

Kirill Shamin

Verkkopalvelun suunnittelu ja toteutus

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikka

Insinööriytyö

27.9.2016

Tekijä(t) Otsikko	Kirill Shamin Verkkopalvelun suunnittelu ja toteutus
Sivumäärä Aika	37 sivua + 1 liite 27.9.2016
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Ohjelmistotekniikka
Ohjaaja(t)	Ohjaaja Simo Silander
<p>Insinöörityön tarkoituksena on suunnitella ja toteuttaa verkkopalvelu, joka toimisi oman suunnittelutoimiston kotisivuna.</p> <p>Koska suunnittelutoimiston pääpalveluna on verkkosivujen myynti sekä yrityksille että yksityisille, työympäristöön ja tarvittaviin teknologioihin pitää syventyä mahdollisimman tarkoin. Tämän takia työssä ei käytetä mitään CMS-järjestelmää, vaan kaikki toteutetaan itsenäisesti.</p> <p>Työ vaatii HTML:n ja CSS:n ymmärrystä ja osaamista. Myös Adobe Illustratorin, Photoshopin tai muun vastaavan kuvankäsittelyohjelmiston tuntemus on tarpeen.</p> <p>Lopputuloksena on mielenkiintoinen, selkeä ja kilpailukykyinen suunnittelutoimiston yrityssivu, joka toteutuessaan synnyttäisi luottamuksellisen kuvan vierailta ja halun tilata palveluita kyseiseltä yritykseltä.</p>	
Avainsanat	verkkosivu, kotisivu, yrityssivu, html, css

Author(s) Title	Kirill Shamin Web page design and development
Number of Pages Date	37 pages + 1 attachment 27 September 2016
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructor(s)	Simo Silander, Project Manager
<p>The purpose of this thesis work was to describe the design and implementation of a web-site for my own web design agency.</p> <p>Since the main service provided by a web design agency is to design and develop webpages, the main point of this project was to deeply immerse into the working environment and the technologies used. Therefore, everything presented in this work has been made personally, without using any CMS-system.</p> <p>The main technologies implemented in this work were HTML and CSS. Knowledge of graphical editors, such as Adobe Illustrator or Adobe Photoshop was also required.</p> <p>As a result, an interesting and competitive business page of a web agency, capable of impressing visitors and creating a positive attitude towards the services of the company, was developed.</p>	
Keywords	web page, www, design, html, css

Sisällys

Lyhenteet

1	Johdanto	1
2	Käytettävät teknologiat	2
2.1	HTML	2
2.2	CSS	2
2.3	Sovellukset	3
2.3.1	Teksti- tai lähdekoodieditori	3
2.3.2	FTP-asiakasohjelma	3
2.3.3	Piirto-ohjelma	4
3	CMS-järjestelmät	5
3.1	Johdanto	5
3.2	Vahvuudet	5
3.3	Heikkoudet	6
3.4	Yhteenveto	6
4	Suunnittelu	8
4.1	Johdanto	8
4.2	Verkkosivujen määrittely	8
4.3	Yritysilme	9
4.4	"ETUSIVU"	11
4.5	"PALVELUMME"	12
4.6	"PORTFOLIO"	14
4.7	"YHTEYSTIEDOT"	15
4.8	"TILAA PROJEKTI"	16
5	Toteutus	17
5.1	Johdanto	17
5.2	Doctype ja html -elementit	18
5.3	Head-elementti	18
5.4	Robots.txt	21
5.5	Sitemap.xml	22
5.6	Navigointi	23
5.7	"ETUSIVU"	27

5.8	"PALVELUMME"	29
5.9	"PORTFOLIO"	30
5.10	"YHTEYSTIEDOT"	32
5.11	"TILAA PROJEKTI"	34
6	Yhteenveto	37
	Lähteet	38
	Liitteet	
	Liite 1. Linkki "Suunnittelutoimisto K&K" verkkosivulle	

Lyhenteet

HTML	HyperText Markup Language, merkintäkieli verkkosivujen rakenteen määrittelyyn.
CSS	Cascading Style Sheets. Esityskieli verkkosivujen ulkoasun määrittelyyn.
CMS	Content Management System. Sisällönhallintajärjestelmä.
FTP	File Transfer Protocol. Tiedostonsiirtomenetelmä kahden tietokoneen välille.
SVG	Scalable Vector Graphics. Kaksiulotteisten vektorikuvien kuvauskieli.
URL	Uniform Resource Locator. Merkkijono, jolla kerrotaan tietyn tiedon/tiedoston paikka.
Tagi (Tag)	Tunniste.

1 Johdanto

Tämän opinnäytetyön tarkoituksena on näyttää, miten verkkosivustoa suunnitellaan ja toteutetaan alusta lähtien. Vaikka lopputuloksena saadaan suunnittelutoimiston yrityskotisivu, tässä insinöörityössä olevia ohjeita voidaan käyttää minkä tahansa verkkosivuston tekemiseen.

Opinnäytetyössä käydään läpi esimerkiksi seuraavia asioita: mistä verkkosivuston teko kannattaa aloittaa, mitä teknologioita ja ohjelmistoja tulee käyttää ja miten tehdään suunnittelu ja toteutus eri vaiheissa. Tätä projektia tehtäessä ei käytetä mitään CMS-järjestelmää, vaan kaikki toteutetaan itsenäisesti, joten työn ohessa on mahdollisuus oppia uusia asioita ja kehittää itseään. Kuitenkin käydään lyhyesti läpi, miksi CMS-järjestelmiä on olemassa ja mitä vahvuuksia ja heikkouksia siinä on.

Pääsääntöisesti insinöörityö jakaantuu kahteen pääosaan: suunnitteluun ja toteutukseen. Suunnitteluvaiheessa määritellään mahdollisimman tarkasti yritysilmettä ja sivujen sisältöä. Toteutuksessa rakennetaan kotisivua ja sen ulkoasua, kuten oli suunniteltu.

Lopuksi saadaan valmiiksi täysin toimiva ja käyttäjäystävällinen verkkosivusto, jonka tarkoitus on luoda luotettava ja positiivinen kuva asiakkaille.

2 Käytettävät teknologiat

2.1 HTML

Tärkein ja eniten käytetty teknologia tässä opinnäytetyössä on HTML eli Hypertext Markup Language [1]. Lähes jokainen internetsivu on rakennettu sen avulla. HTML on kuvauskieli, joka on suunniteltu erityisesti verkkosivujen merkintään. Juuri HTML tarjoaa tarvittavat ohjeet selaimelle siitä, miten erilaisia elementtejä ja tekstejä näytetään käyttäjälle. Myös sen avulla määritellään, missä järjestyksessä elementtejä näytetään. Aina pitää ottaa huomioon se, että eri selaimilla sama koodi voi tuottaa erilaisen visuaalisen lopputuloksen.

HTML-koodi koostuu tunnisteista eli tageista. Niiden avulla kerrotaan selaimelle, miten sisältöä näytetään. Tunnisteet ovat aina kulmasulkeissa muodossa

```
<tunniste> sisältö </tunniste> [1]
```

Esimerkiksi tekstiä lihavoidaan käyttämällä b-tunnistetta näin:

```
<b> Lihavoitu teksti </b>
```

Koodin tuottamiseen voidaan käyttää mikä tahansa teksti- tai lähdekoodieditoria. HTML-tiedostojen päätte on ".html", mutta saa käyttää myös ".php"-päätettä, mikäli verkkosivu sisältää lisäksi PHP-koodia.

2.2 CSS

CSS eli Cascading Style Sheets on tyyliä kuvaava säännöstö [2]. CSS on periaatteessa HTML:n laajennus, joka huomattavasti laajentaa kuvauskielen mahdollisuuksia. Tyyli-tiedostojen käyttö on välttämätöntä graafisessa suunnittelussa. CSS:n tärkeimmät ja käytetyimmät ominaisuudet ovat:

- leveyden ja korkeuden asettaminen
- reunaviivat ja täytteet

- kirjaisintyytit eli fontit ja niiden ominaisuudet (mm. värit, koko, alleviivaus)
- asemointi
- animointi (uudessa CSS3:ssa). [2.]

CSS-tiedostossa olevat perussyntaksit ovat muodossa:

```
valitsin {ominaisuus: arvo;} [2]
```

Esimerkiksi h1-otsikkoelementin tekstin väri määritellään näin:

```
h1 {color: #fff;}
```

Tyylitiedoston tarkoituksena on erottaa dokumentin rakenne ja sen ulkoasu, joten jatkossa olisi helpompi ja mukavampi muokata ja ylläpitää verkkosivustoa. On suositeltava pitää kaikki css-tiedostot omassa kansiossa (mm. `"/style/"`).

2.3 Sovellukset

2.3.1 Teksti- tai lähdekoodieditori

Koodin tuottamista varten tarvitaan teksti- tai lähdekoodieditori. Näitä on olemassa tosi paljon, joten on vaikea suositella jotain tiettyä editoria, koska se on makuasia. Itse olen käyttänyt Atom-sovellusta tätä projektia kirjoittaessani. Hyvät editorit, joita kannattaa ottaa myös huomioon, ovat Sublime ja Notepad++.

2.3.2 FTP-asiakasohjelma

Kaikki HTML- ja CSS-tiedostot sijaitsevat palvelimella. Jos haluaa lisätä, poistaa tai muokkaa näitä tiedostoja, pitää käyttää jotain FTP-asiakasohjelmaa. Tämän ohjelman tarkoitus on muodostaa yhteys tietokoneen ja palvelimen välissä. Itse olen käyttänyt FileZilla-sovellusta, joka on minun mielestäni yksi parhaimmista olemassa olevista FTP-asiakasohjelmista.

2.3.3 Piirto-ohjelma

Sivun graafista suunnittelua varten joudutaan käyttämään piirto- tai kuvankäsittelyohjelmaa. Hyvä valinta on esimerkiksi Adobe Illustrator, joka on vektorigraafikkaan perustuva piirto-ohjelma, joka sopii hyvin esimerkiksi sivun ulkoasun suunnitteluun ja logojen valmisteluun. Nykyään hyvä ja suositeltu tapa on käyttää SVG- eli vektorikuvia tavallisten JPG- tai GIF-kuvien sijaan.

SVG:n vahvuudet ovat:

- parempi skaalautuvuus
- selkeämpi näkyvyys retina-näytöillä
- pienempi koko, esimerkiksi PNG-tiedostoon verrattuna
- kuvan sisältö kirjoitettu XML-kielellä, joten muokausmahdollisuus myös tekstieditorilla
- tiedostoon sisään voidaan lisätä CSS tai JavaScript.

3 CMS-järjestelmät

3.1 Johdanto

Yleensä, kun verkkosivulle halutaan lisätä esimerkiksi tekstiä tai mediaa, joudutaan kirjoittamaan koodia HTML- ja CSS-tiedostoihin. Mutta tällä hetkellä on olemassa aputyökalu niille, jotka eivät osaa koodata tai osaavat sitä huonosti. Sen nimi on CMS-järjestelmä. CMS eli Content Management System on alusta verkkosivuille, joka voi helpottaa ja nopeuttaa verkkosivun tuottamista. [4] Sillä pystytään esimerkiksi muokata sisältöä käyttämällä graafista editoria, joten henkilön ei tarvitse kirjoittaa yhtään koodia. Nykyään suurin osa internetsivuista on toteutettu jollain CMS-järjestelmällä, joita on olemassa tällä hetkellä tosi paljon. Ne ovat joko maksullisia tai maksuttomia. Tunnetuimmat niistä ovat WordPress, Bootstrap, Joomla ja Drupal. Tässä työssä ei katsota, miten erilaiset CMS-järjestelmät eroavat toisistaan. Järjestelmiä kuitenkin vertaillaan, mitä vahvuuksia ja heikkouksia niillä on käsin kirjoitettuun koodiin verrattuna.

3.2 Vahvuudet

CMS-järjestelmien vahvuudet ovat:

- Verkkosivuston sisällön muokkaus tehdään graafisesti. Ei vaadi koodin ymmärrystä.
- Siinä on koodatut tyylitiedostot (sivun standardit).
- Useimmillaan tyylitiedostoissa on toteutettu responsiivisuus (mm. mobiililaitteille).
- Nopeuttaa verkkosivun toteutusta, koska sisältää erilaiset valmiiksi toteutetut moduulit ja lomakkeet.
- On mahdollisuus käyttää valmiiksi koodattuja teemoja ja lisäosia.

- Ongelmatilanteissa aina löytyy vastaus internetistä suuren käyttäjämäärän takia.

3.3 Heikkoudet

Seuraavana on CMS-järjestelmien heikkouksia käsin kirjoitettuihin verkkosivustoihin verrattuna.

- Sivua ei ole eksklusiivinen eli ainutlaatuinen.
- Sivua on vaikeampi adoptoida omille tarpeille.
- Siinä on heikompi sivun optimointi.
- Verkkosivu luo epäluottamuksellisen kuvan ammattilaisilla vierailta.
- Tekijän tietämättömyys sivun rakenteesta ja muista toiminnoista.
- Huonompi verkkosivun turvallisuus.

3.4 Yhteenveto

Nyt, kun on saatu selville, miten CMS-järjestelmällä tehty verkkosivu eroaa käsin kirjoitetusta verkkosivusta, voidaan päätellä, mikä olisi sopivampi ja parempi vaihtoehto tähän työhön. Olen tehnyt yksinkertaisen taulukon, joka voisi helpottaa loppupäätöstä siitä, tehdäänkö sivua käsin vai ei.

Taulukko 1. Vertailu: CMS-järjestelmän käyttö verrattuna käsin kirjoitettuun koodiin.

Ominaisuus	CMS	Käsin kirjoitettu
Nopeampi toteutus	X	
Toteutuksen helppous	X	
WWW-sivun ainutlaatuisuus		X

WWW-sivun turvallisuus		X
Opettavuus		X

Koska suunnittelutoimiston pääpalveluna on verkkosivujen suunnittelu ja toteutus, tärkeintä on ymmärtää koodia ja osata tehdä internetsivuja asiakkaan tarpeiden mukaan. Tämän takia tässä työssä päätetään toteuttaa kaikkea alusta lähtien käsin, ilman mitään CMS-järjestelmää, jolloin on mahdollisuus oppia uutta.

4 Suunnittelu

4.1 Johdanto

Suunnittelun ensimmäisenä vaiheena olisi hyvä määritellä kotisivun tyyppi. Tässä on kaksi vaihtoehtoa: joko tehdään yksisivuinen nettisivu eli landing page tai monisivuinen nettisivu. Mikäli sivulle ei tule liian paljon erilaista sisältöä ja verkkosivu halutaan tehdä mahdollisimman yksinkertaiseksi, landing page on oikea valinta. Alkuvaiheessa aloin toteuttaa juuri sitä, koska ensimmäisessä suunnitteluvaiheessa oli vähemmän ideoita ja tietoa, mitä sisältöä sivulle tulee. Mutta lopuksi, kun yksisivuinen malli oli valmis, ymmärsin, että se ei ole paras vaihtoehto ja parempi olisi tehdä monisivuinen yrityssivu, jossa olisi enemmän sisältöä ja joka olisi vähän mielenkiintoisempi.

Varsinaista graafista suunnittelua on mahdollista tehdä eri tavoilla: joko käyttämällä kuvankäsittelyohjelmaa (Photoshop, Illustrator), esitysgrafiikkaohjelmaa (PowerPoint), prototyypiohjelmaa tai pelkkä kynää ja paperia. Tässä opinnäytetyössä ei katsota, miten työskennellään jossain kuvankäsittelyohjelmassa. Työn tärkeimmät kohdat käydään läpi luvussa "Toteutus", missä kiinnitetään huomiota itse koodiin ja esimerkkeihin.

4.2 Verkkosivujen määrittely

Kun verkkosivun tyyppi on saatu selville, ensimmäiseksi tulee määrittää, mitä sivuja toteutetaan. Tärkein elementti on aloitussivu, josta käyttäjä lähtee liikkeelle. Jollain sivulla on myös hyvä kertoa yrityksestä ja sen taustasta, mutta koska tällä hetkellä mitään yritystä ei ole vielä olemassa, tämä osa jätetään pois ja lisätään vasta silloin, kun yritys alkaa toimintansa.

Tietenkin palveluiden tai tuotteiden kuvaaminen on välttämätöntä. Tätä varten tehdään oma sivu, jossa luetellaan yrityksen tarjoamat palvelut.

Mitä muuta tärkeätä tulisi olla suunnittelutoimiston sivulla? Varmasti, mikäli asiakas päätyy sivulle, taustan jälkeen häntä kiinnostaisi, mitä asioita yritys on jo tehnyt. Tällaista sivua kutsutaan portfolioiksi, ja se voi olla juuri se sivu, joka auttaa asiakkaan tekemään päätöksen siitä, tilataanko palveluita vai ei.

Jos asiakkaalla on esimerkiksi kysymyksiä, hänen on löydettävä yrityksen yhteystiedot, joista näkyisi puhelinnumero, sähköpostiosoite, mahdollinen käyntiosoite ja yhteydenottolomake.

Hyvä myynti vaatii mahdollisimman helppoa tuotteiden ostamista tai palveluiden tilaamista. Tämän takia toteutan tilauslomakkeen, josta käyttäjä voisi tilata palveluita ”yhden napin painalluksella”.

Yhteenvedona on se, että tässä projektissa olevat sivut ovat

- pääsivu (aloitussivu)
- yrityksen palvelut
- portfolio
- yhteystiedot
- projektin tilauslomake.

4.3 Yritysilme

Ennen kuin aloitetaan graafisesti suunnitella verkkosivuja, olisi hyvä määritellä yritysilme. Ilmesuunnitteluun sisältyvät:

- logon ja tunnuksen suunnittelu
- graafiset elementit
- varimaailman, kuvamaailman, muotokielen ja typografian määrittely
- graafinen ohjeistus.

Jos voitaisiin lähteä suunnittelemaan ja toteuttamaan logoa, pitäisi tietää tulevan yrityksen nimi. Sen on oltava helposti luettava ja yksinkertainen. Itse olen kiinnittänyt

huomiota vielä siihen, että yrityksen nimen tulisi olla kansainvälisessä muodossa, joten yritystoiminnan alue ei olisi rajoitettu pelkästään Suomen alueelle. Olen päättänyt antaa yritykselle nimen "K&K Design" tai "Suunnittelutoimisto K&K". Eli logolle käytetään lyhennettä "K&K" (kuva 1).

Tässä työssä ei käydä läpi, miten toimitaan kuvankäsittelyohjelman kanssa, koska kiinnitetään huomiota verkkosivujen toteuttamiseen eli koodiin. Tietenkään ei ole pakko osata Photoshopin, Illustratorin tai muun vastaavan ohjelman käyttöä, koska tarvittaessa on mahdollista tilata logon suunnittelu ammattilaisilta.

Logosuunnittelun yhteydessä on myös hyvä samalla miettiä tulevaa värimaailmaa. Yleensä verkkosivulla on 2-3 pääväriä. Kuitenkin on parempi miettiä eteenpäin ja määrittellä vähintään 5 väriä, joita joudutaan tarvittaessa käyttämään erilaisissa elementeissä.



Kuva 1. Yrityksen logo

Kuvassa 1 näkyy valmis logo. Olen määrittänyt yritykselle myös värimaailman. Päävärit, joiden kanssa työskennellään, ovat

- #14171C ■

- #DF0859 ■

Lopuksi valitaan fontit. Suosittelen käyttämään vähintään kahta erilaista. Olen valinnut tähän projektiin "Sans-Serif" ja "Segoe UI".

Koska esimerkiksi "Segoe UI" ei ole tyypillinen HTML-fontti, vaan se on Microsoftin tekemä, sitä tulee ostaa ja liittää itsenäisesti css-tyylitiedostoon, jossa se olisi käytettävissä. Se tehdään näin:

```
@font-face {
  font-family: "Segoe UI";
  src: url("../fonts/SegoeUIRegular/SegoeUIRegular.eot");
  src: url("../fonts/SegoeUIRegular/SegoeUIRegular.eot?#iefix") format("embedded-open type"),
       url("../fonts/SegoeUIRegular/SegoeUIRegular.woff") format("woff"),
       url("../fonts/SegoeUIRegular/SegoeUIRegular.ttf") format("truetype");
}
```

Kuva 2. Erikoistekstin lisääminen. Tyylitiedoston koodiesimerkki.

Kuva 2 näyttää, miten lisätään erikoisfontti tyylitiedostoon.

- "font-family" – määrittelee fontin nimen, johon viitataan silloin, kun halutaan käyttää kyseistä tekstiä.
- "src" - sisältää osoitteet eli URL:t, joista tekstiä ladataan.

Nyt kun kaikki tässä vaiheessa tarvittavat yritysilmmeen kohteet on määritelty, voidaan alkaa suunnittelemaan verkkosivujen ulkoasua eli siirrytään graafiseen suunnitteluun.

4.4 "ETUSIVU"

Niin kuin on sanottu aikaisemmin, tässä opinnäytetyössä ei lähdetä katsomaan, miten kuvankäsittelyohjelmassa tehdään prototyyppiä tai piirretään logoja yms. Tämän työn tarkoituksena on syventää enemmän toteutukseen eli koodiin ja sen toimivuuteen, joten suunnitteluvaiheessa mietitään jokaisen sivun rakennetta.

Olen päättänyt tehdä aloitussivun, joka olisi ainutlaatuinen ja mielenkiintoinen. Ideana on integroida kauniita videoita sivun taustalle, ja keskelle näyttöä tulisi yrityksen logo.

Sivun yläosan tulee navigointipalkki, jossa olisivat seuraavat linkit:

- "ETUSIVU"
- "PALVELUMME"
- "PORTFOLIO"
- "YHTEYSTIEDOT"
- "TILAA PROJEKTI".

Ja navigoinnin oikealla puolella olisi kielenvaihtolaatikko.

Tässä on etusivun suunnitelma. Vaikka se on aika yksinkertainen, tuloksena saadaan ainutlaatuinen aloitussivu, joka voisi herättää kiinnostusta muilta käyttäjiltä.

4.5 "PALVELUMME"

Tämän sivun tarkoituksena on kuvata yrityksen mahdolliset palvelut.

Erilaiset yritykset keksivät oman ratkaisun tämän verkkosivun toteuttamiseen. On mahdollista kuvata palveluita joko yhdellä sivulla tai vaikka tehdä oma sivu jokaista palvelua varten. Viimeinen vaihtoehto on hyvä silloin, kun löytyy paljon myyvää tekstiä. Meidän tapauksessa ei kiinnitetä huomiota tekstiin. Sen takia laitetaan kaikki samalle sivulle.

Käytetään etusivun pohjaa ja muokataan sitä tarpeiden mukaan, eli logon sijaan laitetaan palveluiden kuvakkeet.



Kuva 3. Palvelu 1: Verkkosivujen suunnittelu



Kuva 4. Palvelu 2: Markkinointi



Kuva 5. Palvelu 3: Yritysilmeen kehitys



Kuva 6. Palvelu 4: Valokuvat ja painotuotteet



Kuva 7. Palvelu 5: Ohjelmistokehitys



Kuva 8. Palvelu 6: Mobiilisovellukset

Kuvakkeet (Kuvat 3-8) olen tehnyt itse, jotta ne noudattaisivat täydellisesti ulkoasua ja yritysilmettä. Jokaisen kuvakkeen alle kirjoitetaan kyseisen palvelun nimi. On myös hyvä laatia myyvää tekstiä ja kertoa asiakkaalle mahdollisimman tarkasti jokaisesta palvelusta. Tämä osa jätetään tekstittäjille ja siirrytään eteenpäin suunnittelemaan portfolioa.

4.6 "PORTFOLIO"

Portfolion tarkoituksena on näyttää asiakkaalle, mitä asioita yritys on tehnyt. Aikaisemmin olen tehnyt verkkosivuja ja logoja, joten haluaisin näyttää näitä tällä sivulla. Mutta minkälainen rakenne portfolioille olisi sopiva?



Kuva 9. Portfolio. Esimerkki.

Kuvasta 9 näkyy ainutlaatuinen ratkaisu, jonka olen keksinyt. Eli ideana on se, että tehdään SVG-kuva, jonka keskellä olisi asiakkaan logo ja logon alla olisi linkki sivulle, jos asiakas on tilannut verkkosivun suunnittelun. Mikäli kyseessä oli logon suunnittelu, niin alle tulee teksti: "Logon suunnittelu".

Sivun alle lisätään alatunnistetta, johon tulevat puhelinnumero ja sähköpostiosoite. Tulevaisuudessa tähän lisätään myös käyntiosoite ja y-tunnus.

4.7 "YHTEYSTIEDOT"

Yhteystiedot-sivulle liitetään yhteydenottolomake, jotta asiakas voisi pyytää ottamaan yhteyttä. Lomakkeen kentät ovat:

- nimi
- sähköposti
- puhelin
- yritys
- viesti.

Lomakkeen alle sijoitetaan Google-kartta, johon merkataan toimiston osoite. Ei kirjoiteta puhelinnumeroa ja sähköpostiosoitetta, koska se löytyy alatunnisteesta.

4.8 "TILAA PROJEKTI"

Tälle sivulle toteutetaan projektin tilauslomake, jotta asiakas voisi helposti ja nopeasti jättää tarjouspyynnön mahdollisten toiveiden kanssa. Lomakkeen kentät ovat

- nimi
- sähköposti
- puhelin
- yritys
- projektin tyyppi
- asiakkaan/asiakasyrityksen tausta
- projektille toivottu määräaika
- arvioitu budjetti.

Alatunniste on sama kuin "PORTFOLIO"- ja "YHTEYSTIEDOT"-sivuilla.

5 Toteutus

5.1 Johdanto

Suunnitteluvaiheen jälkeen on aika siirtyä varsinaiseen vaiheeseen eli toteutukseen. Mutta ennen koodin kirjoittamista pitää rekisteröidä domain eli verkkotunnus, sekä vuokrata webhotelli eli palvelin, jolla verkkosivu pyörisi. Näitä voi kätevästi hoitaa miltä tahansa palveluntarjoalta/verkkotunnusvälittäjältä.

Sen jälkeen, kun verkkotunnus on rekisteröity ja pääsy palvelimelle FTP-ohjelman kautta on hoidettu, voidaan aloittaa toteuttamaan suunnitelmia. Tyypillinen sivun rakenne näkyy kuvasta 10.

```
<html>
<head>
<!--
  title-elementti
  meta-elemntit
  link-elemntit
  skriptit
-->
</head>
<body>
<!--
  Verkkosivun sisältö
-->
</body>
</html>
```

Kuva 10. Verkkosivun perusrakenne

Seuraavissa luvuissa käydään läpi vähän tarkemmin jokaista elementtiä ja sen sisältöä.

5.2 Doctype ja html-elementit

Ihan ensimmäisenä pitää kertoa selaimelle mahdollisimman tarkasti avattavasta verkkosivusta. Ensimmäiselle riville on hyvä laittaa ohjetta avattavasta tiedostosta ja sen HTML-versiosta. Ja se tehdään näin:

```
<!DOCTYPE html>
```

mikäli kyseessä on HTML 5:n versio, tai esimerkiksi:

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

mikäli kyseessä on HTML 4.01. [3]

Tämän jälkeen voidaan alkaa kirjoittamaan html-koodia. Ensimmäiseksi tulee aina html-tagi, joka on tiedoston alkuperä ja joka kertoo selaimelle, että tästä alkaa HTML-tiedosto. Attribuutiksi tulee "lang", joka kertoo, millä kielellä verkkosivun sisältö on kirjoitettu. Koska meidän sivu on kirjoitettu suomeksi, tunnisteeseen tulee

```
<html lang="fi">
```

html-tunnisteen jälkeen tulee tiedoston tärkein tekninen osa - head.

Tyypillinen ja yksinkertainen sivuston rakenne näyttää seuraavalta:

5.3 Head-elementti

Head-tagin sisällä ovat yleensä seuraavat asiat:

- <title> - sivun otsikko
- <meta> - meta-tiedot
- <link> - viittaukset tyylitiedostoihin
- <script> - skriptit.

Otsikko, joka näytetään selaimen yläpalkissa ja hakukoneen tuloksissa, määritellään title-tunnisteen sisällä.

```
<title> Suunnittelutoimisto K&K </title>
```

Meta-tiedot sisältävät tietoa itse dokumentista ja sen datasta. Yleisimmät meta-tiedot, joita käytetään lähes kaikilla sivustoilla, ovat

```
<meta charset="UTF-8">
```

Uudessa HTML5:ssa löytyy uusi attribuutti, jonka nimi on 'charset'. Tämän avulla määritellään tiedoston koodaustapa.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0" />
```

Jotta verkkosivu osaisi paremmin mukautua erilaisiin näyttölaitteisiin, tulee lisätä "viewport"-niminen meta-taggi. Esimerkiksi "width=device-width" asettaa sivun leveydelle käytettävän laitteen leveyden. 'initial-scale' asettaa oletuszoomaustason ja "maximum-scale" maksimizoomaustason. "user-scalable" attribuutti joko sallii tai estää käyttäjän zoomauksen.

```
<meta name="author" content="Kirill Shamin" />
```

"author"-niminen metatieto asettaa verkkosivun tekijän.

```
<meta name="description" content="Suunnittelutoimisto K&K tarjoaa palvelut verkkosivustojen suunnitteluun ja toteutukseen, niiden mainontaan sekä liiketoiminnan visuaalisen imagon luomiseen." />
```

```
<meta name="keywords" content="suunnittelutoimisto, yrityskotisivut, kotisivut, kotisivujen suunnittelu, yritysilme, logon suunnittelu, verkkokaupan suunnittelu, hakukoneoptimointi" />
```

Hakukoneoptimointia varten kannattaa lisätä "description"- sekä "keywords"-metatunnisteet.

“*description*” asettaa verkkosivun kuvauksen, joka näkyy esimerkiksi Googlen hakutuloksissa. Kuvauksen pitää parhaiten kertoa verkkosivusta ja sen sisällöstä. Sen on oltava mahdollisimman selkeä ja tehokas markkinoinnin näkökulmasta.

“*keywords*” asettaa verkkosivun avainsanoja. Tähän ei missään nimessä saa syöttää liikaa avainsanoja, koska se vaan heikentää hakukoneoptimointia. Hyvä tapa on kirjoittaa tämän tunnisteeseen sisälle alle 10 tärkeintä sanaa.

```
<meta name="robots" content="index, follow" />
```

```
<meta name="revisit-after" content="3 days" />
```

“*robots*”-tagi ohjaa hakukonerobottia. Tämän meta-tagin avulla määritellään, halutaanko, että sivu indeksoidaan, tai seurataanko sillä olevia linkkejä. Hakukoneilla on omat indeksit eli hakemistot, joissa on tiedot kaikista hakukoneella löytyvistä sivuista. Indeksointi eli hakemiston luominen ja täydentäminen tapahtuu siten, että hakukoneet seuraavat sivuilta löytyviä linkkejä, kulkevat niitä seuraten ympäri internetiä ja tallentavat löytyvät sivut välimuistiinsa. Kun joku tekee haun hakukoneessa, kone hakee valtavasta välimuististaan hakua vastaavat sivut, ja asettaa ne hakutuloksiin vaikuttavien kriteeriensä mukaiseen paremmuusjärjestykseen hakutulossivulla. [5]

“*revisit-after*” asettaa, kuinka usein hakurobotin tulee vierailla tällä sivulla.

Meta-tunnisteiden jälkeen sijoitetaan link-tagit, joilla määritellään tiedoston suhdetta muihin tiedostoihin. Esimerkiksi juuri link-elementin avulla viitataan ulkoiseen tyylitiedostoon.

```
<link rel="stylesheet" type="text/css" href="//kk-design.fi/style/style.css" />
```

“*rel*” määrittää linkkityyppiä, “*type*” sisältötyyppiä ja “*href*” sijaintia.

Jokaisella verkkosivulla on oma ikoni eli pieni logo, joka näkyy esimerkiksi selaimen osoiterivillä. Tämä kuvake liitetään käyttämällä link-tunnistetta:

```
<link rel="icon" type="image/png" href="//kk-design.fi/images/icon/favicon-32x32.png" sizes="32x32" />
```

```
<link rel="icon" type="image/png" href="//kk-design.fi/images/icon/favicon-16x16.png" sizes="16x16" />
```

Lopussa head-elementtiin liitetään mahdolliset jaskriptit. Esimerkiksi ”jQuery”:a lisätään näin:

```
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js" />
```

Yhteenvedona tämän projektin jokaisen sivun tulee sisältää koodia, joka näkyy kuvassa 11.

```
<!DOCTYPE html>
<html lang="fi">
<head>
  <title> Suunnittelutoimisto K&amp;K </title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0" />
  <meta name="author" content="Kirill Shamin" />
  <meta name="description" content="Suunnittelutoimisto K&K tarjoaa palvelut verkkosivustojen suunnitteluun ja toteutukseen, niiden mainontaan sekä liiketoiminnan visuaalisen imagon luomiseen." />
  <meta name="keywords" content="suunnittelutoimisto, yrityssivut, kotisivujen suunnittelu, yritysilm, logon suunnittelu, verkkokaupan suunnittelu, hakukoneoptimointi" />
  <meta name="robots" content="index, follow" />
  <meta name="revisit-after" content="3 days" />
  <link rel="stylesheet" type="text/css" href="//kk-design.fi/style/style.css" />
  <link rel="icon" type="image/png" href="//kk-design.fi/images/icon/favicon-32x32.png" sizes="32x32" />
  <link rel="icon" type="image/png" href="//kk-design.fi/images/icon/favicon-16x16.png" sizes="16x16" />
  <script src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js" />
</head>
<body>
  <!--
    Navigointipalkki
    Verkkosivun sisältö
  -->
</body>
</html>
```

Kuva 11. HTML-tiedosto. Pohja.

5.4 Robots.txt

Vaikka hakukoneroboteille voidaan antaa ohjeita meta-elementtien sisällä, yleensä tätä kuitenkin tehdään omassa ”robots.txt”-nimisessä tiedostossa, jonka tulee olla verkkosivun juuri-kansiossa. Kuvassa 12 on tiedoston esimerkki.

```
User-agent: *  
Disallow: /cgi-bin  
Sitemap: http://kk-design.fi/sitemap.xml
```

Kuva 12. robots.txt

'*User-agent*' on botin nimi, jolle kohdistetaan käskyjä. Mikäli halutaan kohdistaa käskyt kaikille mahdollisille boteille, tulee käyttää tähtimerkkiä.

'*Disallow*' kertoo botille, mitä tiedostoa tai hakemistoa ei saa indeksoida.

'*Sitemap*' kertoo, mistä löytyy sivukartta.

5.5 Sitemap.xml

Hyvä hakukoneoptimointi vaatii sivukartan, joka sisältää polut sivuille, joita tulee indeksoida. Lisäksi jokaiselle sivulle voidaan antaa seuraavia vaihtoehtoisia lisätietoja:

- Verkkosivun tärkeys (asteikko 0.0 – 1.0 eli mitä suurempi, sitä tärkeämpi)

```
<priority>arvo</priority>
```

- Kuinka usein sivua päivitetään (mm. 'weekly', 'monthly', 'always')

```
<changefreq>arvo</changefreq>
```

- Milloin sivu oli viimeksi päivitetty ('vvvv-kk-pp')

```
<lastmod>2016-09-27</lastmod>
```

Tämän projektin sivukartta näkyy kuvassa 13.

```

<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>http://kk-design.fi/</loc>
    <priority>1</priority>
  </url>
  <url>
    <loc>http://kk-design.fi/services.php</loc>
    <priority>0.9</priority>
  </url>
  <url>
    <loc>http://kk-design.fi/portfolio.php</loc>
    <priority>0.8</priority>
    <changefreq>weekly</changefreq>
  </url>
  <url>
    <loc>http://kk-design.fi/contacts.php</loc>
  </url>
  <url>
    <loc>http://kk-design.fi/projectform.php</loc>
  </url>
</urlset>

```

Kuva 13. Sivukartta. XML-tiedosto.

5.6 Navigointi

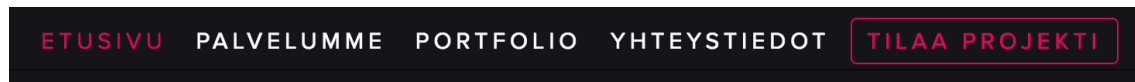
Tärkeä elementti, joka löytyy varmasti jokaiselta sivulta, on navigointipalkki. Olen suunnitellut ja piirtänyt kuvankäsittelyohjelmassa kuvassa 14 olevan mallin, joka tulee toteuttaa.



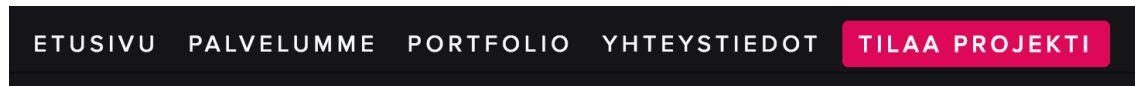
Kuva 14. Navigointipalkki. Näkymä. Esimerkki 1.

Ideana on staattinen navigointi, joka olisi näkyvä sivun yläosassa. Taustaväri on läpinäkyvä (kaikkien verkkosivujen taustaväri on #14171C ■).

Silloin, kun hiiri menee linkin päälle, tekstin väri muuttuu (kuvat 15 ja 16).



Kuva 15. Navigointipalkki. Näkymä. Esimerkki 2.



Kuva 16. Navigointipalkki. Näkymä. Esimerkki 3.

Toteutuksen ensimmäisenä vaiheena tulee aina rakentaa fyysisesti suunniteltu elementti ja vasta sen jälkeen aloittaa muokkaamaan sen ulkoasua.

Jokainen navigoinnin linkki on yksi elementti eli yhteensä saadaan 5 elementtiä, joita tulee koota yhteen, joten lopussa ne kaikki olisivat yhdessä nav-elementissä. Nav-elementti eli `<nav>` on HTML5:n uusi tunniste, jota käytetään sivun navigaation merkitsemiseen. Linkit muodostetaan käyttämällä a-tagia (`teksti`).

```
<nav>
  <a href="//kk-design.fi/index.php">ETUSIVU</a>
  <a href="//kk-design.fi/services.php">PALVELUMME</a>
  <a href="//kk-design.fi/portfolio.php">PORTFOLIO</a>
  <a href="//kk-design.fi/contacts.php">YHTEYSTIEDOT</a>
  <a href="//kk-design.fi/projectform.php">TILAA PROJEKTI</a>
</nav>
```

Kuva 17. Navigointipalkki. Koodiesimerkki 1.

Koska navigointi-elementille ei ole tehty vielä yhtään tyylisääntöä, se näyttää aika tylsältä (kuva 18).

[ETUSIVU](#) [PALVELUMME](#) [PORTFOLIO](#) [YHTEYSTIEDOT](#) [TILAA PROJEKTI](#)

Kuva 18. Navigointipalkki. Näkymä ilman tyylisääntöjä.

Jotta olisi helpompi muokata elementtien ulkoasua, määritellään niille oma luokka. Jokaiselle luokalle luodaan sääntöjä css-tiedostossa. Aloitetaan siitä, että määritellään jokaiselle linkille luokka "menu_link" ja nav-elementille "menu". Koska "TILAA

PROJEKTI"-linkille tehdään vähän erikoinen ulkoasu (kuva 16), määritellään sille myös toinen luokka – "project_link". Navigointipalkin lopullinen koodi näkyy kuvassa 19.

```
<nav class="menu">
  <a class="menu_link" href="//kk-design.fi/index.php">ETUSIVU</a>
  <a class="menu_link" href="//kk-design.fi/services.php">PALVELUMME</a>
  <a class="menu_link" href="//kk-design.fi/portfolio.php">PORTFOLIO</a>
  <a class="menu_link" href="//kk-design.fi/contacts.php">YHTEYSTIEDOT</a>
  <a class="menu_link project_link" href="//kk-design.fi/projectform.php">TILAA PROJEKTI</a>
</nav>
```

Kuva 19. Navigointipalkki. Koodiesimerkki 2.

Ominaisuudet, jotka lisätään "menu_link"-luokalle (kuva 20):

- "text-decoration" – tämän avulla voidaan poistaa esimerkiksi linkin alleviivaus.
- "color" – muuttaa tekstin väriä.
- "padding" – lisää täytealuetta (sisällön ja reunaviivan välinen tyhjätila).
- "letter-spacing" – asettaa sanoissa olevien merkkien välin.

```
a.menu_link {
  text-decoration: none;
  color: #FFF;
  padding-right: 6px;
  letter-spacing: 0.2em;
}
```

Kuva 20. Navigointilinkeille tehdyt tyylisäännöt.

Lisäksi toteutetaan värin muuttaminen silloin, kun hiiri on linkin päällä. Tätä varten tehdään oma sääntö, jonka alkuosa on tarvittava luokka ja loppuosa on "hover"-valitsin (kuva 21).

```
a.menu_link:hover {
  color: #DF0F5A;
}
```

Kuva 21. Navigointilinkeille tehty tyylisääntö.

”TILAA PROJEKTI” -linkin ulkoasu muutetaan hieman erilaiseksi. Väriin ja täytealueen lisäksi tulevat seuraavat ominaisuudet (kuva 22):

- ”border” – asettaa elementille rajan (parametrit: leveys, ulkoasu, väri).
- ”border-radius” – pyöristää rajan kulmat.
- ”font-weight” – määrittää tekstin lihavoitua (mm. ”normal”, ”bold” tai käyttämällä asteikkoa 100 – 900).

```
a.project_link {
  color: #DF0859;
  border: 1px solid #DF0859;
  padding: 5px 10px;
  border-radius: 4px;
  font-weight: 600;
}
```

Kuva 22. Projektin tilauslinkin tyylisäännöt.

Nav-elementti tulee alentaa sivun yläosasta lisäämällä sille täytealuetta. Lisäksi käytetään seuraavia ominaisuuksia:

- ”text-align” – tekstin vaakasuora tasaus.
- ”font-family” – asettaa kirjasintyyppin.

```
nav.menu {  
  text-align: center;  
  padding: 1.2% auto;  
  font-family: ProximaNovaRegular;  
}
```

Kuva 23. Nav-elementille tehdyt tyylisäännöt.

Kuva 23 näyttää "menu"-luokan tyylisäännöt.

5.7 "ETUSIVU"

Aloitussivun esimerkki näkyy kuvassa 24. Eli etusivu oli suunniteltu niin, että keskelle tulisi logo ja taustalla olisi video.



Kuva 24. Etusivu. Esimerkki.

Videon integrointi onnistuu käyttämällä HTML5:n uutta tunnistetta – video (kuva 25).
Attribuuteiksi annetaan:

- "autoplay" – käynnistää videon automaattisesti, kun sivu on ladattu.

- "loop" – toistaa videota jatkuvasti.

```
<video autoplay loop class="video">  
  <source src="//kk-design.fi/media/video.mp4" type="video/mp4">  
  <source src="//kk-design.fi/media/video.webm" type="video/webm">  
</video>
```

Kuva 25. Videon integrointi. Koodiesimerkki.

Video-tagin sisällä ovat linkit videon sijaintiin. Varmuuden vuoksi kannattaa lisätä kaksi eri tiedostotyyppiä (".mp4", ".webm" tai ".ogg"), mikäli selain tai laite ei tue yhden niistä.

Jotta video olisi läpinäkyvä, annetaan "video"-luokalle tyylisääntö "opacity: 0.4". Lisäksi määritetään kiinteä sijainti: "position: absolute".

- "opacity" – määrittää elementin läpikuultavuutta.
- "position" – määrittää elementin asettelumallia.

Videon jälkeen lisätään logo. SVG-tiedosto voidaan lisätä käyttämällä embed-tunnistetta (kuva 26).

```
<embed src="//kk-design.fi/images/logo.svg" class="logosvg" alt="logo">
```

Kuva 26. Logon integrointi. Koodiesimerkki.

Logon kokoa voidaan hallita "width"-ominaisuudella.

- "width" – määrittää elementin leveyden. (esim. "width: 50%" puolittaa kuvan koon).

Etusivun lopputulos on kuvassa 27.



Kuva 27. Valmis etusivu.

5.8 "PALVELUMME"

Tämä sivu toteutetaan melkein samalla tavalla kuin etusivu. Eli sivun ylhäällä on navigointipalkki ja sivun taustalla on integroitu video, mutta logon sijaan lisätään palveluiden kuvakkeet. Sivun rakenne kuvataan kuvalla 28.



Kuva 28. 'PALVELUMME'-sivu. Elementtien rakenne.

Yhteensä on kuusi kuvaketta (img-elementtiä) ja jokaisen kuvakkeen alla on teksti (span-elementti). Eli tehdään kaksi riviä (div-elementtiä), niin jokaiselle riville tulee 3 kuvaketta. Lopuksi nämä rivit laitetaan yhteen div-elementtiin. Tämän osan toteutus näkyy kuvassa 29.

```

<div>
  <div class="linecontainer">
    <div class="iconcontainer">
      
      <span class="servicetext">YRITYSSIVUJEN SUUNNITTELU</span>
    </div>

    <div class="iconcontainer">
      
      <span class="servicetext">MARKKINOINTI</span>
    </div>

    <div class="iconcontainer">
      
      <span class="servicetext">YRITYSILMEEN KEHITYS</span>
    </div>
  </div>

  <div class="linecontainer">
    <div class="iconcontainer">
      
      <span class="servicetext">PAINOTUOTTEET JA VALOKUVAT</span>
    </div>

    <div class="iconcontainer">
      
      <span class="servicetext">OHJELMISTOKEHITYS</span>
    </div>

    <div class="iconcontainer">
      
      <span class="servicetext">MOBIILISOVELLUKSET</span>
    </div>
  </div>
</div>

```

Kuva 29. 'PALVELUMME'-sivu. Koodiesimerkki.

Jotta kuvakkeet olisivat yhdellä rivillä, "iconcontainer"-luokalle tulee antaa tyylisääntö "display: inline-table".

- "display" – määrittelee, miten elementti näytetään. Jotta lohkot olisivat vierekkäin, parametriksi annetaan "inline-table".

Tekstin koko muutetaan käyttämällä "font-size"-ominaisuutta.

5.9 "PORTFOLIO"

Kuvasta 9 näkyy suunniteltu sivumalli. Sen toteutus löytyy kuvasta 30.

```

<div class="portfolio whitebg">
  <object class="portfoliosvg" data="//kk-design.ru/images/bellaportfolio.svg" type="image/svg+xml" alt="bella logo"></object>
  <a class="openbutton" href="//bella-trade.ru" target="_blank">AVAA VERKKOSIVUSTO</a>
</div>
<div class="portfolio vapellabg">
  <object class="portfoliosvg" data="//kk-design.ru/images/vapellaportfolio.svg" type="image/svg+xml" alt="vapella logo"></object>
  <a class="openbutton" href="//vapella.com" target="_blank">AVAA VERKKOSIVUSTO</a>
</div>

```

Kuva 30. Portfolio. Koodiesimerkki.

SVG-kuva voidaan liittää käyttämällä esimerkiksi object-tunnistetta. Sen attribuutit ovat:

- "data"-objektin URL-osoite.
- "type" – tiedoston tyyppi.

Objektin alle lisätään linkki. Tyylisääntöjen avulla yksinkertaisesta linkistä voidaan tehdä mukavannäköinen nappi (kuva 31).



Kuva 31. Nappi. Esimerkki.

```

a.openbutton {
  width:300px;
  border:1px solid #F0823A;
  font-size:.6em;
  padding:10px;
  background-color:#fff;
  color:#F0823A
}

```

Kuva 32. Nappi. Koodiesimerkki.

Kuvalla 32 ovat kaikki tyylisäännöt, joiden avulla voidaan tehdä oletuslinkistä mukava nappi.

5.10 "YHTEYSTIEDOT"

Varmasti jokaiselta sivulta, josta löytyy yhteystiedot, löytyy myöskin yhteydenottolomake, jonka avulla asiakas voi jättää yhteydenottopyynnön. Katsotaan, mistä yksi kenttä (kuva 33) koostuu.



Kuva 33. Kenttä: Nimi. Esimerkki.

Ylhäällä oleva esimerkki koostuu kolmesta elementistä:

- `<section>` - määrittelee osan.
- `<i>` - tämän tagin kanssa voidaan liittää Bootstrapin ikoneita.
- `<input>` - lomake-elementti.

```
<section>
  <i class="fa fa-append fa-user"></i>
  <input type="text" name="name" placeholder="Nimi*" required />
</section>
```

Kuva 34. Tekstikentän toteutus.

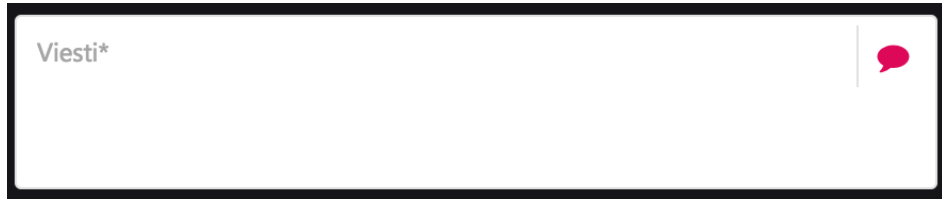
Kuvassa 34 on tekstikentän koodiesimerkki.

Input-tagin attribuutit ovat:

- "type" – lomake-elementin tyyppi (mm. tekstikenttä, valintaruutu tai valintanappi)
- "name" – nimi lomakekäsittelyä varten.
- "placeholder" – syöttökentässä oleva oletusteksti

- "required" – määrittelee, onko kentän täyttäminen pakollista vai ei.

Suuremman tekstikentän (kuva 35) voidaan tehdä käyttämällä <textarea>-tagia <input>:n sijaan (kuva 36).



Kuva 35. Viesti-kenttä. Esimerkki.

```
<section>
  <i class="fa fa-append fa-comment"></i>
  <textarea name="message" id="message" rows="4" placeholder="Viesti*" required></textarea>
</section>
```

Kuva 36. Viesti-kentän toteutus.

- "rows"-attribuutti määrittelee tekstikentän rivien lukumäärän.

Kenttien ulkoasua muokataan jo tässä työssä kerrotuilla tyylisäännöillä, kuten esimerkiksi: "width", "height", "padding", "border", "border-radius" ja "color".

Elementin taustaväriä muutetaan käyttämällä "background-color"-ominaisuutta.

Lomakkeen runko, jonka sisälle tulevat lomake-elementit, näkyy kuvassa 37.

```
<form action="send.php" method="post">
  <!--
    Lomake-elementit
  -->
</form>
```

Kuva 37. Lomakkeen runko. Koodiesimerkki.

- "action"-attribuutti määrittelee, minne lomakkeen tiedot lähetetään käsittelyä varten.

- "method"-attribuutti määrittelee lomakkeen lähetystavan (GET tai POST).

Lähetysnappia toteutetaan niin, että esimerkiksi <button> tai <input> -tunnisteen sisälle kirjoitetaan 'type="submit"'.

Lomakkeen lopullinen koodiesimerkki on kuvassa 38.

```
<div class="contact">
  <form action="send.php" method="post">
    <fieldset>
      <div class="row">
        <section>
          <i class="fa fa-append fa-user"></i>
          <input type="text" name="name" id="name" placeholder="Nimi*" required>
        </section>
        <section>
          <i class="fa fa-append fa-at"></i>
          <input type="email" name="email" id="email" placeholder="Sähköposti*" required>
        </section>
        <section>
          <i class="fa fa-append fa-phone"></i>
          <input type="text" name="phone" id="phone" placeholder="Puhelin">
        </section>
        <section>
          <i class="fa fa-append fa-suitcase"></i>
          <input type="text" name="company" id="company" placeholder="Yritys">
        </section>
        <section>
          <i class="fa fa-append fa-comment"></i>
          <textarea name="message" id="message" rows="4" placeholder="Viesti*" required></textarea>
        </section>
        <button type="submit" class="button"></button>
      </div>
    </fieldset>
  </form>
</div>
```

Kuva 38. Yhteydenottolomake. Koodiesimerkki.

- <fieldset> - ryhmittelee syöttökenttiä.

Lomakkeiden käsittelyä tässä työssä ei katsota, koska se on PHP:n aihealue.

Ohjeet Google-kartan liittämiseen löytyvät täältä:

<https://developers.google.com/maps/documentation/embed/>

5.11 "TILAA PROJEKTI"

Projektin tilaus -sivu on nähtävissä kuvassa 39.

ETUSIVU PALVELUMME PORTFOLIO YHTEYSTIEDOT **TILAA PROJEKTI**

PROJEKTIN TILAUSLOMAKE

Olkaa hyvää ja täyttäkää tätä lomaketta! Valmistamme tarjouksen juuri Teille.

Nimi*

Sähköposti*

Puhelin

Yritys

Projektin tyyppi:

Landing Page

Yrityssivu

Verkkokauppa

Kuva 39. "TILAA PROJEKTI"-sivu.

Tämän sivun toteutus on lähellä luvun 5.10 toteutusta. Yksi poikkeus tässä on sivun taustakuva. Taustakuvaa on mahdollista hallinnoida seuraavilla tyylisäännöillä:

- "background-image" – määrittelee taustakuvan.
- "background-position" – määrittelee taustakuvan sijainnin.
- "background-size" – määrittelee taustakuvan koon.

Koska taustakuvan tulee olla koko sivulla, tyylisääntöjä lisätään body-elementille (kuva 40).

```
body.background {  
  background-image: url(http://kk-design.fi/images/background.jpg);  
  background-size: 100% 295%;  
  background-position: center;  
}
```

Kuva 40. Body-elementille tehdyt tyylisäännöt.

6 Yhteenveto

Tässä opinnäytetyössä käytiin läpi tärkeimmät asiat, jotka liittyvät verkkosivujen suunnitteluun ja toteutukseen. Erityisesti kiinnitettiin huomiota tekniseen osaan: html-rakenteeseen ja css-tyylisääntöihin. Työ sisältää runsaasti esimerkkejä, joten lukijalla on mahdollisuus oppia uutta asiaa.

PHP:n ja JavaScript:n osuutta ei kuvata tässä insinöörityössä, koska se liittyy Back End -kehitykseen.

Lähteet

1. HTML – Wiki. Verkkodokumentti. <<https://fi.wikipedia.org/wiki/HTML>>. Luettu 1.10.2016.
2. CSS – Wiki. Verkkodokumentti. <https://fi.wikipedia.org/wiki/Cascading_Style_Sheets>. Luettu 1.10.2016.
3. HTML Doctype Declaration – W3. Verkkodokumentti. <http://www.w3schools.com/tags/tag_doctype.asp>. Luettu 5.10.2016.
4. CMS – Wiki. Verkkodokumentti. <<https://fi.wikipedia.org/wiki/Sis%C3%A4ll%C3%B6nhallintaj%C3%A4rjestelm%C3%A4>>. Luettu 7.10.2016.
5. Indeksointi – Omni Partners. Verkkodokumentti. <<http://omnipartners.fi/sanakirja/indeksointi/>>. Luettu 15.10.2016.

Linkki ”Suunnittelutoimisto K&K” verkkosivulle

<http://kk-design.fi>