolme, koska kaikki numerosarjat oli merkitty lomakekäsittelijään. Kuvia laitettiin kuitenkin enemmän tunnistukselle luotettavamman ja turvallisemman ulkonäön saavuttamiseksi. Mitä enemmän kuvia oli, sitä pienempi mahdollisuus saman kuvan näyttämiseksi oli. Lomakkeen avautuessa numeroa vastaava, PHP-ohjelmassa muuttujaksi määritelty, kuvatiedosto haettiin kuville varatusta kansioista ja tulostettiin HTML-sivulle. Sivua näytettiin lomakesivun taulukkorakenteen solussa. Tunnistuksessa pyydettyä merkkijonoa käsiteltiin salasanana, joka tarkistettiin verkkopalvelimen lomakekäsittelyohjelmalla. Mikäli salasana oli kirjoitettu lomakekenttään oikein,

lomaketietojen lähetys postitusohjelman lomakekäsittelijälle onnistui. Väärän merkkijonon käytöstä seurasi virheilmoitus ja postituksen keskeytys.

4.6 PHP-posti- ja lomakekäsittelijä

Yhteydenotto- ja tilauslomakkeiden postituksessa ja lomakekäsittelyssä käytettiin PHP-ohjelmointikielellä toteutettuja ohjelmia, joiden tehtävinä oli sähköpostien lähetyksen lisäksi tekstikenttien tarkastus ja CAPTCHA-tunnistuksen valvonta. Lomakkeissa oli eroavaisuuksia kenttien ja niiden täyttövaatimusten sekä lähetettävien sähköpostisisältöjen ja -osoitteiden suhteen ja siksi oli helpompi tehdä kaksi erillistä lomakekäsittelyohjelmaa. Ohjelmien toimintaperiaatteet ja -mallit olivat käytännössä samanlaisia, muuttujat ja virheviestit erosivat toisistaan.

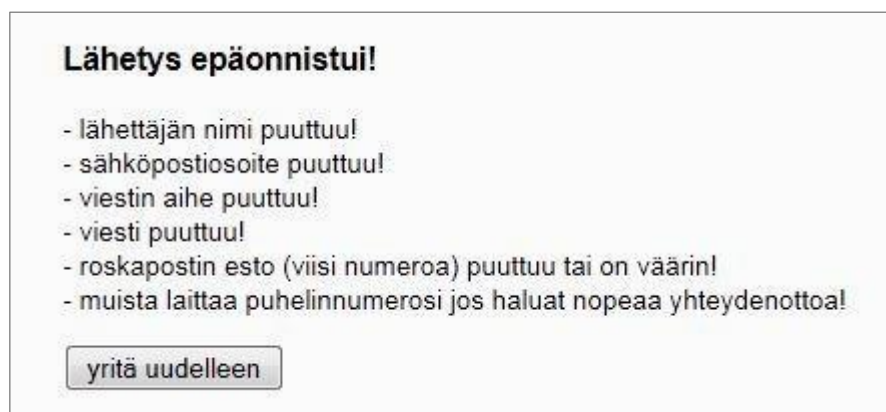
Yhteydenottolomakkeen käsittelyn alussa palvelimen ja selaimen välille muodostettiin dynaaminen yhteys eli istunto `SESSION_START()`; -komennolla ja istunnon tunnistemuuttujaksi määriteltiin ”\$tunniste”. Istunnon tallentuminen välimuisteihin estettiin `CACHE-CONTROL` -ohjauslauseella. Muuttujat määriteltiin lomakekäsittelijöiden ohjelmatiedostoissa ja niille annettiin arvot (Liite 6.2). Virhetilanteita varten määriteltiin ”\$virhe” ja virheilmoituksia varten ”\$virheviesti”. Arvoiksi annettiin ohjelman aloituksessa tilanne, jossa virhettä ei ole olemassa ja virheviesti on tyhjä. Muuttujiksi määriteltiin myös sähköpostin lähetysosoite, postituksen onnistumisesta ilmoittavan tiedoston polku ja lomakkeelta kerättävien tietojen nimet. Yhteydenottolomakkeelta kerättyjen tietojen muuttujiksi nimettiin ”\$nimi”, ”\$puhno”, ”\$email”, ”\$aihe”, ”\$viesti” ja ”\$kapsa”.

Yhteydenottolomakkeessa tekstikentille pakollisiksi määriteltyjä ehtoja olivat lähettäjän nimi, sähköpostiosoite, viestin aihe ja itse viesti. Lisäksi vapaaehtoiseksi kentäksi määriteltiin puhelinnumero ja sille muuttuja viestin sisältöä varten. Puhelinnumeron puuttuminen määriteltiin myös virheviestiksi, jotta sen puuttumisesta voitiin muistuttaa muiden virheviestien yhteydessä. Lomakkeen tarkistuksesta ei haluttu tehdä liian tarkkaa ja raskasta määräämällä viestikentille useita ehtoja, riitti kun tarkistettiin oliko niihin kirjoitettu ylipäänsä mitään ja löytyikö sähköpostiosoitteelle varatusta kentästä @-merkki. Viestikentän merkkien enimmäismääräksi annettiin tuhat PHP-skriptissä, koska HTML-kuvauskielellä ei lomakkeen `TEXTAREA`-kentän merkkimäärää pystynyt

rajaamaan. Muiden kenttien merkkien enimmäismäärä valvottiin suoraan HTML-lomakkeessa. CAPTCHA-tunnistuksen ”\$kapsa”-muuttujalle annettiin etukäteen määritetyt merkkisarjat, joita verrattiin lomakkeelle syötettyihin merkkeihin. Mikäli virheitä ei löytynyt, viesti lähetettiin eteenpäin.

Lomakkeen viestiä eli ”\$viesti”-muuttujaa muokattiin ennen lähettämistä lisäämällä mukaan tieto siitä, oliko lähettäjä antanut puhelinnumeron, ”Viesti:”-otsikko ja viestin lähetysosoite getenv (”REMOTE_ADDR”) -komennolla. Lähettäjän IP-osoite haluttiin liittää viesteihin tunnistetiedoksi, koska lomakkeelle oli mahdollista kirjoittaa mikä tahansa sähköpostiosoite. Varsinkin tilauskaavakkeessa tieto katsottiin erittäin tarpeelliseksi. Lähetettävään viestiin liitettiin myös muuttujina HEADER-otsikkotiedot. Otsikkotiedoissa määriteltiin viestin lisäotsikko, tässä tapauksessa lähettäjän ilmoittamat nimi ja sähköpostiosoite, viestin rungon muodon määrittelevä MIME-versio ja ilmoitettiin postipalvelimelle kyseessä olevan PHP-postinkäsittelyohjelma. Tekstiksi määriteltiin yksinkertainen perusteksti ja CHARSET-parametrilla ISO-8859-15 merkistö, joka ääkkösten lisäksi tunnistaa euron merkin. Lomakkeen tiedot lähetettiin eteenpäin postipalvelimelle MAIL-komennolla: ”\$virheet = ! mail (\$vastottaja, \$aihe, \$viesti, \$headers);”. Ennen komennon suorittamista tarkistettiin, että ”\$virhe”-muuttuja ei ollut muuttunut ohjelman alussa annetusta ”false”-arvosta.

Virhetilanteiden määrittelyssä ei haluttu käyttää ohjelman keskeytystä kenttäkohtaisesti esimerkiksi EXIT-komennolla, vaan koko lomake tarkistettiin ennen lähetystä ja mikäli virheitä löytyi, postitus keskeytettiin. Kaikista virhetilanteista annettiin virheilmoitus samanaikaisesti (Kuva 10).



Kuva 10. Virheilmoitus tyhjän yhteydenottolomakkeen lähetysohjelma-ikkunasta.

Virheilmoitukset tulostettiin käyttäjälle käyttämällä kahta erillistä PHP-dokumenttiin upotettua HTML-osaa, joiden väliin PHP-virheviestit lisättiin. Näin saatiin varmistettua yhteensopiva CSS-tyylimuotoilu muun sivuston kanssa (Liite 6.2). Viestin lähetyksen onnistuminen ilmoitettiin ohjaamalla käyttäjä istunnon lopettavan PHP-ohjelman kautta HTML-kiitosviestiin. Kaikki virheviestit näytettiin allekkain vaihtamalla viestien lopussa riviä ”
”-tagilla eli komennolla ja viestin loppuun lisättiin ”yritä uudelleen”-linkkipainike uutta lähetysoyrytystä varten. Linkkiä käytettäessä suoritettiin PHP-ohjelma, joka ohjasi takaisin viestisivulle (Liite 6.3). Ohjelma avasi SESSION-istunnon SESSION_START()- komennolla, poisti saman tien SESSION-muuttujan SESSION_UNSET()- komennolla ja tuhosi lopuksi istunnon SESSION_DESTROY()-komennolla. SESSION-istunto oli avattava ensin skriptin alussa ennen kuin se voitiin tuhota. Välimuisteihin tallennuksen estolla ja lomakkeiden kenttien pakkotyhjennyksen yhdistelmällä estettiin lomakkeiden toistuva lähetys selaimen työkalupalkin eteenpäin ja taaksepäin osoittavia nuolia käyttämällä.

Tilauslomake toteutettiin samalla perusrakenteella kuin yhteydenottolomake, mutta siihen lisättiin tekstikentät yritysnimelle, posti- ja toimitusosoitteelle, y-tunnukselle ja viitenumerolle. Lisäksi ”viestin aihe”-kenttä vaihdettiin alaspövalikoksi (Kuva 11).

Tilaus/Tilausvahvistus

Täytä tilaus-/tilausvahvistuskaavake ja tilaa tuotteita tai palveluita. Tilaukset taltioidaan ja ne vahvistetaan aina sähköpostitse tai tekstiviestillä käsittelyn jälkeen. Muista, että viestisi ja osoitetietosi tallentuvat järjestelmäämme!

Kirjoita viesti (kaikki tähdellä merkityt kentät ovat pakollisia):

AP Nurminen Oy
Tilaus/Tilausvahvistus

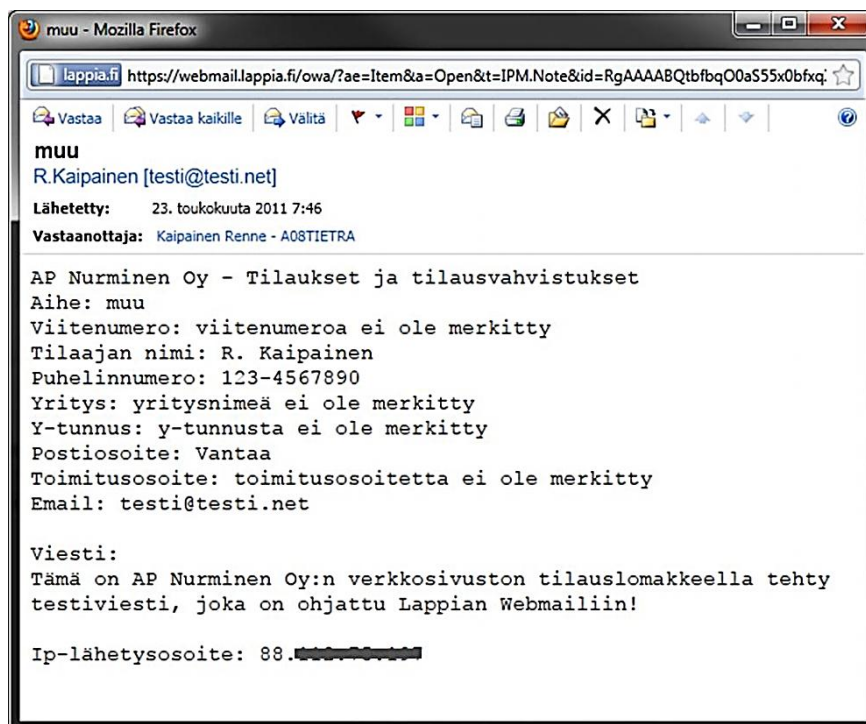
vastuuhenkilön nimi* <input type="text" value="R. Kaipainen"/>	puhelinnumero* <input type="text" value="123-4567890"/>
yrityksen nimi <input type="text"/>	sähköpostiosoite* <input type="text" value="testi@testaus.koe"/>
postiosoite* <input type="text" value="Vantaa"/>	toimitusosoite, jos eri sama <input type="text"/>
y-tunnus <input type="text"/>	valitse viestin aihe* <input type="text" value="muu"/>
viite-/tilausnumero <input type="text"/>	

- muista ilmoittaa myös toimitusosoite jos se on eri kuin postiosoite!
 - kirjoita alla olevaan kenttään viestisi tai tilauksen/vahvistuksen lisätiedot:

Tämä on AP Nurminen Oy:n tilauslomakkeen testiviesti, joka on ohjattu Lappian Webmailiin!

Kuva 11. Postivalmis lomake

Tilausviestit haluttiin muotoilla ulkoasuiltaan ja sisällöltään sellaisiksi, että ne voitiin tulostaa paperille suoraan sähköpostista. Viestit haluttiin pitää samannäköisinä ja näin ollen helppolukuisina. Viesteihin lisättiin HEADER-otsikkotietoina kaikki lomakkeesta saatavat tiedot, vapaaehtoisten kenttien nimet ja tieto siitä, että niitä ei ollut täytetty. Edellisen sivun tilauslomake näytti lähetyksen jälkeen sähköpostissa alla olevan kuvan mukaiselta (Kuva 12).



Kuva 12. Tulostusvalmis tilausviesti

4.7 Kirjautuminen tilauspalveluun

Verkkosivuston tilauspalvelu oli tarkoitettu vain rajoitetulle määrälle sopimusasiakkaita ja yhteistyökumppaneita ja sen käyttö edellytti käyttäjätunnusta ja salasanaa. Palvelun kautta ei hoidettu maksuliikennettä ja siksi varmennettua tunnistautumista ei tarvittu. Omatoimista rekisteröitymismahdollisuutta ei haluttu tässä vaiheessa käyttää ja tietokantaan perustuvasta käyttäjätunnus- ja salasanarekisteristä luovuttiin toistaiseksi. Verkkosivusto haluttiin pitää toiminnallisesti erittäin kevyenä ja sen hallinta helppona ja siksi kirjautumistunnisteet upotettiin suoraan kirjautumislomaketta valvovaan PHP-tiedostoon (Liite 6.3). Käyttötarkoitukseen sopivia tunnuksia ja salasanoja luotiin valmiiksi kaikille nykyisille yhteistyökumppaneille ja lisäksi saman verran varattiin

odottamaan käyttöönottoa. Tilauspalveluun kirjautumista varten tehtiin oma HTML-lomakesivu (Kuva 13).

Kirjautuminen tilauspalveluun:

kirjautuessasi tarvitset asiakastunnuksen ja salasanan.

asiakastunnus:

salasana:

Kirjautuminen onnistuu vain asiakastunnuksella ja salasanalla! Jos sinulla ei ole tunnusta, tai olet unohtanut salasanasi, ota yhteyttä [asiakaspalveluumme](#).

Tilauspalvelu on tarkoitettu yhteistyökumppaneille ja sopimusasiakkaille!

Kuva 13. Kirjautumislomake

Lomakkeen POST-metodi kutsui PHP-ohjelmaa ja lähetti käyttäjätunnuksen ja salananan palvelimelle. PHP ohjelma vertasi suoraan lomakkeelta saatuja tietoja PHP-skriptin ehtoihin ja aloitti SESSION-istunnon, mikäli ehdot toteutuivat. Muussa tapauksessa käyttäjä ohjattiin takaisin HTML-lomakesivulle virheilmoituksen kanssa yrittämään uudelleen. Istuntoa aloitettaessa sen tunnistetiedot tallentuivat palvelimen välimuistiin lyhytkestoisina SESSION-COOKIE -evästeinä, jossa ne pysyivät niin kauan kunnes tilauslomakkeen lähetys onnistui tai istunto lopetettiin kirjautumalla ulos. Molemmissa tapauksissa kutsuttiin PHP-ohjelmaa, joka keskeytti istunnot ja ohjasi käyttäjän asianmukaisille HTML-kiitos- ja ilmoitussivulle. Selaimen sulkeminen keskeytti myös istunnon määräajan jälkeen.

5 VERKKOSIVUSTON TESTAUS JA SUORITUSKYKY

Verkkosivuston suunnittelun lähtökohtana oli tehdä siitä mahdollisimman kevyt ja suorituskykyinen. Varsinkin hitaan verkkoyhteyden vaikutus sivuston käytettävyyteen haluttiin selvittää ohjelmistotestauksella. Toimeksiantajan halusi sivuston olevan käytettävissä alueella, jossa käytettiin yleisesti puhelinmodeemeja Internet-yhteyksien luomiseen. Tavoitteeksi asetettiin sivuston kohtuullinen toiminta melko alhaisella 56 kbps-nopeudella. Sivustoa testattiin jatkuvasti eri kehitysvaiheissa ja mikäli suorituskyky laski liikaa, uusi ja kevyempi versio otettiin työn alle. Verkkosivustoa rakennettiin siis evoluutiomallin mukaisesti.

5.1 Testaustyökalut ja -metodit

Verkkoyhteysnopeuden vaikutusta sivuston toimintaan testattiin muuttamalla laajakaistamodeemin nopeusasetuksia portaittain. Tällä tavoin yritettiin löytää kriittinen latausnopeus eli piste, jossa sivuston käyttö muuttui epämiellyttäväksi hitaan latausnopeuden vuoksi. Testiselaimena käytettiin Firefox 3.6 -selainta, jonka välimuistit oli helppo tyhjentää jokaisen testin jälkeen. Arviointityökaluna käytettiin digitaalista sekundaattoria ja testikriteeriksi valittiin: jos sivuston etusivun lataus kestää yli kymmenen sekuntia, se luokitellaan käyttökelvottomaksi. Testaus aloitettiin suuntaa antavana LAN-verkon puolella. Todellisuudessa nopeudet paikallisverkossa ja Internetissä eivät olleet vertailukelpoisia, koska WAN-verkon edestakainen liikenne ja nopeuden vaihtelut tekivät siitä todellisuudessa paljon hitaamman.

Testitietokoneina käytettiin kahta työasemaa, joista toiseen koneeseen asennettiin Linux-pohjainen Ubuntu-palvelinkäyttäjärjestelmä, ja toiseen Microsoftin Windows XP. Verkkosivusto asennettiin Ubuntu-käyttäjärjestelmän Apache-palvelinohjelmiston WWW-kotisivuksi. Myöhemmin myös palvelinkoneen käyttäjärjestelmäksi vaihdettiin XP ja palvelinohjelmistoksi XAMPP-server. Verkkosivuston dokumenttien siirto ja käsittely sujui huomattavasti nopeammin Windowsin graafisen käyttöliittymän avulla, kuin Ubuntu-palvelinkäyttäjärjestelmän komentoriviä käyttäen.

5.2 Havainnot ja evoluutiomalli

Verkkoyhteysnopeuden vaihtelulla oli erittäin suuri vaikutus sivuston käytettävyyteen. Nopeuden laskiessa erittäin alhaiseksi, sivuston lataus saattoi keskeytyä ja kaataa selaimen. Verkkoyhteysnopeudella oli sivuston toimintaan suurempi vaikutus kuin tietokonelaitteiston teholla. Sivustoa koekäytettiin testauksen aikana yksitoista vuotta vanhalla, 400 Mhz Celeron prosessorilla varustetulla, kannettavalla ja eroa ei juurikaan huomannut verrattaessa 3 Ghz Pentium 4 tietokoneeseen.

Sivuston toinen versio yksinetoista raameineen ja kuvatäyttöineen oli melko raskas ja sen käytettävyys loppui käytännössä kokonaan LAN-verkkonopeuden lähestyessä 20 kbps alarajaa. Navigoitaessa sivustolla kuvatäytöistä hävisi suuri osa ja sivut muuttuivat käytännössä käyttökelvottomiksi. Versio toimi moitteettomasti, jos LAN-verkkonopeus oli yli 50 kbps. Elisan yrityspalvelimen käyttöönoton myötä testaaminen siirrettiin takaisin Internetiin paremman kokonaiskuvan saamiseksi. WAN-verkossa käytettävyys loppui täydellisesti nopeuden ollessa 80 kbps tai alle. Sivusto toimi hyvin 220 kbps tai sitä suuremmilla nopeuksilla.

Version rakenteesta alettiin poistaa osia vähän kerrallaan ja kuvatäyttöjen resoluutioita pienennettiin ulkonäön kustannuksella. Toimenpiteiden välillä sivustoa testattiin eri verkkonopeuksilla ja suorituskykyä verrattiin alkuperäisen version suorituskykyyn. Vertailemalla varmistettiin, etteivät mahdolliset erot suorituskyvyssä johdu normaalista verkkonopeuden vaihtelusta. Lopulta kerätyn tiedon perusteella rakenne hylättiin ja taulukkorakenne otettiin käyttöön.

Kolmas ja viimeinen versio perustui mahdollisimman yksinkertaiseen rakenteeseen ja ulkonäköön. Raamit hylättiin lukuunottamatta sisältöosaa, kuvatäytöt vaihdettiin tasaisiin taustaväreihin ja kuvatiedostoja koitettiin avata sivuston latautuessa mahdollisimman vähän. Poikkeuksena oli HEADER-OTSIKKO -kuva yrityslogoineen, pienehkö kuva yrityksen kuljetuskalustosta ja muutama logokuva-linkki. Sivustoa kevennettiin siirtymällä tabloidi-ulkoasuun. Ensimmäisellä testikerralla voitiin havaita suorituskyvyn parantuneen neljänneksellä. WAN-verkossa päästiin heti lähelle tavoitetta eli 60 kbps miniminopeutta. Sivuston toiminta oli lähes viiveetöntä, mikäli WAN-nopeus oli 150 kbps tai yli.

6 YHTEENVETO

Tämän opinnäytetyön tarkoituksena oli suunnitella ja tehdä toimeksiantajalle käyttäjäystävällinen, kevyt ja turvallinen verkkosivusto. Sivusto piti toteuttaa ilman maksullisia, verkkosivujen toteutukseen käytettäviä ohjelmia tai editointiympäristöjä. Sivustolle haluttiin markkinoivan ulkonäön lisäksi kevyt käyttöliittymä, viestintälomakkeet ja roskapostituksen esto lomakkeiden kautta. Sivuston tyyllillisen ulkoasun piti sopia myös yrityksen yhteistyökumppaneille.

Verkkosivustoon oli tarkoitus laittaa yrityksen oman sisällön lisäksi yhteistyökumppaneiden tuottamaa sisältöä ja kovalinkkejä vastikkeeksi toimeksiantajan vastaavasta sisällöstä partnereiden verkkosivustoissa. Muutama yhteistyökumppani oli äskettäin uusinut yritysilmemuutoksen myötä yrityslogonsa eivätkä he halunneet antaa niitä käytettäväksi ennen kuin sivuston ulkoasu oli niiden kanssa yhteensopiva. Kaikki partnerit päättivät lopulta toimittaa sisältöä ja kuvamateriaalia vasta sitten, kun ulkoasu ja värimaailma oli hyväksyttävä ja niitä sai käyttää julkisesti vasta kun kaikilta osapuolilta oli saatu yhteinen hyväksyntä. Järjestely aiheutti pienen ongelman: kuinka hyväksyttävä ulkoasu ilman sivustolle laitettavaa sisältöä! Ratkaisuksi keksittiin privaattiverkkosivu, jota pyöritettiin palvelimella vain yhteistyökumppaneita varten. Samanaikaisesti AP Nurminen Oy:llä oli käytössä väliaikainen tiedotesivu. Virallisesta sivustosta tehtiin kaiken kaikkiaan kymmenkunta tyylimallia ennen kuin kaikki olivat yksimielisiä ja hyväksyivät sivuston ulkoasun. Parhaaksi valittu malli oli ensimmäinen, reilua kuukautta aiemmin tehty ja esitelty tyylimalli. Uuden tyylimallin myötä käyttöön otettiin allekirjoittaneen tekemä uusi yrityslogo sivuston otsikossa. Kaikki osapuolet olivat lopulta sivuston ulkoasuun erittäin tyytyväisiä.

Sivuston tekeminen kesti huomattavasti odotettua pidempään. Alkujaan hankkeelle oli varattu aikaa kahdesta kolmeen kuukauteen, mutta se saatiin venytettyä kaksinkertaiseksi. Aikataulun pitkittymiseen vaikuttivat sivuston ulkonäön hienosäädön lisäksi muun muassa verkkopalvelinpalvelun tarjoajan pitkät toimitusajat ja ensimmäisellä kerralla puuttunut PHP-tuki. Sivuston ensimmäisiä versioita pyöritettiin kutakuinkin kuukauden ajan kotisivupalvelimella ja vihdoin ja viimein, kun yrityksen yrityspalvelintila saatiin käyttöön, huomattiin, että se olikin laitettu samalle kotisivupalvelimelle missä ei edelleenkään ollut tukea PHP-ohjelmoinnille. Ainoastaan kuukausimaksu oli huomattavasti korkeampi. Kotisivupalvelimet ovat yleensä pelkästään tiedostojen

latausta varten toisin kuin yrityksille suunnatut yrityspalvelimet, joissa sallitaan verkkosivustojen palvelinohjelmien suorittaminen. Siksi yrityspalveluiden palvelinlaitteistot ovat huomattavasti tehokkaampia kuin kotisivupalvelinten ja tehon lisäyksen myötä käyttökustannukset vastaavasti samassa suhteessa korkeampia. Palveluntarjoajalta meni melkein kaksi viikkoa siihen, että kotisivutila muutettiin fyysisesti toiselle palvelimelle, josta PHP-tuki sitten vihdoinkin löytyi. Palveluntarjoaja oli koko prosessin ajan erittäin avulias eikä laskuttanut ”harharetkien” ajalta mitään. Ongelmat johtuivat kesäloma-ajankohdasta ja sen aiheuttamasta henkilöstövajauksesta.

Koko sivuston lähdekoodi kirjoitettiin käyttämällä Windows-käyttöjärjestelmien mukana tulevaa kevyttä Notepad-tekstieditoria. Sivuston ja lomakkeiden toimivuutta testattiin ja parannettiin jatkuvasti koko kehitystyön ajan. Valmis verkkosivusto on ollut käytössä tämänhetkisessä muodossaan jo puolisen vuotta ja kyseisenä aikana ei yrityksen uusiin, sivustoa varten varattuihin sähköpostiosoitteisiin ole tullut ainuttakaan roskapostia tai häiriköintiviestiä, muuta asiakaspostia tosin sitäkin enemmän. Sivustoa on myös käytetty modeemiyhteydellä alueelta, jossa muuta verkkoyhteyksmahdollisuutta ei ollut. Kaikissa tavoitteissa onnistuttiin siis paremmin kuin hyvin. Toimeksiantajakin on ollut sivustoonsa erittäin tyytyväinen.

AP Nurminen Oy:n Verkkosivustoa (Kuva 14) on tarkoitus kehittää edelleen ja varmistaa sivuston toimivuus uusien selaimien kanssa. Internet Explorer 9 ja Firefox 4 näyttäisivät nopeuttavan toimintaansa pakkaamalla kaiken mahdollisen valmiiksi välimuisteihinsa estoyrityksistä huolimatta ja pitävän saamistaan selailutiedoista kiinni melko ponnekkaasti. Esimerkiksi verkkosivuston suorituskykyä testattaessa ei voitu käyttää uusimpia selainversioita, koska niiden välimuistien täydellinen tyhjennys ei onnistunut, ei edes asiaa varten tehdyillä ohjelmilla. Testitulokset eivät olleet vertailukelpoisia esimerkiksi Firefox 3.6-version tulosten kanssa. Uudet selaimet saattavatkin aiheuttaa tietoturvariskejä PHP-evästeitä käytettäessä ja siksi palvelimien palvelinohjelmistoja voidaan joutua tulevaisuudessa muokkaamaan ottamalla käyttöön muita ohjelmointitekniikoita.

Tervetuloa tutustumaan
AP Nurminen Oy:n uudistettuihin
verkkosivuihin!

AP Nurminen Oy on korkealuokkaisten Thermia
maalämpöpumppujen virallinen jälleenmyyjä
Etelä-Suomessa.

Hanki sinäkin tulevaisuuden lämmitysratkaisu
kotisi jo tänään!



Etusivu Palvelut ja tuotteet Tarjoukset Ota yhteyttä Tilaukset



Uutisia

Palvelut ja
tuotteet

Toimitukset ja
tarvikkeet

Pikaviesti

Tilaa tuotteita,
palveluita tai
katsele muksia

Toimitus- ja,
laskutusehdot

Yhteystiedot

Yritysinfo

Sivustoa päivitetään -
valitsemme häiriötä
klytössä!



Site under construction -
sorry for any
inconveniences!

AP Nurminen Oy

suorittaa lämpö-, vesijohto- ja
ilmastointityöt nopeasti,
laadukkaasti ja edullisesti!

Ota yhteyttä, ja kysy tarjouta,
olipa kysymyksessä sitten
kylpyhuoneremontti, omakotitalon
putkisaneraus, moniasuntoisen
uudisrakennuksen LVI-urakointityö
suunnitelmiseen tai teollisuus-
rakennuksen ilmastointityöt!

Palvelumme kattavat kaiken alaan
kuuluvan.

LVI-palveluita, -tuotteita ja
-suunnitelmia jo vuodesta 1994!

Tulevaisuuden LVI-tekniikka-
asiantuntijakin suosittelevat AP
Nurminen Oy:tä!

PARASTA



A-LUOKKAA!

AP Nurminen Oy

toimii kattavasti ja joustavasti
koko pääkaupunkiseudun
alueella ja sen lähiympäristössä!



Päätoimipaikkamme sijaitsee
Vantaalla, toimialueemme
keskeisessä pisteessä nopeiden
kulkuyhteyksien varrella.



Kuljetuskalustoakin on laidasta
laitaan, jopa työmaalle ajettava
liikkuva verstaas varastointeen.

Käytössämme ovat viimeisimmät
työkalut ja varusteet kaikkein
vaativimpiinkin LVI-toihin.



Thermotech

hansgrohe

IDO

uponor

Viikon vinkki -
pidä hiiren osoittama
linkkien ja kuvien päällä,
noin saat lisätietoja!



Muista lukea myös
tuotevalmistajien
verkkosivut - klikkaa
logoja!

AP Nurminen Oy - Kokkokalliontie 5 D 4 - 01380 Vantaa - +358 (0) 40045564

rk the builder ©

Kuva 14. AP Nurminen Oy - verkkosivusto

LÄHTEET

Painamattomat

ApacheFriends 2010. XAMPP. Luettu 12.11.2010.

< <http://www.apachefriends.org/en/xampp.html>>

CAPTCHA 2010. CAPTCHA: Telling Humans and Computers Apart Automatically.

Luettu 10.11.2010.

<<http://www.captcha.net/>>

LeechFTP 2010. LeechFTP. Luettu 11.11.2010. < <http://www.leechftp.org/>>

Nielsen, Jacob 2010. Ten Usability Heuristics. Luettu 29.10.2010.

<http://www.useit.com/papers/heuristic/heuristic_list.html>

php. PHP Manual. Luettu 10.11.2010. <<http://php.net/manual/en/index.php>>

The Apache Software Foundation 2010. Apache HTTP Server Project. Luettu 12.11.2010.

< httpd.apache.org >

Ubuntu Suomi 2010. Mikä on Ubuntu? Luettu 11.11.2010.

< <http://www.ubuntu-fi.org/>>

W3C 2010. World Wide Web Consortium (W3C) Standards. Luettu 12.11.2010.

<<http://www.w3.org/standards/>>

w3schools 2010a. CSS Introduction. Luettu 10.11.2010.

<http://www.w3schools.com/html/css_intro.asp>

w3schools 2010b. HTML Introduction. Luettu 10.11.2010.

<http://www.w3schools.com/html/html_intro.asp>

w3schools 2010c. JavaScript Introduction. Luettu 10.11.2010.

<http://www.w3schools.com/html/js_intro.asp>

w3schools 2010d. PHP Introduction. Luettu 10.11.2010.

<http://www.w3schools.com/php/php_intro.asp>

Ryabov, Vladimir 2010. Presentation, Research methods Part II. Luentomateriaali.

Kemi-Tornion ammattikorkeakoulu. Liiketalouden ja tietotekniikan koulutusyksikkö.

LIITTEET

Haastattelukysymykset:

Liite 1 (1/1)

* Kymmenen kysymystä verkkosivuja varten *

1. Onko yrityksellä aikaisempaa verkkosivustoa tai -sivua?
2. Mitä mieltä olet aikaisemmasta verkkosivusta?
3. Millaiselta uuden sivuston pitäisi näyttää ja mitä sen pitäisi sisältää?
4. Minkälaista yrityskuvaa verkkosivuilla halutaan korostaa?
5. Mitä ja minkälaisia asioita sivustolla pitää näkyä/ei pidä näkyä?
6. Minkälaisia palveluita ja toimintoja sivustolle tarvitaan?
7. Mikä on sivuston pääasiallinen käyttötarkoitus tällä hetkellä?
8. Mikä on sivuston käyttötarkoitus tulevaisuudessa?
9. Minkälaista liiketoimintaa yritys harjoittaa tällä hetkellä?
10. Millaisia tulevaisuudensuunnitelmia yrityksellä on?

1. HTML-iframe

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
<title>AP Nurminen Oy - LVI-palvelut</title>
</head>
<iframe src="etusivu.html" Width="870" height="820" frameborder="0" scrolling="no"></iframe>
</html>
```

2. Frameset-rakenne

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">

<html><head><title>kuusi_raamia</title></head>
<frameset rows="178,33,*" framespacing="0" frameborder="0">
<frame src="otsikko.html" noresize name="header">
<frame src="valikko1.html" noresize name="keskipalkki">
<frameset cols="140,*,140" framespacing="0" frameborder="0">
<frame src="valikko2.html" noresize name="vasenpalkki">
<frameset rows="*,33" framespacing="0" frameborder="0">
<frame src="sisältö.html" name="keskiosa">
<frame src="otsikko.html" noresize name="header">
<frame src="valikko3.html" name="ala" noresize>
</frameset>
<frame src="valikko4.html" name="oikea" noresize>
</frameset>
</frameset>
</html>
```

3. Valmis iframe-perusrakenne

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html><head><title>iframes</title></head>
<iframe src="header.html" Width="860" height="211" name="header" frameborder="0" scrolling="no">
</iframe>
<iframe src="sisältö.html" Width="860" height="600" name="sisältö" frameborder="0" scrolling="no">
</iframe></html>
```

4. Otsikon table-rakenne

Liite 2 (2 /2)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html><head><title>otsikko</title></head><body>
<table align="center" width="860" border="0" cellspacing="0" cellpadding="0">
<tr><td width="300" height="178" colspan="2">
</td><td width="560" height="178"></td></tr>
<tr><td width="140" height="33"></td>
<td width="160" height="33"></td>
<td colspan="2" width="560" height="33"></td>
</tr></table></body></html>
```

5. Sisältöosan frameset-rakenne

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">

<html><head><title>sisältö</title></head>
<frameset cols="140,*,140" frameborder="0" framespacing="0">
<frame src="vasenvalikko.html" name="linkit" scrolling="no">
<frameset rows="567,33">
<frame src="etusivu.html" name="keskiosa" scrolling="no">
<frame src="alalappi.html" name="alalappi" scrolling="no">
</frameset>
<frame src="oikeavalikko.html" name="kuvalinkit" scrolling="no">
</frameset></html>
```

1. CSS-tyylit HTML-dokumentissa

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>oikea valikko</title>
<link rel="stylesheet" type="text/css" href="tyyli1.css">
<style type="text/css">
body { background-color: #4682B4; margin-top: 15px; overflow: hidden;}
</style>
</head>
<body>
</body>
</html>
```

2. Valikoiden muotoilussa käytetty CSS-tyylitiedosto

```
/*Valikkosivujen tyylitiedosto tyyli1.css*/
```

```
body          {margin-top: 5px; color:black; font-family: arial; font-size: 90%}
.navi_level1  {font-size: 100%, font-weight: bold; text-decoration: none}
.navi_level2  {font-size: 100%, color: white; text-decoration: none}
a:hover       {color: #C00000}
```

1. JavaScript upotettuna linkkisivuun

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html><head><title>kuvalinkki</title>
<script type="text/javascript">
function popup(){
newwin = window.open('','ruutu','width=530,height=450,resizable=1');}
</script>
</head>
<body>
<p>
<a href="kuvasivu1.html" target="ruutu" onclick="popup();">kuva</a>
<a href="kuvasivu2.html" ...
<a href="kuvasivu3.html" ...
</p>
</body>
</html>
```

2. JavaScript kuvasivussa

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
<title>kuvasivu1</title>
<body onblur="self.close()">
<p>

</p>
</body>
</html>
```

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html><head>
<title>ylomake</title>
<meta http-equiv="Content-type" content="text/html;charset=ISO-8859-1">
<link rel="stylesheet" type="text/css" href="lomake.css"/>
<script type="text/javascript">
function puhdistus()
{
var i;
for (i = 0; (i < document.forms.length); i++)
{
document.forms[i].reset();
}
}
</script>
</head>
<body onload="puhdistus()" onunload="puhdistus()">
<h3>Yhteydenotto</h3>
<p>Jos haluat, että otamme...</p>
<form action="postitus.php" method="post">
<p>Kirjoita viesti (* kaikki...):</p>
<table width="470" align="center" border="0" cellspacing="0" cellpadding="0">
<tr height="45">
<td valign="top" width="255">nimi* <br><input type="text" name="nimi" size="30" maxlength="50"></td>
<td valign="top">sähköpostiosoite* <br><input type="text" name="email" size="30" maxlength="50"></td>
</tr>
<tr height="45">
<td valign="top">viestin aihe* <br><input type="text" name="aihe" size="30" maxlength="50"></td>
<td valign="top">puhelinnumero <br><input type="text" name="puhno" size="30" maxlength="20"></td>
</tr>
<tr height="150">
<td colspan="2">- kirjoita viestisi...
<br>- muista kirjoittaa...<br><textarea rows="8" cols="55" name="viesti"></textarea></td>
</tr>
<tr height="50">
<td align="center" valign="top"><iframe src="captchakuvat.php"></iframe></td>
<td align="center"><input type="reset" value="tyhjennä...></td>
</tr>
<tr height="30">
<td align="center">Kirjoita kuvan...</td>
<td align="center"><input type="text" name="captcha" size="4" maxlength="5">
<input type="submit" name="send" value="Lähetä viesti"></td>
</tr>
<tr height="30">
<td colspan="2" align="center"><b>Lähetys ei onnistu...</b></td>
</tr>
</table>
</form>
</body>
</html>

```

1. captchakuvat.php - satunnaisten captcha-kuvien arvonta

```

<?php
if (!$satku)
$satku = rand (1,n);
if ($satku == 1)
{$kuva="<img src=' ./ckuvat/1.jpg'>";}
elseif ($satku ==2)
{$kuva="<img src=' ./ckuvat/2.jpg'>";}
elseif ($satku ==3)
{$kuva="<img src=' ./ckuvat/3.jpg'>";}
elseif...

else
{$kuva="<img src=' ./ckuvat/n.jpg'>";}
?>
<html>
<head>
<title>Captchakuvat</title>
</head>
<body>
<p><?php echo $kuva?></p>
</body>
</html>

```

2. ypostitus.php - yhteydenottolomakkeen postikäsittelijä

```

<?php
session_start();
$tunniste=$_SESSION["tunniste"];
header("cache-control: no-store, no-cache, must-revalidate");
$virhe = false;
$virheviesti = "";
$vastottaja = 'nimi@toimialue.verkkotunnus';
$uohjaus = 'postivastaus.php';
$nimi = $_POST['nimi'] ;
$puhno = $_POST['puhno'] ;
$email = $_POST['email'] ;
$aihe = $_POST['aihe'] ;
$viesti = $_POST['viesti'] ;
$kapsa = $_POST['kapsa'] ;

if (empty($nimi))
{
$virhe = true;
$virheviesti .= "- nimesi puuttuu!<br/>";
}

if (!ereg("@", $email))
{
$virhe = true;
$virheviesti .= "- email osoite puuttuu!<br/>";
}

if (empty($viesti))
{
$virhe = true;
$virheviesti .= "- viesti puuttuu!<br/>";
}

```

postikäsittelijä...

Liite 6 (2 /3)

```

if (strlen($viesti) > 1000)
{
$virhe = true;
$virheviesti .= "- viesti on liian pitkä, 1000 merkkiä on maksimi!<br/>";
}

if ($kapsa != xxxx)
if ($kapsa != xxxx)
if ($kapsa...
if ($kapsa != xxxx)
{
$virhe = true;
$virheviesti .= "- roskapostin esto (viisi numeroa) puuttuu tai on väärin!<br/>";
}

if (empty($puhno))
{
$puhno = "numeroa ei ole laitettu viestiin!";
$virheviesti .= "- muista laittaa puhelinnumerosi jos haluat nopeaa yhteydenottoa!<br/>";
}

$viesti = "Lähettäjän puh.no: $puhno \n\nViesti:\n\n" . $viesti . "\n\rIp-os: " . getenv("REMOTE_ADDR");

if ( ! $virhe )
{
$headers = "From: $nimi <$email>";
$headers .= "\nMIME-Version: 1.0";
$headers .= "\nX-Mailer: PHP/" . phpversion();
$headers .= "\nContent-type: text/plain; charset=\"iso-8859-15\"";

$virhe = ! mail($vastottaja, $aihe, $viesti, $headers);
}

if ( $virhe )
{
echo ("<html><body><p><h3>Lähetys epäonnistui!</h3></p></body></html>");
echo ($virheviesti);
}

else
{
header("location:$uuhjaus");
}
?>

<html>
<head>
<title>Uusintayritys</title>
<style type="text/css">
body
{
margin-top: 15px;
margin-left: 50px;
margin-right: 50px;
font-family: arial;
font-size: 84%;
background-color: #ffffff
}
</style>
</head><body>
<p>&nbsp;<a href="ypostivirhe.php">Yritä uudelleen</a></p>
</body>
</html>

```

3. ypostivirhe.php - postikäsittelijä

Liite 6 (3 /3)

```
<?php
session_start();
session_unset();
session_destroy();
header("location: ../ylomake.html");
?>
```

4. login.php - kirjautuminen tilauspalveluun

```
<?php
if((empty($_POST["tunnus"])) or (empty($_POST["salasana"]))) header("location: ../virhe.html");

else
{
if
(($_POST["tunnus"]=="ktunnus_1" and $_POST["salasana"]=="salasana_1") or
($_POST["tunnus"]=="ktunnus_2" and $_POST["salasana"]=="salasana_2") or
($_POST["tunnus"]=="ktunnus_3" and $_POST["salasana"]=="salasana_3") or
($_POST["tunnus"]...

($_POST["tunnus"]=="ktunnus_n" and $_POST["salasana"]=="salasana_n"))

{
session_start();
$kello=date("d.m.Y - H:i:s", time());
session_register("kello");
header("location:tilaus.html");
}

else
header("location: ../virhe.html");
}
?>
```