



Karelia-ammattikorkeakoulu
Tietojenkäsittely, Tradenomi (AMK)

Verkkosivut yritykselle SL Health

Joni Laakkonen

Opinnäytetyö, joulukuu 2023

www.karelia.fi



OPINNÄYTETYÖ
joulukuu 2023
Tietojenkäsittely, tradenomi koulutus

Tikkarinne 9
80200 JOENSUU
+358 13 260 600 (vaihde)

Tekijä(t)
Joni Laakkonen

Nimeke
Verkkosivut yritykselle SL Health

Sanna Laakkonen
SL Health

Tiivistelmä

Tämän opinnäytetyön aiheena on luoda uudet verkkosivut joensuulaiselle pienyritykselle nimeltä SL Health Oy. Nettisivujen on tarkoitus korvata aiemmin käytössä olleet verkkosivut, jotka ovat vanhentuneet ja tarvitsevat uutta ulkoasua, joka paremmin mukailee yrityksen brändiä. Verkkosivuille oli myös tarpeellista saada ajanvarausjärjestelmä nopeuttamaan yrityksen toimintaa.

Työssä perehdytään aluksi modernien verkkosivujen luomiseen teoreettisesti. Teoriassa tutustutaan käyttökokemukseen yleisellä tasolla, sekä käydään läpi eri tapoja parantaa ulkoasua. Työssä kerrotaan myös työkaluista ja teknologioista, joiden avulla toteutus tapahtui. Yhdistämällä Visual Studio Code ja React-kirjaston mahdollisuuksia, verkkosivujen luominen nopeutuu. Ajanvarausjärjestelmäksi valikoitui Vellon-ajanvarausjärjestelmä, jonka sai upotettua suoraan verkkosivuille. Seuraavaksi verkkosivujen ulkoasusta luotiin suunnitelmat. Niiden pohjalta työn varsinainen toiminnallinen osuus lähti käyntiin.

Ongelmilta työssä ei vältytty, mutta haluttuun lopputulokseen lopulta päästiin. Lopuksi työssä pohditaan työn kulkua, sekä sen aikana heränneitä mietteitä.

Kieli
suomi

Sivuja 35
Liitteet 0
Liitesivumäärä 0

Asiasanat
verkkosivut, nykyaikaiset verkkosivut, responsiivisuus, verkkosivujen ulkoasu



THESIS
December 2023
Degree Programme in Bachelor of Information Technology

Tikkarinne 9
80200 JOENSUU
FINLAND
+ 358 13 260 600 (switchboard)

Author (s)
Joni Laakkonen

Title
Website for company called SL Health

Sanna Laakkonen
SL Health

Abstract

The subject of this thesis is to create a new website for a small company in Joensuu called SL Health Oy. The purpose of the website is to replace the previously used web pages, which are outdated and need a new look that better reflects the company's brand. It was also necessary to get an appointment booking system on the website, to improve the performance of the company.

The thesis starts with a theoretical introduction to the creation of a modern website. The theory explains the user experience and reviews different ways to improve the look and feel of websites. The tools and technology used to make this happen are presented next. By combining Visual Studio Code with the potential of the React library, the creation of web pages can be accelerated. The chosen booking system was a Vello booking system. It allowed to embed it directly into the website. The planning of the websites' appearance was done next. The actual operational part of the work was launched based on these plans.

Some problems did emerge during the development progress, but the desired result was eventually achieved. Finally, the thesis recaps and reflect on the working process and thoughts that emerged.

Language
Finnish

Pages 35
Appendices 0
Pages of Appendices 0

Keywords

Website, Modern Website, Responsive Design, Website Design

Sisältö

1	Johdanto	1
2	Verkkosivun ulkoasu ja käyttökokemus osana liiketoimintaa	2
2.1	Käyttökokemus	2
2.2	Responsiivisuus.....	4
2.3	Verkkosivujen väritys	8
2.4	Kuvien käyttö verkkosivuilla	9
2.5	Kirjaisinten huomiointi verkkosivuilla.....	10
3	Teknologiat ja työkalut	11
3.1	HTML ja CSS.....	11
3.2	JavaScript.....	12
3.3	Visual Studio Code	13
3.4	React.js.....	15
3.5	React Router.....	15
4	Ajanvarausjärjestelmä.....	16
5	Toteutus.....	17
5.1	Suunnitteluvaihe:	18
5.2	Kehitysvaihe	20
5.3	Testaus ja käyttöönotto:.....	32
6	Pohdinta.....	33
	Lähteet.....	36

1 Johdanto

Nykypäivänä verkosta voi löytää yli 1,88 miljardia verkkosivua. Tämä tarkoittaa, että erottuakseen joukosta verkkosivujen tulee olla ajantasaiset, toimivat, käyttäjäystävälliset sekä erottua joukosta. (Sernoff 2022.)

Opinnäytetyön aiheena on luoda uudet ja toimivat verkkosivut terveys- ja hyvinvointialalla toimivalle yritykselle. Toimeksiantaja on todennut verkkosivujen olevan päivityksen ja uuden ilmeen tarpeessa. Verkkosivuilla on myös ollut pitkään haaveena saada ajanvarausjärjestelmä, jotta toimeksiantaja säästäisi aikaansa yritykselle välttämättömiltä ajanvarauspyynnöiltä, jotka ovat tähän mennessä tapahtuneet sähköpostitse ja puhelimitse.

Olen tutkinut useita opinnäytetöitä muun muassa omaa kirjoitustyyliäni etsiessä ja inspiraatiota hakiessa. Niissä sivustot on usein rakennettu hyödyntäen Wixiä tai WordPressiä (Kämäräinen 2019). Nämä ovat verkkosivujen rakennusjärjestelmiä, joilla voi helposti ja nopeasti luoda näyttäviäkin sivustoja käyttämällä valmiita komponentteja (Jukarainen 2020). WordPress kävi monesti mielessä suunnitellessani, mutta halusin haastaa itseni oppimaan enemmän verkkosivujen toiminnallisuudesta kooditasolla.

Raportissa käydään ensin läpi teoreettista pohjatyötä opinnäytetyölle. Teoria pohjassa kerrotaan teknologioista ja työkaluista, joita on hyödynnetty työssä. Teoriaosiossa käsitellään myös tapoja, joilla verkkosivuista saadaan nykyaikaiset ja käyttäjäystävälliset.

Lopuksi raportissa kerrotaan työn toiminnallisen osuuden etenemisestä vaiheittain, minkä jälkeen pohdintaosiossa käydään läpi työtä yleisellä tasolla. Toiminnallinen osuus on jaoteltu kolmeen osaan, ja jokaista vaihetta on käsitelty tarkasti tekijän näkökulmasta, kuvitusta apuna hyödyntäen.

2 Verkkosivun ulkoasu ja käyttökokemus osana liiketoimintaa

Verkkosivut ovat yksinkertaisimmillaankin monimutkaisia teknologisia luomuksia. Jos jotain odottamatonta tapahtuu, käyttäjillä on tapana syyttää siitä itseään. Kuitenkaan ei ole ikinä heidän syytänsä, etteivät verkkosivut toimi oikein. (Garret 2011a.)

Pohjimmiltaan jokaisessa tapauksessa verkkosivut ovat itsepalvelutuotteita. Yksittäisiin sivuihin harvoin löytää mitään ohjeita tai webinaareja, joissa ohjeistetaan, kuinka verkkosivuilla toimitaan ja mitä sieltä löytyy. Tästä syystä huonosti suunniteltu verkkosivu voi johtaa käyttäjän turhautumiseen ja pahimmassa tapauksessa asiakkaan menettämiseen. (Garret 2011b.)

Verkkosivujen ulkoasun suunnittelua voi monessa tapauksessa verrata sisustus suunnitteluun. Molemmissa pyritään luomaan tyyliä, etsimällä elementtien välistä tasapainoa ja harmoniaa. Molemmissa on omia hetkellisiä villityksiä ja ajattomia tyyli valintoja. Viimeisteleättömät verkkosivut ovat kuin tylsä huone. Huone voi olla toiminnallinen ja ajaa asiansa, mutta kuukaan ei viihdy siellä ilman tarvetta. (Beaird & George 2014a.)

Ulkoasu (user interface eli UI) ja käyttökokemus (user experience eli UX) usein sekoitetaan keskenään. Kuitenkin niillä on tärkeä ja selkeä ero. Käyttökokemuksen voisi jakaa vielä useampaan osaan, tuotteen käyttökokemukseen (product experience eli PX) ja käyttökokemuksen parantamiseen (experience transformation eli XT). (Kantamneni 2022a.) Tässä luvussa perehdytään paremmin käyttökokemukseen yleisellä tasolla sekä käydään läpi eri tapoja parantaa ulkoasua. Käyttökokemuksen parantamiseen (XT), perehdytään myöhemmin, Pohdintaluvussa.

2.1 Käyttökokemus

Tässä luvussa käyn läpi mitä käyttökokemus on, sen merkitystä, kuinka sitä voi ottaa huomioon ja kuinka kohdistaa käyttökokemusta. Seuraavassa kappaleessa käydään läpi arkinen tilanne huonosta käyttökokemuksesta. Tämän tarkoitus on antaa käsitys käyttökokemuksen merkityksestä.

Huonoa käyttökokemusta havainnollistaa esimerkki, jossa henkilö laittaa aamulla laittaa kahvinkeitin päälle ja huomaa 10 minuutin päästä, ettei sieltä olekaan valunut vielä mitään. Ei ole käyttäjän syytä, ettei kahvia tullut, varsinkin jos kahvinkeitinissä ei ole käynnistymisen merkiksi mitään merkkivaloa tai ääntä. Lisäksi nappia pitäisi painaa tasan 3 sekuntia käynnistääkseen prosessin. (Garret 2011c.)

Käyttäjäkokeista suunnittelussa on hyvä lähteä liikkeelle tiedostamalla mitä se on. Käyttökokemus ylettyy miellyttävän ulkoasun ja toiminnallisuuden ulkopuolelle. Ulkoasun suunnittelija pitää huolen, että kahvinkeitin käynnistysnappi näyttää ja tuntuu miellyttävälle. Toiminnallisuuden suunnittelija pitää huolen, että nappia painamalla tapahtuu haluttu asia. Käyttökokemuksen suunnittelija taas joutuu miettimään esimerkiksi napin koon ja merkityksen suhdetta, napin paikan ja merkityksen suhdetta, ja saako käyttäjä tarpeeksi tietoa napin painamisesta. (Garret 2011d.)

Käyttäjän toimista informoiminen on tärkeää. Se on osa käyttökokemusta. Verkkosivujen käyttökokemus on usein paljon kahvinkeitintä monimukaisempi. Sitä parantaakseen kuuluu ymmärtää käyttäjiä ja heidän toimintaansa, miksi he tekevät mitään päätöksiä ja suunnitella ohjelmistot niiden perusteella. Suunnittelussa käyttäjän tarpeet ja kipupisteet huomioidaan. Kipupisteillä tarkoitetaan tässä tapauksessa tilanteita, joissa käyttäjät kohtaavat haasteita suoriutua tai ymmärtää. (Kantamneni 2022b.)

Käyttäjien tarpeet ja halut saadaan parhaiten selville ajan myötä, esimerkiksi palautteista ja käyttäjäkokeista koskevista kyselyistä kerätystä datasta. Ilman tällaista dataa kuitenkin voidaan suunnitteluun joko olettaa tietoa tai tarvittavaa dataa käyttäjäryhmien perusteella tai voidaan myös kerätä esimerkiksi kyselyillä tai vaikka puhumalla loppukäyttäjille kasvotusten. (Kantamneni 2022c.)

Käyttäjäryhmät määritellään usein väestörakenteellisten piirteiden, kuten iän, sukupuolen, palkan ja muiden vastaavien piirteiden perusteella. Myös psykograafisten, eli mieltymysten, asenteen ja näkemysten perusteella voidaan jakaa käyttäjät eri ryhmiin. Nämä piirteet usein myös korreloivat monia väestörakenteellisia piirteitä. Saman ikäiset, samalta alueelta ja samalla palkkatasolla olevat

henkilöt usein omaavat yhteisiä näkemyksiä ja asenteita. Kuitenkin samoja väestörakenteellisia piirteitä omaavat henkilöt saattavat erota toisistaan paljonkin psykograafisella tasolla. Tästä syystä on usein käytännöllisempää jakaa käyttäjäryhmiä psykograafisten piirteiden perusteella. (Garret 2011e.)

Käyttäjäryhmien määrittäminen auttaa ymmärtämään käyttäjien tottumuksia ja ymmärrystä vastaavista tuotteesta, sekä millaisia tarpeita ja vaatimuksia käyttäjillä on tuotteen käyttämistä kohtaan. Joskus on kuitenkin tilanteita, kun käyttäjäryhmien tarpeet eroavat toisistaan enemmän kuin on mahdollista ottaa huomioon. Tällöin ei molempien tarpeita voi yksinkertaisilla ratkaisuilla tyydyttää. Täytyy keskittyä luomaan käyttökokemus vain toisen käyttäjäryhmän ympärille tai tarvittaessa luoda kaksi erillistä ympäristöratkaisua kummallekin käyttäjäryhmälle. (Garret 2011f.)

Käyttökokemus on tärkeä huomioida yrityksen digitalisoimisessa, mutta myös yrityksen kasvua suunnitellessa. Se voi olla merkitsevä tekijä yrityksen menestyksessä, sillä usein asiakkaalla on mahdollista itse valita palvelu ja sen tarjoaja. (Kantamneni 2022d.)

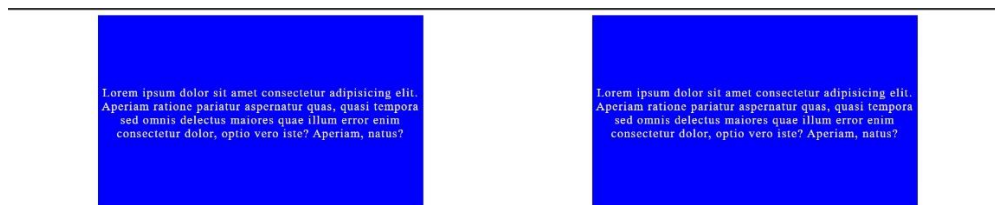
2.2 Responsiivisuus

Responsiivisuus on käsite, joka lähti liikkeelle käyttäjien lisääntyneestä verkkosurffailusta eri laitteilta. Mobiililaitteet ovat usein rajoittuneita näytön koon suhteen ja uudemmat suuret televisiot tai 4k näytöt skaalaavat verkkosivut kaikki hieman eri tavoilla (kuva 1). Responsiivisilla verkkosivuilla nämä asiat on otettu huomioon ja käyttökokemus optimoidaan kaikille laitteille sopivaksi. (LePage & Anrew 2022.)

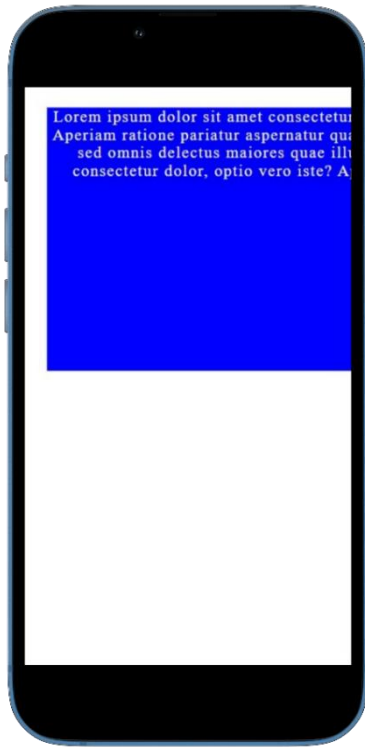


Kuva 1. Responsiivinen verkkosivu esitys (Velázquez 2023.)

Verkkosivuja voi usein selata vain alaspäin, jos on tarvetta. Se on vain ajan myötä muodostunut käytäntö. Useaan suuntaan jatkuvat sivut ovat vaikeasi hahmotettavissa ja monesti käyttäjiltä jääkin helposti jotain näkemättä. Monesti tietokoneen näytölle helposti mahtuvat vierekkäiset laatikot (kuva 2) tai isommat kuvat eivät mahdu mobiililaitteen pienelle näytölle (kuva 3). (LePage & Andrew 2022.)



Kuva 2. Kaksi laatikkoa aseteltuna vierekkäin: Tietokoneen näyttö.



Kuva 3. Kaksi laatikkoa aseteltuna vierekkäin: Mobiililaitteen näyttö



Kuva 4. Kaksi laatikkoa aseteltuna allekkain: Mobiililaitteen näyttö

Yllä mainituissa tapauksissa responsiivisesta asettelusta on hyötyä. Responsiiviset verkkosivut saadaan toimimaan niin, että tietokoneen näytöllä selkeästi näkyvät vierekkäiset laatikot näkyvät mobiililaitteilla allekkain ja liian suuri kuva skaalautuu pienemmäksi (kuva 4). Kuvan mukaista asettelua voidaan toteuttaa löytämällä verkkosivun elementeistä katkaisukohtia. Katkaisukohtien avulla voidaan nettisivua pilkkoa pienemmäksi, eri näyttökokojen perusteella. (LePage & Andrew 2022.)

Teksti on monesti iso osa sivua. Tekstiä ollessa enemmän on mietittävä responsiivisuutta senkin puolesta. Luettavuusteorian mukaan ihanteellisen sarakkeen tulisi sisältää 70–80 merkkiä rivillä. Mobiililaitteilla teksti on kuitenkin suurempaa, jotta teksti näkyy paremmin pienemmältä näytöltä ja tällöin on hyvä tasata

teksti lyhemmin välein ja mahdollisesti lyhentää tekstiä sopivan pituiseksi. (LePage & Andrew 2022.)

Verkkosivujen optimoiminen eri näytöille vaatii meta viewport -tunnisteen määrittämisen HTML-tiedoston head-tunnisteen sisälle. Meta-tunniste (kuva 5) kertoo selaimelle, miten verkkosivu tulee skaalautua, esimerkiksi ohjeistamalla sivun vastaamaan näytön leveyttä laitteesta riippumattomina pikseleinä. (LePage & Andrew 2022.)

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
```

Kuva 5. Viewport-niminen metaelementti kertoo selaimelle, että sivusto tulee skaalata laitteesta riippumattomina pikseleinä.

ChatGPT vastasi kysymykseen, mitä avaintekijöitä responsiivinen verkkosivun suunnittelussa tulee ottaa huomioon. ChatGPT luetteli seuraavat avaintekijät:

1. Joustavat ruudukot ja asettelut: Käytä joustavia ruudukkoja ja asetteluja, jotka mukautuvat laitteen ja näytön kokoon ja tarjoavat yhtenäisen käyttökokemuksen kaikilla laitteilla.
2. Joustavat kuvat ja media: Käytä kuvia ja mediaa, jotka voidaan skaalata laitteen näytön koon mukaan ilman, että laatu heikkenee tai vääristyy.
3. Responsiivinen typografia: Käytä typografiaa, joka mukautuu näytön kokoon ja tarjoaa optimaalisen luettavuuden ja luettavuuden kaikilla laitteilla.
4. Mobiiliystävällinen navigointi: Tarjoa navigointi, jota on helppo käyttää mobiililaitteella, kuten hampurilaisvalikko tai yksinkertaistettu navigointipalkki.
5. Nopeat latausajat: Varmista, että verkkosivustosi latautuu nopeasti kaikilla laitteilla optimoimalla kuvat, pakkaamalla tiedostot ja minimoimalla HTTP-pyyntö.
6. Käyttäjätystävälliset lomakkeet: Varmista, että verkkosivustosi lomakkeet on optimoitu mobiililaitteille ja että niissä on helppokäyttöiset syöttökentät ja selkeät kehotukset toimia.

7. Saavutettavuus: Varmista, että verkkosivustosi on kaikkien käyttäjien käytettävissä laitteesta tai kyvyistä riippumatta, käyttämällä sopivaa kontrastia, tekstikokoja ja kuvien alt-tekstejä.” (ChatGPT 2023a.)

2.3 Verkkosivujen värit

SL Healthilla on valmiina väripaletti, jota verkkosivuilla voisi hyödyntää. Väripaletti sisältää useita hyviä värejä, jotka liittyvät logoon. Väripaletti sisältää kuitenkin montaa eri väriä, ja verkkosivusta tulisi nopeasti sotkuinen tai tylsä tai herättäisi vääränlaisen kuvan, jos värejä käytettäisiin ajattelemattomasti verkkosivun eri elementeissä. (Beaird & George 2014b.)

Käyttöliittymä on kaiken tiedon lähde, johon käyttäjä koskettaa käyttäessään verkkosivua. Tästä näkökulmasta tärkein ja nopein tapa herättää käyttäjän kiinnostus on käyttää huomiota herättäviä värejä halutulla tavalla. Värit tarjoavat valtavia viestintämahdollisuuksia ja ne ovat maailmanlaajuinen, visuaaliseen viestintään perustuva kieli. (Li, Li & Jae 2020, sivu 1.)

Värejä voi miettiä myös psykologisesta näkökulmasta. Tällaista lähestymistapaa kutsutaan väripsykologiaksi. Väripsykologia on tieteenala, joka tutkii värien ja väriyhdistelmien vaikutusta tunteisiin ja käyttäytymiseen. Esimerkiksi kaupallisen verkkosivun ylläpitäjä voi haluta tietää, mitä värejä hyödyntämällä asiakas saadaan kuluttamaan eniten rahaa. Ravintolayrittäjät puolestaan voivat haluta tietää, mikä väri korostaa asiakkaan näläntunnetta. (Beaird & George 2014c.)

Kun ChatGPT-tekoälyltä kysyttiin kysymys, kuinka valita oikeat värit verkkosivuille ja millä suhteilla ne tulisi valita, vastasi tekoäly vastaavasti: ” Verkkosivuston värejä valittaessa on tärkeää ottaa huomioon verkkosivuston tarkoitus, kohdeyleisö ja brändi-identiteetti tai tyyliopas”. ChatGPT antoi myös muutamia vinkkejä, joita pitää mielessä värejä miettiessä. Väripaletin tulisi olla visuaalisesti houkutteleva ja käyttöyhteyteen sopia. Värejä voi käyttää hierarkian ja kontrastin luomiseen. Tämän avulla voidaan ohjata käyttäjän huomiota tärkeisiin elementteihin ja helpottaa verkkosivuilla navigointia. Värejä valitessa tulisi myös huomioida, että tekstit ovat luettavissa taustaväriä vasten, ja että kontrasti on riittävä. (ChatGPT 2023b.)

Lisäksi ChatGPT muistutti käyttämään värejä säästeliäästi ja niin, että ne tukevat brändi-identiteettiä, mikäli verkkosivun luodaan jollekin brändätylle kohteelle. Tämä auttaa luomaan yhtenäisen brändi kokemuksen kaikissa kosketuspisteissä. Värisuhteista ChatGPT kehotti pitämään kontrastin mielessä. Tekoäly painotti, että kontrastisuhteen sen tulisi olla vähintään 4,5:1 runkotekstiä ja 3:1 isossa tekstissä tai otsikoissa. ChatGPT myös vinkkasi muutamia verkkotyökaluja, joiden avulla kahden värin välisen kontrastisuhteen voi tarkistaa. Näihin työkaluihin lukeutuivat muun muassa, WebAIM Contrast Checker, Adobe Color tai Coolers. (ChatGPT 2023b.)

2.4 Kuvien käyttö verkkosivuilla

Yksi kuva on 1000 sanan arvoinen, sanonta ei ole liioittelua. Kuvat ovat erinomainen tapa tuoda tunnelmaa verkkosivuille. Kuvat myös auttavat käyttäjiä muistamaan paremmin, millä sivuilla he ovat käyneet. Verkkosivuilla kuvia voi käyttää esimerkiksi tukeakseen jotain selitystä, parantaakseen verkkosivun ulkoasua, mainoksissa tai tuotteen kuvauksessa. (Butler 2018.)

Verkkosivuilla kuvia valitessa tulee miettiä mitä tarvitsee ja millaiseen käyttöön kuva tulee. Kuvien tulee olla hyvälaatuisia, selkeitä ja tarkoituksenmukaisia. Jos ravintoloissa esitetyt annosten tai reseptien viitekuvat eivät onnistu luomaan asiakkaalle näläntunnetta tai halua maistaa ruokaa, asiakas todennäköisesti ei kokeile kyseistä ruokaa. (Beaird & George 2014d.)

Kuvan voi asettaa suoraan sellaisenaan verkkosivuille ja toivoa parasta. Usein kuvasta saa kuitenkin tarpeisiin soveliaamman hieman kohdistamalla, rajaamalla tai kontrastia säätämällä. Kohdistamalla ja rajaamalla kuvan, saadaan kuvasta esille vain tarpeellinen osa. Kontrastia säätämällä värejä voidaan korostaa ja esimerkiksi tummentaa taustaa, jolloin haluttu elementti erottuu paremmin kuvassa. (Beaird & George 2014e.)

Kuvia voi myös muokata kuvankäsittelyohjelmilla, jolloin kuvat voidaan asettaa verkkosivuilla käyttövalmiina. Kuvia voi myös löytää valmiiksi muokattuna tai kuvissa voi olla käytetty suodattimia, jotka kuvan ottohetkenä saa kuvaan luotua erilaisia efektejä. Joskus kuitenkin kuvan käsittelyyn riittää yksinkertaisemmat toiminnot. Jos esimerkiksi kuvasta halutaan tummempi, jotta teksti erottuu

paremmin sen päällä tai sen reunoja halutaan pyöristää, voivat yksinkertaiset CSS-komennot riittää. (Beaird & George 2014f.)

Kuvia on monenlaisia, kuten 3D-kuvia, animaatioita, piirroksia ja valokuvia. Niitä voi itse luoda erilaisilla ohjelmilla tai valokuvata itse. Kuvien tuottaminen itse luo mahdollisuuksia saada juuri halutunlainen kuva verkkosivuille. Jos kuitenkin kuvien luominen itse ei onnistu, tai muuten aika rajoittaa tekemistä, saa verkosta ladattua kuvia eri kuvapankkipalveluista. Kuvapankeissa kuvat ovat lähes aina maksullisia ja vain harvoin täysin ilmaisia. Joskus vaikka kuva olisi ilmainen, saattaa sen käyttöoikeuksia olla rajattu niin, ettei sitä voi verkkosivulla käyttää. (Beaird & George 2014g.)

2.5 Kirjaisinten huomiointi verkkosivuilla

Niin kuin kuvat ja värimaailma, myös kirjaisimet eli fontit voivat herättää lukijassa tietynlaisia tunteita. Fontit ovat kirjaimien erilaisia, yhtenäisiä kirjain tyynejä, joita hyödyntämällä tekstistä saadaan visuaalisesti erinäköinen. Tästä syystä myös fonttien käyttöä on hyvä miettiä verkkosivuilla.

Verkossa nykypäivänä on useita enemmän tai vähemmän toisistaan eroavia fontteja. Tyylitellyllä tekstillä voidaan esimerkiksi täydentää kuvia tai korostaa tärkeitä asioita teksteissä. CSS-kielellä voi tehokkaasti muokata tekstiä ja fontteja, niiden painoa tai kokoa. Jos kuitenkin jotain CSS-kielellä ylitsepääsemättöntyä tulee vastaan, voi tekstin muokata haluamansalaiseksi kuvanmuokkausohjelmilla ja tuoda sen verkkosivuille kuvatiedostona. (Beaird & George 2014h.)

Fontteja voi asentaa verkosta. Esimerkiksi <https://fonts.google.com/>-sivustolta voi helposti asentaa haluamansa fontit tai vain etsiä inspiraatioita eri fontteihin. Fontit voi asentaa paketteina ja liittää verkkosivustoon, mutta useat fontit myös tukevat suoraa viittausta CSS- tai HTML-tiedostoon, jolloin niitä ei tarvitse edes ladata. Jotkin laadukkaammat ja viimeistellymmät fontit saattavat vaatia lisen sin, niitä käyttääkseen. Tämä on hyvä tarkistaa ennen fonttien julkista tai kaupallista käyttöä. (Beaird & George 2014i.)

Kaikki selaimet eivät kuitenkaan tue kaikkia fontteja. Vanhemmat selainversiot tukevat usein vain rajoitetusti erilaisia fontteja, ja eri käyttöjärjestelmien tukemat

selaimet saattavat tukea hieman erilaisia fontteja kuin toiset. Tästä syystä fontteja käyttäessä on hyvä viitata vaihtoehtoisiin fontteihin ja käyttää fonttipinoa. Vaihtoehtoiset fontit kirjataan fontteja määrittäessä peräkkäisjärjestykseen ja selain lataa aina ensimmäisen fontin, jonka se tukee. (Beaird & George 2014j.)

3 Teknologiat ja työkalut

Tässä luvussa esitellään työssä käytettyjä teknologioita ja työkaluja. Luvussa selvitetään ensin työssä käytetyt työkalut, sekä verkkosivujen ohjelmoinnissa yleisimmin käytetyt merkintä- ja ohjelmointikielet. Luvussa käydään läpi lyhyesti näiden teknologioiden ja työkalujen asennus ja selitetään myös miksi kyseinen teknologia ja työkalu on valittu työn toteuttamiseen.

Oma verkkosivujen kehittäjä polkuni lähti JavaScriptin opettelusta, josta siirryin React kirjastoon, laajentamaan ymmärrystäni verkkosivuista ja niiden kehityksestä. Tästä syystä ymmärrän hyvin Reactin vahvuuksia ja heikkouksia, sekä osaan hyödyntää sen tehoa kattavasti. Se sisältää paljon React Routerin tapaisia tehokkaita lisäkirjastoja, joiden avulla voi parantaa koodia huomattavasti ilman vaivaa.

Visual Studio Code on yleisesti suosittu ja paljon verkkosivujen kehityksessä käytetty ilmainen API. Käytämme sitä työharjoittelussanikin. Se sisältää paljon muokkaus ominaisuuksia ja lisäosia, joilla kukin saa siitä omanlaisensa. Se sisältää myös version hallinta työkaluja, joilla on esimerkiksi helppo GitLab yhteydellä tallentaa tai peruuttaa tekemiään muutoksia.

3.1 HTML ja CSS

HTML on merkintäkieli, joka tulee sanoista Hypertext Markup Language. Se mahdollistaa verkkosivujen perusrakenteen luomisen hyperlinkkien avulla, tätä myös kutsutaan verkkosivujen rungoksi. HTML-rakennetta on mahdollista parantamalla millä tahansa ohjelmointikielellä. (Popovic 2022a.)

CSS on tyyliomakekieli, joka tulee englannin kielen sanoista cascading style sheets, ja on kehitetty täysin HTML-elementtien esitystavan parantamiseen.

CSS-koodissa viitataan HTML-elementteihin, elementin, nimen tai id:n perusteella ja siten voidaan kohdistaa haluttu muotoilu oikeaan elementtiin. Näin saadaan verkkosivut näyttämään visuaalisesti miellyttäviltä ja moderneilta. (Popovic 2022b.)

Inline CSS, internal CSS ja external CSS ovat kaikki toimivia eri tapoja liittää CSS-muotoilu HTML-koodiin. CSS on kuitenkin kieli, jossa hierarkiat ovat oleellinen osa muotoilua ja vaikuttavat siten koodin toiminnallisuuteen, mahdollisesti jopa vahingossa ohittaen toisella tavalla luodun muotoilun. (Popovic 2022b.)

Inline CSS on tavoista hierarkiassa ylimpänä. Se ohittaa kaikki muut eri tavoilla luodut tyylimuotoilut. Inline CSS -muotoilu tarkoittaa, että haluttu muotoilu määritetään suoraan elementin sisään, style-muuttujassa. Tämä ei kuitenkaan tarkoita, etteikö se olisi CSS-kieltä, vaikka se määritetäänkin HTML-elementin sisältämässä muuttujassa. Keskimäisenä CSS-muotoilutavoista on internal CSS. Näin se ohittaa vain external CSS-muotoilun. Internal CSS -muotoilu asetetaan HTML-koodiin, style-elementin sisään, joka tulee olla head-elementin alla. Alimpana CSS-hierarkiassa on external CSS. Tämän tavan kanssa muotoilu on mahdollisesti kaikista selkeintä. Se tapahtuu omassa .CSS-päätteisessä tiedostossaan, sekoittumatta muuhun koodiin. Tähän tiedostoon viitataan HTML-tiedostossa, Link-elementin sisällä ja head-elementin alla. (Popovic 2022b.)

HTML ja CSS on tehokas ja välttämätön yhdistelmä nykyaikaisia verkkosivuja luodessa. On kuitenkin hyvä tietää, että HTML ja CSS eivät ole ohjelmointikieliä itsessään, ne ovat merkintäkieliä. Kun verkkosivuja lähdetään luomaan ja halutaan tuoda elementtejä näkyviin, käytetään HTML-koodia. Kun HTML-elementeistä halutaan saada visuaalisesti näyttäviä tai muuten parannella niiden ulkoasua, on CSS-kieli välttämätön siihen. Jos taas halutaan luoda toiminnallisuuksia tai efektejä sivustolle, tarvitaan jotain ohjelmointikieliä. (Popovic 2022a.)

3.2 JavaScript

Verkkosivujen toiminnallisuuden luomiseen tarvitaan jotain ohjelmointikieltä. Ohjelmointiin on mahdollista valita lukuisista eli kielistä mitä vain, mutta verkkosivustoja luodessa mm. PHP-, Python- ja JavaScript-kielet ovat yleisimpiä. (Webb 2023). JavaScript ja PHP ovat molemmat kevyitä ja tehokkaita

ohjelmointikieliä, jotka molemmat on luotu varta vasten verkkosivujen kehitykseen (Popovic 2022c). PHP skaalautuu hieman paremmin backend-kehitykseen, mutta JavaScript-kielellä kirjoitettu node.js lyö PHP:n backend-koodauksen asynkronisuudellaan (node.js 2023).

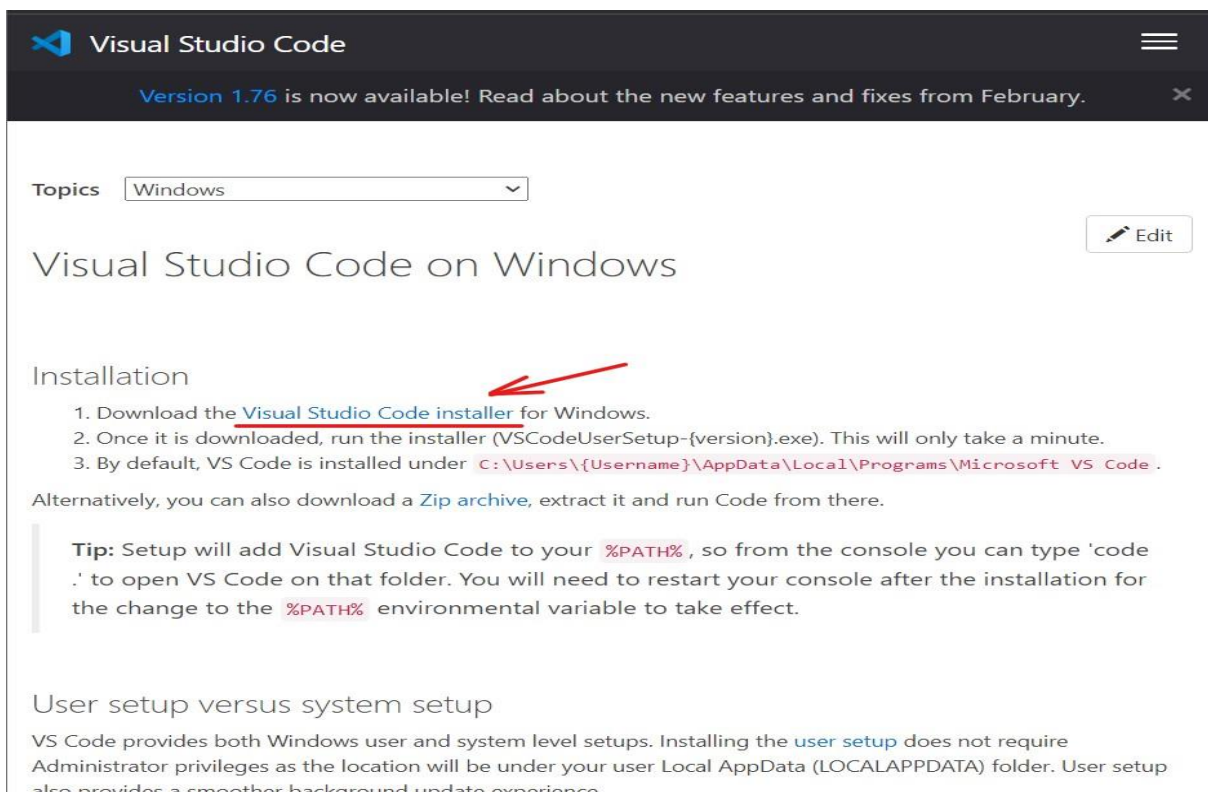
JavaScript on viehättänyt monia ohjelmointiosaajia ja sillä on luotu myös useita eri kirjastoja helpottamaan verkkosivujen ohjelmointia (Webb 2023). Ruby, jQuery ja React ovat hyvä esimerkkejä näistä (JavaScripting 2023). JavaScript-kielellä saa helposti ja nopeasti luotua interaktiivisia verkkosivuja. Sen oppiminen ei ole liian vaativaa suhteessa muihin, raskaampiin ohjelmointikieliin. Verkosta myös löytyy laajasti JavaScript-dokumentaatiota tukemaan kehitystä. (Popovic 2022c.)

JavaScript-kieltä käytetään tässä työssä. Kieli on valittu sen tehokkuuden, keveyden ja laajan dokumentaation takia. Kieltä ei kuitenkaan toteuteta sen varsinaisessa muodossa, vaan React.js-kirjaston yhteydessä. Tämä mahdollistaa keskittymisen ongelmanratkaisuun, kun React.js-kirjaston valmiit komponentit nopeuttavat perustoiminnallisuuden luomista (React 2023).

3.3 Visual Studio Code

Visual Studio Code on kevyt, mutta tehokas avoimen lähdekoodin editointiohjelma. Sitä voi käyttää eri käyttöjärjestelmissä, kuten Windowsissa, Linuxissa ja macOS:ssa. Alustavasti se tukee JavaScript-kieltä, TypeScript-kieltä ja Node.js-palvelimen hallintaympäristöä. Visual Studio Codesta löytyy myös lukuisia lisäosia eri ohjelmointikielille ja kirjastoille. (Visual Studio Code 2023a.)

Visual Studio Coden asentaminen Windowsille on helppoa ja nopeaa. Käyttäjän tulee ensin ladata asennuspaketti osoitteesta <https://code.visualstudio.com/docs/setup/windows>, klikkaamalla Visual Studio installer -linkkiä. (Visual Studio Code 2023b.)



Visual Studio Code

Version 1.76 is now available! Read about the new features and fixes from February.

Topics

Visual Studio Code on Windows Edit

Installation

1. Download the Visual Studio Code installer for Windows.
2. Once it is downloaded, run the installer (VSCodeUserSetup-{version}.exe). This will only take a minute.
3. By default, VS Code is installed under `C:\Users\{Username}\AppData\Local\Programs\Microsoft VS Code`.

Alternatively, you can also download a [Zip archive](#), extract it and run Code from there.

Tip: Setup will add Visual Studio Code to your `%PATH%`, so from the console you can type `'code .'` to open VS Code on that folder. You will need to restart your console after the installation for the change to the `%PATH%` environmental variable to take effect.

User setup versus system setup

VS Code provides both Windows user and system level setups. Installing the [user setup](#) does not require Administrator privileges as the location will be under your user Local AppData (LOCALAPPDATA) folder. User setup also provides a smoother background update experience.

Kuva 6. Punainen nuoli osoittaa, mistä Visual Studio Code Installer ladataan.

Kun lataus on valmis, käyttäjän tulee käynnistää juuri ladattu VSCodeUserSetup-{version}.exe asentaakseen Visual Studio Coden tietokoneelleen. Käyttäjän on hyvä lisätä Visual Studio Code tietokoneen polkuihin `%PATH%`. Näin ohjelman voi avata komentokehotteesta, `code .`-komennolla. Tämä nopeuttaa useissa tapauksissa projektin avaamista suoraan komentokehotteesta. (Visual Studio Code 2023b.)

Visual Studio Code saa kuukausittain päivityksiä, mukaan lukien bugikorjaukset, ja se on yksi käytetyimmistä avoimen lähdekoodin kehitysympäristöistä (Visual Studio Code 2023b). Tämän takia Visual Studio Code on erinomainen valinta tämän työn työalustaksi. Visual Studio Codeen on myös helppo asentaa lisäosia ja sen käyttöliittymän ulkoasua voi muokata mieltymystensä mukaan. Ohjelman mukana tulee myös valmiita teemoja ja niitä pystyy helposti lataamaan lisää Visual Studio Coden omasta tietojenjakokeskuksesta. Olen itse käyttänyt Visual Studio Codea jo 5 vuotta ohjelmointitarkoituksiin ja sen takia päädyin valitsemaan sen tämän työn kehitysympäristöksi.

3.4 React.js

React on JavaScript-kirjasto. Reactilla luodaan helposti koottavia käyttöliittymiä. Lyhyesti selitettynä, Reactin idea on luoda käyttöliittymiä yhdistelemällä komponentteja keskenään. (Chiarelli 2018, 6.) Nämä komponentit voivat olla esimerkiksi tykkäuspainikkeita, tekstipalstoja, videoiden pienoiskuvia tai jopa koko käyttöliittymä. React rohkaiseekin luomaan komponentteja niiden yksinkertaisuuden ja uudelleenkäytettävyyden kautta. (React 2023.)

Koska React on JavaScript-kirjasto, sitä voidaan käytännössä lisätä suoraan HTML-koodiin script-elementillä. Tämä ei ole kuitenkaan paras vaihtoehto, sillä ohjelmoija jäisi paitsi monista nykyajan mukavuuksista ja se estäisi muun muassa `let` ja `const` -komentojen käyttöä vanhemmilla selaimilla. React vaatii useita säätöjä, että näitä pystyisi hyödyntämään tehokkaasti. (Chiarelli 2018, 7.)

Onneksi `create-react-app`-komennon avulla React-projektin luonti sujuu helposti. Tämän komennon avulla kaikki tarvittavat säädökset ja asetukset luodaan projektikansioon. Komennon `create-react-app` voi asentaa komentokehokkeesta, komennolla `npm install -g create-react-app`. Npm-kutsu (node package manager) asentuu, asentamalla `node.js`. (Chiarelli 2018, 8.)

Nyt kun `create-react-app` on asennettu, on React-projektien luonti helppoa. Uuden React-projektin voi käyttäjä luoda komentokehokkeelta komennolla `npx create-react-app my-app`. `create-react-app` on aiemmin asennettu komento, jolla luodaan React-projekti ja `my-app` on projektin nimi, joka voi olla mikä vain käyttäjän haluama nimi, tietyn merkki rajoituksin. `Npx`-komento (node package extractor), `npm`-komennon tavoin vaatii Node.js asennuksen. (create react app 2023.)

3.5 React Router

React Router tai React Router DOM, on React-kirjasto. React Router mahdollistaa client-side-reitityksen, mikä parantaa käyttäjäkokemusta ja suorituskykyä verkkosivujen kehityksessä. (React Router 2023.)

Reititys verkkosivujen kehityksessä kuvaa prosessia, jossa käyttäjä linkkiä painamalla ohjataan uudelle sivulle, jolla on oma URL-osoite. Tavallisesti tämä vaatii pyynnön tekemistä palvelimelle, joka sitten siirtää käyttäjän uuteen HTML-dokumenttiin. Client-side-reitityksessä sovellus voi vaihtaa sivustolla URL-osoitetta ja näyttää uuden sivun ilman uusia palvelupyyntöjä. Tämä lisää suorituskykyä, ja React Router hyödyntää tätä. (Ahmed 2023.)

Client-side-reitityksen mahdollistumiseksi kehittäjän on luotava reititys React Routerin avulla. Tätä varten kehittäjä voi käyttää "Router" ja "Route" -elementtejä. Näiden avulla voidaan määrittää, mitkä komponentit liittyvät mihinkin reititykseen ja kuinka eri URL-osoitteet kasaantuvat verkkosivustolla. (React Router 2023.)

React Router tarjoaa myös muita tehokkaita ja helppokäyttöisiä ominaisuuksia. Niitä on muun muassa dynaaminen reititys, joka mahdollistaa tietojen hakemisen URL-osoitteesta, erilaisia linkitysvaihtoehtoja, kuten Link-elementin, ja vierityksen palautuksen, joka varmistaa vierityksen asennon säilymisen navigoidessa uudelle sivulle. Myös hieman edistyneempää ajattelua vaativa data loading on tehokas ominaisuus. Tämä mahdollistaa datan lataamisen reititysprosessin yhteydessä, jolloin tiedot ovat saatavilla komponenteille jo navigoinnin aikana. (React Router 2023.)

Työkalujen ja teknologioiden valinta verkkosivujen kehityksessä riippuu lopulta jokaisen kehittäjän oman mieltymyksistä. Kirjastot, kuten React helpottavat ohjelmointi ja julkaisu -prosessia, mutta niiden käyttö edellyttää niitä pyörittävän kielen ymmärtämistä. Tässä tapauksessa JavaScriptin.

4 Ajanvarausjärjestelmä

Aluksi oli ideana luoda järjestelmä osana opinnäytetyötä, mutta lyhyen tutkimuksen perusteella se paljastui liian laajaksi projektiksi sivuston rinnalle. Ajanvarausjärjestelmän toteuttaminen olisi edellyttänyt perusteellista tietokantaratkaisujen suunnittelua, kyberturvallisuuden huomioon ottamista ja peruskäytettävyyden huomioimista. Tämä olisi vienyt suuren määrän aikaa ja vastuullisuuden

näkökulmasta se olisi vaatinut myös jatkokehitystä ja tietoturvan säännöllistä ylläpitoa.

Sivustolle valittiin kolmannen osapuolen ajanvaraus järjestelmä. Ajanvarausjärjestelmiä on saatavilla suomeksi monia kymmeniä. Tunnetuimpia mainintoja on muun muassa Vello, Timma ja Visma. Useimmat näistä järjestelmistä tarjoavat samanlaiset perusominaisuudet, ja niiden takana on erilaisia hintamalleja ja sopimuksia.

Jotkin järjestelmät tarjoavat paketteja, jotka sisältävät esimerkiksi maksupäätämahdollisuuksia tai yhteyden omakanta-asiakasrekisteriin. Monissa suosituimmissa järjestelmissä, kuten Vellossa ja Timmassa on myös jo valmis asiakas-kunta, josta yritykset voivat hyötyä. (Vello 2023.) Näiden ajanvarausjärjestelmien verkkosivuilta käyttäjät voivat valita palveluita tarjoavilta yrityksiltä, mikä lisää yritysten näkyvyyttä.

Työn osana loin listan tunnetuista ja toimivimmiksi osoittaneimmista järjestelmistä. Listaani valikoitui järjestelmiä, mitkä toteuttivat vaatimukset, joita toimeksiantaja oli antanut. Listaani oli korostettu yrityksen käyttöön sopivimmat ja hintalaatu suhteeltaan parhaat vaihtoehdot. Toimeksiantaja arvioi listan ja tutustui itse syvemmin häntä kiinnostaviin järjestelmiin. Listasta valikoitui kaksi järjestelmää, johon toimeksiantajan sitten soitti ja kyseli lisätietoja. Näistä kahdesta Vello osoittautui paremmaksi vaihtoehdoksi ja siten se valittiin verkkosivuston ajanvarausjärjestelmäksi.

5 Toteutus

Verkkosivujen luominen toimeksiantajalle, SL Healthille lähti liikkeelle toimeksiantajan tarpeesta päivittää yrityksen näkyvyyttä ja lisätä ajanvarausjärjestelmä verkkosivuille. Kuitenkin tarkkaa kuvaa siitä, millaiset verkkosivut yritys tarvitsi, ei ollut vielä selvillä.

Tieto yrityksen halusta päivittää verkkosivuja, ei monesti riitä luomaan verkkosivustoja, jotka vastaavat täysin toimeksiantajan odotuksia. Lähdin liikkeelle selvittämällä, mitä vanhoissa verkkosivuissa oli vikana. Laadin kyselyitä ja

haastatteluja, joita kävimme toimeksiantajan kanssa läpi. Näiden kysymysten tarkoituksena oli hahmottaa karkeasti, mitä yritys halusi verkkosivujen saavuttavan ja mitä ominaisuuksia verkkosivuilta odotettiin. Kyselyiden kysymykset perustuivat työssä laadittuun tietopohjaan. Kysymysten tarkoituksena oli kartoittaa, mitä yritys haluaa. Niissä kysyttiin muun muassa, millaisia tunteita yritys haluaa verkkosivujen herättävän, onko väri toiveita, verkkosivujen kohdeyleisöä ja saavutettavuus vaatimuksia.

Kyselyt lähetin sähköpostitse toimeksiantajalle. Näin toimeksiantaja pystyi pohtimaan kysymyksiä itsekseen. Kävimme kyselyt vielä läpi kasvotusten tai puhelimitse. Näin pääsimme varmuuteen siitä, että kysymysten merkitys ja tarkoitus selvisi, sekä pystyin tarvittaessa avaamaan asioita laajemmin. Tällöin en kuitenkaan vielä ymmärtänyt aikataulutuksen merkityksellisyyttä ja jäin hieman katuimaan, ettemme sopineet siitä tarkemmin heti alkuun.

Pidimme alkuun palavereita toimeksiantajan kanssa usein ja myöhemmin tapaamiset harvenivat, kun asiat alkoivat selkenemään. Tarkkaa lukua en pitänyt tapaamisten määrästä, mutta suunnittelun alkuvaiheissa pyrimme tapaamaan toimeksiantajan kanssa kerran viikossa. Tapaamiset pyrimme pitämään kasvotusten. Näin kesti suunnilleen kuukauden ajan ja sitten tapaamiset harvenivat sekä siirtyivät enemmän puhelintapaamisiin.

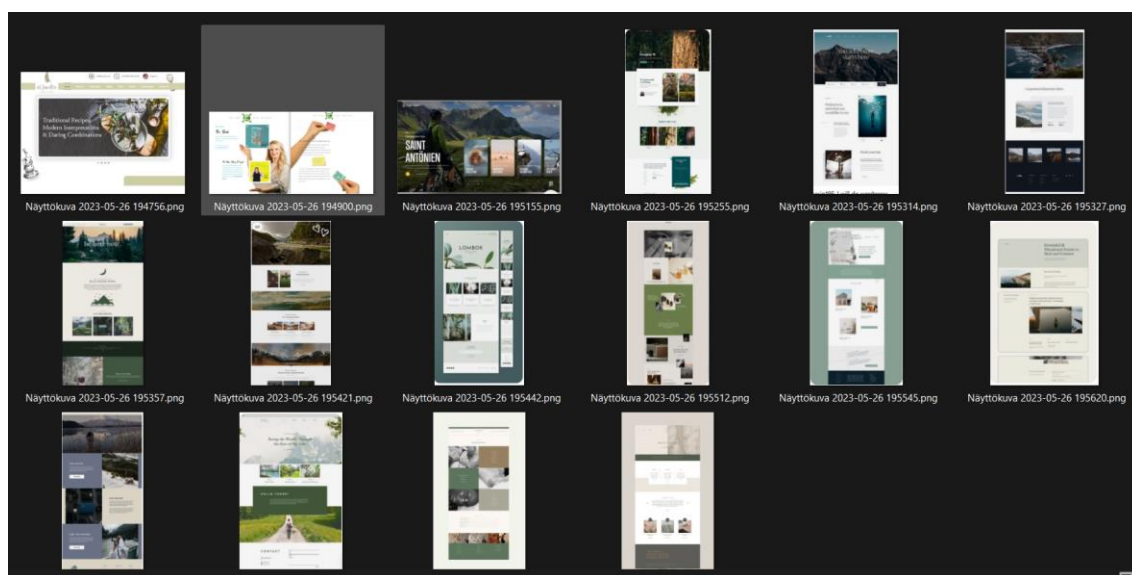
5.1 Suunnitteluvaihe:

Kun toimeksiantajan kanssa oli päästy yhteisymmärrykseen siitä, mitä yritys haluaa ja mihin tähdätään, lähdin aloittamaan verkkosivujen visuaalisia hahmotelmia. Aloitin suunnitteluprosessin luomalla rautalanka mallia uusien verkkosivujen etusivusta. Tavoitteenani oli lähteä liikkeelle löytämällä etusivut, joista toimeksiantaja pitäisi ja sen perusteella luoda loput sivut.

Sivustojen suunnittelussa käytin hyväkseni Justinmind-suunnitteluympäristöä. Sen avulla saatoin luoda nopeita hahmotelmia sivusto rakenteista ja visuaalisuuksista. Tämän ympäristön avulla voisin luoda nopeasti hahmotelmia, jotka vastaisivat lopullista tuotetta.

Ensimmäisen version etusivuista loin sen pohjalta, mitä toimeksiantaja oli vastaanottanut haastatteluissa. Versio oli oma näkemys visuaalisuudesta sen hetkisten tietojen perusteella. Tämä versio päädyttiin hylkäämään kokonaan, mutta toimeksiantaja otti ensimmäisen version nähtyään pohdintaan, kuinka haluaisi yrityksen logon esiintyvän sivuilla.

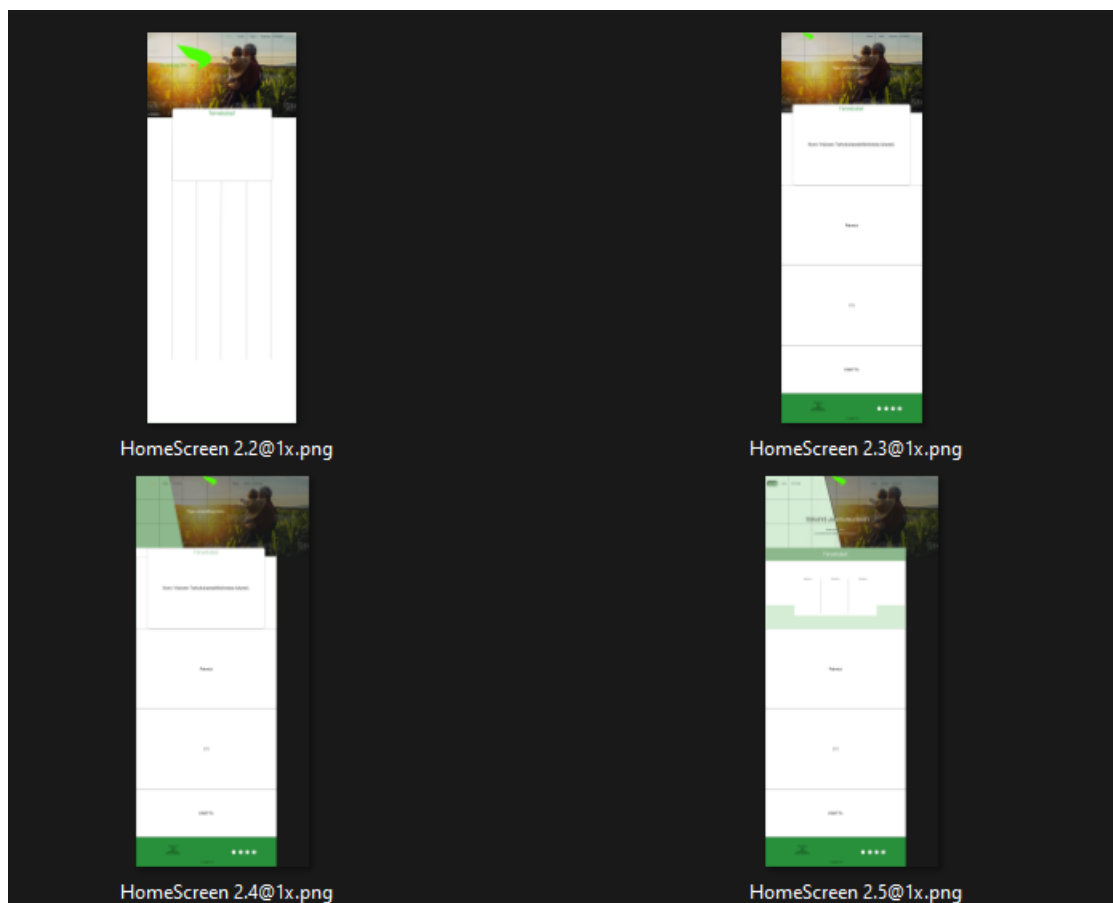
Jatkoimme suunnittelua ja otin käyttöön kuvakollaasin (kuva 7), johon keräsin erilaisia esimerkkejä verkkosivuista. Kollaasiin valitut verkkosivut pyrkivät vastaamaan kyselyjen perusteella saatuja yrityksen toiveita ja asioita minkä ajattelin sopivan parhaiten yrityksen imagoon. Pyrin varmistamaan, että valitut verkkosivut eroavat toisistaan mahdollisimman paljon, jotta voisimme harkita erilaisia vaihtoehtoja. Kun olin kerännyt riittävästi vaihtoehtoja, kävimme kollaasin läpi toimeksiantajan kanssa.



Kuva 7. Näyttökuvaa kansioista, johon kerätty erilaisia verkkosivujen ulkoasuja.

Kollaasin läpikäyminen toimeksiantajan kanssa auttoi meitä löytämään yhteisymmärryksen siitä, millaiset verkkosivut tukisivat yrityksen imagoa parhaiten. Valitsimme useita erilaisia ominaisuuksia eri vaihtoehdoista ja lopulta pääsimme sopivaan ulkoasuun etusivua varten. Esimerkiksi yhdestä väritys ja asettelu, yhdestä navigaatio valikko ja parista eri vaihtoehdosta otimme vielä muutaman nosto laatikon verkkosivuille.

Kokosin vielä useamman erilaisen ratkaisun (kuva 8) Justinmind -sovelluksella, esittelin ne toimeksiantajalle ja löysimme vihdoinkin sopivan ulkoasun yrityksen etusivuille. Etusivujen jälkeen loput sivut oli helppo hahmotella vastaamaan samaa tyyliä, mutta omia tarkoituksiaan. Näissä pääsimme sittemmin toimeksiantajan kanssa nopeammin yhteisymmärrykseen.



Kuva 8. Justinmindilla tehtyjä ratkaisuja etusivusta

5.2 Kehitysvaihe

Viimeistelin kaikki hahmotelmat eri sivuista ja siirryin varsinaiseen kehitysvaiheeseen. Kehitysprosessin toteutin Reactilla, joka on minulle tuttu kieli ja jolla olen tehnyt lukuisia projekteja viime vuosina. Ymmärrän hyvin sen rajoja ja mahdollisuuksia ja olen innokas oppimaan siitä lisää jokaisen projektin myötä. Reactilla on laaja yhteisö ja hyvät mahdollisuudet valmiisiin komponentteihin, jotka nopeuttavat tekemistä huomattavasti.

Muita vaihtoehtoja olisi ollut muun muassa PHP, python ja muita Reactin tyyppisiä JavaScript kirjastoja. Näin laajassa projektissa uuden kielen opetteluun olisi melko iso kynnys. Eikä kaikki vaihtoehdot soveltuisikaan tähän projektiin yhtä

hyvin. Vaikka pythonilla on mahdollista toimia web kehitys kielenä, ei se ole niin yleistä ja kynnyks on monesti isompi senkin takia. Lisäksi PHP:n hankalasti ymmärrettävä syntaksi olisi Reactiin verrattaessa hidastanut omaa tekemistä.

Navigaatiovalikko esiintyy jokaisella sivulla. Tässä projektissa sen on tarkoitus olla hieman erilainen etusivulla, kuin muilla sivuilla. Aloitin koodausprosessin etusivun navigaatiovalikosta oman mieltymykseni takia sen ollessa yleisestikin ensimmäinen elementti verkkosivuilla. Pystyn hyödyntää etusivun valikkoa myöhemmin muillakin sivuilla tekemällä vain pieniä muutoksia ulkoasuun.

Valikon luominen on melko suoraviivainen prosessi. Ensin luon nav-elementin, jonka sisään listataan linkit halutuille sivuille. Reactilla työskennellessä, käytin hyväkseni React Router Domia. Kutsuin React Router Domia app.js -tiedostossa (kuva 9).

```
import { Etusivu, Tietoja, Otayhteytta, Palvelut } from "../src/pages/index";

function App() {
  return (
    <>
      <Routes>
        <Route path="/slhealthwebsite/" element={<Etusivu />} />
        <Route path="/slhealthwebsite/tietoja" element={<Tietoja />} />
        <Route path="/slhealthwebsite/otayhteytta" element={<Otayhteytta />} />
        <Route path="/slhealthwebsite/palvelut" element={<Palvelut />} />
      </Routes>
    </>
  );
}

export default App;
```

Kuva 9. React-router-domin käyttö App.js -tiedostossa

React Router Domin avulla pystyn muun muassa muuttamaan sivulla näytettävää sisältöä aiheuttamatta koko sivun uudelleenlatausta. Tämä luo sujuvamman ja nopeamman käyttökokemuksen. React Router Dom tukee myös HTML5 history API:a, jonka avulla esimerkiksi eteen ja takaisinpäin nuolien käyttö ei aiheuta koko sivun uudelleen latausta, parantaen edelleen käyttökokemusta.

Linkit sivuille voisi listata suoraan React-koodiin. Kuitenkin jatkokehitystä ja uudelleen käytettävyyttä ajatellen loin JSON-tiedoston, joka sisälsi tiedot eri sivuista sivustolla. Tämän JSON-tiedoston sitten toin navigaatiovalikkoon ja kävin sen läpi listaamalla sen (kuva 10) sisältämät tiedot (kuva 11) oikeisiin paikkoihin. Näin koodista tuli myös helpommin ymmärrettävää eikä se vienyt niin montaa koodiriviä.

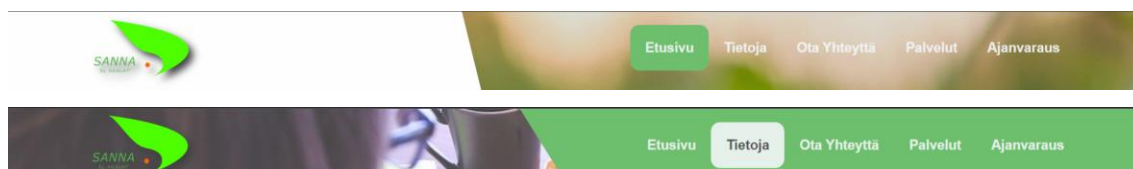
```
<ul className={navOpen ? "nav-items active" : "nav-items"}>
  <div className="nav-options">
    {navItems.map(item => {
      return(
        <li key={item.id} className={item.cName} id={item.itemId}
onClick={() => activateLink(item)} >
          <Link id="navA" to={item.path}>{item.title}</Link>
        </li>
      );
    })}
  </div>
</ul>
```

Kuva 10. Navigaatio valikon linkkien listaus.

```
export const ServiceItems = [
  {
    id: 1,
    title: "Etusivu",
    path: "/...",
    cName: "nav-item",
    position: "left",
    itemId: "unactive"
  },
  {
```

Kuva 11. JSON-tiedoston rakenne.

Navigaatiovalikon ulkoasu vaati myös logon korostaakseen yrityksen imagoa. Logosta ei ollut versiota, jossa tausta olisi läpinäkyvä, joten jouduin ensin leikkaamaan sen irti taustasta. Logo tuli navigaatiovalikkossa vasempaan reunaan ja itse sivuston valikko tuli oikeaan reunaan (kuva 12).

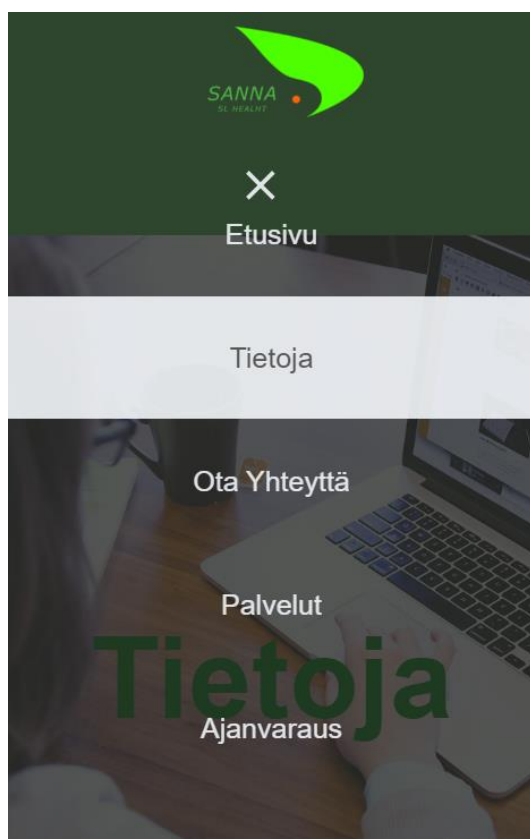


Kuva 12. Ylempänä navigaatio valikko etusivulla ja alempana muille sivuille muotoiltuna.

Mobiili näkymässä navigaatiovalikko on hieman erilainen. Valikko pienenee kahteen osaan, jotka asettuvat päällekkäin, logo päällimmäisenä ja valikko kuvake sen alapuolella (kuva 13). Valikon logo vaihtuu valkeaksi, jotta se erottuu paremmin taustasta. Valikon kuvaketta painamalla tippuu ylhäältä päin valikon linkit (kuva 14).



Kuva 13. Navigaatio valikko mobiili näkymässä



Kuva 14. Navigaatiovalikko auki mobiili näkymässä

Lopuksi pyrin saamaan asettelun kuntoon CSS-koodilla. Tähän Reactin live-palvelin oli hyödyllinen. Muutokset näkyivät reaaliajassa sitä mukaan, kun muokkasin koodia. Live palvelimen saa käyntiin, kun komentokehokkeella ajaa komennon, npm start. Asettelu on tärkeä pitää kunnossa alusta asti, jossa muutkin

sivuston elementit asettuvat paikoilleen. Lopussa on sitten helppo tehdä pieniä muutoksia, mikäli joku asia ei tyydytä omaa tai asiakkaan silmää.

Pidin myös koko ajan huolta sivuston responsiivisuudesta. CSS:n flexbox on helppo elementti responsiivisuuden kannalta ja käytin sitä myös navigaatio valikossa. Flexboxin lisäksi käytin responsiivisia arvoja, kuten prosentteja ja remittayksikköä esimerkiksi marginaaleja luodessa. Fontteja säädettyä on enemmän mielipidekysymys millä kukin kokee fontin koon säätyvän parhaiten. Käytin navigaatiovalikossa pikseleitä, sillä ne ovat helposti ymmärrettäviä ja näissä tapauksissa yleisesti käytetty 18–22 pikseliä monesti riittää takaaman helpon luettavuuden.

Etusivulle oli tarkoitus tulla myös linkkejä ajanvaraukseen ja palveluihin. Palveluita varten oli tarkoitus luoda samanlainen JSON-tiedosto, kuin navigointi valikkoa. Tästä syystä oli hyvä aika luoda se heti ensimmäisenä, jotta sitä voisi käyttää hyväksi alusta alkaen aina, kun palveluihin viitataan muissakin komponenteissa.

Rakenne JSON-olioilla on sama kuin navigointi valikossa, mutta muuttujia on enemmän. Yrityksen palveluihin viitatessa haluttiin myös käyttää kuvia, joten loin JSON-tiedostoon olioille myös kohdat, jotka sisältäisivät kuvien URL osoitteet. Näin JSON-tiedoston koko pysyy pienenä ja se auttaa nopeampaan tiedoston käsittelyyn.

Ensimmäisenä loin etusivulle pääkuvan. Kuvan tarkoituksena on toivottaa käyttäjä tervetulleeksi sivustolle. Tämän takia lisäsin kuvaan myös tekstiä ja taustaväriä, että teksti erottuisi kuvasta paremmin.

Palveluiden ja ajan varauksen esille tuonti oli yksi päätekijä verkkosivujen päivitykseen. Tästä syystä etusivulla heti pääkuvan alapuolella on hyvä paikka laittaa viittauksia niihin. Loin tätä varten palveluille korttityylisen komponentti ratkaisun (kuva 15). Korttityyppisestä komponentista saa helposti sellaisen, että ne saavuttavat käyttäjän huomion.



Kuva 15. Palvelukortit rivissä

Korteissa oli oleellista olla linkki ajanvaraukseen ja muuten sain aika vapaat kädet niiden suunnitteluun. Kunkin kortin merkityksen selventämiseksi, toin palveluiden nimet, niitä vastaavan kortin päälle värikkääseen palloon (kuva 16). Korttien taustalle tulee palvelua edustava kuva ja niitä klikkaamalla pääsee siirtymään palvelua vastaavalle sivulle, josta voi sitten lukea palvelusta lisää.



Kuva 16. Palvelukortin nimi värikkäässä pallossa

Sivut tarvitsivat myös kevennystä. Pitääkseni yrityksen imagon sivustolla maanläheisempänä, loin palvelukorttien alle tekstikomponentin (kuva 17). Tähän komponenttiin yritys voi kirjoittaa erimerkiksi tunteisiin vetoavan siteerauksen tai kertoa lyhyesti yrityksestä. Tekstin taustan pidin valkoisena, pyöristin reunat ja alueen taustalle asetin kuvan. Kuvan tyyllittelin niin, että kuva pysyy paikallaan, vaikka sivua rullataan. Tämä onnistui muuttamalla CSS:llä, kuvan background-attachment arvo, arvoksi fixed.



Kuva 17. Tekstikomponentti täytetty satunnaisella tekstillä.

Viimeistelläkseni etusivut, loin sivuille yhteenveto laatikon. Laatikoon tulee yritykseltä lyhyt läpikäynti palveluista, yrityksestä ja ajanvarauksesta. Sähköinen ajanvaraus on yrityksellä uusi ominaisuus, ja sen esille tuominen on ollut toiveena koko sivuston luomisprosessin ajan. Ylimääräiset esilletuomiset eivät siis ole pahitteeksi.

Yhteenvedolle loin korttityylisen ratkaisun (kuva 18). Tämä on kuitenkin hieman erilainen korttityyppi, kuin palveluilla. Yhteenvedot kustakin aiheesta tuli vierekkäin, valkoiselle pohjalle, tumman viivan erottamana. Jotta valkoinen pohja erottuisi taustasta, loin vihreän taustan, joka kuitenkin peittäisi vain alle puolet taustasta, luodakseni hieman tyyliä. Tämä myös mahdollistaisi peittää loppu tausta taustakuvalla, mutta se ei ole tälle projektille oleellista ja tehdään vain, jos toimeksiantaja niin haluaa myöhemmin.



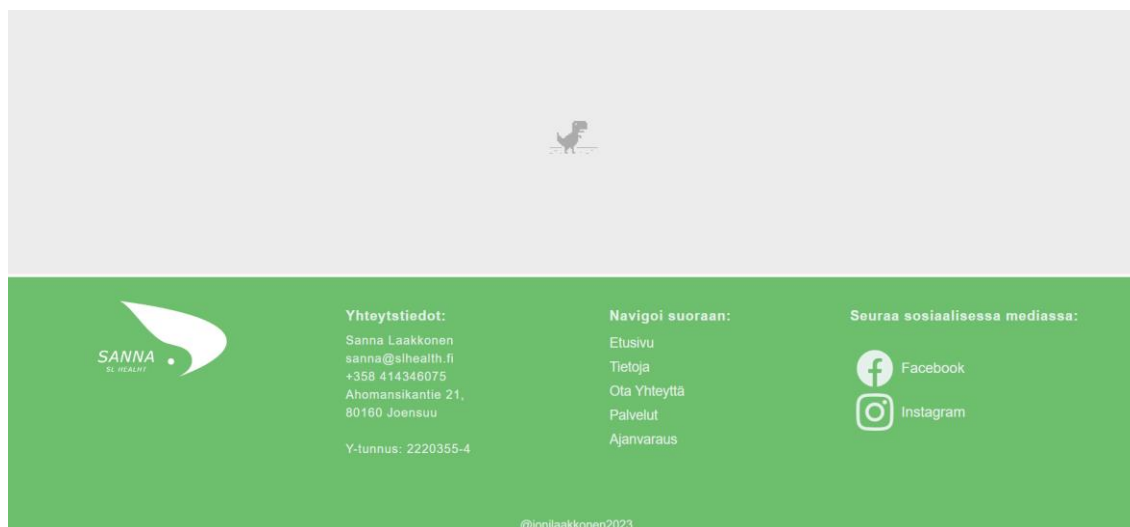
Kuva 18. Yhteenvetokortit

Viimeisenä komponenttina sivustolle tulee kartta ja footer-elementti. Kartalle tulisi merkitä SL-Healthin toimipiste ja se tulisi kohdistaa siihen, riittävälle etäisyydelle. Footer-elementtiin vaatimuksina on tuoda yhteystiedot ja sosiaalisen median kuvakkeet.

Kartan toteuttamiseksi, hain google kartoista iframe-elementin. Tämä on helppo tapa tuoda kartta, kun se tuo mukanaan automaattisesti halutun zoomauksen ja merkityt paikat. Lopuksi sievensin karttaelementin CSS-koodilla, sivustoon sopivaksi.

Footer komponentin rakensin yhdestä flexbox-elementistä (kuva 19). Näin saa säädettyä footerin sisältämien tietojen asettelua helposti esimerkiksi katkaisupisteillä, eli CSS:än media tagilla, jolla voi säätää sivuston asetteluja esimerkiksi näytön leveyden perusteella. Footeriin toin logon, tein yhden laatikon yhteystiedoille ja sosiaaliselle medialle, sekä muulle mahdolliselle footer-sisällölle ja viimeisenä alimmaksi footeriin jäi paikka vielä tekijän oikeuksille.

Keskimmäisen laatikon footeriin tein myös flexbox-elementistä. Näin saan asettelun tulemaan vierekkäin isommalla näytöllä, ja sitten tippumaan allekkaiseksi järjestelyksi, kun näytön koko kapenee tarpeeksi. Yhteystietojen ja sosiaalisen median kuvakkeiksi lisäksi toin tähän laatikkoon myös navigointi valikon linkit, käyttäjä kokemusta ajatellen.



Kuva 19. Footer jossa päälimmäisenä kartta elementti ja alla linkit sekä muu footerissa esitetty informaatio

Loput sivuista sisältävät kaikki navigaatio valikon, header-kuvan, kartan ja footerin. Navigaatio valikko on samankaltainen kuin pääsivulla, vain värit tulevat eri tavalla (kuva 12). Header-kuva valitaan jokaiselle sivulle erikseen vastaamaan kunkin sivun tarpeita ja kuvan keskellä lukee isolla tekstillä kyseisen sivun nimi (Palvelut, Tietoja, jne.). Kartta ja footer-komponentit asettuvat sivun alimmiksi komponenteiksi päällekkäin, samalla tavalla kuin etusivulla.

Sivujen sisältö riippuu sivu vaatimuksista. Sivuja tulee olla palveluiden listaukselle, yksittäisten palveluiden tarkemmalle erittelylle, yrityksen esittelylle, yhteyden otto lomakkeelle ja ajanvaraukseen.

Ensimmäiseksi lähdin etusivun jälkeen toteuttamaan tietoja sivun. Sen sisältö oli nopein toteuttaa, eikä vaatinut mitään erillisiä toiminnallisuuksia. Sivulle tuli ainoastaan laatikko, johon tulee tietoa yrityksestä ja osio, jossa on tietoa henkilökunnasta.

Tein laatikon, joka kulkee samassa leveydessä tiettyyn pisteeseen asti. Sitten se supistuu sivustolle responsiivikseksi tarvittavin keinoin, kuten jakautumalla useampaan osaan. Koska yritys on yhden henkilön yritys, erillisiä henkilöstö kortteja ei tarvinnut tehdä. Näin on sija helpommin korostaa yrittäjän omaa tarinaa.

Pääosan sivusta varasin kuitenkin yrityksen esittelylle ja tarinalle. Toisena laattikkona tämän rinnalle tuli henkilökortti, hieman eri sävyisellä taustalla ja henkilökuvalla. Tyyllittely oli yleisesti hyvin neutraalia ja väritys mukaili vain yrityksen logon oranssia väriä pienissä määrin (kuva 20).



Kuva 20. Tietoja sivun sisältö

Yrityksen verkkosivut tarvitsivat lomakkeen yhteyden ottoa varten. Lomake lähetäisi sähköpostin verkkosivuilta yrityksen sähköpostiin, halutulla tavalla muotoiltuna. Lomakkeen rakensin form-elementin sisään. Tämä toimi lomakkeen pohjana, jonka sijoitin sitten keskelle näyttöä ja muotoilin sen halutunlaiseksi CSS-koodin avulla. Lomakkeen viereen toin vielä listauksen yrityksen tarkemmista yhteystiedoista (kuva 21).

Kuva 21. Yhteydenotto-lomake

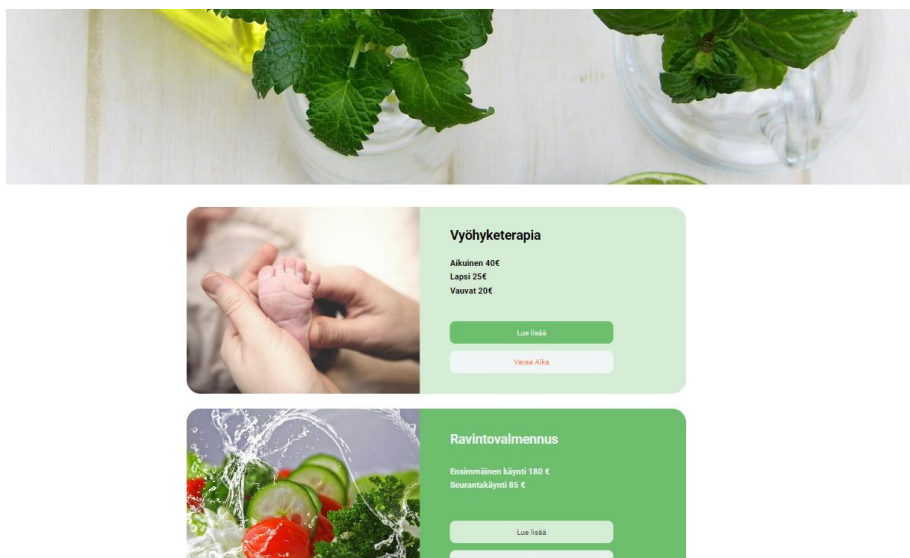
Palveluja on monta ja yhden sivun tarkoitus onkin listata nämä helposti käyttäjän tarkasteltavaksi. Käyttäjän olisi myös hyvä päästä helposti tarkastelemaan

paremmin haluamaansa yksittäistä palvelua tai varaamaan aika palveluun, mikä häntä kiinnostaa.

Aloitin tuomalla palvelut JSON-tiedostosta ja listaamalla ne sivustolle (kuva 22). Toin palveluista vain haluamani tiedot HTML-elementteihin ja tyyllittelin nämä elementit sitten CSS-koodin avulla halutunlaiseksi. Tein elementeistä toistuvasti erilaisia niin, että vain joka toinen elementti oli samanlainen. Yleinen tyyli pysyi kyllä samana, mutta värit vaihtuvat even ja odd -luokkanimien perusteella, tunnelman luomiseksi (kuva 23).

```
{ServiceItems.map((item, index) => {
  const isEven = index % 2 === 0;
  const containerClassName = isEven ? 'service_container
even' : 'service_container odd';
  const pricesArray = item.prices.split(',');
  const textArray = item.text.split(';');
  return(
    . . .
  )
})}
```

Kuva 22. Palveluiden listaus

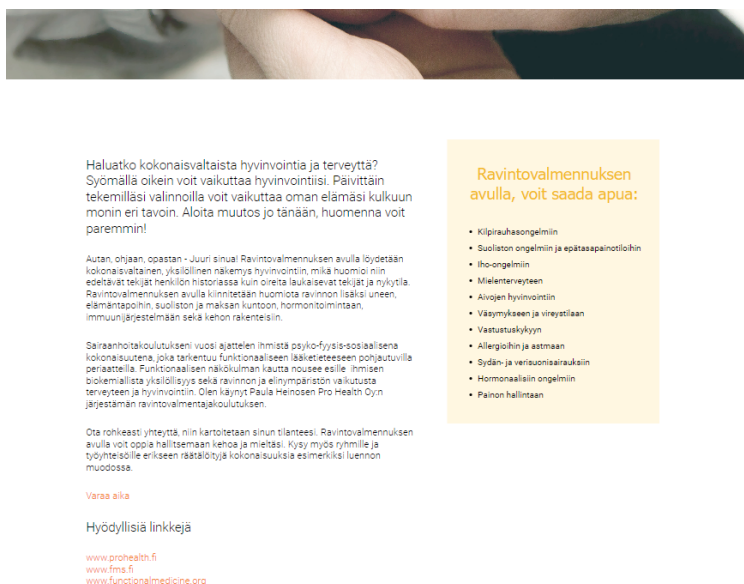


Kuva 23. Palvelut sivu

Palveluille loin kaikille omat sivut, jotka toimivat kaikki samalla pohjalla niin, että vain sisältö vaihtui. Palveluiden sivut toimivat samalla rakenteella, kuin muutkin sivut, eli ylimpänä header-kuva, sitten sisältö ja alimmaisena footer. Sivulle oli tarkoitus saada näkyviin vain tietoja palvelusta ja erillinen laatikko, jossa on faktoja esimerkiksi palvelun hyödyistä. Tämän takia käytin hyväkseni tietoja sivun

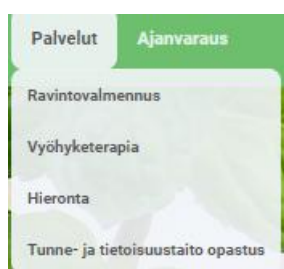
muotoilua ja rakennetta, jotta tyylejä ja asetteluja ei tarvitse luoda kokonaan uudestaan.

Palveluille tuli tietoja sivun tapaan yksi isompi laatikko, johon saa kirjattua palvelun tiedot ja hinnaston ja viereen pienempi info laatikko, jossa käytin henkilökortin rakennetta (kuva 24). Responsiivisuus ja väritkin mukailee tietoja sivun komponentteja.



Kuva 24. Yksittäisen palvelun sivusto rakenne

Navigaatio valikkoon loin palveluille oman paikan palvelut linkin alapuolelle. Yksittäisiin palveluihin pääsee viemällä hiiren palvelu linkin päälle, jolloin sen alle listautuu valikko palveluista (kuva 25). Palveluihin pääsee myös linkeistä palvelut sivulta tai etusivun palvelu korteista.



Kuva 25. Palvelut listattuna Palvelut linkin alle.

Ajanhallintatyökalun upotus sivustoille tapahtui nopeasti. Sen jälkeen, kun toimeksiantaja oli ostanut haluamansa Vellon ajanvarausjärjestelmä paketin, sai sisään kirjautuneena tilaajana koodinpätkän, jolla pystyi upottamaan sivustolle

Vellon ajanvaraus kalenterin. Tällä kalenterilla on suora yhteys Velloon ja siten aikoja voi varata joko Vellon tai yrityksen verkkosivujen kautta.

Yrityksen ajanhallinta ja maksut siirtyvät Vellon alle. Tämä tarkoittaa, että toimeksiantajan ajanvarauksiin käytetty aika sähköpostien sekä puheluiden välityksellä minimoituu. Lisäksi yrityksen lomien ja vapaapäivien hallinta helpottuu, kun järjestelmään voi etukäteen valita tunnit tai päivät, milloin ei voi varata aikoja. On myös mainittava, että maksujen hallinta helpottuu, kun toimeksiantajan ei tarvitse enää käsin kirjata maksutapahtumia ylös.

5.3 Testaus ja käyttöönotto:

Verkkosivut alkoivat lähentyä haluttua lopputulosta. Kun sivustot olivat valmiit. Pyysin toimeksiantajaa tarkastamaan ja testaamaan sivut, sekä lähettämään kuvat ja tekstit, joita sivuille tulisi.

Sisällön toimeksiantaja valitsi itse. Kuvat ladattiin netistä kuvagallerioista, lain puitteissa ja toimeksiantajalla oli muutamia omia kuvia jo valmiina. Tekstit toimeksiantaja kirjoitti sivustolle, kun hän näki millaisia tekstejä ne vaativat. Kiireellisen aikataulunsa takia, toimeksiantajalla kesti toimittaa sisältöä odotettua kauemmin, viivästäen hieman opinnäytetyön kulkuakin.

Kun sivustot alkoivat olla valmiit, ajoin sivustot verkkoon kaikkien katseltavaksi Githubin palvelimelle. Näin pystyin vielä jakamaan sivuston linkkiä, haluamilleni henkilöille. Nämä henkilöt sitten kykenivät testaamaan sivuston toiminnallisuutta ja antamaan kehitysehdotuksia.

Testaajina toimivat toimeksiantajan ystävät. Testaajat tarkastivat sivustojen toiminnallisuutta ja ulkoasua. Heille ei annettu erillistä ohjetta testaukseen, vaan testaus tapahtui täysin itsenäisesti ja omilla ehdoillaan. Näin testaajilta tuli omia näkemyksiä verkkosivujen hyvistä ja huonoista piirteistä. Palautteiden perusteella sivustoa korjattiin vielä kertaalleen yhdessä toimeksiantajan kanssa, bugien ja kirjoitusvirheiden osalta.

Kun toimeksiantaja oli tyytyväinen sivustoihin ja ajanvarauskalenteri upotettu sivustoille, oli aika ladata verkkosivut julkaisupalveluun. Webhotelliksi ja

julkaisupalveluksi valittiin Elisan yritysweb (Elisa 2023a). Elisan yritysweb valinta tapahtui kokonaan toimeksiantajan toimesta. Toimeksiantajalla oli vielä tilausta jäljellä yrityswebiin, eikä hänelle tämän valinta ollut niin merkityksellistä.

Yrityswebiin julkaisu tapahtui ensin asentamalla jokin tiedoston siirto sovellus, joka tukee FTP-palvelimeen siirtoa. Sovelluksen kautta tiedostot siirtyivät antamalla haluttu osoite ja tiedostot. Kun yrityswebiin on ladattu haluamansa verkkosivujen tiedostot, niitä pääsee vielä kerran tarkastelemaan testivww-osoitteesta. Kun sielläkin sivustot näyttivät oikealle ja toimivat oikein, soitettiin elisan asiakaspalveluun, mistä sivustot siirrettiin sitten oikean www. osoitteen alle. (Elisa 2023b.)

6 Pohdinta

Opinnäytetyön alkuperäinen tavoite oli luoda verkkosivuille lisäksi oma ajanvarausjärjestelmä ja verkkosivuston hallintajärjestelmä. Hyvin nopeasti tuli kuitenkin selväksi, että hanke olisi ollut liian kunnianhimoinen aikaan, joka oli käytettävissä. Selvisi, että kummastakin järjestelmästä yksinään voisi tehdä opinnäytetyön, vaikka kyberturvallisuudestakin erikseen. Tästä syystä päätimme toimeksiantajan kanssa etsiä sopiva, valmis ajanvarausjärjestelmä, mistä opinnäytetyön aihe sitten muodostui.

Työn alkuvaiheissa tuli selväksi, että minulla on vain rajoitettu määrä tunteja viikossa käytettäväksi. Aluksi yritin ohittaa tämän tosiasian ja tehdä niin paljon kuin mahdollista aina kun siihen oli aikaa. Nopeasti tämä johti kuitenkin useisiin tilanteisiin, missä poltin itseni loppuun. Tajusin tämän vasta, kun sain hetken tauon kaikista koulutehtävistä. Tästä viisaampana, aloin työstää opinnäytetyötäni harkitummin yksi askel kerrallaan.

Työn edetessä huomasin myös, että toimeksiantaja oli työssä mukana paljon vähemmän, mitä aluksi ajattelin. Huomasin joutuvani lukemaan toimeksiantajaa paljon rivien välistä, opettelemaan olemaan omatoiminen tiettyjen päätösten kanssa, jotta työ etenisi ja tekemään monia erilaisia suunnitelmia valmiiksi, että toimeksiantaja pystyi antamaan edes suuntaa antavia mielipiteitä.

Alkuun minulla ei ollut erityisemmin sellainen olo, että olisin työn aikana kehitty-
nyt ohjelmoijana. Jälkeenpäin kuitenkin kuvia ja tekstejä liittäessä, olen huo-
mannut, koodissani paljon pientä viilattavaa. Koodi on toisaalta kirjoitettu pitkän
aikavälin kuluessa ja sekin voi olla osittain syynä epäsymmetrisyyksiin. CSS-
koodin olen oppinut siistimään ja tarkistamaan kertaalleen jälkeenpäin. Sinne
tulee helposti päällekkäisyyksiä ja asioita, kuten responsiivisuutta tukevia kat-
kaisukohtia jää helposti miettimättä.

Eniten olen huomannut kehittyneeni kirjoittamisessa. Raportointi, koodin kirjoin-
tus ja suunnittelu vaativat kaikki omanlaista tapaa kirjoittaa ja miettiä asioita. Tä-
hän asti olen aina mennyt siitä mistä aita on matalin, mutta tämä työ on pakotta-
nut ja totuttanut kirjoittamaan asiat huolella ja omin sanoin.

Kuten jo johdannossa on mainittu, kuinka verkkosivustojen luonti järjestelmiä
kuten WordPressin hyödyntäminen kävi alkutaipaleilla mielessä. Halusin kuiten-
kin haastaa itseni oppimaan enemmän verkkosivujen toiminnallisuudesta koodi-
tasolla. Omasta mielestä olen paljon oppinutkin. Työn tutkiminen ja luominen
antoi paljon uutta näkemystä verkkosivujen ominaisuuksista, kuten turvallisuus-
desta, julkaisusta ja ylläpidosta. Olen oppinut paljon omasta koodistani, sen hy-
vistä ja huonoista puolista ja käytänteistä, joilla voisin mahdollisesti parantaa
omaa koodia jatkossa.

Myös työ harjoittelun mukana olen perehtynyt verkkosivujen kehitykseen sy-
vemmin. Työharjoittelussa olen päässyt näkemään useita erilaisia verkkosivu
toteutuksia ja työskentelemään niiden parissa. Erilaisten ulkoasujen näkeminen
ja verkkosivuilla käytettyjen komponenttien toiminnallisuudet saivat minut vielä
loppua kohti näkemään useita korjauksen kohteita opinnäytetyössäni. Harjoitte-
lustani sain myös useita hyviä vinkkejä, miten ajatella ja miten tehdä joitain asi-
oita työssäni. Yksi näistä asioista oli katkaisukohtien käyttö fonttien -ja kuvien
koon muuntamiseen.

Tulevaisuudessa ongelma tilanteiden välttämiseksi, paremmat suunnitelmat
ajankäyttöön, projektin etenemiseen ja toimeksiantajan kanssa toimimiseen
ovat välttämättömiä. Ongelma tilanteet ja niistä etenkin ajankäytön ongelmat oli-
vat työssä suuri ajan hukka.

Jos jossain kuitenkin onnistuin niin sen, miten paljon tietoa verkossa on saatavilla. Olen tiedostanut, että verkosta helposti löytää vastauksen useimpiin kysymyksiin. Tietopohjaa työhön hankkiessani kuitenkin tajusin kuinka laajasti ja tarkasti asiat on monesti selitetty, kun vain hieman jaksaa etsiä.

Verkkosivut jäivät kokonaisuudessaan kuitenkin hieman vajaaksi. Opinnäytetyön osalta työ saatettiin maaliin asti, mutta siitä jäi uupumaan tärkeä ominaisuus käytettävyyden kannalta. Verkkosivut vaativat vielä hallintajärjestelmän, jotta toimeksiantaja voisi helposti seurata sivuston toimintaa ja päivittää sivuja tarpeitansa vastaaviksi. Tämä onkin seuraava asia, mitä rupean kehittämään, kun saan opinnäytetyön valmiiksi. Hallintajärjestelmä ei kuitenkaan ollut niin tärkeä etappi, kun ei verkkosivuja koko aikaa olla päivittämässä.

Lähteet

- Ahmed A. 2023. A Basic introduction to client-side routing using React Router. <https://medium.com/@aadilahmed0/a-basic-intro-to-client-side-routing-using-react-router-e9effe7cab4d>. 1.11.2023.
- Beaird, J. & George, J. 2014a. The Principles of Beautiful Web Design, 3rd Edition. <https://urly.fi/39U7>. 17.5.2023.
- Beaird, J. & George, J. 2014b. The Principles of Beautiful Web Design, 3rd Edition. <https://urly.fi/39B5>. 21.4.2023.
- Beaird, J. & George, J. 2014c. The Principles of Beautiful Web Design, 3rd Edition. <https://urly.fi/39B6>. 21.4.2023.
- Beaird, J. & George, J. 2014d. The Principles of Beautiful Web Design, 3rd Edition. <https://urly.fi/39B8>. 21.4.2023.
- Beaird, J. & George, J. 2014e. The Principles of Beautiful Web Design, 3rd Edition. <https://urly.fi/39B9>. 21.4.2023.
- Beaird, J. & George, J. 2014f. The Principles of Beautiful Web Design, 3rd Edition. <https://urly.fi/39Ba>. 21.4.2023.
- Beaird, J. & George, J. 2014g. The Principles of Beautiful Web Design, 3rd Edition. <https://urly.fi/39Bb>. 21.4.2023.
- Beaird, J. & George, J. 2014h. The Principles of Beautiful Web Design, 3rd Edition. <https://urly.fi/39Bq>. 21.4.2023.
- Beaird, J. & George, J. 2014i. The Principles of Beautiful Web Design, 3rd Edition. <https://urly.fi/39TS>. 17.5.2023.
- Beaird, J. & George, J. 2014j. The Principles of Beautiful Web Design, 3rd Edition. <https://urly.fi/39Bc>. 21.4.2023.
- Butler, M. 2018. The Importance of Images on a Website. <https://www.carmelon-digital.com/digital-marketing-magazine/the-importance-of-images-on-a-website/>. 28.4.2023.
- Chiarelli, A. 2018. Beginning React: Simplify Your Frontend Development Workflow and Enhance the User Experience of Your Applications with React. E-Book Central. 27.2.2023
- ChatGPT 2023a. Mitkä ovat responsiivisen verkkosivun avaintekijöitä?. Open AI. <https://chat.openai.com>. 18.4.2023.
- ChatGPT 2023b. Kuinka valita oikeat värit verkkosivuille ja millä suhteilla ne tulisi valita?. Open AI. <https://chat.openai.com>. 18.4.2023.

- create react app 2023. Create React App. <https://create-react-app.dev/>.
27.3.2023.
- Elisa 2023b. Elisa Yritysweb – ohjeet ja vinkit yrityksille.
<https://yrityksille.elisa.fi/ohjeet/yritysweb>. 6.11.2023.
- Elisa 2023a. Elisa Yrityswebin Testwww-sivusto.
<https://yrityksille.elisa.fi/ohje/elisa-yrityswebin-testwww-sivusto>.
6.11.2023.
- Garret, J. 2011a. Elements of User Experience: User-Centered Design for the Web and Beyond, Second Edition. <https://urly.fi/39Br>. 7.4.2023.
- Garret, J. 2011b. Elements of User Experience: User-Centered Design for the Web and Beyond, Second Edition. <https://urly.fi/39Bt>. 7.4.2023.
- Garret, J. 2011c. Elements of User Experience: User-Centered Design for the Web and Beyond, Second Edition. <https://urly.fi/39Bw>. 7.4.2023.
- Garret, J. 2011d. Elements of User Experience: User-Centered Design for the Web and Beyond, Second Edition. <https://urly.fi/39Bx>. 7.4.2023.
- Garret, J. 2011e. Elements of User Experience: User-Centered Design for the Web and Beyond, Second Edition. <https://urly.fi/39BB>. 7.4.2023.
- Garret, J. 2011f. Elements of User Experience: User-Centered Design for the Web and Beyond, Second Edition. <https://urly.fi/39BE>. 7.4.2023.
- JavaScripting 2023. The definitive source of the best JavaScript libraries, frameworks, and plugins. <https://www.javascripting.com/>. 3.4.2023.
- Jukarainen, M. 2020. Nettisivujen teko 2020 – Hanki hyvät WordPress kotisivut. <https://mikaeljukarainen.com/nettisivut/>. 1.1.2023.
- Kantamneni, S. 2022a. User Experience Design. <https://urly.fi/39BJ>. 6.4.2023.
- Kantamneni, S. 2022b. User Experience Design. <https://urly.fi/39BT>. 6.4.2023.
- Kantamneni, S. 2022c. User Experience Design. <https://urly.fi/39BU>. 6.4.2023.
- Kantamneni, S. 2022d. User Experience Design. <https://urly.fi/39BV>. 6.4.2023.
- Kämäräinen, A. 2019. Verkkosivut toiminimiyrityksen markkinointikanavana – Case toiminimi Antti Henrik.
https://www.theseus.fi/bitstream/handle/10024/265313/K%c3%a4m%c3%a4r%c3%a4inen_Antti.pdf?sequence=2&isAllowed=y.
1.1.2023.
- LePage, p & Andrew, R. 2022. Responsive web design basics. <https://web.dev/responsive-web-design-basics/>. 13.4.2023.
- Li, X., Li, Y. & Jae M. 2020. Neural network's selection of color in UI design of social software. <https://urly.fi/37sX>. 18.4.2023.

node.js 2023. Overview of Blocking vs non-Blocking.

<https://nodejs.org/en/docs/guides/blocking-vs-non-blocking>.

27.3.2023.

Popovic, B. 2022a. Front End Development Course, HTML. Brainnest yrityksen pitämä webinaari. 26.6.2022.

Popovic, B. 2022b. Front End Development Course, CSS. Brainnest yrityksen pitämä webinaari. 3.7.2022.

Popovic, B. 2022c. Front End Development Course, JavaScript. Brainnest yrityksen pitämä webinaari. 11.7.2022.

React 2023 React. <https://react.dev/>. 27.3.2023.

React Router 2023. Feature overview.

<https://reactrouter.com/en/main/start/overview>. 20.9.2023.

Sernoff, L. 10.4.2022. 11 Elements of Modern Website Design.

<https://www.wix.com/blog/2022/03/modern-website-design/>.

7.5.2023.

Vello 2023. Vertaile Vello-paketteja. <https://www.vello.fi/fi/vertaile-paketteja>.

11.1.2023.

Velázquez, D. 2023. Responsive Web Illustration.

<https://pixabay.com/illustrations/responsive-illustration-page-seo-2044932/>. 13.4.2023.

Visual Studio Code 2023a. Getting Started. <https://code.visualstudio.com/docs>. 27.3.2023.

Visual Studio Code 2023b. Visual Studio Code on Windows.

<https://code.visualstudio.com/docs/setup/windows>. 27.3.2023.

Webb, H. 2023. Best Programming Languages for Web Development.

<https://www.computerscience.org/bootcamps/guides/programming-languages-web-development/>. 3.4.2023.