



MYNNIN TYÖKALU VERKKOPALVELULLE

Landing page

Allan Palmroth

Opinnäytetyö
Marraskuu 2013
Tietojenkäsittely

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietojenkäsittely
Proakatemia

PALMROTH, ALLAN:
Myynnin työkalut verkkopalvelulle
Landing page

Opinnäytetyö 41 sivua, joista liitteitä 0 sivua
Marraskuu 2013

Tämän opinnäytetyön toimeksiantona oli koota tietoa nykyaikaisen ja toimivan myyntiä ja markkinointia avustavan WWW-myyntisivun rakenteesta, ulkoasusta ja keskeisimmistä periaatteista. Tutkimuksessa kartoitettiin piirteitä, joita menestyvältä markkinointisivulta tulisi löytyä. Tutkimustehtävänä oli selvittää eri tietolähteistä toimivia ratkaisuja markkinointisivun teoriaan, ja mahdollisuus viedä opittu tieto käytäntöön helposti. Tutkimuksen jälkeen tehtiin ulkoasusuunnitelma markkinointisivulle löydetyn teorian pohjalta.

Opinnäytetyön toimeksiantajana toimi Raine Luntta Extempo osuuskunnasta. Luntta on osallisena projektissa, jossa toteutettiin verkkopalvelu muun muassa Tampereen ammattikorkeakoulun yrittäjyyden yksikölle, Proakatemialle. Verkkopalvelun nimi on Alumni Galleria, ja sen tavoite on edistää alumnien tiedon ja taidon välitystä nykyisille Proakatemian opiskelijoille. Verkkopalvelu tuoteistetaan myöhemmässä vaiheessa Kojako Galleriaksi, ja sitä tullaan myymään eri korkeakouluille ja yrityksille.

Tutkimuksesta kävi ilmi, että vaikka markkinointisivu ei varsinkaan Suomessa ole vielä vakiintunut markkinointiväline, sen toiminnallisuudesta ja teoriasta on olemassa riittävästi tietoa esimerkiksi ulkomaisissa blogikirjoituksissa. Ulkoasusuunnitelmaa tehtäessä on käytetty löydettyä tietoa markkinointisivun keskeisimmistä ohjeistuksista.

ABSTRACT

Tampere University of Applied Sciences
Degree Programme in Business Information Systems
Proacademy

PALMROTH, ALLAN:
Selling tool for webservice
Landing page

Bachelor's thesis 41 pages, appendices 0 pages
November 2013

The purpose of this thesis was to collect and present pragmatic information about a modern, functional and successful webpage built for sales and marketing purposes. Information such as landing pages' structure and layout needed to be presented. Mobile users of the internet were also taken into account, hence information about how to make webpages responsive was gathered.

The study and design for a website was made for Raine Luntta, a member of Extempo co-op. Mr. Luntta is participating in a project in which an informational web application will be created for the Proacademy unit at Tampere University of Applied Sciences.

According to the results, there are not many web pages for Finnish products that include a landing page. However, the concept of landing page was introduced as a great sales and marketing tool several years ago internationally. This study provides necessary information and examples about building a good landing page in a way that makes it easily usable.

Key words: xhtml marketing landing page responsive design

SISÄLLYS

1	JOHDANTO.....	7
2	LANDING PAGE	8
2.1	Landing pagen tarkoitus.....	8
2.1.1	Clickthrough raten kasvattaminen.....	8
2.1.2	Rekisteröityneiden käyttäjien määrän kasvattaminen	10
2.1.3	Tuotteen ostavien asiakkaiden määrän kasvattaminen	11
2.1.4	Uusien asiakkaiden määrän kasvattaminen.....	12
2.2	Esimerkki landing pagesta	13
3	KÄYTETTÄVIÄ TEKNIIKOITA.....	17
3.1	HTML	17
3.1.1	Syntaksi.....	18
3.2	XML.....	18
3.3	XHTML	19
3.3.1	Esimerkki	20
3.4	CSS	20
3.4.1	Syntaksi.....	22
3.4.2	Luokat ja aliluokat	22
3.4.3	DIV-elementti	24
3.5	PHP	26
3.5.1	PHP:n lyhyt historia	27
3.5.2	Syntaksi.....	28
3.5.3	PHP:n käyttämisen haittapuolet	29
3.6	JavaScript.....	30
3.7	Responsiivinen suunnittelu	30
3.7.1	Taustaa	31
3.7.2	Responsiivisen suunnittelun ja käytön hyödyt.....	32
3.7.3	Responsiivinen toteutus	33
4	SUUNNITELMA LANDING PAGEN TOTEUTUKSESTA	35
4.1	Kojako Galleria esimerkki 1	35
4.1.1	Rakenne.....	36
4.2	Kojako Galleria esimerkki 2	37
4.2.1	Rakenne.....	38
	LÄHTEET.....	40

ERITYISSANASTO

Landing page	Myynti- tai markkinointisivu, joka on erillään tietyn yrityksen varsinaisista kotisivuista, ja jonka päätarkoituksena on mainostaa tiettyä tuotetta tai palvelua
Liidi	Mahdollinen uusi asiakas
Konversio	Tapahtuma, jossa jokin sisältö, tieto, tai tiettyyn segmenttiin kuuluva yksilö muuttuu toiseen käyttötarkoitukseen
CTR	Clickthrough rate. Lukema, jota käytetään mittaamaan mainoskampanjan menestystä tietyn mainoksen tai linkin saamien napautusten mukaan
CTA	Call-To-Action. Painike, jossa kehoitetaan vierailijaa painamaan sitä, minkä tuloksena tapahtuu konversio
Konversio-lukema	Konversiota mittaava ja kuvaava lukema
Twiittaus, twiitti	Sosiaalisessa verkkopalvelu Twitterissä julkaistu tilapäivitys
Dropbox	Verkkopalvelu, joka tarjoaa käyttäjilleen ilmaista ja maksullista tallennustilaa
CRO	Conversion rate optimization. Konversion optimointi
Slogan	Ytimekäs mainoslause
HTML	HyperText Markup Language
XHTML	Extensive HyperText Markup Language
CSS	Cascading Style Sheets. Tyyliohje, joka määrittelee sivuston ulkoasun
PHP	Hypertext Preprocessor -ohjelmointikieli
JavaScript	Selainpohjainen ohjelmointikieli
WWW	World Wide Web
ASCII	American Standard Code for Information Interchange. Tietokoneiden merkistö, joka sisältää ensisijaisesti amerikanenglannissa tarvittavat kirjaimet, numerot, väli- ja erikoismerkkejä sekä eräitä ohjauskoodeja
Validointi	Dokumentin kieliopin oikeellisuuden tarkastaminen dokumenttityypin ja yleisten säännösten mukaan

W3C	World Wide Web Consortium. Kansainvälinen yritysten ja yhteisöjen yhteenliittymä, joka ylläpitää ja kehittää WWW-standardeja ja suosituksia
DTD	Data type Defining. Yleinen tapa kuvata dokumentin kielioppi
URL-osoite	Uniform Resource Locator. Käytetään osoittamaan WWW-sivuja
Hakkeri	Innokka tietokonealan harrastaja, joka omaa paljon tietoa esimerkiksi tietystä tietojärjestelmästä tai koodikielestä. Suomessa yleisesti käytetty termi esimerkiksi tietoturvarikollisista

1 JOHDANTO

Tampereen ammattikorkeakoulun yrittäjyyden yksikössä, Proakatemiolla, oppiminen on käytännönläheistä luennoille osallistumisen sijaan. Proakatemia on perustettu vuonna 1999, ja siellä opiskelee yhtäaikaaisesti 6 tiimiyritystä. Proakatemian pitkästä historiasta johtuen alumnien määrä on suuri, mutta nykyisillä tiimiyrittäjillä, tai Proakatemiolla, ei ole ollut rekisteriä, jossa olisi listattu alumnit sekä heidän yhteistietonsa. Tähän tarpeeseen luotu on verkkopalvelu, Alumni Galleria. Se tarjoaa nykyisten opiskelijoiden sekä valmentajien käyttöön tiedot kaikista suostumuksensa antaneista alumneista sekä valmentajista; heidän yhteystietonsa, heidän aikaisemmat ja nykyiset työtehtävänsä sekä heidän osaamisensa toimialoittain. Alumni Gallerian toteuttivat Raine Luntta ja Antti Vuento. Luntan ja Vuennon luoma palvelu on prototyyppi Kojako Galleriasta. Kojako Galleria toimii samalla periaatteella kuin Alumni Galleria, mutta sen asiakaskuntaan kuuluvat myös yritykset ja erilaiset yhteisöt oppilaitosten lisäksi. Yrityskäyttöön tuotteistettava versio palvelusta keskittyy esimerkiksi alihankkijoiden jäsentelyyn heidän toimialansa mukaan.

Opinnäytetyön tavoitteena on selvittää keskeinen teoria, johon täytyy kiinnittää huomiota Kojako Gallerialle luotavassa markkinointisivussa, sitä suunniteltaessa ja toteutettaessa. Markkinointisivu toimii myyntiä edistävänä ja helpottavana työkaluna tarjoten tietoa palvelusta.

Opinnäytetyön tarkoituksena on etsiä tarvittava teoria ja koota se loogisesti, jotta teoria olisi helppo saattaa käytäntöön. Tarkoituksena on antaa esimerkkejä olemassaolevista ja toimivista markkinointisivuista, sekä antaa toimeksiantajalle teorian pohjalta luotu malli Kojako Gallerialle suunnitellusta markkinointisivusta.

2 LANDING PAGE

Landing page nimitys tulee sanasta (*to*) *land* (*on something*) (=laskeutua). Tämän määritelmän mukaan mikä tahansa internetsivu on periaatteessa *landing page*. Markkinoinnin alalla *landing page* kuitenkin viittaa yksittäiseen sivuun, joka on erillään yrityksen varsinaisista kotisivuista, ja joka on suunniteltu yhteen tiettyyn tavoitteeseen pääsemiseksi. (Unbounce, What is a landing page?.)

2.1 Landing pagen tarkoitus

Landing page on tietylle tuotteelle, palvelulle tai sivustolle tehty yksittäinen myyntisivu, jonne usein ohjaututaan käyttäjän painettua mainosta, linkkiä, tai linkkiä jonka hakukone antaa tuloksena tietylle haulle. *Landing pagen* perimmäinen tarkoitus on muuttaa sivustolla käyvä vierailija asiakkaaksi tai *liidiksi*. (Wikipedia, Landing page.) Tätä muuntumista kutsutaan myös *konversioksi*, joka tulee englanninkielisestä sanasta (*to*) *convert* (=muuttua, muuntua).

Kun puhutaan internetmarkkinoinnin suunnittelusta, on ensisijaisen tärkeää selvittää, mikä sivun tarkoitus on, ja mitä vierailijan halutaan tekevän. Halutaanko vierailijan kirjautuvan palvelun käyttäjäksi, lataavan jotakin, ostavan jotakin, tilaamaan esimerkiksi uutiskirjeen tai jakamaan jotain omille verkostoilleen. (Longanecker, 2012.)

Landing pagen käyttämisen suurin hyöty on, että sen avulla vierailijan huomio kiinnitetään ainoastaan myytävään tuotteeseen. Vierailijalle ei anneta ylimääräisiä tai myytävän tuotteen kannalta turhia virikkeitä, jotta esimerkiksi vierailijan impulsiivinen ostohalu ei katoa tai tukahdu.

2.1.1 Clickthrough raten kasvattaminen

Jos sivuston tarkoituksena on esimerkiksi saavuttaa ennalta määrätty mainoksen painelukertojen määrä, siihen yritetään vaikuttaa eri keinoilla. Muun muassa seuraavaksi esitellyt ohjeet todistetusti lisäävät *clickthrough ratea* (*CTR*) Longaneckerin (2012) mukaan.

Ensinnäkin käyttäjiä pitäisi johdatella (tuotteen tai palvelun) hyödyillä. Suunnittele selkeät ja houkuttelevat mainoslauseet, joissa kerrotaan (esimerkiksi tuotteen) konkreettisia hyötyjä asiakkaalle. (Longanecker, 2012.) Myös asiakkaiden antamat suositukset toimivat mainoslauseina, sillä niissä usein käy ilmi juuri tietyn asiakkaan saamat hyödyt. Suositukset myös lisäävät Gardnerin (101 Landing Page Optimization Tips) mukaan luottamusta tarjoamaasi palvelua tai tuotetta kohtaan, mikä taas parantaa *konversiota*.

Toinen ohje suuremman *CTR*:n saavuttamiseksi on Longaneckerin (2012) mukaan selkeä *kutsu*, *Call-To-Action* (*CTA*). *CTA*:n tulee olla selkeä, esiintyöntyvä ja "mehukas". Esimerkiksi erilaisten visuaalisten muotojen, kuten nuolien, käyttäminen *CTA*:na tekevät painikkeiden painamisesta houkuttelevampaa. (Longanecker, 2012.) Formstackin (Anatomy of a perfect landing page) mukaan *CTA*:n tulisi olla iso, kirkas, väriltään oranssi tai keltainen, ja sijaita sivulla sellaisessa kohdassa, että vierailijan ei tarvitse liu'uttaa sivua alaspäin sen nähdäkseen.

Gardnerin (Unbounce) mukaan *CTA*-painikkeen tulisi olla monessakin eri kohdassa sivua alaspäin liikuttaessa erityisesti pitkillä sivuilla joissa on paljon tekstiä. Tämä erityisesti sen takia, että erilaiset ihmiset tekevät ostopäätöksen eri aikoihin ja eri syistä. Jos jonkun vierailijan täytyy lukea kaksi kolmasosaa sivulla olevasta tekstistä ennen ostopäätöstä, on hyvä että sivulla on *CTA*-painike juuri siinä kohtaa, missä vierailijan tunteeseen perustuva ostopäätös kohtaa sivun sanoman kanssa.

Kolmantena ohjeena Longanecker (2012) kehoittaa käyttämään ajatusta visuaalisen hierarkisuuden suunnitteluun. Mikä informaatio sivulla on ensisijaista? Mitä haluat vierailijan näkevän ja sisäistävän? Hänen mukaansa typografian, fontin koon, ja fontin painotuksen tulisi yhdessä visuaalisten vihjeiden kanssa omalta osaltaan vaikuttaa vierailijan vetämisessä puoleensa. (Longanecker, 2012.)

Viimeisimpänä ohjeena Longanecker (2012) kehoittaa muistamaan *AIDAn*, mikä tulee sanoista *awareness* (=tietoisuus), *interest* (=mielenkiinto), *decision* (=päätös), *action* (=toiminta). Voita ensimmäiset kolme, niin saat neljännen. Luo tietoisuus ja mielenkiinto tuotetta tai palvelua kohtaan, sen jälkeen tarjoa lisäinformaatiota päätöksen tekemisen tueksi. (Longanecker, 2012.)

2.1.2 Rekisteröityneiden käyttäjien määrän kasvattaminen

Jos ollaan julkistamassa uutta websovellusta tai -palvelua, on todennäköistä, että markkinointiin ei ole käytettävissä suuria summia rahaa. Rahan puutteen vuoksi vertaismarkkinointikampanjan tehokkuus näyttelee kriittistä osaa tuotteen tai palvelun menestyksessä. On ehdottoman tärkeää, että yritys suunnittelee mainoskampanjan levittämisen jo olemassa olevien asiakkaiden kautta uusille potentiaalisille asiakkaille. (Longanecker, 2012.)

Longaneckerin (2012) mukaan seuraavana luetellut tekniikat auttavat käyttäjälähtöisen markkinointikampanjan levittämistä.

Landing page, tai pikemminkin *konversion* optimointi. Vierailijoilta tulisi kysyä ainoastaan sellaista tietoa, joka on ehdottoman tärkeää palvelun tai tuotteen myyjätaholle. Mitä vähemmän täytettäviä tekstikenttiä esimerkiksi rekisteröitymislomakkeessa on, sitä suuremmat *konversio-lukemat* saavutetaan. (Longanecker, 2012.) Myös Paulin-Daiglen (Checkout Champions: 10 Tips To Win More Ecommerce Sales) mukaan mitä enemmän tekstikenttiä asiakas joutuu täyttämään, sitä enemmän aikaa hän joutuu siihen kuluttamaan. Vanhan sananlaskunkin mukaan aika on rahaa, joten mitä enemmän aikaa rekisteröitymiseen kuluu, sitä varmemmin se jää kokonaan tekemättä.

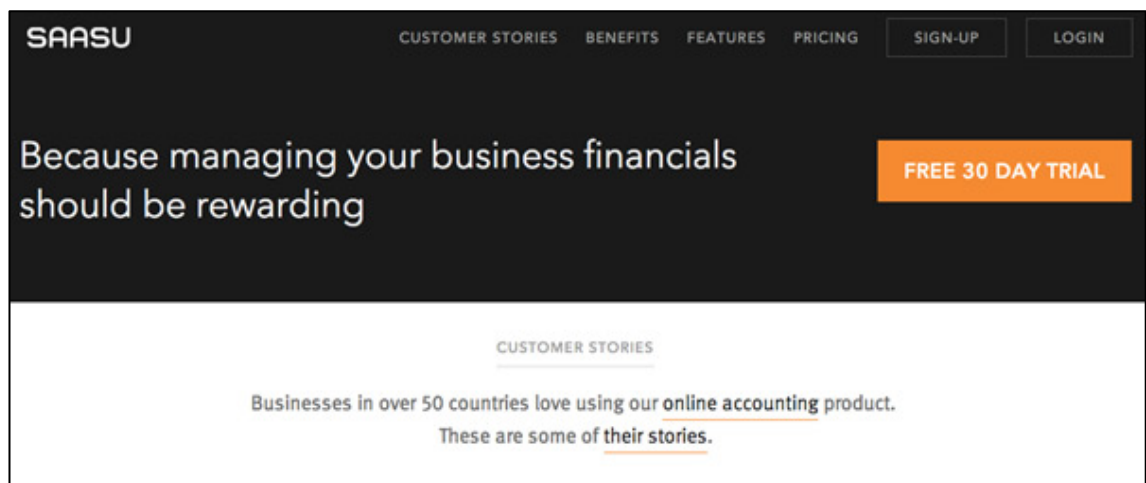
Jos tavoitteena on rekisteröityneiden käyttäjien lukumäärän kasvattaminen, on hyvä miettiä sosiaalisen median hyväksikäyttämistä rekisteröitymistilanteessa. Esimerkiksi Facebookin tilin käyttäminen rekisteröinnissä ei pelkästään tee rekisteröitymisprosessista helpompaa, vaan auttaa sisällyttämään tapahtumaan suosituksia sekä antaa rekisteröitymisprosessiin luotettavuutta. Nämä kaikki yhdessä mahdollistavat rekisteröitymisvauhdin ja lukumäärän kasvattamisen. (Longanecker, 2012.) Sosiaalisen median hyväksikäyttäminen sivulla on muutenkin hyödyllistä, sillä se kasvattaa sivun luokitusta hakukoneissa, ja esimerkiksi toisten tutkimusten mukaan myös mitä enemmän sivua on *twiitattu*, sitä paremman luokituksen ja näkyvyyden se saa hakukoneissa (DeMers, [How To] Create Landing Pages that Convert & Rank Well in Search).

Eräs tärkeä seikka vertaismarkkinointikampanjan onnistumisessa on olemassaolevien käyttäjien palkitseminen heidän tekemästään työstä kampanjan levityksessä. Esimerkiksi *Dropbox* tarjoaa lisää tallennustilaa käyttäjille, jotka hankkivat palvelulle lisää käyttäjiä. (Longanecker, 2012.)

2.1.3 Tuotteen ostavien asiakkaiden määrän kasvattaminen

Kun *landing page* tarkoituksena on kasvattaa maksavien asiakkaiden määrää, huomion tulisi keskittyä kahteen eri seikkaan: selkeyteen ja kitkan vähentämiseen. *Landing page* muotoilun tulisi tarjota asiakkaalle helposti ja selkeästi tiedon siitä, mitä ollaan myymässä, sekä tuotteen tai palvelun tarjoamat hyödyt asiakkaalle. (Longanecker, 2012.)

Lojaalius ja luottamus ovat kestävän bisneskumppanuuden perusta. Kun tietää asiakkaan ongelman tai haasteen ja tarjoaa siihen ratkaisun, asiakas näkee myyjän luotettavana kumppanina ennemmin kuin vain yhtenä myyjänä muiden joukossa. Ennenkuin kiirehtii myymään, pitää synnyttää myyjän ja asiakkaan välille luottamus. Yksinkertaisimmillaan tämä toteutetaan siten, että näytettävien myyntilauseiden sijaan sivulla esitettäisiin ensin asiakkaan ongelman ratkaisemiseen johdatteleva lause, kuten "Koska yrityksesi talouden hoitamisen tulisi olla palkitsevaa". (Passiak, Seven Steps to Selling SaaS like Socrates – it's About Solutions, Not Products!.) Seuraavassa kuviossa näkyy, miten kyseinen lause voidaan sisällyttää internetsivulle. Kuviossa on myös eräs toinen *konversiota* edistävä tekijä, ilmainen kokeilujakso tuotteelle.



KUVIO 1. Luottamusta herättävän sanoman ilmaiseminen myyntilauseiden sijaan. Kuva: Passiak, Seven Steps to Selling SaaS like Socrates – it's About Solutions, Not Products!

2.1.4 Uusien asiakkaiden määrän kasvattaminen

Landing page perimmäinen tarkoitus saattaa olla myös pelkästään mahdollisten uusien uutiskirjeen tilaajien, rekisteröityneiden käyttäjien tai asiakkaiden kartoittaminen ja saaminen. Esimerkiksi uutiskirjeiden tilaajista käytetään sanaa *liidi*, sillä he eivät välttämättä vielä osta tarjottavaa palvelua tai tuotetta, mutta antavat kuitenkin esimerkiksi sähköpostiosoitteensa vastaanottaakseen uutiskirjeitä. Nick Martinin (Kissmetrics, How To Identify And Stop Emotion Based Conversion Killers) sanoin: "If they're not going to get me to buy now, they'll get me eventually." On jopa olemassa kokonainen analytiikan osa-alue, *conversion rate optimization (CRO)*, jonka tarkoitus on parantaa kirjautumis- ja rekisteröitymistilanteissa tapahtuvaa *konversiota* (Longanecker, 2012).

Longanecker (2012) mainitsee muutamia vinkkejä joiden avulla yllämainittua konversiota voidaan parantaa:

1. Kerro tarina. Ilmaise, kuinka tuotteesi tai palvelusi on ratkaissut asiakkaan ongelman, jolloin ainoa looginen seuraava askel sivustolla vierailijalle on kirjautua sisään tai rekisteröityä.

2. Käytä selkeää hierarkiaa. Ohjaa vierailijat koko prosessin läpi, ja tee heille helpoksi tarttua pääkohtiin ja tunnistaa CTA:t.

3. Ole ytimekäs. Älä täytä sivua tai sivuja turhalla tai ylimääräisellä informaatiolla. Ole säälimätön päättäessäsi mikä ansaitsee olla näkyvillä sivulla ja mikä ei.

4. Valitse mukaansatempaavia kuvia. Tarinaasi ei kerrota yksin sanojen avulla, varmista siis että kuvat, joita sivullasi on, kertovat samaa tarinaa sanojen kanssa.

5. Rajoita tarvittun tiedon määrää. Mahdolliselta tulevalta asiakkaalta vaadittava informaatio rekisteröitymistilanteessa kannattaa pitää absoluuttisessa minimissä, jolla kuitenkin vielä saavutetaan tarvittavat tulokset.

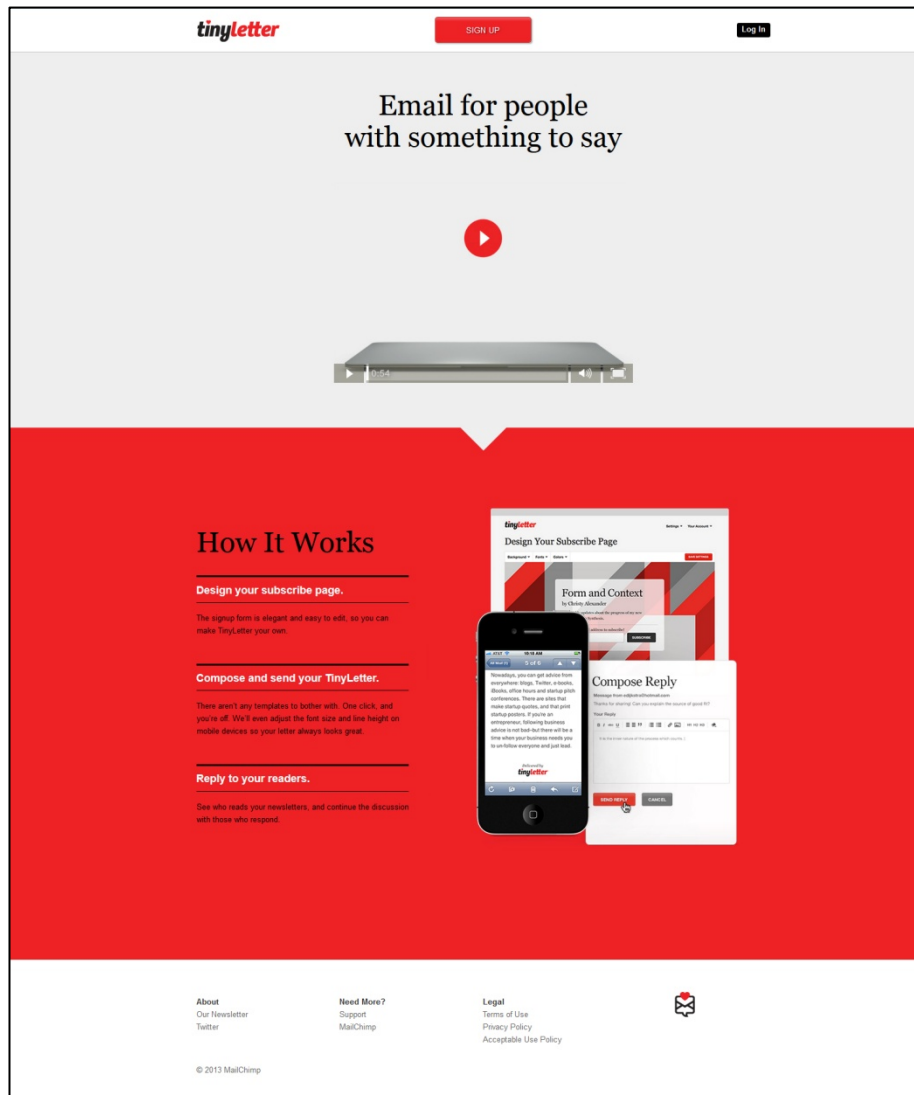
(Longanecker, 2012.)

2.2 Esimerkki landing pagesta

Tinyletter on työkalu, joka on suunniteltu ihmisille, jotka laativat sähköpostin välityksellä jaettavia uutiskirjeitä. Tinyletteri avulla käyttäjä pystyy muun muassa suunnittelemaan haluamansa näköisen uutiskirjeen helposti ja lisäämään henkilöitä postituslistoille. Tinyletter myös kerää dataa uutiskirjeen lukijoista uutiskirjeen lähettäjälle. Kerättävää dataa ovat muun muassa uutiskirjeen avauksien määrä, eri painikkeiden painelukertojen määrä sekä määrä, joka uutiskirjettä on lähetetty. Seuraavassa kuviossa (KUVIO 2) on ruutukaappaus Tinyletteriin *landing pagesta*.

Tinyletteriin *landing page* (KUVIO 2) on oiva esimerkki hyvin suunnitellusta *landing pagesta*. Se koostuu neljästä osasta. Ylimpänä olevassa valkoisessa osassa on punainen ja erottuva CTA-painike, "Sign up". Se pysyy näkyvissä sivun ylälaidassa huolimatta siitä, vieritetäänkö sivua alaspäin. Seuraavassa, vaalean harmaassa osiossa on tuotteen mainoslause eli *slogan* "Email for people with something to say" (=sähköposti ihmisille, joilla on jotain sanottavanaan). Lause tiivistää työkalun käyttötarkoituksen, ja kertoo kenelle se on suunniteltu. *Sloganin* alla on video (KUVIO 2), jossa kerrotaan tarkemmin Tinyletteristä. Video kestää vain alle minuutin, mutta siitä saa hyvän kuvan työkalusta nopeammin kuin lukemalla. Videossa näytetään myös Tinyletteriin toimintoja

käytännössä. Kolmantena osiona, punaisella pohjalla on enemmän tietoa tuotteesta niille, jotka eivät katso videota. Tässä osiossa on myös kuva kahdesta Tinyletterin toiminnosta sekä näkymästä älypuhelimessa (KUVIO 3). Viimeisessä valkoisessa osiossa on käyttäjälle vähemmän tärkeää tietoa, kuten käyttöehdot ja linkki tukisivustolle.

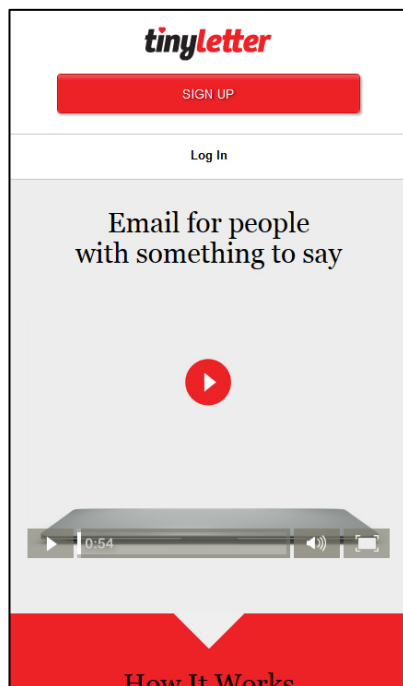


KUVIO 2. Tinyletter palvelun landing page kokonaisuudessaan. Ruutukaappaus



KUVIO 3. Tinyletterin sivulla punaisessa osiossa näkyvät kuvat tuotteen toiminnoista. Ruutukaappaus

Tinyletterin *landing page* on täysin responsiivinen (KUVIO 4). Se mukautuu sen mukaan, minkä kokoisella laitteella tai näytöllä sitä katsotaan. Esimerkiksi punaisella osiolla näkyvä kuva (KUVIO 3) ei näy ollenkaan selattaessa sivua älypuhelimella. Todennäköisesti syy on, että kuvalla ei ole suurta merkitystä tuotteen esittelyn kannalta, vaikka se antaakin katsauksen siitä, mitä tuote sisältää.



KUVIO 4. Tinyletterin landing pagen asettelu ja ulkoasu katsottaessa sivua mobiililaitteella tai selaimen ikkuna pienennettynä. Ruutukaappaus

Tinyletteriin sivulta puuttuu myös perinteinen navigointivalikko. Kaikki oleellinen tieto tuotteesta esitellään yhdellä sivulla. Tällöin sivulla vierailijan huomio keskittyy itse tuotteeseen. CTA-painike on punainen ja siksi erottuva valkoisella pohjalla. Se seuraa mukana sivun yläreunassa sivua alaspäin vieritettäessä, jolloin se on aina lähellä, riippumatta siitä milloin vierailija tekee ostopäätöksen. Slogan kertoo ytimekkäästi kenelle tuote on suunnattu, ja sloganin alla olevasta videosta saa nopeasti kuvan tuotteen toiminnoista. Erityisen hyvänä koen, että video ei lähde toistamaan heti sivulle mentäessä, vaan on vierailijan päätettävissä katsoa video. Sen varalta että vierailija ei halua katsoa videota, on alempana sivulla kerrottu tuotteesta tekstimuodossa, joten myös lukijatyypin ihmiset on huomioitu *landing pagella*. Sivua ei ole ylikuormitettu turhalla tiedolla, vaan siinä kerrotaan lyhyesti tärkeimmät seikat tuotteesta.

3 KÄYTETTÄVIÄ TEKNIIKOITA

Koska *landing page* on kuin mikä tahansa internetsivu rakenteeltaan, sen koodissa ei ole mitään erityisiä tekniikoita verrattuna esimerkiksi jonkin yrityksen kotisivuihin. Se luodaan käyttäen HTML- tai XHTML-merkkauškieliä, ja sivun ulkoasun tyylitiedot määritellään CSS-tiedostossa. Sivun toiminnallisuudet voidaan suorittaa pääosin esimerkiksi PHP-ohjelmointikielellä. Myös JavaScript on suosittu ohjelmointikieli. Tässä luvussa esitellään tekniikoita, joita käytetään *landing pagen* luomisessa.

3.1 HTML

HTML-kieli on WWW-sivujen perusrakennusmateriaali. Internetsivuilla näytettävä data sijaitsee palvelimella HTML-dokumentissa. HTML-dokumentti sisältää HTML-kielen muotoilukomentoja, ja muodostaa ohjeet internetselaimessa esitettävien WWW-sivujen loogiseen tai fyysiseen muotoiluun (Linjama, 2001:32).

HTML-dokumentin luomiseen ei tarvita erikoisia ohjelmia, vaan mikä tahansa ASCII-tekstieditori kelpaa. Itse käytän HTML-dokumentin tekoon Notepad++ -ohjelmaa, sillä siinä olevat asetukset helpottavat HTML-koodin lukemista ja kirjoittamista, ja koodin tarkistaminen mahdollisten virheiden varalta on huomattavasti mielekkäämpää. HTML-koodi kannattaa kuitenkin aina *validoida* varmistaakseen koodin virheettömyys.

Internetselaimet eivät tarkista tekstin tarkkaa rakennemäärittelyä, vaan hyväksyvät myös rakenteeltaan virheelliset HTML-dokumentit. Selaimet kuitenkin näyttävät virheellisen HTML-koodin olettamiansa korjausten avulla, mikä voi aiheuttaa sivun virheellisen näkymisen. (Linjama, 2001:67.)

Tyypillisiä virheitä HTML-koodauksessa ovat muun muassa pakollisen päättävän tagin puuttuminen, tagi-parit on osoitettu ristiin, elementin attribuutteja ei ole erotettu välilyönnillä tai attribuutin arvon lainausmerkeistä puuttuu toinen (Linjama, 2001:66).

3.1.1 Syntaksi

Yksinkertaisimmillaan HTML-dokumentiin kuuluu vain muutama elementti. Pakolliset elementit ovat <html>, <head> ja <body> elementit. Seuraavassa kuviossa näkyy HTML-dokumentin perusrakenne.

```
testi.html
<html>
  <head> Sivun otsikko </head>
  <body>

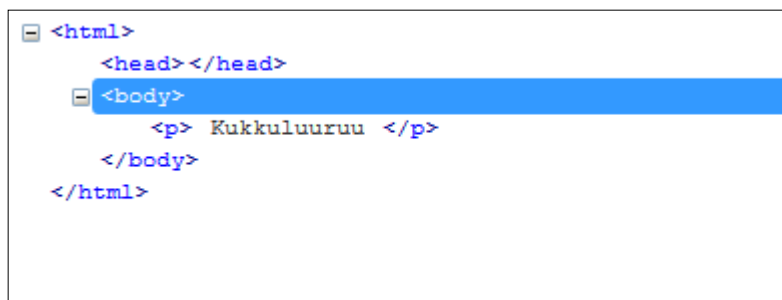
    <h1>Ensimmäisen tason otsikko</h1>

    <p>Tekstielementti</p>

  </body>
</html>
```

KUVIO 5. Esimerkki testi.html -tiedoston sisällöstä

Pakollisia elementtejä ei kuitenkaan tarvitse kirjoittaa itse dokumenttiin, sillä jotkin selaimet lisäävät tarvittavat elementit automaattisesti. Seuraava kuvankaappaus (KUVA 1) Firebug-ohjelmalla näkyvästä testiverkkosivun rakenteesta, kun dokumentissa oli vain <p> -elementti. Dokumentti on avattu Mozilla Firefox selaimen 22.0 versiolla.



```
<html>
  <head></head>
  <body>
    <p> Kukkuluuruu </p>
  </body>
</html>
```

KUVA 1. Kuvankaappaus Firebug-ohjelman koodista

3.2 XML

XML (eXtensible Markup Language) esiteltiin vuonna 1998. Se on kielioppi, joka itsessään ei määrittele kuin dokumentin yleisen rakenteen, jossa esitellään vain kieliversio ja dokumentin juurielementti. (Linjama, 2001:23.) Sen kehitti W3C, ja alunperin se kehitettiin vastaamaan laajamittaisen elektronisen julkaisemisen haasteisiin (W3C, XML).

XML:n käyttämisellä tavoitellaan muun muassa riippumattomuutta tietystä ohjelmistotoimittajasta, integraation helpottamista, sisältövirheiden välttämistä, käsittelyvaiheiden automatisointia ja tiedon hakemisen helpottamista (Wikipedia, XML).

3.3 XHTML

XHTML on XML johdannainen, joka on HTML-hypertekstin kuvauskieltä, ja on vain kirjoitettu uudelleen käyttäen XML-kielioppia (Linjama, 2001:24). Perinteisen HTML-kielen komentojen lisäksi XHTML-dokumenttiin voidaan liittää XML-standardin mukaan tehtyjä omia elementtejä (Linjama, 2001:9). Virallisesti XHTML 1.0 on esitelty vuonna 2000 tarkoituksenaan korvata vanhat HTML-standardit (Linjama, 2001:854).

XHTML-dokumentti aloitetaan XML-kieliversion esittelemisellä, jonka jälkeen esitellään HTML-kieliversio ja linkitetään XHTML-DTD. DTD määrittelee elementit, attribuutit ja näiden arvotyypit. XML-kieliversiota ei kuitenkaan tarvitse esitellä, jos dokumentin tiedostopäätte on .htm tai .html. (Linjama, 2001:845 ja 854.) Seuraavassa kuviossa näkyy esimerkki XHTML-dokumentin oikeaoppisesta XML- ja DTD-määrittelystä.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

KUVIO 6. Esimerkki XHTML-dokumentin oikeaoppisesta XML- ja DTD-määrittelystä

HTML-elementti on aina XHTML-dokumentin juurielementti, jossa kerrotaan dokumentin kielityyppi xmlns-attribuutilla, ja jonka arvona on W3C:n XHTML-määrittelyn URL-osoite. Seuraavassa kuviossa näkyy miten kyseinen koodi kirjoitetaan XHTML-dokumenttiin.

```
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="fi">
```

KUVIO 7. W3C:n XHTML-määrittelyn URL-osoitteen oikeamuotoinen ilmaiseminen XHTML-koodissa

3.3.1 Esimerkki

Seuraavassa kuviossa on esimerkki XHTML-dokumentista yksinkertaisimmillaan ilman CSS-tyylimääreitä.

```
testi.html
<?xml version="1.0" encoding="iso-8859-1"?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="fi">
<head>
<title> Esimerkkisivu </title>
</head>
<body>
    <h1> Tämä on tason 1 otsikko! </h1>
    <p> Ja tähän tulee sitten leipätekstiä. </p>
</body>
</html>
```

KUVIO 8. XHTML-dokumentti yksinkertaisimmillaan ilman CSS-tyylimääreitä

3.4 CSS

CSS on erityisesti WWW-dokumenteille kehitetty tyyliohje. Tyyliomakkeita käyttämällä sivuston tekijän ei tarvitse kiinnittää huomiota sivujen ulkoasuun, vaan hän voi keskittyä kokonaan dokumentin sisällön ja rakenteen luomiseen. Tyylitiedostojen käytön yksi huomattava etu onkin ulkoasun yhtenäisyys sekä helppo muokattavuus. (Linjama, 2001:502.)

Tyylitiedostoa käyttämällä sivuston hallinta ja muokattavuus tehostuu, sillä sen avulla (X)HTML-dokumentista voidaan jättää pois kaikki tyylimäärittelyt. Yhdellä muutoksella tyylitiedostoon saadaan esimerkiksi muutettua kaikkien dokumentissa esiintyvien tietyllä tagilla merkittyjen elementtien väri tai koko. Etuna on myös se, että kun tyylitiedosto on erillisenä, varsinaisen HTML-dokumentin särkemisen riski pienenee, kun siihen ei tarvitse koskea.

Särkemisellä tässä tapauksessa tarkoitan esimerkiksi tahattomasti tehtyä koodin muokkausta, jonka seurauksena sivu näkyy virheellisesti.

Tyylit voidaan määritellä suoraan HTML-dokumentin head-osioon (sisäinen tyylitiedosto), tai tyylit voivat sijaita erillisessä dokumenttiin linkitetystä .css-tiedostossa (ulkoinen tyylitiedosto). Stockuksen (2007) mukaan sisäisiä tyylitiedostoja ei kuitenkaan suositella käytettävän muun muassa sen tähden, että siinä menetetään tyylitiedostojen käyttämisen suurimmat hyödyt; tehtyjen muutoksien vaikuttaminen kerralla kaikkeen sekä tiedostojen koon suuruuden vuoksi. Seuraavassa kuviossa on esimerkki HTML-dokumentin head-osiossa sijaitsevasta sisäisestä tyyli-määritteestä:

```
testi.html
<head>
<style type="text/css">
  h1 {border-width:1px; border:dotted;}
</style>
</head>
```

KUVIO 9. Esimerkki HTML-dokumentin head-osiossa sijaitsevasta sisäisestä tyylimääritteestä

Seuraavassa kuviossa näkyy esimerkki ulkoisen tyylitiedoston linkittämisestä HTML-dokumenttiin dokumentin head-osiossa.

```
testi.html
<head>
<link rel="stylesheet" type="text/css" href="tyylit.css">
</head>
```

KUVIO 10. Esimerkki ulkoisen CSS-tiedoston linkittämisestä HTML-dokumentin head-osiossa

Itse käytän sisäisiä tyylitiedostoja erityisesti sivuston testivaiheessa, sillä siinä sivuston tyylejä sekä rakennetta pystyy muokkaamaan vain yhtä tiedostoa muokkaamalla. Lopullisiin sivuihin kuitenkin poistan sisäisen tyylitiedoston HTML-dokumentista, ja sijoitan sen palvelimelle ulkoiseen tyylit.css tiedostoon.

3.4.1 Syntaksi

Wikipedian (CSS) mukaan CSS:n perussyntaksi muodostuu *valitsimesta*, *ominaisuudesta* ja *arvosta*. Kaikki ominaisuudet tulevat aaltosulkeiden "{}" sisään, ja useampia ominaisuuksia ja arvoja lisätessä ne erotellaan toisistaan puolipisteellä ";". Ominaisuus ja arvo erotellaan toisistaan kaksoispisteellä ":". Seuraavassa kuviossa näkyy CSS:n perussyntaksi ja koodin oikeaoppinen asettelu tyylitiedostossa.

```
valitsin {  
    ominaisuus1: arvo;  
    ominaisuus2: arvo;  
}
```

KUVIO 11. CSS:n perussyntaksi ja koodin oikeaoppinen asettelu tyylitiedostossa

Valitsin voi olla mikä tahansa HTML-dokumentin elementti (*body*, *h1*, *table*, *div*, ym...). Valitsimien ominaisuuksia (*color*, *border*, *width*, *padding-left*, ym...) voidaan muuttaa vaihtamalla arvoja (*red*, *dotted*, *200px*, *3em*, ym...). (Leinonen, 2008.)

3.4.2 Luokat ja aliluokat

Vaikka HTML-dokumenttia luotaessa voidaan jättää tyylit huomioimatta tyylitiedostoa käytettäessä, kannattaa sen luomisessa kuitenkin muistaa tyylien olemassaolo, koska dokumentissa voi olla tarpeellista myös käyttää tyylien aliluokkia. Aliluokkia voidaan tarvittaessa käyttää dokumentin ulkoasun hiomiseen, ja niiden avulla jokaiselle elementille saadaan tarpeellinen ulkoasu käyttötarkoituksen mukaan rikkomatta dokumentin rakennetta. (Linjama, 2001:517.)

Aliluokat merkitään HTML-dokumentissa "class"-luokkamerkinnällä. Seuraavassa kuviossa näkyy miten class-merkintää sovelletaan HTML-dokumentissa.

```

testi.html
<body>
<p class="green"> Tämän tekstin haluan vihreäksi</p>
<p class="red"> ja tämän punaiseksi </p>
</body>

```

KUVIO 12. Esimerkki class-merkinnän soveltamisesta HTML-dokumentissa

Class-aliluokkaan viitataan CSS-tiedostossa käyttämällä pistettä (.). Seuraavassa kuviossa näkyy class-aliluokan merkintä CSS-tiedostossa.

```

tyylit.css
p.green {
    color:green;
}
p.red {
    color:red;
}

```

KUVIO 13. Class-aliluokan merkintä CSS-tiedostossa

Samaa luokkaa voi käyttää eri valitsimissa, ja luokkiin voidaan viitata CSS-tyylitiedostossa myös määrittämättä erikseen valitsinta, jolloin se vaikuttaa kaikkiin elementteihin kyseisessä luokassa. Seuraavissa kuvioissa näkyy saman aliluokan merkitseminen useampaan elementtiin ja niihin viittaaminen CSS-tyylitiedostossa.

```

testi.html
<body>
    <h1 class="green"> Vihreä otsikko </h1>
    <p class="green"> Tässä on vihreää leipätekstiä </p>
</body>

```

KUVIO 14. Class-elementin merkitseminen useammalle elementille

```

tyylit.css
.green {
    color:green;
}

```

KUVIO 15. Class-elementin viittaus CSS-tiedostossa siten, että määrite vaikuttaa kaikkiin "green"-aliluokan elementteihin

Vaihtoehtona "class"-attribuutin käytölle voidaan käyttää myös id-attribuuttia, mutta toisin kuin *classia*, *id:tä* voidaan käyttää dokumentissa vain kerran. Kun samaa aliluokkaa halutaan käyttää useammassa elementissä, pitää id-attribuutti korvata class-attribuutilla. *Id:tä* käytetään esimerkiksi erottamaan jokin tietty h1-tason otsikko muista h1-tason otsikoista. (Linjama, 2001:519.) Seuraavassa kuviossa näkyy esimerkki id-attribuutin merkitsemisestä HTML-koodissa.

```
testi.html
<body>
  <h1 id="vasen"> Tämä otsikko näytetään sivun vasemmassa laidassa
</h1>
  <h1> Kun taas kaikki muut otsikot näkyvät sivun oikeassa reunassa
</h1>
</body>
```

KUVIO 16. Esimerkki id-attribuutin merkitsemisestä HTML-koodissa

Id-luokka ilmaistaan CSS-tyylitiedostossa "#" -merkillä. Seuraavassa kuviossa näkyy esimerkki *vasen*-nimisen id-attribuutin merkintä CSS-tiedostossa.

```
tyylit.css
h1 {
  text-align:right;
}
#vasen {
  text-align:left;
}
```

KUVIO 17. *Vasen*-nimisen id-attribuutin merkintä CSS-tiedostossa

3.4.3 DIV-elementti

"Div on yksi tärkeimmistä HTML-standardin elementeistä. Pääasiassa tällä elementillä ohjataan tyylilomakkeiden arvojen kohdistumista HTML-dokumentin alueella." (Linjama, 2001:202). Div-elementti on asemointilohkon muodostava elementti, jolla rajataan ne HTML-elementit joihin halutaan kohdistaa erilaisia tyylejä. Se tarkoitettiin alunperin korvaamaan center-elementti, mutta tyylilomakkeiden tulon jälkeen siitä on muodostunut yksi tärkeimmistä HTML-elementeistä. (Linjama,2001:202.)

Linjaman (2001:203) mukaan div-elementillä rajataan siis tietty lohko, jolle voidaan määritellä useita erilaisia arvoja, kuten sijainti selaimessa, kehyksen ulkoasu, lohkoon kirjoitetun tekstin ulkoasu ja niin edelleen.

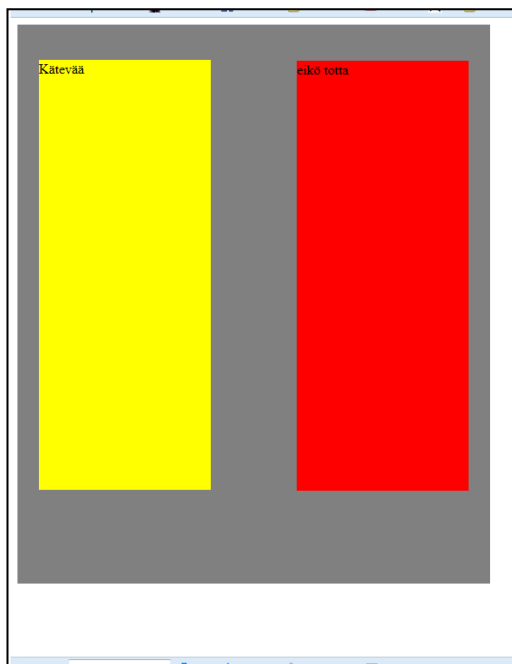
Itse käytän hyvin paljon div-elementtiä erityisesti sivuston ulkoasun rakenteelliseen luomiseen. Seuraavissa kuvioissa esimerkit HTML-dokumentista, CSS-tyylitiedostosta sekä kuvankaappaus selaimen ikkunasta näytettäessä kyseiset dokumentit.

```
testi.html  
<div id="sisalto">  
<div id="vasen"> <p> Kätevää </p> </div>  
<div id="oikea"> <p> eikö totta </p> </div>  
</div>
```

KUVIO 18. Esimerkki HTML-dokumentista

```
tyylit.css  
#sisalto {  
    height: 600px;  
    width:500px;  
    background-color:grey;  
    padding:25px;}  
#vasen {  
    background-color:yellow;  
    height:500px;  
    width: 200px;}  
#oikea {  
    background-color:red;  
    height:500px;  
    width:200px;  
    margin-left:300px;  
    margin-top:-515px;}
```

KUVIO 19. Esimerkki CSS-tyylitiedostosta



KUVIO 20. Kuvankaappaus Mozilla Firefox-selaimen ikkunasta

3.5 PHP

PHP on palvelinpuolella suoritettava ohjelmointikieli. PHP on laajasti käytetty yleispätevä koodikieli joka sopii erityisesti web-kehitykseen ja on helposti käytettävissä (X)HTML:n kanssa upotetussa tilassa (Php.net).

PHP-koodi käsitellään www-palvelimella, jonka jälkeen palvelin parsii koodista näytettäväksi vain sillä tuotetun sisällön sivustolle. Jäljelle jäävä sisältö muutetaan HTML-koodiksi, jonka jälkeen se ladataan asiakkaan selaimessa. Asiakas ei siis näe omasta selaimestaan PHP:llä suoritettavia komentoja.

PHP tarjoaa useita hyviä puolia web-kehityksessä. Sen avulla voidaan välttää muun muassa turhaa (X)HTML-koodin toistamista, ja näin vähentää internet-sivustojen luomiseen kulutettavaa aikaa. PHP:n avulla voidaan luoda ja näyttää myös erittäin paljon dynaamista sisältöä, kuten päivämääriä, kellonaikoja, esimerkiksi erilaisia mainoksia perustuen käyttäjän sijaintiin, ja paljon muuta.

3.5.1 PHP:n lyhyt historia

PHP:n ensimmäinen versio oli nimeltään Personal Home Page Tools (PHP Tools) kun Rasmus Lendorf julkaisi sen vuonna 1995. Vaikka se oli kooltaan vain 26 kilobittia, eli pikkuruinen nykypäivän standardien mukaan, saattoivat käyttäjät sisällyttää vieraskirjoja ja muita lomakkeen prosessointisovelluksia sivustoilleen sen avulla. (Ide, 2013.)

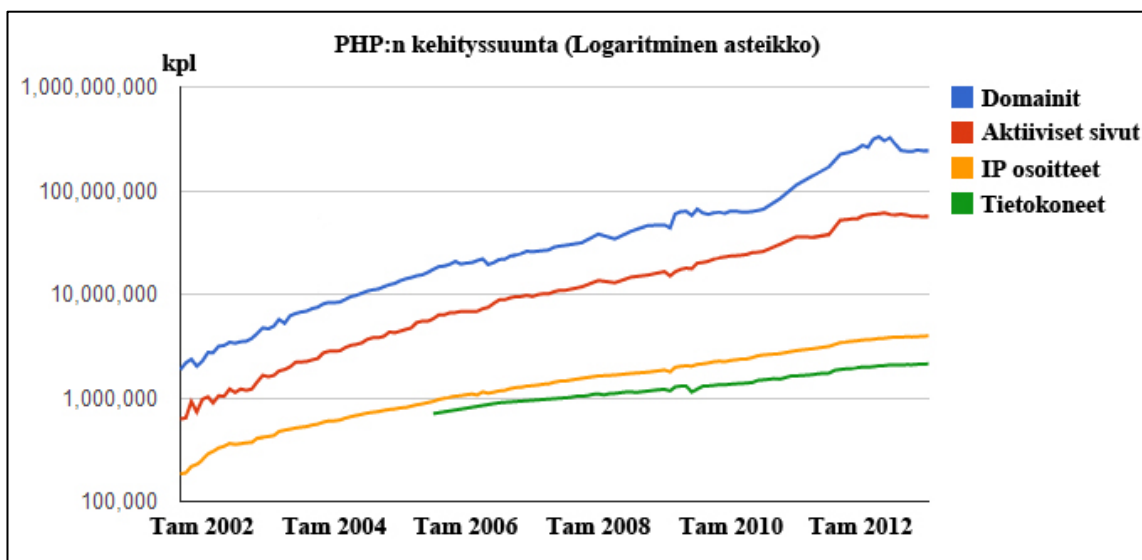
PHP 2:ssa esiteltiin sisäänrakennettu tuki tietokantoihin pääsyyn, evästeiden käsittelyyn ja käyttäjän määrittämiin funktioihin. PHP 2 julkaistiin vuonna 1997, ja seuraavana vuonna noin 1% internet-sivustoista käytti PHP:ta. (Ide, 2013.)

PHP 3 muistutti jo enemmän nykypäivän PHP:ta. Andi Gutmanin ja Zeev Suraskin uudelleenkirjoitus PHP:n perustana olevaan jäsentelyyn johti uudenlaiseen (ohjelmointi)kieleen, jonka jälkeen kieli myös nimettiin pelkäksi PHP:ksi, lyhenne sanasta Hypertext Preprocessor. PHP 3 julkaistiin vuonna 1998, ja kielen laajentamisen helppous houkutti kymmenittäin kehittäjiä työskentelemään ja julkaisemaan erilaisia moduuleita kieleen. (Ide, 2013.)

Gutman ja Suraski jatkoivat PHP:n ydinkoodin uudelleenkirjoittamista parantaakseen suorituskyykyä ja laajentaakseen moduuleista koostuvaa koodikirjastoa. Tämän johdosta luotiin Zend Engine, jota PHP 4:ssä käytettiin kun se julkaistiin vuonna 2000. Syyskuussa 2001 Netcraftin Web Server Survey -tutkimuksessa todettiin 1.8 miljoonan sivuston käyttävän PHP:tä. (Ide, 2013.)

PHP 5 julkaistiin vuonna 2004, ja sen versio 5.4.11 (julkaistu 17. tammikuuta 2013) on uusin käytössä oleva PHP:n versio. Tammikuussa 2013 PHP:tä käytettiin 244 miljoonalla sivustolla, mikä tarkoittaa 39% Netcraftin Web Server Survey -tutkimukseen osallistuneista sivustoista. (Ide, 2013.)

Seuraavassa kuviossa näkyy PHP:n suosion kasvu 10 vuoden ajalta.



KUVIO 21. PHP:n suosion kasvu vuoden 2002 tammikuusta vuoden 2012 tammikuuhun (Ide 2013, muokattu)

3.5.2 Syntaksi

PHP:tä käytetään yleisimmin upotettuna (X)HTML-sivujen sisälle. PHP-koodin aloitus ja lopetustagit ovat "<?php" ja ">". Seuraavassa kuviossa näkyy miten php-koodi merkitään HTML-dokumentin body-osiossa.

```
testi.php
<html>
<head>
    <title> PHP Syntaksi </title>
</head>
<body>
    <?php echo "Hello World!"; ?>
</body>
</html>
```

KUVIO 22. Php-koodin merkintä HTML-dokumentissa

Kun käyttäjän selain lähettää pyynnön yllä olevasta *testi.php* sivusta, palvelin käyttää koodin PHP-tulkilla. Tulkki käsittelee vain PHP:n aloitus ja lopetustagien sisällä olevan

tekstin, jonka jälkeen se palauttaa valmiin sivun palvelimelle, joka lähettää sen edelleen selaimelle. (Wikipedia, PHP.)

PHP:n mahdollisuudet ovat miltei rajattomat, joten PHP:n syvempi esittely on mahdotonta tässä opinnäytetyössä. PHP:llä luotu dynaaminen sisältö antaa sivuston kehittäjälle esimerkiksi mahdollisuuksia palvella sivuston käyttäjiä henkilökohtaisemmin kuin pelkällä staattisella sivulla.

Seuraavassa kuviossa olevassa esimerkissä käyttäjää tervehditään sivustolla eri tavoin kellonajasta riippuen.

```
testi.php
<html>
<head>
    <title> PHP tervehdys </title>
</head>
<body>
<?php
    $t=date("H");
    if ($t<"10")
    {
        echo "Hyvää huomenta!";
    }elseif ($t<"20"){
        echo "Hyvää päivää!";
    }else {
        echo "Hyvää yötä!";
    }
?>
</body>
</html>
```

KUVIO 23. Php:llä toteutettu käyttäjän tervehtiminen kellonajasta riippuen

3.5.3 PHP:n käyttämisen haittapuolet

PHP:stä on tavallaan tullut oman menestyksensä uhri. Koska PHP:tä käytetään miljoonilla palvelimilla ja internetsivuilla, hakkereilla on edessään laaja ja houkutteleva hyökkäysalusta. PHP:n koodaamisen aloitus on erittäin helppoa sen suosion ja laajan

yhteisön avulla, mikä houkuttaa kaikenlaisia koodaajia kokeilemaan kieltä. Varsinkin aloittelevilla PHP-koodareilla ei välttämättä ole tarvittavaa tietoa tai kokemusta PHP:n tietoturvan parantamiseksi. (Ide, 2013.)

Toisaalta suuri ja avulias yhteisö on tukemassa kaikenlaisia koodareita, ja aloittelevalle PHP-koodarille löytyykin rutkasti tietoa koodin haavoittuvuuksien ehkäisemiseksi.

3.6 JavaScript

JavaScript on Netscapen vuonna 1995 Navigator 2 -selaimen mukana julkaisema ohjelmointikieli. Se mahdollisti ensimmäistä kertaa interaktiivisten HTML-dokumenttien tuottamisen. (Linjama, 2001:604.)

JavaScript-kieltä voi kirjoittaa millä tahansa ASCII-tekstieditorilla. Ohjelmakoodi kirjoitetaan *script*-elementin sisään, ja se sijaitsee tavallisesti HTML-dokumentin alussa dokumentin *head*-osiossa. Se voidaan kuitenkin sijoittaa myös dokumentin *body*-osioon. (Linjama, 2001:611.) JavaScript-koodi voidaan kirjoittaa myös ulkoisen JavaScript-tiedostoon, joka haetaan komennolla `<script src="testi.js"></script>`.

JavaScript-kielen mahdollisuudet, samoin kuin PHP:n, ovat miltei rajattomat. Leinosen (2008) mukaan JavaScriptin toimintaidea on luoda dynaamista toiminnallisuutta sivustolle. Sitä ei kuitenkaan pidä sekoittaa dynaamisen sisällön tuottamiseen palvelinpuolella. JavaScript lataa tarvitsemansa tiedot selaimeen, jolloin käyttäjä voi kokea interaktiivista toimintaa ottamatta välillä yhteyttä palvelimeen. Varsinkin JavaScriptin alkutaipaleella selaintuki kielelle oli erittäin puutteellista, mutta kieli on kärsinyt tuen puutteesta koko olemassaolonsa ajan. Nykyäänkin käyttäjä voi halutessaan sulkea JavaScriptin suorituksen selaimestaan, minkä vuoksi JavaScript-koodia ei tulisi käyttää sivuston käytön kannalta olennaisten toimintojen luomiseen. (Leinonen, 2008.)

3.7 Responsiivinen suunnittelu

Korpelan (2012) mukaan responsiivisella suunnittelulla (myös mukautuva suunnittelu) pyritään mukauttamaan internetsivun tai -sovelluksen sisältö, sen rakenne ja asettelu

siten, että se sopii hyvin kullekin laitteelle ja sen normaalille käyttötavalle. Tällöin keskeisintä on laitteen näytön koko.

Verkkosivun tai -palvelun mukauttaminen erikokoisille laitteille on tärkeää hyvän käyttökokemuksen takaamiseksi. Kuluttajilla on todennäköisesti eri tarpeet pöytäkoneella ja läppäriä käytettäessä kuin älypuhelinta käytettäessä. Älypuhelimella ei esimerkiksi välttämättä haluta selailla sivustoa, vaan hakea jotakin tiettyä sisältöä, kuten yhteystietoja tai aukioloaikoja. Pöytäkoneella ja läppäriä taas käytetään älypuhelinta enemmän yleiseen surffailuun. (Malinen, 2013:5.)

3.7.1 Taustaa

Internetin käyttö ei enää, toisin kuin kymmenen vuotta sitten, rajoitu miltei ainoastaan työpöytäkäyttöön. Mobiililaitteet (älypuhelimet, tabletit ja phabletit) kasvattavat suosiotaan, ja internet on yhä enenevässä määrin saavutettavampi. Jo pelkästään Suomessa, joka kansainvälisellä mittapuulla on useissakin muoti-ilmiöissä ajastaan vähintäänkin muutaman vuoden suunnannäyttäjiä jäljessä, kannettavilla laitteilla tapahtuva internetin käyttö on alkaa muodostaa huomattavaa osaa internetin käytöstä kokonaisuudessaan.

Tilastokeskuksen Tieto- ja viestintätekniikan käyttö -tutkimuksen (2011) mukaan väestöstä 26 % käyttää internetiä kannettavalla tietokoneella, ja 33 % käyttää sitä matkapuhelimella kodin ja työpaikan ulkopuolella.

Kasvava mobiilikäyttäjien määrä on laittanut internetsivujen kehittäjät ja suunnittelijat uudenlaisen ongelman eteen; Miten tarjota sivusto, joka toimii ja näyttää hyvältä myös mobiilikäyttäjille? Aiemmin tapana oli tehdä erillinen mobiilisivusto sivustoille, joilla oli suuri mobiilikäyttäjäkanta. Erillisen mobiilisivuston käyttämisellä kuitenkin on haittapuolia, kuten vaativa ylläpitäminen ja mahdollisesti identtinen sisältö varsinaisen sivuston kanssa, jonka hakukoneet, kuten Google, saattavat tulkita tuplasisällöksi. (Lemminki, 2013.)

3.7.2 Responssiivisen suunnittelun ja käytön hyödyt

Responssiivisen suunnittelun merkittäviä hyötyjä on Malisen (2013) mukaan useita. Yksi käyttöliittymä kaikkeen käyttöön madaltaa ylläpito- ja kehityskuluja, mikä merkitsee säästöjä. Kun verkkosivut ovat selkeät ja johdonmukaiset jokaisella laitteella, se lisää myyntiä ja sivuilla on parempi konversio, sillä asiakas löytää helpommin etsimänsä tai haluamansa tiedon. Responssiivisuus takaa paremman hakukonenäkyvyyden verrattuna erilliseen mobiilisivustoon, ja esimerkiksi Google suosittelee responssiivista toteutustapaa. Responssiivisuus antaa lisäksi mahdollisuuden sivuston täyteen saatavuuteen myös mobiilisti, toisin kuin erillinen mobiilisivusto, joka usein on vain "riisuttu" versio varsinaisesta sivustosta. Sivuston käyttökokemusta parantaa myös esimerkiksi se, että erillisen mobiilisivuston nappeihin ja linkkeihin saattaa olla vaikea osua kosketusnäyttöpäätellä, kun taas responssiivinen sivusto on suunniteltu näitä käyttäjiä ajatellen. (Malinen, 2013). Seuraavassa kuviossa esimerkki responssiivisesti toteutetun sivuston näkymästä erikokoisilla näytöillä ja laitteilla.



KUVIO 24. The Boston Globe -uutissivuston näkymä eri päätelaitteilla (Kuva: Arora 2013)

3.7.3 Responsiivinen toteutus

Responsiivinen toteutus tapahtuu HTML 5 -kuvauskieltä ja CSS 3 -tyylimääritteitä. Jo CSS 2.1:stä lähtien tyylitiedostossa on pystytty vastaamaan jossain määrin käyttötarkoituksen tai päätelaitteen asettamiin vaatimuksiin *media* -tarkennuksen avulla. (Marcotte, 2010.) Seuraavassa kuviossa näkyy Marcotten (2010) esimerkki, jossa määritellään sivu käyttämään eri CSS-tyylitiedostoa eri tarkoituksiin.

```
<link rel="stylesheet" type="text/css" href="core.css"
      media="screen" />
<link rel="stylesheet" type="text/css" href="print.css"
      media="print" />
```

KUVIO 25. Marcotten (2010) esimerkki eri css-tiedoston linkittämisestä eri käyttötarkoitusta ajatellen

Esimerkiksi *media="print"* -tarkennus on ensisijaisesti tarkoitettu kuvaamaan sivuston sisällön sivutetussa muodossa ja dokumenteissa, jotka näkyvät ruudulla *tulostuksen esikatselu* tilassa (W3C, Media Types). Useimmat selaimet ja laitteet eivät kuitenkaan koskaan omaksuneet tätä tarkennusta vartenotettavana tapana parantaa sisällön skaalautuvuutta, jättäen monet media-tyypit huonosti toteutetuiksi tai kokonaan huomiotta. W3C kuitenkin sisällytti CSS 3:n määrittelyihin mediarajoituksia (*media queries*). *Media query* mahdollistaa sisällön kohdentamisen päätelaitteen fyysisten ominaisuuksien, kuten näytön leveyden, mukaan. (Marcotte, 2010). Seuraavassa kuviossa Marcotten (2010) esimerkki *media query*n liittämistä *media*-attribuuttiin.

```
<link rel="stylesheet" type="text/css"
      media="screen and (max-device-width: 480px)"
      href="shetland.css" />
```

KUVIO 26. Marcotten (2010) esimerkki *media query*n käytöstä

Ylläolevassa esimerkissä on kaksi osatekijää:

1. Media-tyyppi *screen*, ja
2. varsinainen kysely, joka määrittää tarkasteltavan toiminnon *max-device-width* ja siihen kohdistuvan arvon *480px*.

Toisinsanoen ylläoleva koodi kysyy laitteelta sen horisontaalista enimmäisleveyttä (*max-device-width*), ja onko se yhtäsuuri tai pienempi kuin 480 pikseliä. Jos kyselyn

arvo toteutuu, eli jos katsomme sivua laitteella jonka enimmäisleveys on 480 pikseliä, kuten iPhonella, selain lataa *shetland.css* -tyylitiedoston. Jos kyselyn arvo ei toteudu, linkki ohitetaan. (Marcotte, 2010.)

CSS-koodin soveltaminen rajataan usein sellaisiin ikkunoihin tai laitteisiin, joiden leveys on tietyllä välillä. Tyypillisesti tarvitaan muutama tällainen rakenne, jolloin on selvintä sijoittaa näyttölajikohtaiset koodit omiin CSS-tiedostoihinsa. (Korpela 2012.) Seuraavassa kuviossa Korpelan (2012) esimerkki css-tiedoston liittäminen vain tietyn kokoista näyttöä käytettäessä.

```
<link rel="stylesheet" href="pienehkot.css" media= "only screen and
(min-width: 320px) and (max-width: 640px)">
```

KUVIO 27. Korpelan (2012) esimerkki css-tiedoston käytöstä tietyn kokoiselle näytölle suunnattuna

Kun käytettävä laite saadaan määritettyä media-kyselyn avulla ja kunkin kokoiselle näytölle saadaan ladattua oma tyylinsä, voidaan sivun asettelu muuttaa näytön koon mukaan käyttäjäystävällisimmäksi. Seuraavassa kuviossa mukailen Marcotten (2010) esimerkkejä eri näyttökokoja varten tehdyistä css-koodeista.

```
@media screen and (min-width: 600px) {
    .column {
        float: none;
    }
}

@media screen and (max-width: 600px) {
    .column {
        float: left;
    }
}
```

KUVIO 28. Marcotten (2010) esimerkkejä mukaileva css-koodi

Tyylitiedostoissa kannattaa myös määrittää esimerkiksi sivustolla oleville kuville ja painikkeille erilaiset koot erilaisten näyttöpäätteiden kokojen mukaan. Tällöin voidaan muun muassa välttää painikkeiden huono käytettävyys esimerkiksi mobiililaitetta käytettäessä.

4 SUUNNITELMA LANDING PAGEN TOTEUTUKSESTA

Tein *landing pagesta* kaksi erilaista esimerkkivaihtoehtoa, jotta palvelun valmistuttua *landing pagen* toteuttajalla olisi mahdollisimman hyvät lähtökohdat toteuttaa sivu nopeasti ja pienehköllä vaivalla. Sivun suunnitelmat ovat suuntaa antavia, ja toteuttajan tulisikin soveltaa tässä opinnäytetyössä esiteltyjä ohjeita sivun sisällön lopullisen luomisen yhteydessä.

Koska tuotettava verkkopalvelu Kojako Galleria on huomattavan vaikeaselkoinen konsepti selittää henkilölle jolla ei ole palvelusta taustatietoa, mielestäni *landing pagella* on parasta keskittyä palvelun esittelyyn jonkinlaisen demon muodossa.

Kummassakaan vaihtoehtoista sivulle ei ole tehty eri välilehtiä, vaan kaikki tarjottava informaatio on sijoitettu yhdelle sivulle. Tekstin asettelu on tarjottavan informaation määrästä riippuvainen. Jos sivulle tulee esimerkiksi paljon tekstiä jonkin otsikon alle, voidaan tarvittaessa hyödyntää *collapsing div*-tekniikkaa, jossa teksti sijoitetaan eri *div*-elementteihin, ja niistä näkyvät esimerkiksi pelkät otsikot. Otsikkoa painettaessa *div*-elementti aukeaa paljastaen sen sisältämän tekstin. Sivulla olevan datan määrä vaikuttaa hakukonenäkyvyyteen. *Collapsing div*-tekniikkaa käytetään muun muassa sen vuoksi, että siten saadaan sijoitettua sivulle suuri määrä dataa, mutta samalla pystytään säilyttämään pelkistetty ja yksinkertainen ulkoasu.

4.1 Kojako Galleria esimerkki 1

Sivulla on hyvä antaa jotain informaatiota palvelusta, jotta saataisiin sivulla vierailija jäämään sivulle sen sijaan, että hän vain poistuisi sivulta sen enempää ajattelematta. Seuraavassa kuviossa näkyy esimerkkisuunnitelma Kojako Gallerialle suunnittelemani *landing pagesta*. Sivua suunnitellessani kiinnitin erityisesti huomiota sivun selkeään rakenteeseen, ja siihen että palvelun mielestäni selkeä käyttöliittymä tulee selvästi esille. Seuraavassa kuviossa on esiteltynä suunnittelemani ulkoasu.



KUVIO 29. Esimerkkisuunnitelma Kojako Gallerian landing pagen ulkoasusta

4.1.1 Rakenne

Sivun rakenne on yksinkertainen. Ylhäällä olevassa vihreässä palkissa on vain tuotteen nimi. Sivun vasemmassa reunassa on mainoslause, ja tietoa tuotteesta. Esimerkissä käytin *Lorem ipsumia* näyttääkseni, miltä sivu voisi näyttää.

Sijoitin *CTA*-painikkeen palvelun esittelytekstin alapuolelle, ja koska sivulle ei tietääkseni ole tulossa tietoa palvelusta muutamia lauseita enempää, en kokenut tarpeelliseksi tehdä sivulle useampia *CTA*-painikkeita. Jos sivulle kuitenkin tehdään enemmän sisältöä, tulee miettiä useampien *CTA*-painikkeiden lisäämistä sivun eri kohtiin. Toinen vaihtoehto useammille *CTA*-painikkeille on sijoittaa painike sellaiseen kohtaan, jossa se on aina näkyvässä, myös sivua vieritettäessä alaspäin, kuten Kuvio 2:ssa näkyvässä Tinyletlerin sivussa.

Sivun oikealla puolella näkyy palvelun käyttöliittymä tietokoneen näytöllä ja älypuhelimella. Mielestäni useampien näyttökokojen hyödyntäminen esittelyssä on tärkeää, jotta mahdollinen asiakas saa heti käsityksen siitä, että palvelu toimii hyvin myös mobiilisti.

Suomen lain määräämät rekisteriselosteet ja muu lakiteksti voidaan sijoittaa sivun alaosassa sijaitsevaan tummempaan osioon, sillä tavanomaista käyttäjää tiedot tuskin kiinnostavat, tai ne eivät ole hänelle olennaisia.

4.2 Kojako Galleria esimerkki 2

Seuraavassa kuviossa esitellään Kojako Gallerialle suunnittelemani vaihtoehtoinen ulkoasu. Tässä esimerkissä on käytetty hieman eri värimaailmaa esimerkin vuoksi. Myös sivun rakenne poikkeaa esimerkistä 1.



KUVIO 30. Vaihtoehtoinen ulkoasusuunnitelma Kojako Gallerialle

4.2.1 Rakenne

käyttäjän huomio kiinnitetään vihreällä isolla otsikolla. Otsikon alla on toisen tason otsikko, jossa kerrotaan palvelusta. Tämän jälkeen seuraa pieni esittelyteksti, jossa kerrotaan tuotteen tarkoitus ja hyödyt asiakkaalle.

Sivun oikeassa reunassa on kuva palvelun käyttöliittymästä kuten ensimmäisessäkin vaihtoehdossa. Tässä vaihtoehdossa CTA-painike on sijoitettu melko keskelle sivua, kuitenkin niin että se näkyy sivulle tultaessa ilman että sivua joutuu vierittämään alaspäin sen nähdäkseen.

Sivun alaosa on jaettu esimerkissä kahteen sarakkeeseen, mutta tarvittaessa niitä voi olla useampi. Sarakkeiden sisältämästä tekstin määrästä riippuen siinä ei tarvitse olla kokonaista tekstiä, vaan tekstin voi osioida siten, että vain tekstin alkuosa on näkyvillä. Käyttäjä voi kuitenkin painaa kohdasta *lue lisää*, jolloin koko teksti tulee näkyviin.

Tässä esimerkissä on myös tehty suunnitelma *landing page*n näkymästä katsottaessa sitä pienemmällä näytöllä, kuten mobiililaitteella. Mobiilinäkymä on esitelty seuraavassa kuviossa. Mobiilinäkymässä kuvat eivät näy sivun käyttäjille, vaan ne piilotetaan CSS-tyylitiedoston *media query*n avulla.



KUVIO 31. Mobiilikäyttäjälle näkyvä näkymä Kojako Gallerian responsiivisesta ulkoasusta

LÄHTEET

Arora, D. 2013. Responsive Design vs. Mobile Sites vs. Native Mobile Apps: Which One's the Best?. [online]. Julkaistu 19.2.2013. Luettu 11.9.2013. <http://vandelaydesign.com/blog/web-development/responsive-mobile-native/>.

DeMers, J. [How To] Create Landing Pages that Convert & Rank Well in Search. [online]. N.d. Luettu 11.9.2013. <http://unbounce.com/landing-pages/that-convert-and-rank-well/>.

Formstack, Anatomy of a perfect landing page. Anatomy of a Perfect Landing Page. [online]. N.d. Luettu 14.7.2013. <http://www.formstack.com/the-anatomy-of-a-perfect-landing-page>.

Gardner, O. 101 Landing Page Optimization Tips. [online]. N.d. Luettu 10.9.2013. <http://unbounce.com/101-landing-page-optimization-tips/>.

Ide, A. 2013. PHP just grows and grows. [online]. Julkaistu 31.1.2013. Luettu 28.8.2013. <http://news.netcraft.com/archives/2013/01/31/php-just-grows-grows.html>.

Korpela, J. 2012. Responsiivinen suunnittelu. [online]. Julkaistu 2.8.2012. Luettu 5.9.2013. <http://html5kirja.fi/2012/08/02/responsiivinen-suunnittelu/>.

Leinonen, J. 2008. Www-koulutussivuston kehittäminen Case Inspecta Oy. Tietojenkäsittelyn koulutusohjelma. Tampereen ammattikorkeakoulu. Opinnäytetyö.

Lemminki, R-M. 2013. Responsiiviset verkkosivut korvaavat mobiilisivut. [online]. Julkaistu 19.4.2013. Luettu 5.9.2013. <http://www.jm-emotion.fi/blogi/2-blogi/117-responsiiviset-verkkosivut-korvaavat-mobiilisivut>.

Linjama, T. 2001. XHTML. Jyväskylä: Docendo Finland Oy.

Longanecker, C. 2012. Design is Marketing. [online]. Julkaistu 16.2.2012. Luettu 11.9.2013. <http://www.quicksprout.com/2012/02/16/design-is-marketing/>.

Malinen, T. 2013. Responsiivinen suunnittelu - Varmista, että verkkopalvelusi toimii myös mobiilisti. [online]. Julkaistu 2013. Luettu 5.9.2013. http://www.sofokus.com/wp-content/uploads/sofokus_responsiivinen_suunnittelu_pikaopas-web.pdf.

Marcotte, E. 2010. Responsive web design. [online]. Julkaistu 25.5.2010. Luettu 5.9.2013. <http://alistapart.com/article/responsive-web-design>.

Martin, N. How To Identify And Stop Emotion Based Conversion Killers. [online]. N.d. Luettu 11.9.2013. <http://blog.kissmetrics.com/emotion-based-conversion-killers/>.

Passiak, D.

Seven Steps to Selling SaaS like Socrates – it's About Solutions, Not Products!.

[online]. N.d. Luettu 11.9.2013. <http://blog.kissmetrics.com/selling-saas-like-socrates/>.

Paulin-Daigle, R. Checkout Champions: 10 Tips To Win More Ecommerce Sales. [online]. N.d. Luettu 11.9.2013. <http://blog.kissmetrics.com/win-more-ecommerce-sales/>.

Php.net. What is php?. [online]. Päivitetty 28.8.2013. Luettu 28.8.2013. <http://www.php.net/>.

Stockus, S. 2007. WWW-sivuston toteutussuunnitelma. Tietojenkäsittelyn koulutusohjelma. Tampereen ammattikorkeakoulu. Opinnäytetyö.

Suomen virallinen tilasto (SVT): Väestön tieto- ja viestintätekniikan käyttö. 2011. Internetin mobiilikäyttö ja mobiilin käytön useus kannettavilla laitteilla iän, toiminnan, koulutusasteen, asuinpaikan kaupunkimaisuuden ja sukupuolen mukaan 2011, %-osuus väestöstä. [online]. Päivitetty 14.6.2012. Luettu 5.9.2013. http://tilastokeskus.fi/til/sutivi/2011/sutivi_2011_2011-11-02_tau_006_fi.html.

Unbounce, What is a landing page?. 2013. What is a landing page?. [online]. Luettu 14.7.2013. <http://unbounce.com/landing-page-articles/what-is-a-landing-page/>.

Wikipedia, Conversion rate. 2013. Conversion rate. [online]. Päivitetty 11.5.2013. Luettu 14.7.2013. http://en.wikipedia.org/wiki/Conversion_rate.

Wikipedia, Landing page. 2013. Landing page. [online]. Päivitetty 25.6.2013. Luettu 27.6.2013. http://en.wikipedia.org/wiki/Landing_page.

Wikipedia, PHP. 2013. PHP. [online]. Päivitetty 15.3.2013. Luettu 28.6.2013. <http://fi.wikipedia.org/wiki/PHP>.

Wikipedia, XML. 2013. XML. [online]. Päivitetty 7.3.2013. Luettu 1.8.2013. <http://fi.wikipedia.org/wiki/XML>.

W3C. Extensible Markup Language (XML). [online]. Päivitetty 16.6.2013. Luettu 1.8.2013. <http://www.w3.org/XML/>.

W3C. Media types. [online]. N.d. Luettu 5.9.2013. <http://www.w3.org/TR/CSS21/media.html#media-types>.