

Antti Ronkanen

VIDEOVERKKOPALVELUN TOTEUTTAMINEN

Opinnäytetyö
Tietojenkäsittelyn koulutusohjelma


Maaliskuu 2010




MIKKELIN AMMATTIKORKEAKOULU

Mikkeli University of Applied Sciences

KUVAILULEHTI

 <p>MIKKELIN AMMATTIKORKEAKOULU Mikkeli University of Applied Sciences</p>	<p>Opinnäytetyön päivämäärä</p> <p>8. maaliskuuta 2010</p>		
<p>Tekijä(t) Antti Ronkanen</p>	<p>Koulutusohjelma ja suuntautuminen Tietojenkäsittely</p>		
<p>Nimeke Videoverkkopalvelun toteuttaminen</p>			
<p>Tiivistelmä</p> <p>Tämän opinnäytetyön tarkoituksena oli toteuttaa Mikkelin ammattikorkeakoulun CampusTV:lle uusi verkkopalvelu sekä verkkosovellus videoiden tallennusta ja niiden tietojen hallintaa varten. Ensisijaisena tavoitteena oli toteuttaa nykyaikainen sovellus ja saada vanhat videot toimimaan myös uudessa järjestelmässä. Taustatutkimusta työhön tehtiin aiemmista CampusTV:n uudistussuunnitelmista ja siitä, miksei aiempia uudistuksia oltu otettu käyttöön.</p> <p>Tutkimuksen aikana selvisi, että työ on melko haastava toteuttaa, joten oli tutustuttava videoiden käyttöön erilaisissa sovelluksissa. Työssä tutustutaan ensin www-sovellusten ohjelmointitekniikoihin ja tietokantoihin ja miten niillä on mahdollista toteuttaa verkossa toimivia verkkopalveluita. Esittely kattaa myös yleisimmät videon striimaukseen tarkoitetut protokollat sekä videooittimet ja yleisimmät videomuodot ja -standardit. Työssä esitellään myös palvelun toteutus, eli miten näitä edellä mainittuja asioita käytettiin ja sovellettiin.</p> <p>Yksi iso asia videoiden käytössä verkkosivuilla on se, mitä tietoja videosta näytetään ja mistä ne haetaan. Esimerkiksi tässä työssä videot jouduttiin tallentamaan erilliselle tiedostopalvelimelle, kun taas videoiden tiedot tallennettiin omalle tietokantapalvelimelleen. Tämä aiheutti joitakin haasteita, joiden selvittämiseen meni melko paljon aikaa.</p> <p>Ohjelmointikielien ja oheisohjelmien kehitys on tarjonnut uusia mahdollisuuksia Internet-sivujen tekoon, ja ne tekevät sitä myös tulevaisuudessa. Tulevaisuudessa voi ehkä välttyä ainakin osittain tässä työssä kohdatuilta ongelmilta, koska ohjelmointikielien tulossa olevat versiot mahdollistavat esimerkiksi videon näyttämisen Internet-sivuilla ilman tietojen hakemisen hankaluutta.</p> <p>Työ onnistui tavoitteiden mukaisesti, niin sivuista kuin videoiden hallinnasta tuli sellainen kuin haluttiin, ja uusi järjestelmä ja sivut julkaistaan lähiaikana.</p>			
<p>Asiasanat (avainsanat) verkko-ohjelmointi, verkkopalvelut, video</p>			
<p>Sivumäärä 33 s.</p>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">Kieli Suomi</td> <td style="width: 33%;">URN URN:NBN:fi:mamk-opinn2010A2453</td> </tr> </table>	Kieli Suomi	URN URN:NBN:fi:mamk-opinn2010A2453
Kieli Suomi	URN URN:NBN:fi:mamk-opinn2010A2453		
<p>Huomautus (huomautukset liitteistä)</p>			
<p>Ohjaavan opettajan nimi Janne Turunen</p>	<p>Opinnäytetyön toimeksiantaja Mikkelin ammattikorkeakoulu/CampusTV</p>		

DESCRIPTION

 <p>MIKKELIN AMMATTIKORKEAKOULU Mikkeli University of Applied Sciences</p>		Date of the bachelor's thesis 8 March 2010
Author(s) Antti Ronkanen	Degree programme and option Business Information Technology	
Name of the bachelor's thesis Implementation of online video service		
Abstract <p>The purpose of this bachelor's thesis was to create new websites to CampusTV of Mikkeli University of Applied Sciences and an application that could be used to add videos to the websites and to control the saved information of these videos. The main priority was to create a modern application and also to make the old videos work in the new system. I had to study the previous renewal plans of CampusTV, and to find out why they were not implemented.</p> <p>During the study I found out that this work was fairly difficult to accomplish, and therefore I had to study how videos were used in different applications. This bachelor's thesis introduced first the techniques and databases, and how they made creating an online video service possible. I also introduced the most common protocols used in streaming, video players and some of video forms and standards.</p> <p>One major feature of using videos on web sites was the information shown about the video, and where this information came from. For example, in this application the videos had to be saved in a separate file server, and information about the videos was saved in a separate database server. This caused some challenges, and it took quite a lot of time to solve the problems.</p> <p>The development of programming languages and applications has provided a lot of new possibilities for making websites, and they will do so also in the future. In the future you can maybe avoid some of the problems faced in this study, because the new, still unpublished versions of programming languages permit the use of video on websites without the difficulty of getting the information from different locations.</p> <p>The study succeeded as planned. The new websites and the video application will be published soon.</p>		
Subject headings, (keywords) web programming, web services, video		
Pages 33 p.	Language Finnish	URN URN:NBN:fi:mamk-opinn2010A2453
Remarks, notes on appendices		
Tutor Janne Turunen	Bachelor's thesis assigned by Mikkeli University of Applied Sciences/CampusTV	

SISÄLTÖ

1	JOHDANTO	1
2	WWW-SOVELLUSTEN OHJELMOINTITEKNIIKAT	2
2.1	HTML, XHTML ja HTML5.....	3
2.2	CSS	5
2.3	JavaScript.....	7
2.4	PHP	8
2.5	Java	9
2.6	ASP.NET	11
2.7	MySQL-tietokannat	12
3	VIDEOIDEN NÄYTTÄMINEN SELAIMELLA	13
3.1	Streamaus-protokollat.....	13
3.2	Windows Media Player.....	14
3.3	Quicktime	16
3.4	Flash-video	17
3.5	Videostandardit.....	18
3.6	Digitaaliset videot.....	19
4	TOTEUTUS	20
4.1	Suunnittelu.....	20
4.2	Sivujen toteutus	21
4.3	Ongelmat.....	23
5	VIDEOKIRJASTO	25
5.1	Videon tallennus flash-palvelimelle	25
5.2	Videon haku flash-palvelimelta.....	26
6	TESTAUS JA KÄYTTÖÖNOTTO	28
7	PÄÄTÄNTÖ	29
	LÄHTEET	31

1 JOHDANTO

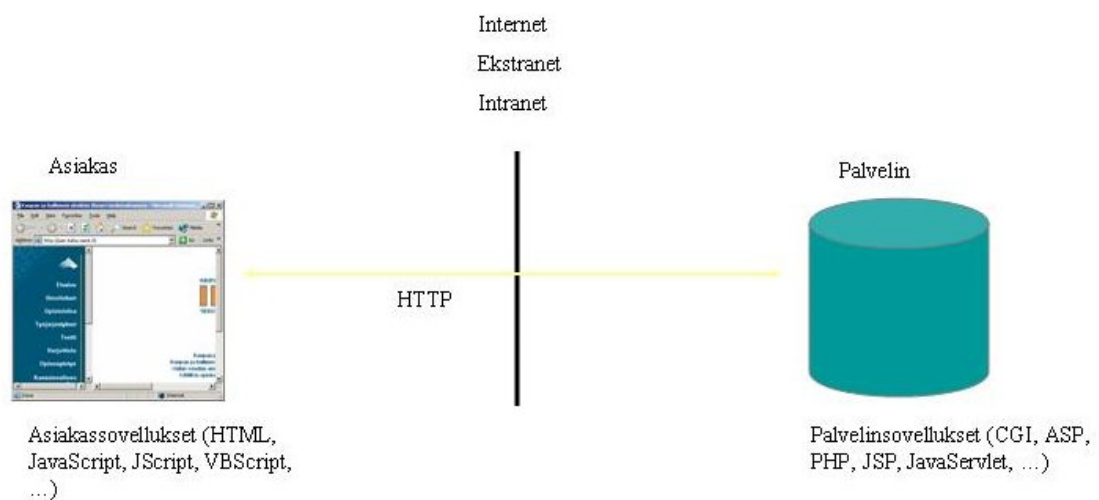
Tämän opinnäytetyön tarkoituksena on toteuttaa Mikkelin ammattikorkeakoulun CampusTV:lle, myöhemmin tässä työssä CTV, uusi julkaisujärjestelmä ja videokirjasto sekä ratkaista videoiden tallennukseen ja hakuun liittyviä ongelmia. Internetsivujen toteuttaminen on nykyään mielenkiintoinen projekti, koska tapoja toteutukseen ja esim. mielipiteitä ulkoasusta on yhtä monta kuin tekijöitäkin. Tämän opinnäytetyön parissa pyrin toteuttamaan sellaisen sovelluksen, joka palvelee käyttäjää ja sivujen omistajaa kaikkein parhaiten. Tavoitteena on, että sivuista tulisi kevyet, vaikka tämä asettaa haasteita videoiden käytön takia, sillä jo ne itsessään hidastavat sivuja.

Kun aletaan toteuttaa verkkopalveluja, pitää miettiä, mitä ohjelmointikieltä käytetään, koska vaihtoehtoja on runsaasti, esim. Java, ASP.NET, PHP sekä C-kieli. Nykyään yksi yleisimmistä verkkopalvelujen tekemiseen käytetyistä ohjelmointikielistä on PHP. PHP on hyvä valinta verkkopalvelun toteutukseen, koska se on vertailussa yksi johtavista dynaamisten web-palveluiden toteuttamiseen tarkoitetuista kielistä. Muilla edellä mainituilla ohjelmointikielillä toteutetaan enemmän ohjelmia kuin verkkopalveluja.

Luvussa kaksi esittelen yleisimmät ohjelmointikieliset, kaikista on myös pienet esimerkit kuvien muodossa. Luvussa kolme esitellään videoiden näyttämistä selaimella sekä kerrotaan yleistä tietoa yleisimmistä videoformaateista sekä mediasoittimista. Luvussa neljä esitellään toteutusratkaisu, mitä ratkaisuja on tehty ohjelmointikielen ongelmien suhteen. Luku viisi keskittyy kokonaan tässä työssä toteutettavan videokirjaston esittelyyn, mitä erikoista sen suunnittelu vaati, mihin ongelmiin törmättiin ja kuinka videoiden näyttämisen uusilla Campustv.fi- internetsivuilla ratkaistiin sekä miten videot tallennettiin tietokantaan ja kuinka toteutettiin videoiden haku tietokannasta.

2 WWW-SOVELLUSTEN OHJELMOINTITEKNIIKAT

Internetissä olevat WWW-palvelut eli verkkopalvelut ovat yhä useammin interaktiivisia eli sisältävät monenlaisia toimintoja. Eri ohjelmointikieliä on tuhansia, mutta vain osa niistä on levinnyt niin laajaan käyttöön, että kielten standardisointi on ollut aiheellista. Vanhimpia standardisoituja kieliä ovat Fortran ja COBOL, joiden ensimmäiset standardit laadittiin jo 40 vuotta sitten. Ohjelmointikieliä kehitettäessä syntyy helposti ns. omia murteita, joka ei kuitenkaan toimi kaikissa ympäristöissä, koska standardisoidut kielet ovat sanastoltaan ja kieliopiltaan tarkoin määritellyt. (Ohjelmointikielet, niiden ympäristöt ja varusohjelmisto 2009).



KUVA 1. Toimintaperiaate (Web-sovellusten ohjelmointi, johdanto 2009)

Web-palvelin eli web server on järjestelmä tai laite, jolle voidaan tallentaa web-dokumentteja tai se voi myös tuottaa niitä. Web-palvelimen perustehtävänä on ottaa vastaan pyyntöjä selaimilta ja vastata niihin, eli näyttää käyttäjän haluamia sivuja. Tämä toimii niin, että selain pyytää palvelimelta jotain sivua osoitteen tai muun tunnistimen perusteella (kuva 1), ja palvelin vastaa tähän lähettämällä sivun ja tarvittaessa muita tietoja sivusta. (Korpela & Linjama 2004, 14.)

Monen verkkopalvelun taustalla toimii jonkinlainen palvelinsovellus, joka suorittaa esim. tietokantahakuja, tallentaa tietoja tai lähettää sähköpostia. Palvelinsovellukset ovat erittäin tärkeä osa ohjelmointia, koska niiden avulla toteutetaan kaikki se, mihin tavalliset julkaisukielet eivät pysty. Palvelinsovellusten tekemiseen käytettyjä ohjel-

mointikieliä ovat esim. ASP, Java ja PHP. Palvelinsovelluksia vaativa verkkosovellus voi olla esim. verkkopankki, verkkokauppa, hakukone tai keskustelupalsta. (Keränen ym. 2005, 338.)

2.1 HTML, XHTML ja HTML5

HTML on lyhenne sanoista Hypertext Markup Language, joka siis tarkoittaa sellaista tekstiä, joka sisältää viittauksia muuhun tekstiin. Näillä viittauksilla on oma nimensä, eli hyperlinkki, normaalissa puhekielessä linkki. HTML tunnetaan myös kielenä, jonka avulla rakennetaan web-sivuja. HTML-sivujen tekemisessä suurin osa työstä on määrittää eri tekstien näyttäminen, koska tehdessä näitä sivuja sinun täytyy määrittää erikseen otsikko, dokumentin runko, lihavoidut tekstit jne. HTML:n runko ei sisällä mitään muuta kuin tekstiä ja merkkejä (kuva 2). Jos haluat esimerkiksi saada kuvan näkyviin sivuillasi, sinun täytyy tallentaa kuva palvelimellesi ja linkittää se koodiin. (Staflin 1996, 27 – 28.)

```
1 <html>
2 <head>
3   <title>Testailua</title>
4 </head>
5 <body>
6   <h1>Tähän tulee pääotsikko</h1>
7   <p>Tähän voit kirjoittaa normaalia tekstiä</p>
8 </body>
9 </html>
```

KUVA 2. Yksinkertainen HTML-esimerkki

Vuonna 1986 kansainvälinen standardoimisjärjestö ISO teki päätöksen standardista, joka oli tarkoitettu asiakirjan sisällön ja rakenteen merkitsemiseen. Tämä oli SGML (Standard Generalized Markup Language). Kielen tavoitteena oli, että sen avulla voitaisiin luoda omia merkintäkieliä omiin tarkoituksiin. Vuonna 1990 haluttiin toteuttaa hypertextijärjestelmä, joka oli alustasta riippumaton, eli sitä piti pystyä käyttämään erilaisista tietokoneista. Tällöin ryhmä nimeltään World Wide Web Initiative loi hypertextijärjestelmän käyttäen sen pohjana SGML:ää, näin siis syntyi HTML. (Staflin 1996, 41.)

HTML on kehitetty web-sivujen tekemiseen. Aluksi sivut sisälsivät pelkästään HTML:ää, mutta nykyään niihin sisältyy paljon muutakin, kuten kuvia, tyyliohjeita ja erilaisia ohjelmoituja toimintoja, mutta edelleen HTML on säilynyt sivun runkona. Selain näyttää HTML-sivut siten, että se aloittaa HTML-dokumentista ja liittää siinä olevien viittausten perusteella mukaan muita elementtejä. (Korpela & Linjama 2004, 3.)

XHTML:ää käytetään HTML:n uusien versioiden yleisenä nimityksenä. Kirjain X tulee Englannin kielen sanasta eXtensible, eli laajennettavissa oleva. Todellisuudessa kuitenkin XHTML:ää ei voi laajentaa sen enempää kuin tavallista HTML:ää, vaan kirjain X viittaa XML:ään. Eli XHTML:n muotosäännöt määritellään niin että siitä tulee yhteensopiva XML:n kanssa. (Korpela & Linjama 2004, 26.)

Eli käytännössä XHTML:n laajennettavuus tarkoittaa ainoastaan sitä, että voidaan laatia sekakielisiä dokumentteja XML:n kanssa (kuva 3), eli että osa dokumentista on tavallista HTML:ää ja osa taas XML:ää. (Korpela & Linjama 2004, 27.)

```

1  <xml version="1.0" encoding="ISO-8859-1"?>
2  <!DOCTYPE html public "-//W3C/DTD XHTML 1.0 Strict//EN"
3     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4  <html xmlns="http://w3.org/1999/xhtml">
5  <head>
6  <title>Testailua</title>
7  </head>
8  <body>
9     <h1>Tähän tulee pääotsikko</h1>
10    <p>Tähän voit kirjoittaa normaalia tekstiä</p>
11 </body>
12 </html>

```

KUVA 3. XHTML- esimerkki

HTML:n tulossa oleva versio on nimeltään HTML5. HTML-työryhmä rakentaa HTML5:sta avointa ja lisenssivapaata määrittystä web-sisällölle ja –sovelluksille. HTML-kielen uusista piirteistä mielenkiintoisimpia ovat mahdollisuus kaksiulotteisen grafiikan piirtämiseen, audio- ja videosisällön upottamisen mahdollisuus web-sivuille. (Web-sisällön tulevaisuutta tehdään nyt: W3C julkaisee HTML 5 –luonnoksen 2010.)

2.2 CSS

Jo HTML:n ensimmäisessä virallisessa määrittelyssä (HTML 2.0 vuonna 1995) mainittiin, että sivun ulkoasu riippuu tyyliohjeista. Tyyliohjeiden määrittelemine ja toteuttaminen etenivät hitaasti, joten selainten valmistajat lisäsivät HTML:ään erilaisia ulkoasun muokkauksen piirteitä, jolloin sivujen tekijät opettelivat käyttämään niitä ja tottuivat niihin. Varhaiset graafiset selaimet esittivät sivut yleensä melko karussa muodossa, yleensä pelkällä harmaalla taustavärillä, ilman mitään muotoiluja. (Korpela 2003, 467.)

CSS (Cascading Style Sheets) on yleisesti sanottuna dokumentin ulkoasua koskeva ehdotus, johon tarkoitukseen CSS on kehitetty. Kun dokumenttia tehdessä käytetään CSS:ää, ei ohjelmakoodiin tarvitse kirjoittaa erillisiä tyylimäärittelyjä. CSS:n käyttö tarkoittaa sitä, että HTML-koodissa jollekin elementille annetaan class-arvo, jonka avulla ohjelma osaa hakea tälle elementille tarkoitetun tyylin erillisestä CSS-tiedostosta. (Korpela 2009.)

Vuonna 2000 CSS:n käyttö oli jo huomattavasti yleistynyt. Tähän vaikutti muun muassa se että web-sivujen tekijät olivat alkaneet omaksua CSS:n käytön sivujen tekemisessä, vaikkakin sitä käytettiin siihen aikaan esim. linkkien alleviivausten poistamiseen. Tähän aikaan myös oppilaitoksissa ja oppikirjoissa alettiin puhua enemmän CSS:stä, joka edisti sen yleistymistä. Nykyään lähes kaikki web-sivut sisältävät jonkinlaisen tyylimäärittelyn CSS:n muodossa. (Korpela 2003, 468 – 469.)

CSS:n merkitys on kasvanut kahdesta syystä. Se tarjoaa web-sivujen tekijöille joustavan mahdollisuuden sivujen ulkoasun tekemiseen ja niiden muokkaukseen. CSS:n osaaminen on siis melkein välttämätöntä jos teet web-sivuja. Käytännössä CSS:ssä kyse on yleensä web-sivuista, yleensä HTML-muotoisista dokumenteista, joita web-selaimet esittävät. Yksi CSS:n keskeisistä ajatuksista on, että CSS-tyyliohje esittää kaikki ulkoasuehdotukset erillisenä kokonaisuutena, jonka selain ottaa huomioon, tai ei ota huomioon. Jos selain ei huomioi ulkoasuehdotuksia, niin web-sivu näkyy selaimella ilman värejä sekä ilman muita muotoiluja. (Korpela 2003, 2.)

Käytettäessä HTML:n kanssa, CSS:n tarkoituksena on siis vaikuttaa vain sivun ulkoasun joihinkin piirteisiin (kuva 4). Yleensä näissä tapauksissa ei siis kuvata ulkoasua

kovinkaan yksityiskohtaisesti, vaan asetetaan ainoastaan joitakin ulkoasun ominaisuuksia, ja muilta osin selain käyttää sen oletusasetuksia. (Korpela 2003, 3.)

```
.palaute {
    background-color: #333333;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 11px;
    line-height: 155%;
    color: #ffffff;
}

.yhteystiedot {
    background-color: #333333;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 11px;
    line-height: 155%;
    color: #ffffff;
}
```

KUVA 4. Esimerkki CSS-tyylitiedostosta

```
<title>Esimerkki</title>
<link rel="StyleSheet" href="includes/tyyli.css"/>
</head>
<body>
```

KUVA 5. Tyylitiedosto linkitettyä HTML - koodiin

Kuten kuvasta 5 näkyy, CSS-tyyliohje linkitetään HTML-sivuun. Tämä mahdollistaa helposti usean erilaisen tyyliohjeen käytön, johon viitataan eri sivuilta. Tällä tekniikalla vähennetään suuresti kirjoitustyötä siihen verrattuna, että jokaiselle HTML-sivulle kirjoitettaisiin tyylimäärittelyn suoraan HTML-koodin sekaan. Näin karsitaan myös kirjoitusvirheiden mahdollisuuksia. (Korpela 2003, 21.)

2.3 JavaScript

JavaScript on ohjelmointikieli, jota selain tulkkaa muodostaessaan dokumenttia.

Yleensä JavaScript-ohjelmointikieltä käytetään erilaisten toiminnallisuuden tekemiseen, toisin sanoen tehdään sellaisia ominaisuuksia, joita ei yleensä tehdä tai pystytä tekemään ns. normaaleilla julkaisukielillä, esim. tarkastetaan onko jonkin lomakkeen kaikki tekstikentät täytetty. (Keränen ym. 2005, 336.)

JavaScript-koodi voidaan muiden kielten tapaan kirjoittaa joko HTML-koodin sekaan (kuva 6) tai erilliseen tiedostoon. JavaScriptiä käytettäessä tulee ottaa huomioon, että siitä on monta versiota, ja jotkin ominaisuudet eivät välttämättä toimi kaikissa selaimissa. (Keränen ym. 2005, 336.)

Useat web-selaimet osaavat tulkita ja suorittaa JavaScript-ohjelmakoodia, mutta tulkinassa on kuitenkin joitain eroja, mistä johtuen JavaScript- kielellä toteutetut sovellukset eivät välttämättä toimi kaikissa selaimissa. Kieltä käytetään interaktiivisten ja dynaamisten sivujen toteuttamiseen. JavaScript on erinomainen väline esim. käyttäjän antamien syötteiden tarkistukseen. (Web-sovellusten ohjelmointi: Asiakastekniikat - JavaScript-kielen perusteita 2009).

```
1  <html>
2  <head>
3    <title></title>
4  </head>
5  <body>
6    <script language="JavaScript">
7      document.writeln("Toimiiko tämä koodi?");
8    </script>
9  </body>
10 </html>
```

KUVA 6. Yksinkertainen esimerkki JavaScript-koodista

2.4 PHP

WWW-kehittäjille riitti aluksi pelkkä HTML-kieli, koska HTML- kielellä tehtyjä dokumentteja voitiin lukea kaikilla käyttöjärjestelmissä toimivilla selaimilla. Staattisten sivujen päivitys oli kuitenkin erittäin hankalaa ja paljon työtä vaativaa, jolloin kehitettiin palvelimella tapahtuva CGI-ohjelmointi, jolla mahdollistettiin palvelimella tapahtuva dynaamisten sivujen generointi. (Peltomäki 1998, 129.)

Web-sivujen kehityksen alkuaikoina sivujen tekemiseen käytetty tekniikka oli melko rajoittunutta. Staattiset HTML-sivut tarjosivat tiedot, jotka haluttiin käyttäjille esittää. Nykyään on erittäin tärkeää osata tehdä dynaamisia sekä vuorovaikutteisia web-sivuja. (Martin & Tomson 2002, 6.)

PHP on lyhenne sanoista PHP: Hypertext Preprocessor. PHP:n ensimmäinen versio on julkaistu jo vuonna 1995, ja nykyään se on vertailussa johtava dynaamisten (tieto voi muuttua esim. tietokannan sisällön tai käyttäjän tekemän toiminnan seurauksena) web-palveluiden tuottamiseen tarkoitettu kieli. PHP on komentosarjakieli, eli sen ohjelmakoodi tulkitaan vasta ohjelman suoritusvaiheessa. (PHP 2009.)

Tämä tarkoittaa sitä, että PHP-sivut ja niiden sisältämä koodi käsitellään www-palvelimella ennen sivun lataamista asiakkaan selaimelle. Kuten muissakin vastaavissa kielissä, PHP:n kanssa käytetään usein HTML/XHTML-kieliä (kuva 7). Tätä kutsutaan upotetuksi tilaksi. PHP:n syntaksi muistuttaa erittäin paljon C-, Java sekä Perl-kieliä, mutta siihen on kuitenkin lisätty joitakin ainutlaatuisia piirteitä. PHP:n tavoitteena on tarjota nopea ja monipuolinen ympäristö web-kehityksen alalle. (PHP 2009.)

PHP:n esitteli ensimmäistä kertaa Rasmus Lerdorf vuonna 1994. Silloin se tunnettiin nimellä PHP/FI (Personal Home Page / Forms Interpreter). Se sisälsi jo silloin samoja toimintoja ja toiminnallisuutta kuin vielä tänäkin päivänä. PHP alkoi yleistyä melko nopeasti ohjelmien tekijöiden keskuudessa, ja vuonna 1997 se todettiin riittämättömäksi joidenkin sovellukseen tekemiseen. Tämän jälkeen sitä alettiin kehittää ja siitä muun muassa löydettiin paljon laajennusmahdollisuuksia, minkä myötä myös tietokantojen käyttö yleistyi ja PHP:n käyttö alan ammattilaisten keskuudessa käynnistyi kunnolla. Tämä versio, eli 3.0, muistutti jo jonkin verran tänä päivänä käytössä olevia PHP:n versioita. (The history of PHP 2009.)

```

1  <html>
2  <head>
3    <title>Testailua</title>
4  </head>
5  <body>
6    <?php
7      print "Toimiiko tämä koodi?";
8    ?>
9  </body>
10 </html>

```

KUVA 7. Yksinkertainen esimerkki PHP-koodista

Kun sivulla oleva PHP-koodi suoritetaan, selaimella näkyy ainoastaan ohjelman tulos. PHP-koodin voi nähdä ainoastaan silloin, kun käsitellään palvelimella olevia tiedostoja esimerkiksi FTP:n kautta, eli koodi ei näy edes silloin jos käyttäjä katsoo sivun lähdekoodin joka sisältää PHP:tä. Tällöin lähdekoodissa näkyy ainoastaan HTML-koodi. (Heinisuo & Rauta 2007, 13.)

2.5 Java

Java-ohjelmointikielen kehitys käynnistyi osana Sun Microsystemsin aloittamaa projektia, ja sen tarkoitus oli kehittää edistynyt ohjelmisto sulautettujen järjestelmien verkottamista varten. Ohjelmiston hallintaan tarvittiin pieneen, luotettavaan, siirrettävään, hajautettuun ja reaaliaikaiseen käyttöympäristöön sopiva ohjelmointikieli. (Peltomäki & Malmirae 2000, 6 – 7.)

Javan on kehittänyt Sun Microsystems, ja se on oliopohjainen ohjelmointikieli. Java on saavuttanut suuren suosion sekä ohjelmistoteollisuudessa että opetuksessa. Javan suosioon on vaikuttanut etenkin voimakas markkinointi ja selkeä olio-ohjelmointia tukeva rakenne. Javaa käytetään etenkin yrityssovellusten toteuttamisessa, koska sillä voidaan tehdä helposti verkkoon hajautettuja komponentteja ja toteuttaa tietokantayhteydet. Java tarjoaa myös muiden ohjelmointikielien tapaan tietokantarajapinnan. Tietokantayhteydet hoidetaan JDBC-rajapinnan avulla. Tässä on se hyvä puoli, että yhteys relaatiotietokantoihin voidaan ottaa suoraan koodista. (Peltomäki & Malmirae 2000, 6 – 7.)

```
1 // Luokka maailma
2 class HeiMaailma
3 {
4     public static void main(String args[]) {
5         System.out.println("Toimiiko tämä koodi?");
6     }
7 }
```

KUVA 8. Yksinkertainen esimerkki java-koodista

Java on monipuolinen ohjelmointikieli, jonka avulla voidaan toteuttaa esim. web-sivuihin liitettäviä ohjelmia (applet). Java ei ole millään tavalla sama kieli kuin JavaScript, vaikkakin niissä on joitakin samanlaisia piirteitä. Java-koodi siis kirjoitetaan johonkin editoriin, jonka jälkeen se käännetään sopivalla ohjelmalla, ja saadaan ns. luokkatiedosto. Nykyään sekä editori että kääntämiseen tarvittu ohjelma sisältyvät yleensä samaan sovellukseen. (Korpela 2009.)

Java on oliopohjainen ohjelmointikieli eli ohjelmat koostuvat erilaisista luokista ja olioista. Java sisältää monia erilaisia valmiita luokkakirjastoja, joita hyödyntämällä luodaan ohjelmia. Valmiit luokat sisältävät esim. luokkia käyttöliittymä ohjelmointiin, tiedostojenkäsittelyyn sekä grafiikan piirtämiseen. Jopa mediatiedostojen käsittelyyn on olemassa luokkia. Javan kehitysympäristö jakautuu kolmeen kokonaisuuteen, J2SE (Java 2 Standard Edition), J2EE (Java 2 Enterprise Edition) ja J2ME (Java 2 Micro Edition). (Keränen ym. 2005, 339.)

Jos Javaa halutaan käyttää palvelinsovellusten ohjelmointiin, palvelimelta täytyy löytyä Java-ympäristö. Tällainen palvelin mahdollistaa verkkopalvelun ohjelmoinnin käyttäen palvelinsovelmia. Javaa tukevia palvelimia ovat mm. Java Web Server, Apache sekä Tomcat. (Keränen ym. 2005, 339 – 340.)

2.6 ASP.NET

ASP (Active Server Pages) on Microsoftin palvelintekniikka, jossa käytetään ohjelmointikielenä VBScript- tai JScript-kieltä. ASP ei siis ole itsessään ohjelmointikieli, vaan eräänlainen tekniikka, joka antaa erilaisia tekniikoita ohjelmoijan käyttöön. Näiden tekniikoiden avulla voidaan suorittaa erilaisia tietokantahakuja, tietojen tulostamista selaimelle, evästeiden käsittelyä jne. (Keränen ym. 2005, 338.)

Microsoft .NET –arkkitehtuuri on kunnianhimoinen yritys yhdistää eri sovellusalueiden tekniikka ja palvelut yhteen yhteiseen aiheeseen. Tavoitteena on tarjota kehittäjille yhteinen ympäristö, jonka eri osa-alueet voisivat toimia yhdessä ilman erillisiä rajapintoja, yhteisten sääntöjen avulla. .NET–arkkitehtuuri koostuu kahdesta osasta: Common Language Runtime (CLR) ja .NET Framework Class Library. Näiden välille tehtävät on jaettu niin, että CLR toimii .NET-ympäristön perustana ja tarjoaa palvelut ajettavien ohjelmien taustalla. CLR:n tehtäviä on muun muassa huolehtiminen yleisestä käännetyn koodin ajamisesta, ajonaikainen turvallisuus ja käyttövarmuus. (Inkinen 2003, 4.)

Ajonaikainen ympäristö CLR siis kytkeytyy varsinaiseen järjestelmään toimien rajapintana, jonka päällä .NET –sovellukset toimivat. CLR tarjoaa myös muita hyödyllisiä mahdollisuuksia, sillä sitä voidaan käsitellä toisista ympäristöistä käsin, mikä taas tarjoaa lisää mahdollisuuksia .NETin käyttöön hajautetuissa toteutuksissa. (Inkinen 2003, 4 – 5.)

ASP-koodi kirjoitetaan HTML-tiedostoon tai ulkoisesti linkitettävään tiedostoon. Palvelin suorittaa koodin aina sivua näytettäessä, eli siis tekniikka on selaimesta riippumaton. ASP kirjoitetaan `<% ja %>` merkkien sisään (kuva 9), ja koodi-osassa voidaan määritellä muuttujat ja proseduurit. (Keränen ym. 2005, 338.)

```
1 <%  
2 Response.Write "Tämähan toimii"  
3 %>
```

KUVA 9. Yksinkertainen esimerkki ASP.NET-koodista

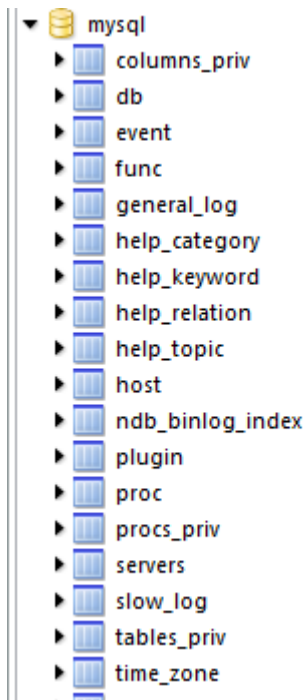
2.7 MySQL-tietokannat

Monien verkkopalveluiden taustalla toimii tietokanta, jonne sivujen sisällöt on tallennettu. Tietokantojen käyttö verkkopalveluiden tietojentallennuksessa on erittäin viisas keksintö varsinkin tiedon määrän kasvaessa. Tietokanta tarjoaa suuria hyötyjä esim. hakutoimintojen toteutukseen koska käytettäessä tietokantaa voidaan hakea tietoa ainoastaan jostain tietystä taulusta. (Keränen ym. 2005, 342.)

MySQL on tiedonhallintajärjestelmä, toiselta nimitykseltä RDBMS (Relational Database Management System), eli relaatiotietokantojen hallintajärjestelmä. MySQL:n avulla on mahdollista käsitellä tehokkaasti suuria tietomääriä, ja tietojen hallintaan on olemassa monia erilaisia työkaluja. MySQL:ää voidaan käyttää PHP:n lisäksi monen muun ohjelmointikielen yhteydessä tietokantana, tätä helpottamaan on kehitetty oma ODBC-ajuri (Open Database Connectivity), jonka avulla muodostetaan yhteys sovelluksesta tietokantaan. Ennen MySQL:ää hallittiin komentorivin avulla, joka ei nykyään olisi kovin käyttäjäystävällistä tai helppoa, joten hallintaan on kehitetty erilaisia graafisia työkaluja. (Web-sovellusten ohjelmointi: MySQL-relaatiotietokannan ylläpito ja tietokannan luominen 2009).

MySQL on tietokantojen hallitsemiseen tarkoitettu järjestelmä, jolla käsitellään relaatiotietokantoja. Tietokannat ovat selvästi suomennettuna varastoja, joihin voidaan tallentaa tietoa ja hakea sitä sieltä. Relaatiotietokannat ovat nykyään ylivoimaisesti eniten käytetty tietokantatyyppejä. Relaatiotietokantojen etu muihin verrattuna on se, että niissä tiedot on järjestetty tauluihin (kuva 10), ja tietokantahaussa voidaan helposti yhdistää tauluja niiden keskinäisillä suhteilla. MySQL:n on toteuttanut ruotsalainen yritys nimeltä MySQL AB, ja se kehitettiin alun perin kyseisen yrityksen sisäiseen käyttöön.

(Heinisuo & Rauta 2007, 37 – 38.)



KUVA 10. MySQL:n taulukkorakenne

3 VIDEOIDEN NÄYTTÄMINEN SELAIMELLA

Videoiden näyttämiseen selaimella on useita eri tapoja. Yleisin tapa on, että Internetissä oleva video ladataan ja se näytetään jonkun tarkoitukseen tehdyn soittimen avulla käyttäjälle. Tulevaisuudessa tämäkin asia mahdollisesti helpottuu, kun HTML5 tulee laajemmin käyttöön. Tässä työssä käytetään termiä ”striimaus”, jossa käyttäjä näkee ja kuulee kuvan ja äänen sitä mukaa kun media siirtyy verkkoa pitkin. Striimatavan videon laatuun vaikuttavat verkon nopeus sekä sen kuormitus. Kuormitukseen varaudutaan niin, että käyttäjän koneelle on ladattu n. 30 sekunnin mittainen puskuri, joka auttaa mahdollisissa ruuhkatilanteissa. (Streaming-videot 2010.)

3.1 Streamaus-protokollat

http-protokollan striimaus mahdollisuudet ovat melko rajalliset, joten striimausta varten on kehitetty protokollia, jotka sopivat huomattavasti paremmin reaaliaikaiseen tiedonsiirtoon. Tällaisten protokollien hyviä puolia on esim. se, että ne laskevat käyttäjän yhteyden nopeuden, ja pystyvät näin tarjoamaan jokaiselle käyttäjälle mahdollisimman hyvälaatuisen streamin. (Hertell 2005.)

Yksi tällaisista protokollista on Quicktime-palvelinten käyttämä RTP-protokolla (Real time Transport Protocol). Se on samantapainen kuin tavallinen http-protokolla, mutta erona on se, että toisin kuin http, RTP ei lataa tiedostoa kokonaan käyttäjän koneelle, vaan se välittää striimiä lähetyksen vaatimalla nopeudella, jolloin striimi käynnistyy melkein välittömästi lähetyksen alkamisen jälkeen. (Hertell 2005.)

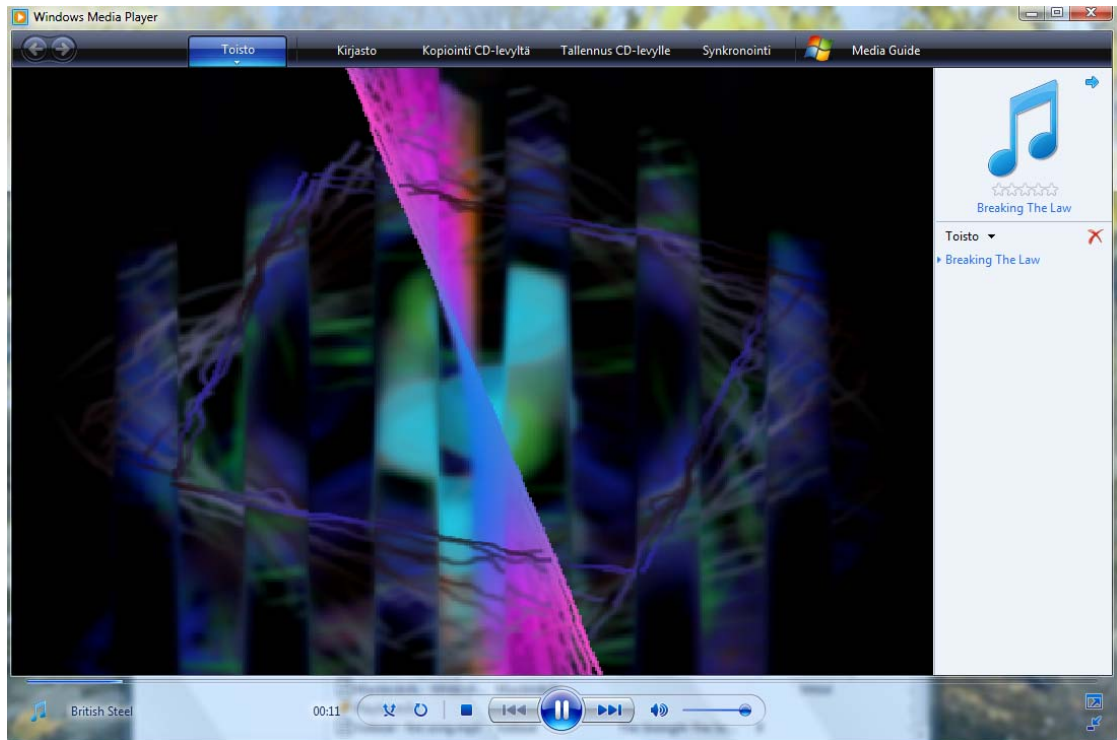
Toinen protokolla on RTSP (Real Time Streaming Protocol). Tämä protokolla toimii yhdessä RTP-protokollan kanssa, mutta RTSP mahdollistaa palvelimen kanssa kommunikoinnin. Tämän ominaisuuden avulla siis pystytään keskustelemaan palvelimen kanssa, joka taas mahdollistaa striimattavan lähetyksen kelaamisen ja hyppimisen kohdasta toiseen. (Sosinsky 2009, 646.)

Kolmas samaan tarkoitukseen oleva protokolla on UDP, joka on ideaalinen protokolla lähetyksien striimaukseen. UDP tärkein periaate on jatkuva tiedon lähettäminen eikä se ole liian tarkka lähetyksen perillemenosta. Tällöin ei synny periaatteessa mitään harmia, vaikka tietopaketteja häviäisikin matkalla, koska UDP vain lähettää tietoa koko ajan. Tällä tavalla lähetykseen ei pääse syntymään suuria katkoja. (Hertell 2005.)

3.2 Windows Media Player

Ensimmäinen Windows media-player tuli markkinoille Windows 3.0 käyttöjärjestelmän mukana. Media playeria ei kuitenkaan ollut käyttöjärjestelmässä automaattisesti, vaan se kuului saatavilla olleisiin multimedialaajennuksiin. Soitinta ei voi mitenkään verrata nykyajan Windows Media Playeriin, sillä siinä oli vain neljä nappia.

Vuonna 1995 julkaistiin Windows 95-käyttöjärjestelmä, joka sisälsi jo huomattavasti paremman soittimen, ja sillä voitiin jo toistaa AVI- sekä MPEG- tiedostoja. (Windows XP Media Player 2009.)



KUVA 11. Windows Vistan Media Player

Windows Media Player on tarkoitettu pääasiassa WMV- ja WMA-videoformaattien toistoon (Windows Media Video ja Windows Media Audio). Soitin kuitenkin mahdollistaa myös muiden formaattien toiston, riippuen mitä kodekkeja käyttäjän koneelle on asennettu. Windows Media palvelu sisältää erillisiä komponentteja striimausta varten. Striimaus-palvelimena toimii Windows Media Services, ja tiedostojen striimaukseen käytetään Windows Media 9 Series Encoder-komponenttia. (Hertell 2005.)

Streamien katsomiseen taas käytetään Windows Media 9 Series Player sovellusta. Windows Median kiistattomia haittapuolia on se, että se toimii käytännössä ainoastaan Windows-koneilla. Myös striimaus on mahdollista ainostaan Windows NT-palvelimien kautta. (Hertell 2005.)

Microsoftia on syytetty useasti Windows-käyttöjärjestelmän aseman väärinkäytöstä. Syyttelyn kohteeksi on joutunut Internet Explorerin lisäksi Windows Media Player. Microsoftin on katsottu vahingoittavan kilpailevien soittimien kehittäjiä. (Hertell 2005.)

3.3 Quicktime

Quicktime on paljon muutakin kuin pelkkä soitin. Se on myös tiedostoformaatti, ympäristö erilaiselle median hallinnoinnille, ja siitä on monia erilaisia sovelluksia median soittamiseen ja toistamiseen. Apple keksi Quicktimen omana formaattinaan multimedia-tiedostoille. Jokainen raita toimittaa eri sisältöelementtejä, kuten videota, audiota, interaktiivisia tiedostoja (flash), HTML:ää jne. Quicktime sisältää myös soittimen, joka toimii sekä Apple Macintoshissa että Microsoft Windowsissa. Quicktime sisältää mm. Streaming Server-nimisen sovelluksen, joka toimii Apple Mac OS X käyttöjärjestelmässä. Sen avulla voidaan ”striimata” mediatiedostoja näkymään Internetissä reaaliajassa. Quicktime tukee myös useimpia standardeja web-striimaukseen, kuten http, rtp ja RTSP, useimpia kuvaformaatteja (JPEG, BMP, PICT, PNG ja GIF). Quicktime 5 avainominaisuudet olivat tuki Macromedia Flash 4:lle sekä ON2 kodekki. (Topic 2002, 96 – 97.)

Virtausjakelu eli ”striimaus”

Virtausjakelulla tarkoitetaan sitä, että multimediaa lähetetään vastaanottajalle jatkuvana virtana, eli vastaanottaminen ei vaadi tiedoston lataamista tietokoneelle, esim. livelähetykset. Mediavirtaus mahdollistaa myös muun tiedon lähettämisen, kuten soittavien kappaleiden nimet sekä mainokset lähetyksen yhteydessä. Käytettäessä virtausjakelua, lähetystä ei ole mahdollista tallentaa tai kopioida. (Keränen ym. 2005, 346 – 347.)

Virtausjakeluun käytetään Quicktimen omaa lähetyspalvelinta. Palvelin käyttää lähetyksessä joko RTSP- tai RTP-protokollaa. Jos lähetyksiä lähetetään virtausjakeluna Quicktimena, lähetysten vastaanottamiseen tarvitaan Quicktime-ohjelma. (Keränen ym. 2005, 352 – 353.)

Striimausmahdollisuus on melko myöhäinen lisäys Quicktimeen. Alun perin käyttäjän oli ladattava koko tiedosto omalle koneelleen ennen sen katselua. Applen käyttämän ”Fast Start”-tekniikan avulla tämä on helpottunut. Kyseinen tekniikka tarkoittaa sitä, että käyttäjä lataa kyseessä olevan Quicktime-videon käyttäen joko http- tai RTMP-protokollaa. Kun tiedosto on alkanut latautua, video käynnistyy käyttäjän koneella, vaikka loppuosa on vasta latautumassa webin välityksellä. (Hertell 2005.)

3.4 Flash-video

Ennen flash-videota Internet-suunnittelijat ja –kehittäjät kohtasivat suuria ongelmia videon saamisessa Internetiin. Videot lataantuivat todella hitaasti, koska nettiyhteyksien nopeus ei ollut lähelläkään nykyistä tasoa. Tällöin käyttäjät joutuivat asentamaan erillisiä soittimia, joilla sitten pystyi katsomaan videoita pop-up-ikkunoissa. (Video technology center 2009.)

Nykyään useimmilla Internetiä käyttävillä ihmisillä on käytössään nopeat laajakaiselyyhteydet, ja suunnittelijat ja kehittäjän ovat käytännössä ottamassa Flash-videota käytettäväksi standardiksi. Videot voidaan toistaa suoraan joltain web-sivulta Adobe Flash Playerin avulla, eikä se tarvitse erillisiä lisäosia tai laajennuksia. (Video technology center 2009.)

Flash-video tarkoittaa Adoben valmistamaa digitaalisen videon pakkaus- ja julkaisuformaattia, jonka ensisijainen tarkoitus on mahdollistaa videon julkaiseminen Internetissä. Flash-videota käyttävät mm. YouTube, MySpace, Google Video, Reuters.com ja Yahoo Video. (Kekoni 2008.)

Flash-video esiteltiin vuonna 2002 Flash MX-nimisen ohjelman ilmestymisen yhteydessä. Kuten muillakin Internetiin tarkoitettujen videoformaattien kehittäjillä, on myös Adobella oma teknologiansa rakennettuna videoformaatin ympärille. Tällä siis tarkoitetaan niitä kaikkia komponentteja, joita tarvitaan videon julkaisuun ja niiden katselemiseen. Flash-video siis koostuu kaikkiaan kolmesta komponentista: julkaistava formaatti eli flash-video, katselijan koneelle asennettava mediasoitin eli Flash Player sekä palvelin. (Kekoni 2008.)

Flash Player on Adoben oma mediasoitin, jonka voi asentaa käyttämänsä Internet-selaimen lisäosaksi halutessaan katsoa esim. videoita selaimella. Flash Player pystyy toistamaan lähestulkoon kaikkia multimedian muotoja, mm. teksti, kuva, ääni, video, vektorit jne. Yksi Flash Playerin suosioon vaikuttaneista seikoista on sen koko, Flash Player 9 oli kooltaan 1,4 Megatavua, kun taas tässäkin työssä käsitelty Windows Media Playerin versio 11 oli kooltaan 24,6Mt. (Kekoni 2008.)

3.5 Videostandardit

M-JPEG perustuu valokuvien pakkaukseen tarkoitettuun JPEG-tekniikkaan, jolla saavutetaan hyvä kuvanlaatu. Siksi se onkin yleisesti käytössä videoeditoinnissa. MPEG-menetelmä (Moving Pictures Expert Group) on tarkoitettu digitaalisen median tallennukseen ja lähettämiseen. MPEG-menetelmällä pystytään pakkaamaan hyvänlaatuisen kuva ja ääni melko pieneen tilaan. MPEG sisältää neljä eri standardia, joissa jokaisessa on erilaiset määritelmät kuvalle ja äänelle. Näistä yleisimmät ovat MPEG-1, MPEG-2 ja MPEG-4. Alun perin näihin kuului myös MPEG-3, joka oli suunniteltu HDTV-käyttöön, mutta MPEG-2 korvasi sen. (Keränen ym. 2005, 221)

MPEG-1 standardi (ISO-11172) on valmistunut vuonna 1995. MPEG-1 on tarkoitettu kuvan ja äänen pakkaamiseen jollekin digitaaliselle tallennusmedialle, esim. CD-ROMille. MPEG-1 koodaus tapahtuu kolmessa osassa, systeemi, video ja ääni. Kun koodataan MPEG-1 muotoon, käytetään kiinteää datavirtaa (CBR).

Videokuvan pakkauksessa suositeltu kuvakoko PAL-järjestelmälle on 352 * 288 kuvapistettä ja kehysnopeus on 25. (Keränen ym. 2005, 221.)

MPEG-2 (ISO-13818) on vuonna 1997 valmistunut standardi yleisradiotasaisen kuvan pakkaukseen. Tätä standardia käytetään mm. lähes kaikissa DVD-levyissä sekä kaapeli- ja satelliittitelevisiossa, digi-tv-lähetyksissä sekä HDTV:ssä. MPEG-2 koodatun kuvan HDTV-tasoinen maksimikoko on 1920*1152. (Digitaalinen video 2010.)

MPEG-4 on kansainvälisen ryhmän kehittämä joukko videon ja äänen pakkaamistapoja. MPEG-4 standardi julkaistiin alkujaan vuonna 1997, ja siitä tuli standardi vuonna 1999. Sen kehitys on kuitenkin jaettu vaiheisiin, joista oleellisimmat ovat DivX:n ja Xvid:n käyttämä MPEG-4 SP/ASP ja Quicktime 7:n ja Blu-rayn käyttämä uudempi MPEG-4 AVC. (Keränen ym. 2005, 221.)

MPEG-4 standardia voidaan käyttää digitaalisessa videossa, interaktiivisissa grafiikkasovelluksissa ja multimediasa. Mm. Sonyn PSP sekä PlayStation 3 tukevat tätä pakkausmenetelmää yhteensopivissa videoissa. (Digitaalinen video 2010.)

3.6 Digitaaliset videot

Videomateriaalien tuottaminen, käsittely sekä tallennus muuttui digitaaliseksi käytännössä kokonaan 1990-luvun aikana. Nykyään koko video-muotoisen ohjelman tuottaminen kuvaamisesta lähettämiseen toteutetaan digitaalisesti. (Digitaalinen video 2010.)

Aiemmin videotekniikka yhdistettiin vain televisioon ja televisiossa esitettyihin ohjelmiin. Kun tietotekniikka kehittyi, se mahdollisti myös videoille uusia käyttötapoja. Nykyään digitaalista video-ohjelmaa voidaan katsella esim. multimediaesityksissä, Internet-sivuilla sekä matkapuhelimissa. (Digitaalinen video 2010.)

Tietokoneella esitettävät videot ovat aina digitaalisia. Videotiedostojen katselu tietokoneella onnistuu useilla erilaisilla videosoittimilla, esim. Microsoft Windowsin mukana tulevalla Media Playerilla. Muita videon toistoon tarkoitettuja soittimia ovat esim. Real Player, Applen Movie Player tai JW-Player. (Digitaalinen video 2010.)

Digitaalisella videolla tarkoitetaan siis videokuvaa, joka voidaan tallentaa, editoida, kopioida ja toistaa täysin digitaalisena niin, ettei laatu heikkene prosessin aikana. Digitaalisessa signaalissa on paljon enemmän monipuolisempaa dataa kuin analogisessa. Digitaalinen video voi myös sisältää erilaista metadataa ja aikakoodeja. Digitaalisen videon hyväksi puoleksi voi laskea myös sen, että se ei kulu käytössä, sitä voi kopioida käytännössä loputtomiin laadun heikkenemättä. (Digitaalinen video 2009.)

Videokuva joka liikkuu tietoverkoissa, on usein joko pakkaamistavasta tai hitaasta tiedonsiirrosta johtuen hidas, tai jopa nykivä. Ihmissilmä tarvitsee vähintään 17 kuvaa sekunnissa, jotta kuva näyttää kunnolla liikkavalta. Videokuva on kuitenkin käytännössä pakko pakata jollain tekniikalla, koska muuten se veisi niin runsaasti tilaa. (Videoeditointi 2009.)

4 TOTEUTUS

Tässä luvussa keskityn esittelemään opinnäytetyön toteutettavan osan suunnittelua ja toteutusta. Toteutukseen kuului uusien web-sivujen toteuttaminen, videokirjaston toteuttaminen sekä videoiden hallintatoimintojen toteuttaminen.

4.1 Suunnittelu

Aivan työn alussa pidimme yhden palaverin ns. projektiryhmän kesken, jossa olivat läsnä kaikki tähän projektiin keskeisesti kuuluvat henkilöt. Palaverissa tehtiin yhteenveto, missä ollaan menossa, koska CTV:n sivuja sekä videonhallintajärjestelmää on paranneltu ennenkin, mutta näitä parannuksia ei ole syystä tai toisesta saatu otettua käyttöön. Tämän jälkeen omien tietojeni ja palaverissa saatujen tietojen perusteella aloin selvittää tämän hetkistä tilannetta sekä uudistusten tärkeyttä ja niiden toteuttamismahdollisuuksia ja -tapoja.

Ensiksi piti selvittää tämän hetkinen palvelintilanne ja niiden uudistustarve, koska sivut tulisivat käyttämään omaa palvelinta web-sivuille ja tietokannalle, ja videot käyttäisivät omaa, jo käytössä olevaa tiedostopalvelintaan. Tämän lisäksi piti myös selvittää nykyään olemassa olevat ohjelmistot, jotta saatiin selville mahdolliset uusien ohjelmien tarpeet. Tämä oli tärkeää, koska vanha videonhallintajärjestelmä ei käyttänyt tietokantana MySQL:ää, joten piti varmistua, että palvelimelta löytyy sille tuki. Palvelimelta täytyi myös löytyä tuki PHP:lle. Johtopäätös oli, että palvelinjärjestelyt pystyttäisiin toteuttamaan nykyään olemassa olevilla palvelinkoneilla, jolloin uusien hankkimiseen tai asennukseen ei tarvitsisi käyttää mitään resursseja.

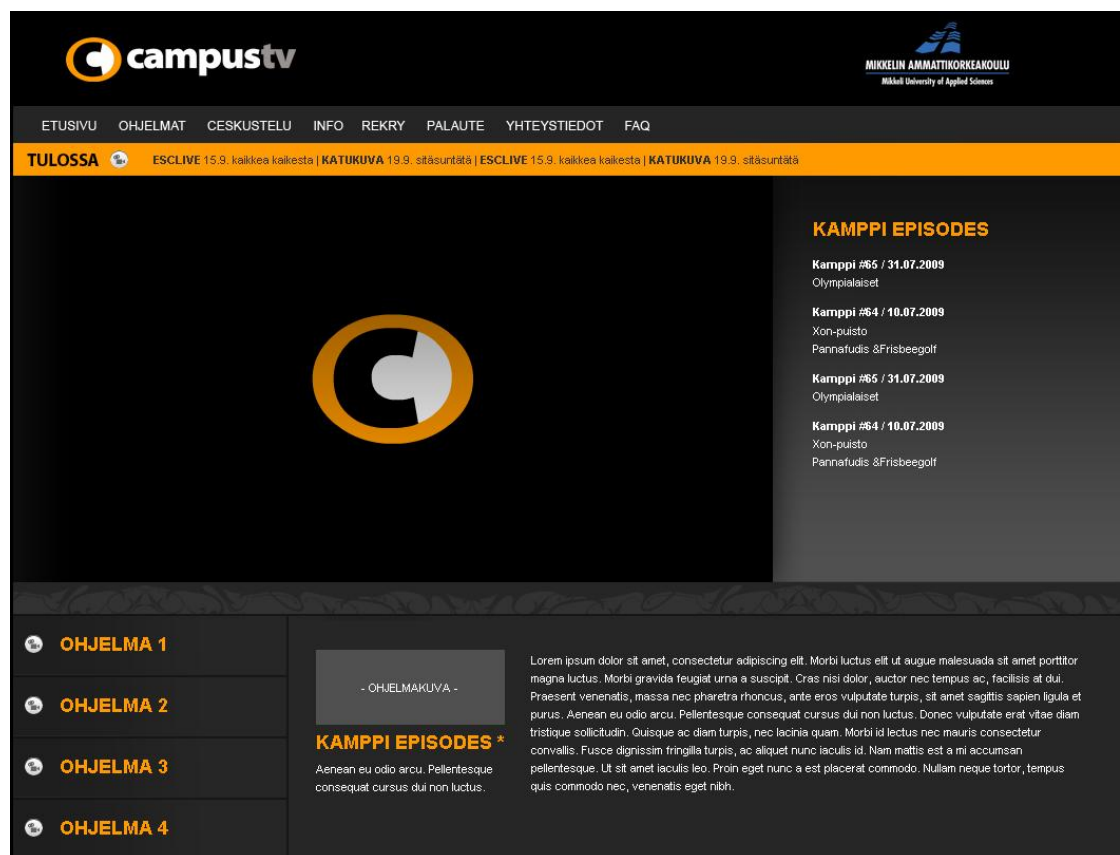
Itse sivujen visuaaliseen suunnitteluun minun ei täytynyt ottaa kauheasti kantaa, koska sivujen layout saatiin projektin ulkopuoliselta taholta. Layout-luonnos saatiin CTV:ltä, koska siellä oltiin jo aiemmin suunniteltu sivujen uudistusta ja tällöin oltiin tehty muutamia erilaisia luonnoksia uusien sivujen mahdollisesta ulkoasusta. Keskeisenä asiana uusiin sivuihin liittyy myös videonhallinta, videoiden haku tietokannasta sekä niiden katselu.

Tämän asian tiimoilta piti miettiä tarkkaan, minkälainen mediasoitin palvelisi parhaiten sivujen käyttäjien tarpeita ja mihin soittimeen olisi mahdollista saada siltä vaadit-

tavat ominaisuudet. Lopulta valittiin ohjelma nimeltä Jw-Player, koska se on helppo-käyttöinen ja helppo liittää sivujen koodiin. Sen hyvänä puolena on myös laajat ulko-asun sekä erilaisten toimintojen muokkausmahdollisuudet.

4.2 Sivujen toteutus

Sivujen toteutukseen valitsin ohjelmointikieleksi PHP:n sekä tietokannaksi MySQL:n. Kuten edellisessä luvussa mainitsin, ei minun tarvinnut alussa puuttua sivujen HTML-koodiin eli runkoon juuri ollenkaan, ja ainoat muutokset olivat ainoastaan joidenkin linkkien määrä, eli sivuilta täytyi poistaa pari linkkiä.

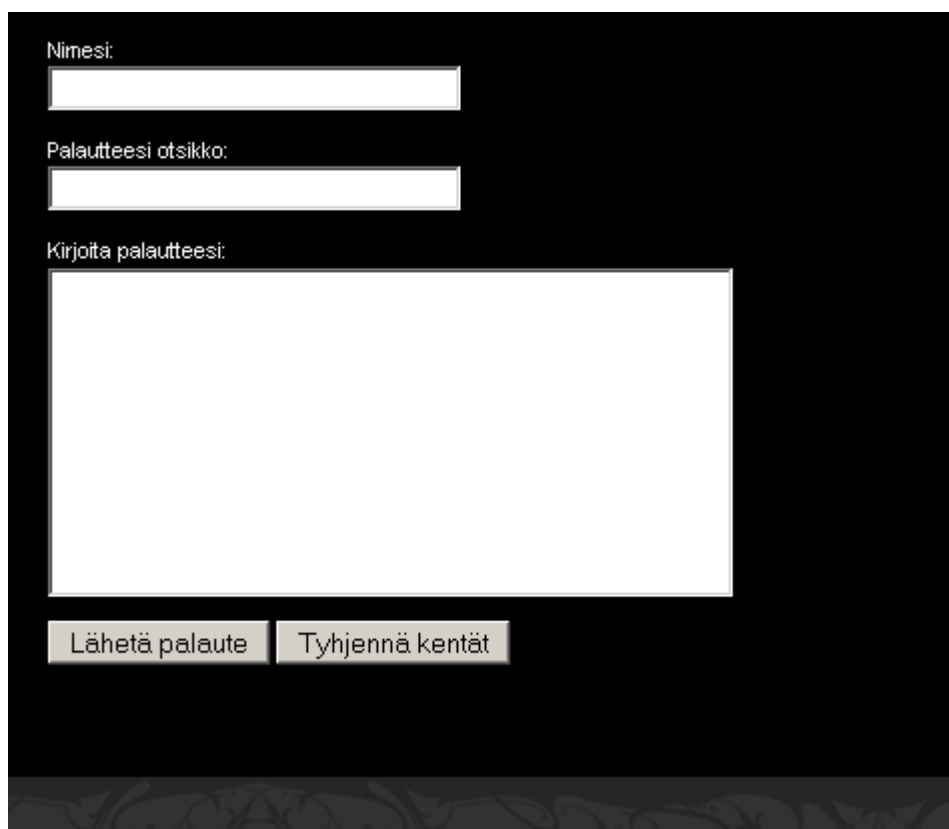


KUVA 12. Graafikolta saatu layout

Kuvassa 12 näkyvä layout siis tulisi olemaan sivujen ulkoinen ilme, ja keskellä sivua tulisi olemaan soittimen paikka. Eli tämä kuva siis on etusivusta. Haku-linkin takaa löytyvät uusimmat ohjelmat sekä tämä opinnäytetyön virallinen aihe, eli videokirjasto. Haku- sivulle siis tulee videoiden hakutoiminto, ja hakuehtoina täytyy pystyä antamaan videon/ohjelman nimi, mihin aiheeseen se liittyy sekä miltä ”kanavalta” se on tullut. Ceskustelu-linkki tullaan poistamaan lopullisesta versiosta, koska nykyisillään

sivuilla on sama linkki, eikä sillä ole käyttöä juuri ollenkaan. Info-linkin taakse tulee yleistä tietoa CTV:stä.

Rekry-linkin taakse tulee tiedot siitä, miten ihmiset voivat hakea CTV:lle harjoitte-
luun sekä mitä harjoittelu pitää sisällään. Palaute-linkin taakse tehdään palautelomake
(kuva 13), jolla voi antaa palautetta esimerkiksi sivuista tai vaikka harjoittelusta sekä
yleisesti CampusTV:n toiminnasta. Kun palautelomake on täytetty ja painetaan lähetä
-nappia, palaute lähetetään automaattisesti CTV:lle rekisteröityyn sähköposti-
osoitteeseen, josta työntekijät tai harjoittelijat voivat käydä sen lukemassa.



The image shows a feedback form with a black background. It contains three input fields: 'Nimesi:' (Name), 'Palautteesi otsikko:' (Subject), and 'Kirjoita palautteesi:' (Write your feedback). Below the fields are two buttons: 'Lähetä palaute' (Send feedback) and 'Tyhjennä kentät' (Clear fields).

KUVA 13. Palautelomake

Yhteystiedot linkin takaa löytyvät yhteydenottokeinot, postiosoitteet, puhelinnumerot, sähköpostiosoitteet sekä yhteyshenkilöiden tiedot. Faq-linkin taakse rakennetaan jonkinlainen usein kysyttyä-palsta. Jokaisen linkin alla olevaa sisältöä voi muokata haluamakseen, sinne voi esimerkiksi laittaa jonkin ajankohtaisen ilmoituksen tai muuta tietoa.

Sivuilla tulee olemaan sekä ns. julkinen puoli että hallintapuoli, jolle pääsemiseen vaaditaan käyttäjälle tehty tunnus ja salasana. Hallintapuolelta pystyy muokkaamaan

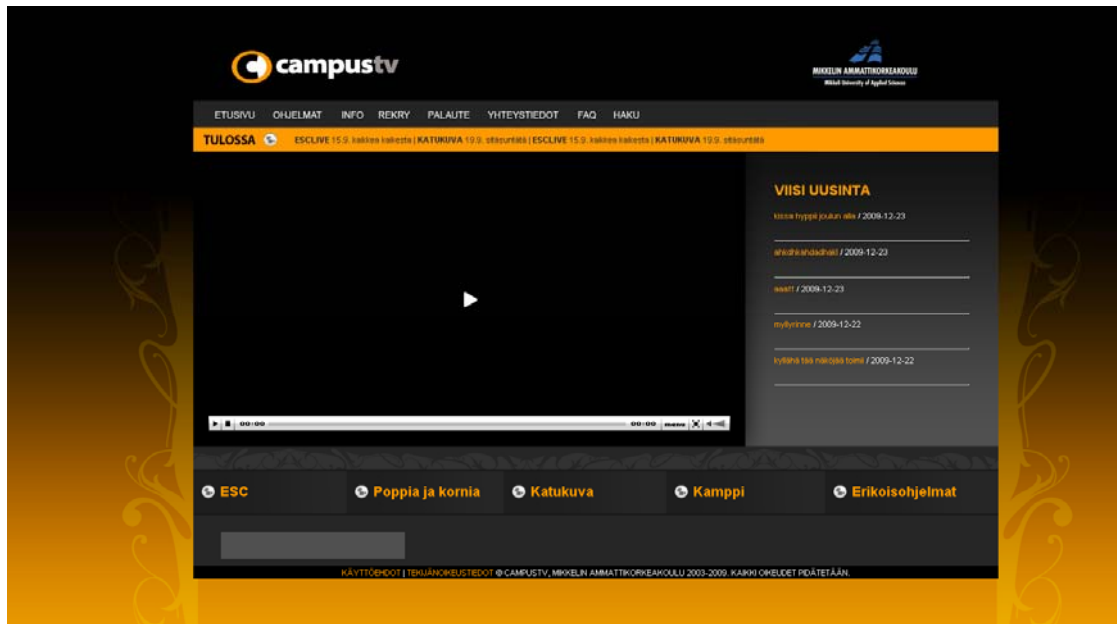
kaikkea sivuilla olevaa informatiivista sisältöä. Hallintasivulta pystyy myös laittamaan päälle eräänlaisen infon kun livelähetys on tulossa. Tämän hyvä puoli on siinä, ettei sivujen sisältöä tarvitse muokata erikseen livelähetystä varten.

4.3 Ongelmat

Videoita lisättäessä itse video tallentuu erilliselle tiedostopalvelimelle ja videosta tallennettavat tiedot, esim. kuvaus, tekijät, aihe yms. tallentuu tietokantaan erilliselle tietokantapalvelimelle, joten videoiden haku tuottaisi ongelmia, koska video ja sen tiedot sijaitsevat eri palvelimilla. Aloin ratkaista ongelmaa seuraavasti: kun video kopioidaan palvelimelle, sen tiedostopolku kopioidaan ja tietokantaan tallennettavien videon tietojen nimeksi annetaan tämä kopioitu tiedostopolku. Näin tiedot ja video saadaan linkitettyä kätevästi toisiinsa, ja haku toimii.

Toinen ongelma ilmeni, kun sivujen olivat normaaleilta toiminnoiltaan melkein valmiiksi. Toimeksiantaja halusi että kuvassa näkyvät ohjelmanlinkit olisivat vierekkäin (kuva 14), eikä allekkain kuten alun perin. Näin siis ohjelman kuvaukselle varattu tila pienenisi ja siirtyisi aivan sivun alareunaan koko sivun levyiseksi. Syynä näihin muutoksiin oli sivun rullaus, alkuperäisessä muodossaan sivu ei mahtunut kerralla näytölle, joten sitä joutui rullaamaan hiirellä alaspäin ja tämä ei ole käyttäjän kannalta mukavaa. Koska layoutin suunnitellut ja toteuttanut graafikko oli toteuttanut sivun ulkoasun tarkoitukseen tehdyllä ohjelmalla, aiheutti ongelmia saada sivun rakennetta muutettua suoraan HTML-koodista.

Seuraavan ongelman kohtasin, kun täytyi miettiä miten videoiden tiedot saataisiin siirrettyä hakutuloksista etusivulle, koska kun video on haettu hakutoiminnolla ja sitä klikataan, sivu ohjautuu automaattisesti etusivulle ja video alkaa pyöriä etusivun soittimessa. Tämähän ei olisi muuten ongelma, mutta en halunnut tehdä etusivulle uutta tietokantahakua videoiden tietoja varten, vaan tavoitteena oli saada ne kulkemaan videon mukana etusivulle.



KUVA 14. Muokattu ulkoasu

Ehkä kaikkein isoin ongelma oli, kun tietokanta oli saatu rakennettua valmiiksi. Tämän työn ja uusien sovellusten valmistuttua kaikkien videoiden tuli olla .flv-muodossa eli flash-videoita, kun taas aiemmin videot olivat .wsx-muodossa. Vanhoilla campustv.fi-verkkosivuilla oli käytetty erilaista soitinta, joka tuki wsx-muotoa. Uusilla sivuilla käytetty jw-player ei kuitenkaan tukenut tätä kyseistä muotoa, joten kaikki videot tulisi kääntää flash-videoiksi, ja videoiden hallintasovelluksen avulla muuttaa jokaisen videon nimestä päätteeksi .flv.

5 VIDEOKIRJASTO

Sivuille oli tarkoitus toteuttaa ns. videokirjasto, josta voisi hakea CTV:n julkaisemia videoita eri hakuehdoilla. Tässä luvussa kerron videokirjaston toteutukseen liittyvistä asioista.

5.1 Videon tallennus flash-palvelimelle

Videon tallennus palvelimelle toteutetaan edelleen manuaalisesti, eli kun video on kuvattu kameran kovalevyllä tms. ja editoitu, se kopioidaan palvelimen kovalevyllä aivan kuten mikä tahansa muu tiedosto. CTV:n jokaiselle kanavalle on olemassa oma kansio, jonka sisään tehdään videolle kansio, joka nimetään tallennuspäivämäärän mukaan, esim. 23-12-2009. Videon muut tiedot, (selite, kuvaaja jne.) tallennetaan tarkoitukseen tehdyllä web- lomakkeella (kuva 15) erilliselle MySQL- tietokantapalvelimelle, josta jokaisen videon tiedot linkitetään oikeaan videoon myöhemmin tässä työssä selitetyllä tavalla.

Tiedostonimi (*)	<input type="text"/>
Selite (*)	<input type="text"/>
Kuvaus	<input type="text"/>
Julkaistaan (*) (vvvv-kk-pp tt:mm:ss)	<input type="text" value="2010-01-19 10:00:00"/>
Ohjelmaan kuuluu	<input type="text"/>
Ohjelmaluokka (*)	Valitse luokka <input type="button" value="v"/>
<input type="button" value="Tallenna"/>	

KUVA 15. Videon tietojen tallennukseen tarkoitettu HTML-lomake

Videoiden lisäys siis on oma sovelluksensa, eikä sitä pysty käyttämään varsinaisten sivujen yhteydessä. Syy tähän on se, että suunnitelmana oli, ettei videoiden lisäys ja hallinta kuormita sivuja ja näin hidasta latausaikoja ja yleistä sivujen käyttöä.

Jokainen video kuuluu tiettyyn ohjelmaan, eli etusivulla näkyviin ESC:iin, Specialiin jne. Kun sivut olisivat valmiit, kaikki olemassa olevat videot siirrettäisiin arkistoon, ja sen jälkeen tehtäisiin ainoastaan ESC:iä ja Live-lähetystä. Eli videoiden hallinnasta pitää pystyä muuttamaan ohjelmaluokkaa.

Sovellus on toteutettu niin, että videon nimen täytyy täsmätä tarkasti videon fyysiseen sijaintiin jos sen halutaan toimivan. Videoiden hallintaan toteutettiin videoiden listaus-toiminto, eli kun video ja sen tiedot lisätään, voi katsoa että kyseinen video on listassa. Listaus-toiminnon yhtenä ominaisuutena on esikatselu, tämä on sen vuoksi tärkeä toiminto, että jos video ei toimi esikatselussa, ei se luultavasti toimi myöskään itse sivuilla, joten lisääjä voi siten heti muuttaa videon nimen niin että se toimii.

5.2 Videon haku flash-palvelimelta

Haku toimii siten, että sivuilla on haku-sivu (kuva 16), josta kävijä voi hakea videoita joko nimen perusteella tai sen perusteella, minkä kanavan ohjelmia haluaa etsiä. Hakutuloksille ei ole määritelty mitään rajoja, eli haku tulostaa kaikki videot, jotka osuvat hakutuloksen sisään.



KUVA 16. Videoiden hakusivu

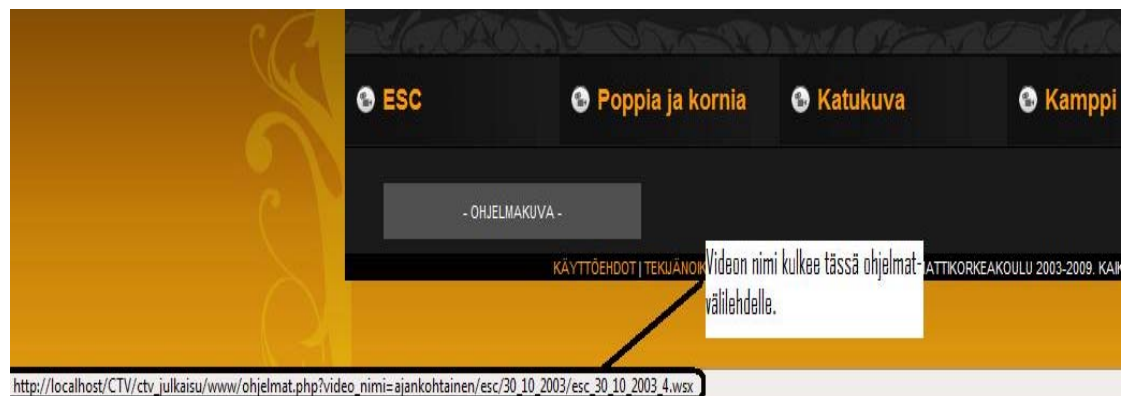
Hakutoiminto on linkitetty tiedostopalvelimeen seuraavasti: Sivujen videosoittimen koodiin on määritetty kaksi http-osoitetta (kuva 17), toinen on soittimen koodin osoite ja toinen osoite on tiedostopalvelimen osoite.

```
<script type='text/javascript'>
var so = new SWFObject('http://localhost/ctv_julkaisu/www/jw-player/player-viral.
so.addParam('allowfullscreen','true');
so.addParam('allowscriptaccess','always');
so.addParam('wmode','opaque');
so.addVariable('streamer','rtmp://flashms.mikkeli.amk.fi:1935/CampusTV/');
so.addVariable('file','ESC/01_08_2007/esc_01_08_2007b');
so.addVariable('type','rtmp');
so.addVariable('controlbar','over');
so.addVariable('stretching','fill');
```

KUVA 17. Videosoittimen koodi

Soittimessa oleva tiedostopalvelimen koodi on määritetty sen vuoksi, että näin saadaan haku toimimaan luvussa 4.3 mainitulla tavalla, eli saadaan haettua videon tiedot tietokantapalvelimelta ja tietoja vastaava video tiedostopalvelimelta. Koska videot on nimetty tiedostopolun perusteella, niin käytännössä nyt videon tiedostopolku määritetään lennosta. Eli tiedostopalvelimen osoite on nyt flashms.mikkeli.amk.fi, ja kun hakutoiminnolla valitaan joku video, ohjelma lisää videon nimen tiedostopalvelimen nimen perään, eli tällöin polku voisi olla esimerkiksi seuraava:

rtmp://flashms.mikkeli.amk.fi/ajankohtainen/esc/23-11-2009/ohjelma.flv



KUVA 18. Videon nimen kuljettaminen

Syy videoiden hakemiseen nimen perusteella on se, että se mahdollistaa videoiden tallentamisen omiin kansioihinsa. Videot olisi voitu tallentaa esim. järjestysnumeron perusteella (1, 2, 3, 4 jne.), mutta siinä tapauksessa jokaisen videon olisi pitänyt sijaita samassa kansiossa, ja silloin niiden selaileminen käsin olisi ollut melko hankalaa.

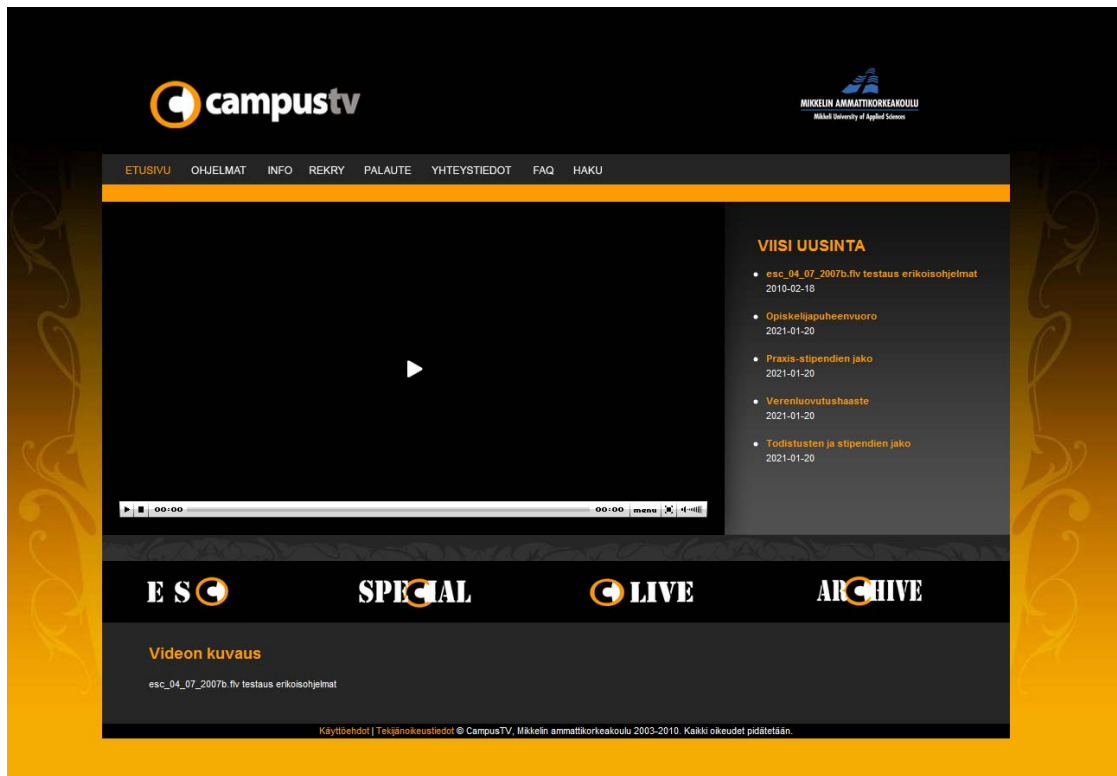
6 TESTAUS JA KÄYTTÖÖNOTTO

Kun sivut olivat muuten valmiit, suoritettiin niille järjestelmätestaus. Tämän testauksen tarkoituksena on lähinnä kokeilla sellaisia virhetilanteita, jotka voidaan tahallaan aiheuttaa tekemällä jotain ns. väärin. Testauksessa löytyi muutamia epäkohtia, esim. jos CSS-tyylit otetaan selaimesta pois käytöstä, jotkin kuvat olivat liian suuria.

Myös hakusivun vapaa-haku laatikosta pystyi manipuloimaan tietokantaa, eli nämä asiat oli korjattava ennen sivujen julkaisua. Sivut oli myös saatava menemään validaattorista läpi. Tämä siis tarkoittaa sitä, että otetaan sivun lähdekoodi näkyviin ja kopioidaan se <http://validator.w3.org>-sivuilla olevaan validaattoriin, joka tutkii onko se validia XHTML:ää.

Tämän jälkeen testattiin sivuja käytännössä, eli lisättiin video sekä sen tiedot MySQL-tietokantaan, ja yritettiin hakea videota selaimella. Testaus on tärkeää tehdä ennen sivujen julkaisua, koska kun sivut ovat julkaistu ja kaikkien käytettävissä, näitä mahdollisia olemassa olevia virheitä voi hyödyntää, jolloin pystytään käyttämään sivuja väärin tarkoituksiin tai jopa mahdollisesti estämään niiden toiminta. Sivujen testaus ei mennyt läpi, koska videoiden nimiä ei ollut muutettu osoittamaan oikeisiin paikkoihin luvussa 4.3 mainitulla tavalla, eli nimet olisi täytynyt muuttaa ja päätteet olisi pitänyt olla muotoa .flv.

Sivujen saamiseksi käyttöön tarvittiin palvelin web-sivuja ja tietokantaa varten. Tarvittava palvelin löytyi koululta, joten sen hankkimiseen ei tarvinnut käyttää muita resursseja. Käyttöön oton yhteydessä oli tärkeää muistaa muuttaa sivujen koodissa olevat http-osoitteet osoittamaan oikeaan paikkaan. Toteutusvaiheessa osoitteet luonnollisesti osoittivat tietokoneeseen, jolla sivuja tehtiin, mutta kun sivut siirrettiin palvelimelle, osoitteiden täytyi osoittaa palvelinkoneeseen. Tämän jälkeen sivut olivat Internetissä käytettävissä (kuva 19).



KUVA 19. Lopullinen ulkoasu

7 PÄÄTÄNTÖ

Opinnäytetyöni aihe löytyi sattumalta, kun eräs opettajamme ehdotti sitä. Innostuin aiheesta siinä vaiheessa kun tajusin, että joudun toteuttamaan sellaisen sovelluksen, jonka kanssa en ollut aiemmin ollut tekemisissä. Työ oli erittäin mielenkiintoinen, koska siinä joutui soveltamaan todella paljon niitä ohjelmoinnin taitoja, joita olin oppinut tunneilla sekä erilaisten projektien parissa, koska en ollut ikinä toteuttanut mitään vastaavaa kuin tämä. Tiesin, ettei kirjoista ole apua itse toteutuksessa, joten toteutusta piti lähteä tekemään ikään kuin kokeilemalla, ja sen jälkeen korjaamalla.

Työ oli siinä mielessä erittäin haastava, että siinä piti toteuttaa eräänlainen videosovellus. Nykyään moni yritys sekä Internet-sivu tarjoaa jonkinlaisia videopalveluja, esim. Youtube, Yle arkisto sekä jossain määrin myös yhteisöpalvelut, esim. Facebook. On mahdollista, että tulevaisuudessa vastaavat sivut yleistyvät, kuten nykyään olemassa olevat tekivät räjähdysmäisesti joitakin vuosia sitten.

Videoiden käyttö Internet-sivuilla tarjoaa mielestäni paljon mahdollisuuksia. Voisiko esim. joissakin kouluissa järjestettävien virtuaalikirssien sisältöä opettaa videoiden

avulla, nykyään ihmiset kuitenkin katsovat mieluummin tietokoneen näyttöä kuin lukevat esimerkiksi PowerPoint-esitystä tai Microsoft Word-dokumenttia. Mielestäni video-sovellusta voisi käyttää hyväksi myös muissa tapauksissa, esimerkiksi uuden työntekijän perehdytyksessä vaikka toimistotyöhön, työntekijä voisi tehdä töitä samalla kun kuulisi tai näkisi ohjeita videolta.

Tehdessäni opinnäytetyötä ja tutkiessani aiheeseen liittyvää materiaalia kiinnostuin tulevaisuuden mahdollisuuksista. Esimerkiksi tulossa oleva HTML:n uusi versio HTML5 tarjoaa Internet-sivujen kehittäjille todella paljon uusia mahdollisuuksia mm. uuden video-ominaisuuden muodossa. Tämä siis tarkoittaa sitä, että ohjelmoijan ei tarvitse linkittää sivuille mitään erillistä soitinta, vaan videot voidaan upottaa suoraan HTML-koodiin eli sivun runkoon.

Tulevaisuuden mahdollisuuksilla siis pystytään toteuttamaan verkkovideopalvelu ilman, että joutuu kohtaamaan hankaluuksia esim. sen muodossa, että videot ja niiden tiedot joudutaan hakemaan eri palvelimilta. Tämä helpottaa paitsi sovelluksen toteutusta, se tarjoaa helpotusta myös ongelmatilanteissa ja sivujen ylläpidossa. Tämä ei kuitenkaan tarjoa automaattisesti ns. helppoa tietä hyvien verkkopalveluiden tekemiseen, vaan alan ammattilaisten on opiskeltava uusia asioita ja tekniikoita, jolloin uudet tekniikat ja keinot yleistyvät, ja samalla ne luonnollisesti leviävät laajemmalle käyttöön.

LÄHTEET

Digitaalinen video. 2009. Digiwiki. WWW-dokumentti.

http://www.digiwiki.fi/fi/index.php?title=Videoiden_digitointi#Digitaalinen_video.
Luettu 17.2.2010.

Digitaalinen video. 2010. Keski-Pohjanmaan aikuisopisto. WWW-dokumentti.

<http://viestinta.kpakk.fi/oppimateriaalit/Digivideo/1-5tietoverkoissa.htm>. Päivitetty
14.3.2005. Luettu 24.01.2010.

Heinisuo, Rami, Rauta, Ilkka 2007. PHP ja MySQL. Helsinki: Gummerus.

Hertell, Janne 2005. Äänen stremaus. Tampereen ammattikorkeakoulu. WWW-
julkaisu. [https://oa.doria.fi/bitstream/handle/10024/5235/-
TMP.objres.51.pdf?sequence=1](https://oa.doria.fi/bitstream/handle/10024/5235/-TMP.objres.51.pdf?sequence=1). Päivitetty 20.4.2005. Luettu 04.02.2010.

History of PHP. 2009. The PHP Group. WWW-dokumentti.

<http://fi2.php.net/manual/en/history.php.php>. Päivitetty 2.10.2009. Luettu 7.10.2009.

Inkinen, Ville 2003. ASP.NET. Jyväskylä: Docendo.

Kekoni, Tuomo 2008. Flash-video ja sen merkitys videon julkaisemiselle Internetissä.
Tampereen ammattikorkeakoulu. WWW-julkaisu. [https://oa.doria.fi-
/bitstream/handle/10024/42288/Kekoni,%20Tuomo.pdf?sequence=1](https://oa.doria.fi/bitstream/handle/10024/42288/Kekoni,%20Tuomo.pdf?sequence=1). Päivitetty
20.5.2008. Luettu 27.1.2010.

Keränen, Vesa, Lamberg, Niko, Penttinen, Jukka 2005. Digitaalinen media. Jyväsky-
lä: Docendo.

Korpela, Jukka K. 2003. CSS-tyylit. Jyväskylä: Docendo.

Korpela, Jukka K, Linjama, Tero 2004. XHTML – käsikirja. Jyväskylä: Docendo.

Korpela, Jukka 2009. Java-sovelmat ja Flash-tekniikka. WWW-dokumentti.
<http://www.cs.tut.fi/~jkorpela/webjulk/3.4.html>. Päivitetty 21.6.2009. Luettu 24.01.2010.

Korpela, Jukka 2009. Värikkyttä: Tyylisäännöt. WWW-dokumentti.
<http://www.cs.tut.fi/~jkorpela/webjulk/2.5.html>. Päivitetty 21.6.2009. Luettu 9.11.2009.

Liron, Marc 2009. Windows XP Media Player. WWW-dokumentti.
<http://www.updatexp.com/windowsxpmmediaplayer.html>. Päivitetty 19.6.2008. Luettu 10.11.2009.

Martin, Joe, Tomson, Brett 2002. ASP.NET – Trainer Kit.

Ohjelmointikielet, niiden ympäristöt ja varusohjelmisto. 2009. SFS. WWW-dokumentti. <http://www.sfs.fi/it/aihealueet/ohjelmointikielet/>. Luettu 21.11.2009.

Peltomäki, Juha, Malmirae, Pekka 2000. Java-ohjelmoinnin peruskirja. Jyväskylä: Teknolit Oy.

Peltomäki, Juha 1998. WWW-ohjelmointi. Jyväskylä: Teknolit Oy.

PHP. Johdatus PHP-ohjelmointiin. 2009. K2-mediat. WWW-dokumentti.
<http://www.2kmediat.com/php/johdanto.asp>. Luettu 6.10.2009.

Sosinsky, Barrie 2009. Networking bible. Indiana: Wiley Publishing.

Staflin, Rolf 1996. HTML-ohjelmointi. Vantaa: Pagina Oy.

Streaming-videot. 2010. Keski-Pohjanmaan aikuisopisto. WWW-dokumentti.
<http://viestinta.kpakk.fi/oppimateriaalit/Digivideo/8-3streaming.htm>. Päivitetty 14.3.2005. Luettu 24.01.2010.

Topic, Michael 2002. Streaming media demystified. McGraw-Hill.

Videoeditointi. 2009. Joensuun yliopisto. WWW-dokumentti. http://www.joensuu.fi/-opetusteknologiakeskus/palvelut/materiaalit/multimedia/sv_video.html. Päivitetty 20.10.2006. Luettu 19.2.2010.

Video technology center. 2009. Adobe. WWW-dokumentti. <http://www.adobe.com/devnet/video/>. Luettu 10.11.2009.

Web-sovellusten ohjelmointi, johdanto. 2009. Oulun seudun ammattikorkeakoulu. WWW-dokumentti. <http://www.oamk.fi/sbc/www/johdanto.php>. Luettu 21.11.2009.

Web-sovellusten ohjelmointi: MySQL-relaatiotietokannan ylläpito ja tietokannan luominen. 2009. Oulun seudun ammattikorkeakoulu 2009. WWW-dokumentti. <http://www.oamk.fi/sbc/-www/mysql.php#ohjelmistot>. Luettu 21.11.2009.

Web-sovellusten ohjelmointi: Asiakastekniikat - JavaScript-kielen perusteita. 2009. Oulun seudun ammattikorkeakoulu 2009. WWW-dokumentti. <http://www.oamk.fi/sbc/-www/javascript.php#johdanto>. Luettu 21.11.2009.

Web-sisällön tulevaisuutta tehdään nyt: W3C julkaisee HTML 5 –luonnoksen. 2010. W3C. WWW-dokumentti. <http://www.w3c.tut.fi/press/2008/0122-html5/index.html>. Luettu 01.02.2010.