



## **Yksityisrittäjän kotisivujen kehitys ja käyttöönotto**

Vilma Laurila

Haaga-Helia ammattikorkeakoulu

Tradenomi

Opinnäytetyö

2025

## Tiivistelmä

<b>Tekijä(t)</b> Vilma Laurila
<b>Tutkinto</b> Tradenomi
<b>Opinnäytetyön nimi</b> Yksityisyrittäjän kotisivujen kehitys ja käyttöönotto
<b>Sivu- ja liitesivumäärä</b> 28 + 4
<p>Tämän opinnäytetyön tarkoituksena on suunnitella, toteuttaa ja julkaista verkkosivut koirahoitopalveluja tarjoavalle yritykselle. Tavoitteena on luoda kotisivut React-kirjastoa ja Material UI-komponenttikirjastoa käyttäen, sekä kerätä analyysidataa Google Firebase-sovellusalustan kautta. Kotisivut toteutettiin Scrumia seuraten. Lopuksi yrittäjä perehdytetään kotisivujen päivittämiseen ja analytiikkadatan lukemiseen, jotta sivusto olisi mahdollisimman pitkään käyttökelpoinen ja ajantasainen.</p> <p>Länsituulen koirahoitopalvelut on yhden henkilön yritys, joka nimensä mukaisesti tarjoaa koirahoitopalveluita sopimusyrittäjyyden kautta. Yrityksellä ei ole ennestään verkkosivuja, ja muidenkin sosiaalisen median palveluiden käyttö on hyvin rajallista. Tämä opinnäytetyö syntyi nimenomaan tarpeesta saada yritykselle toimivat verkkosivut. Opinnäytetyön toteutukseen on varattu aikaa maaliskuusta 2024 saman vuoden joulukuuhun.</p> <p>Tämän opinnäytetyön tekoprosessi vahvistaa tekijän osaamista varsinkin React-kirjaston käytöstä ohjelmistokehityksen näkökulmasta. Osaaminen ja ymmärrys Reactista kasvavat, ja vaikka näissä kotisivuissa ei käytetäkään kovin monimutkaista React-ohjelmointia, kasvaa tekijän itsevarmuus ja perusosaaminen merkittävästi.</p> <p>Opinnäytetyö alkaa johdannolla, jossa kerrotaan, mitä opinnäytetyö pitää sisällään ja millainen on verkkosivustojen tärkeys nykymarkkinoinnissa. Johdannossa myös rajataan pois tehtävät, jotka eivät kuulu tähän opinnäytetyöhön. Tietoperusta rakentuu kehitystyön vaiheista, digitaalisesta markkinoinnista ja verkkopalvelun luonnista, käytettävyydestä ja saavutettavuudesta. Lisäksi esitellään verkkosivuston luonnissa käytettyjä tekniikoita ja teknologioita, kuten esimerkiksi React, Material UI ja Googlen Firebase. Työn empiirisessä osassa käydään tarkasti läpi verkkosivuston suunnittelu, luonti ja julkaisu nimenomaan React-projektin näkökulmasta. Lopuksi esitellään valmis tuotos ja analysoidaan, kuinka sen toteutus onnistui ja mitä voisi mahdollisesti tehdä toisin.</p>
<b>Asiasanat</b> React-kirjasto, web-kehitys, www-sivut

# Sisällys

1	Johdanto .....	1
1.1	Sanasto.....	2
2	Tietoperusta .....	4
2.1	Kehitystyön vaiheet.....	4
2.2	Digitaalinen markkinointi.....	5
2.3	Verkkosivujen luonti.....	6
2.4	Verkkopalvelun käytettävyys ja saavutettavuus.....	7
3	Käytetyt tekniikat ja teknologiat.....	8
3.1	React-kehys.....	8
3.2	Material UI-komponenttikirjasto .....	9
3.3	Google Firebase-sovellusalusta.....	9
3.4	Scrum.....	9
4	Verkkosivujen toteuttamisprosessi.....	11
4.1	Syyt verkkosivujen toteutukselle.....	11
4.2	Verkkosivujen suunnittelu .....	11
4.3	Verkkosivujen toteutus.....	14
4.4	Projektin käyttöönotto .....	21
5	Yhteenveto .....	23
	Lähteet.....	25
	Liitteet .....	29
	Liite 1. Ohje sivuston päivittämiseen GitHubin kautta .....	29

# 1 Johdanto

Tämän opinnäytetyön tavoitteena on toteuttaa kotisivut koirahoitopalveluita tarjoavalle yritykselle nimeltään Länsituulen koirahoitopalvelut. Yrityksellä ei ole ennestään kotisivuja, ja muidenkin sosiaalisten medioiden käyttö on minimaalista, joten tarve kotisivuille on merkittävä, jotta yritys pystyisi tavoittamaan mahdollisimman monta nykyistä ja tulevaa asiakasta. Opinnäytetyön aihe syntyikin nimenomaan tämän tarpeen pohjalta.

Yrittäjä toimii sopimusyrittäjänä koirahoitopalveluita tarjoavassa yrityksessä. Sopimusyrittäjä on yksityisyrittäjä, joka maksaa omasta ansaitsemastaan liikevaihdosta tietyn prosenttiyksikön verran korvausta liikkeelle, jossa hän työskentelee, ja tätä korvausta vastaan hän saa käyttöönsä liikkeen välineet ja laitteet. Tämä eroaa niin sanotusta vuokratuoliyrittäjyydestä niin, ettei omia välineitä tarvitse välttämättä hankkia, eikä kiinteää kuukausivuokraa makseta. (Lakihelppi s.a.)

Opinnäytetyön tarkoituksena on suunnitella, toteuttaa ja julkaista nimenomaan verkkosivut yritykselle. Sivut suunnitellaan niin, että ne olisivat mahdollisimman siistit ja saavutettavat kaikilla laitteilla ja erikokoisilla näytöillä. Mobiilisovellusta ei kuitenkaan ole tarkoitus tehdä, ainoastaan verkkosivu, joka skaalautuu siististi mobiilistikin. Vaikka mobiilisovellus tai jokin muu vastaava digitaalinen palvelu olisi varmasti yrittäjälle hyödyllinen verkkosivujen lisäksi, tulisi opinnäytetyöstä liian laaja, jos sen aikana tehtäisiin kaksi erilaista työtä, varsinkin kun nykypäivänä verkkosivujen tekemiseen kuuluu oleellisesti myös sivuston responsiivisuudesta huolehtiminen. Sivuston julkaisun jälkeen tavoitteena on perehdyttää yrittäjä sivuston ylläpitämiseen, jotta sivusto olisi käyttökelpoinen ja ajantasainen mahdollisimman pitkään.

Nykyaikana verkkosivut ovat oleellinen osa yrityksen myyntiä ja brändiä, ja suurin osa asiakkaista ottaakin selvää yritysten palveluista etukäteen verkosta. Verkkosivut ovat jatkuvasti saatavilla, ja ne antavat suoraan hyvän tai huonon kuvan asiakkaalle. Verkkosivut ovatkin digitaalisen markkinoinnin peruspilari ja perusta, ja on tärkeää, että sivusto on paitsi esteettisesti miellyttävä ja helpokäyttöinen kaikilla laitteilla, myös latautuu nopeasti ja löytyy helposti hakukoneista. (Suomen digimarkkinointi s.a.b.)

Opinnäytetyön aikana työn tekijä saa tarkempaa ja syvällisempää osaamista React-sovelluksen kehittämisestä ja verkkosivustoprojektin eri vaiheista. Tämä työ esittelee tarkasti React-sovelluksen ja verkkosivujen kehityksen eri vaiheita, ja tarjoaa näkökulmaa React-projektin tekoon. React-kehitys on laaja ja sillä on mahdollista tehdä monenlaisia verkkosovelluksia, ja tämä työ esittelee yhdenlaisen sovelluksen teon aina suunnittelusta sivuston julkaisuun asti, koodikirjoituksen perspektiiviä unohtamatta.

## 1.1 Sanasto

Backend	Backend-ohjelmointi on palvelinpuolen ohjelmointia, jossa luodaan sovelluksen tai nettisivun toimintaa – kaikkea, mitä sivun tai sovelluksen käyttäjä ei suoraan näe. Backend-palvelut ovat yleensä kolmannen osapuolen tarjoamia palveluita, joilla tuetaan sovelluksen tai nettisivun backendin toimintaa. (Coursera 3.4.2024; Indeed 27.6.2024.)
DOM	Document Object Model, standardi internetsivustojen rakenteen esittämiseksi. Se voidaan nähdä puuna, joka rakentuu eri objekteista. Erilaiset ohjelmat ja skriptit voivat muuttaa DOM:n rakennetta, näin muodostaen verkkosivun eri osat. (W3Schools s.a.a.)
Frontend	Frontend-ohjelmointi on käyttäjäpuolen ohjelmointia, missä keskitytään siihen, miltä nettisivu tai sovellus näyttää. Sen tärkeimpänä tehtävänä on käyttöliittymä ja sen luonti. (W3Schools s.a.b.)
Hostaus	Palvelu, jonka avulla verkkosivu saadaan näkyviin internetkäyttäjille. Sivujen julkaisija ostaa palvelintilaa toiselta osapuolelta, jossa verkkosivujen dataa säilytetään, ja jonka avulla sivusto yleensä latautuu käyttäjille nopeammin. (Tambe 20.2.2024.)
React-komponentit	Käyttöliittymäkomponentteja, joita voi käyttää uudelleen. Komponentit määritellään yleensä funktioina, ja ne palauttavat uudelleenkäytettäviä käyttöliittymän rakenneosia. (React s.a.; W3Schools s.a.b.)
Product Backlog	Jatkuvasti muotoutuva, jäsenelty lista tilatun tuotteen ominaisuuksista, josta Scrum-tiimi valitsee yhden sprintin aikana tehtävät työt (Scrum Guides 2020).
Props	Tulee sanasta <i>properties</i> , React-komponenttiin sisään tulevaa dataa, jota komponentti ei itse voi muuttaa (Patterns s.a.).
Renderöinti	Sovelluksen käyttöliittymän päivittyminen. React kutsuu komponentteja ja näyttää ne sivulla. (React 2024a.)
Scrum Master	Vastaa Scrum-tiimin työstä tiimin johtajana (Scrum Guides 2020).
Uudelleenrenderöinti	Sovelluksen käyttöliittymän päivittyminen tehtyjen muutosten jälkeen (React 2024a).

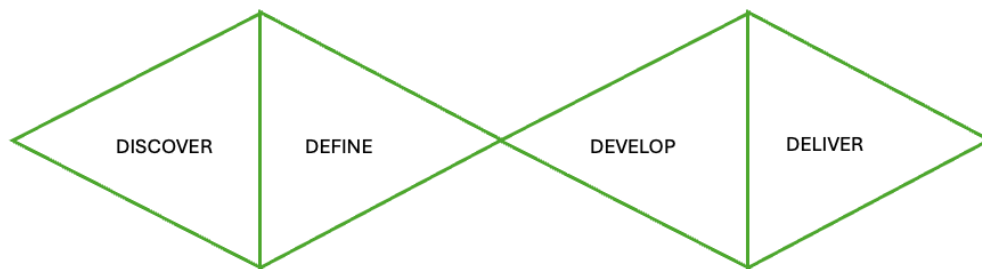
Virtuaalinen DOM	Laitteen muistissa oleva kopio DOM:sta (Santana Roldán 2023, luku 1).
Yhden sivun applikaatio	Single-page application (SPA), on sovellus, joka useiden erilaisten sivujen sijaan lataa ainoastaan yhden verkkosovelluksen, ja tarvittaessa uudelleenkirjoittaa sen sisältöä. Näin ollen palvelimelta ei tarvitse ladata useita sivuja. (MDN Web Docs s.a.)

## 2 Tietoperusta

Verkkosivujen luontiin liittyy useampia askeleita ennen kuin verkkosivut ovat valmiit julkaistavaksi. Verkkosivujenkin luonti on omanlaisensa kehitysprosessi, jonka vaiheita käsitellään tässä luvussa. Luvussa käydään myös tarkemmin läpi verkkosivujen luontiin liittyviä työvaiheita ja kuinka ne vaikuttavat lopulliseen tulokseen. Sen lisäksi tutustutaan digitaaliseen markkinointiin ja palvelun käytettävyyteen ja saavutettavuuteen nimenomaan verkkosivujen näkökulmasta.

### 2.1 Kehitystyön vaiheet

Kehitystyö lähtee yleensä käyntiin työelämästä saadulla toimeksiannolla, jonka tarkoituksena on luoda tuotos ja yleensä myös kehittämisehdotuksia toimeksiantajalle, tässä tapauksessa yrittäjälle (JAMK 27.5.2022). Palvelumuotoilun näkökulmasta katsottuna palveluiden kehittämisellä on yleensä neljä päävaihetta: tutkimus, määrittely, toteuttaminen ja toimittaminen. Tämä perustuu British Design Councilin tuplatimanttiajatteluun (double diamond), jossa ensimmäinen timantti sisältää ongelman ymmärtämisen ja sen ratkaisun, ja toinen timantti ratkaisun kehittämisen. (Palvelumuotoilu Palo 8.12.2018.) Tuplatimanttia ja sen vaiheita on havainnollistettu kuvassa 1.



Kuva 1. Tuplatimantti (mukaillen Palvelumuotoilu Palo 8.12.2018)

Kehittämistyön ensimmäinen vaihe on nykytilanteen kartoitus, jonka tärkeimpinä tavoitteina on löytää kehittämisen tavoitteet yhdessä asiakkaan kanssa, rajata kohderyhmä ja selvittää jo olemassa olevat ratkaisut, ja miten niitä voisi hyödyntää (Palvelumuotoilu Palo 8.12.2018.). Tämä yleensä tarkoittaa olemassa olevan materiaalin läpikäymistä ja sen hyödyntämistä uudessa projektissa sekä kohderyhmän ja heidän tarpeidensa pohdintaa (Humble s.a.; UXPin 5.8.2024).

Seuraavassa vaiheessa tutustutaan tarkemmin kohderyhmään, ja tehdään jatkosuunnitelma heidän perusteellaan. Tärkeintä on kartoittaa kohderyhmän tavoitteet ja tarpeet, ja miten niitä voisi hyödyntää palvelun kehittämisessä. (Palvelumuotoilu Palo 8.12.2018.) Tässä vaiheessa yleensä kartoitetaan asiakkaan halut ja tarpeet, pyritään löytämään projektin tärkein ydin ja luodaan alustava suunnitelma käsillä olevasta projektista (Humble s.a.; UXPin 5.8.2024).

Kolmannessa vaiheessa kehitetään erilaisia ideoita lopputuotteesta, ja kokeillaan erilaisia menetelmiä ja ratkaisuja. Vaiheen aikana ideat jalostuvat ja niitä tarkastellaan eri näkökulmista, jolloin voidaan valita parhaimmat ideat ja lähteä jatkokehittämään niitä. (Palvelumuotoilu Palo 8.12.2018.) Tässä vaiheessa ideoidaan, luodaan prototyyppejä lopullisesta projektista ja tehdään testausta, jotta parhaat mahdolliset vaihtoehdot löydetään (UXPin 5.8.2024).

Viimeisessä vaiheessa viimeistellään edellisessä vaiheessa jatkokehitykseen valitut ideat, ne toteutetaan ja julkaistaan. Osana tätä vaihetta on testaus, jolla varmistetaan kokonaisuuden toimivuus. (Palvelumuotoilu Palo 8.12.2018.) Prototyyppejä ja muita ratkaisuja testataan asiakkailla, jotta palautetta toimivuudesta saadaan mahdollisimman aikaisin. Tässä vaiheessa tärkeintä on valmistella projektia julkaisua varten. (Humble s.a.)

## **2.2 Digitaalinen markkinointi**

Digitaalinen markkinointi on markkinointia, jossa asiakkaita pyritään tavoittamaan erilaisten digitaalisten kanavien, kuten verkkosivujen, verkkokaupan tai erilaisten sovellusten kautta. Digitaalinen mainonta on osa digimarkkinointia, ja sen pyrkimys on nimenomaan saavuttaa asiakkaat esimerkiksi videomainonnan, hakusanamainonnan tai sosiaalisessa mediassa tapahtuvan mainonnan avulla. (Sanoma 16.8.2022.) Digimarkkinoinnin suurin hyöty on sen tavoitettavuuden laajuus: asiakkaita on mahdollista saada ympäri maailman, vaikka yleensä tavoitteena onkin saavuttaa suppeampi, lähellä oleva asiakaskunta. Digimarkkinointia onkin helppo kohdentaa, ja oikein tehtynä kohdentaminen tavoittaa vanhat ja uudet, samanmieliset asiakkaat tehokkaasti ja yleensä perinteistä markkinointia edullisemmin. (Alma Media s.a.; Meio s.a.).

Digimarkkinointi sisältää monia erilaisia tapoja ja keinoja saavuttaa asiakas. Tärkeimpänä pidetään yrityksen verkkosivuja, sillä yleensä palveluiden etsiminen alkaa verkosta. Verkkosivujen tulisi olla selkeät ja ilmeeltään yhteneväiset, toimia sekä tietokoneen että mobiililaitteen näytöllä, ja löytyä hakukoneista. (Alma Media s.a.; Sanoma s.a.) Verkkosivut, kuten muutkin digitaalisen markkinoinnin keinot, mahdollistavat datan ja analytiikan, esimerkiksi kävijämäärän, keräämisen, ja tätä dataa voidaan hyödyntää markkinoinnin ja sen kanavien tehostamisessa ja muokkaamisessa (Alma Media s.a.).

Meio Oy:n (s.a.) mukaan tärkeintä nykypäivänä on ymmärtää, että ilman digitaalista markkinointia moni potentiaalinen asiakas ei löydä yritystä ja sen palveluita, mikä vaikuttaa suoraan heikentävästi yrityksen myyntiin ja liiketoiminnan kasvuun; toisaalta oikein tehtynä digimarkkinointi on kustannustehokasta ja laajasti hyödynnettävissä yrityksen tarpeisiin ja kasvuun.

### 2.3 Verkkosivujen luonti

Verkkosivujen kehittämisvaiheessa tehdään määrittely, johon kuuluu muun muassa sivuston teeman, päämäärän ja kohdeyleisön pohdinta, vaatimukset työn laadusta ja määrästä sekä lopullisen tuotteen – tässä tapauksessa verkkosivun – muodosta. Kehittämisvaiheessa myös pohditaan sivujen lopullista designia, sisältöä ja arkkitehtuuria, ja tähän vaiheeseen kuuluu olennaisesti sivuston sisällön keräämistä ja kirjoittamista. (Ben Rebah, Boukthir, Chédebois 2022, luku 4.1.1.) Tärkeintä sivuston suunnittelussa ja sivujen pohjaratkaisun mietinnässä on pitää mielessä paitsi yhteneväinen ulkoasu, myös kohdeyleisö, ja se, että sivustoa olisi mahdollisimman helppo käyttää (Diggity Solutions Pvt. Ltd. 8.2.2023). Kehittämisvaiheen toinen osa on tekninen kehitys, johon kuuluu sivuston rakentaminen suunnitelmaa seuraten. Sivuston sisältö rakennetaan ja sille luodaan testejä, ja sivuston toimintaa ja ulkoasua testataan paikallisesti kehittäjän omassa selaimessa. (Ben Rebah, Boukthir, Chédebois 2022, luku 4.1.1.4.) Testaus on tärkeä osa sivuston kehittämistä, sillä testien avulla löydetään ohjelmointivirheitä (bugeja) ja muita korjattavia asioita, jotka on hyvä huomioida ennen sivuston julkaisua (Diggity Solutions Pvt. Ltd. 8.2.2023).

Seuraava askel on pohtia sivuston hostaamista, eli sivuston julkaisua internetissä: kuinka suunniteltu ja luotu sivusto saadaan kaikkien näkyville internetiin palvelimen kautta. Hostaamista on erilaisia muotoja: ylläpitäjä voi varata oman maksullisen verkkotunnuksen ja maksaa toisen osapuolen palvelimesta, jotta sivusto saadaan näkyväksi. Toinen vaihtoehto on ilmaiset hostauspalvelut, joihin yleensä liittyy sivustolla näkyvää mainontaa ja/tai verkkotunnusta ei ole mahdollista yksilöidä samalla tavoin. (Ben Rebah, Boukthir, Chédebois 2022, luku 4.1.2.)

Sivuston luojan ja ylläpitäjän tulee myös huolehtia sivuston hakukoneoptimoinnista, eli kuinka saada sivusto näkyväksi hakukoneiden tuloksissa. Kehitysvaiheessa tulee huolehtia, että hakukoneet pystyvät löytämään sivuston, muun muassa rakentamalla sivuston niin, että se noudattaa yleisesti käytössä olevia käytänteitä, kuten esimerkiksi, että sivuston nimi löytyy <title></title> - tunnisteiden välistä. (Ben Rebah, Boukthir, Chédebois 2022, luku 4.1.3.) Esimerkiksi Reactilla luodut sivut ovat hakukoneystävällisiä, sillä hakukoneet suosivat tuloksiinsa sivustoja, jotka latautuvat ja renderöityvät nopeasti (Kundariya, H. 17.12.2021).

Ben Rebahin, Boukthirin ja Chédebois'n (2022, luku 4.1.4.) sekä Diggity Solutions Pvt. Ltd:n (8.2.2023) mukaan verkkosivujen luonnin viimeinen vaihe on sivuston päivittäminen ja päivityksistä huolehtiminen: sivusto tulee pysyä ajantasaisena ja saavutettavana. Päivitysten ei aina välttämättä tarvitse kohdistua sivuston jokaiseen sivuun, vaan ainoastaan päivityksen tarpeessa oleviin, jolloin palvelimelle ladataan vain uudet versiot tietyistä sivuista.

## 2.4 Verkkopalvelun käytettävyys ja saavutettavuus

Palvelun käytettävyydellä tarkoitetaan yleensä sen helppokäyttöisyyttä. Se vastaa muun muassa kysymyksiin siitä, kuinka helposti käyttäjät suoriutuvat palvelun käytöstä ensimmäisellä käyttökeralla, kuinka helppoa palvelun käyttöön on palata tauon jälkeen, ja kuinka paljon virheitä käyttäjät tekevät palvelua käyttäessään. Käytettävyys on osa käyttäjäkokemusta, mutta ne eivät ole synonyymeja: käyttäjäkokemus sisältää myös käyttäjän tunteita ja odotuksia, kun taas käytettävyys keskittyy nimenomaan palvelun helppokäyttöisyyteen, ja kuinka tehokasta palvelun käyttö ylipäätään on. (Haltu 18.10.2023.)

Saavutettava palvelu on suunniteltu ja toteutettu kaikki käyttäjäryhmät huomioiden. Saavutettavan verkkopalvelun sisältö on kaikkien käyttäjien saavutettavissa ja käytettävissä, ja se pitää sisällään muun muassa selkeästi esitetyt sisällöt, ymmärrettävän kieliasun ja intuitiivisen navigoinnin. Jos saavutettavuus on pidetty mielessä verkkopalvelu suunniteltaessa ja toteuttaessa, palvelee se esimerkiksi näkö- ja kuulovammaisia, ikääntyneitä käyttäjiä ja kielivähemmistöjä, ja avaa yritykselle ovia yhä laajempaan asiakaskuntaan. (Haltu 31.3.2023.)

Verkkopalvelun saavutettavuutta tarkasteltaessa otetaan yleensä huomioon sen tekninen saavutettavuus ja sisällön saavutettavuus. Teknisellä saavutettavuudella tarkoitetaan asianmukaista koodausta, ja lupausta siitä, että sivusto toimii myös erilaisilla lukuohjelmilla. Sisällön saavutettavuuteen kuuluu verkkopalvelun selkeä kieliasu ja helppolukuisuus. Jos molemmat saavutettavuuden osa-alueet on otettu huomioon, on verkkopalvelulla mahdollisuudet tarjota sisältöä kaikille, myös niille, joilla on yleensä rajoitteita digipalveluiden käytössä. (Haltu 31.3.2023.)

Saavutettavuus on otettava huomioon verkkopalvelun suunnittelussa ja kehityksessä. Suunnitteluvaiheessa tulee kiinnittää huomiota esimerkiksi käytettyihin väreihin ja fontteihin, kontrasteihin, linkkien ulkoasuun ja liikkuvan tai välkkyvän sisällön välttämiseen. Kehitysvaiheessa tulee muun muassa huomioida eri otsikkotasot – pääotsikot on hyvä olla 

# </h1> - tagien sisällä ja loput otsikot hierarkkisesti sen alla. Kuville tulisi määritellä vaihtoehtoiset tekstit, sillä verkkopalvelua luultavasti tullaan käyttämään eri selaimilla ja laitteilla, ja verkkosivun tulisi responsiivisesti skaalautua suosituimmille näyttökoille. Saavutettavuutta on myös hyvä testata ennen palvelun julkaisua esimerkiksi käyttäjätestauksella, ja testaamalla palvelua eri selaimissa ja eri laitteilla. Näin varmistetaan verkkopalvelun mahdollisimman laaja saavutettavuus. (Haltu 31.3.2023.)

## 3 Käytetyt tekniikat ja teknologiat

Tässä luvussa käydään läpi verkkosivujen luonnissa käytetyt tekniikat ja teknologiat. Ensimmäiseksi perehdytään React-kehykseen ja sen ominaisuuksiin, jonka jälkeen tarkastellaan Material UI-komponenttikirjastoa ja kuinka se keskustelee Reactin kanssa. Lopuksi keskitytään vielä Googlen Firebase-sovellusalueeseen, ja erityisesti sen tarjoamiin analytiikkapalveluihin, sekä Scrum-viitekehykseen.

### 3.1 React-kehys

React on Facebookin kehittämä ja Metan ylläpitämä avoimen lähdekoodin JavaScript-kehys, joka on saavuttanut paljon suosiota web- ja sovelluskehittäjien keskuudessa. Reactin suosio perustuu muun muassa sen helppokäyttöisyyteen, komponenttien uudelleenkäytettävyyteen, hakukoneoptimointiin ja kuinka se hyödyntää DOM:ia. (Kundariya 17.12.2021.) React käyttää virtuaalista DOM:ia, joka vertaa nykyistä DOM:ia tehtyihin muutoksiin, ja näin vähentää DOM:in päivitysten määrää (Santana Roldán 2023, luku 1).

React perustuu komponenttiohjelmointiin, joiden avulla ohjelmoija voi rakentaa kokonaisia, käyttäjätasoisia sivustoja tai sovelluksia (React 2024b). Komponentit ovat itsenäisiä koodin osia, joita voi käyttää uudelleen. Ne vastaanottavat erilaisia syötteitä, propseja, ja palauttavat React-elementtejä. (Patterns s.a.) Elementit ovat koodin osia, jotka kertovat, mitä näytöllä näkyy. Ne luodaan usein JSX-syntaksilla, joka mahdollistaa HTML-koodin kirjoittamisen suoraan JavaScript-koodin sekaan. (Santana Roldán 2023, luku 1.)

Verrattuna Angulariin, joka on myös frontend-ohjelmoinnissa käytetty, suosittu JavaScript-kirjasto, React on kevyempi käyttää ja nopeampi oppia. Koska React sisältää virtuaalisen DOM:in ja renderöinnin eri mahdollisuudet, se tarjoaa tehokkaampaa ohjelmistokehitystä ja vähentää virheiden määrää. (Raval 1.4.2024.) React-kirjastoa ylläpitää paitsi Meta, myös suuri joukko itsenäisiä ohjelmistokehittäjiä ympäri maailman, ja sitä käyttää Metan omistamat palvelut, kuten Facebook ja Instagram, sekä muun muassa Netflix. Toisaalta Angular tarjoaa Googlen tuen ja tarkan dokumentaation, ja sitä käyttääkin moni suuryritys, mukaan lukien Google itse sekä Microsoft ja PayPal. (Powell 29.12.2023.)

Reactia hyödyntävä Next.js on myös vaihtoehto JavaScript-ohjelmoinnille. Next.js perustuu Node.js:ään ja Babeliin, mutta se hyödyntää Reactia, ja sillä pystyy rakentamaan nopeasti lataantuvia ja käyttäjätasoisia yhden sivun applikaatioita. Renderöinti tapahtuu palvelimen puolella, ja staattisten verkkosivujen lataaminen on nopeaa. (Patadiya 1.4.2024.) Sitä voidaan käyttää rakentamalla Reactilla luodun käyttöliittymän päälle erilaisia toimintoja, kuten datan hakua ja sivujen

reititystä (Next.js s.a.). Jos kuitenkin on tarkoitus rakentaa suurempia, useamman sivun kokonaisuuksia, joka käyttää monimutkaisempaa sivujen reititystä, on React yleensä parempi valinta. Staattisten nettisivujen tekoon Next.js on ehdottomasti suositumpi. (Patadiya 1.4.2024.)

### **3.2 Material UI-komponenttikirjasto**

Material UI on avoimen lähdekoodin muotoiluun keskittynyt komponenttikirjasto, joka toteuttaa Googlen Material Designia. Material UI tarjoaa yli neljäkymmentä valmiista komponenttia, kuten painikkeita, taulukkoja ja ikoneja. Tarjolla olevia komponentteja on helppo muokata käyttäjän omiin tarpeisiin, niin ulkonäöllisesti kuin toiminnallisestikin, ja sitä käyttääkin laajasti suuremmatkin yritykset, kuten esimerkiksi Netflix ja NASA. (Material UI 2024.)

Material UI:ta on kehittänyt tuhannet avoimen lähdekoodin kehittäjät, ja sen komponentit ovat valmiina käytettäväksi myös ilman muutoksia tai muokkauksia. Se vaatii erillisen asennuksen, mutta asennuksen jälkeen käyttäjä voi käyttää kaikkia valmiita komponentteja omassa koodissaan. (Material UI s.a.)

### **3.3 Google Firebase-sovellusalusta**

Firebase on Googlen ylläpitämä, pilvessä olevia backend-palveluja tarjoava sovellusalusta. Se mahdollistaa sovellusten rakentamisen ja ylläpidon, ja sen backend-palvelut ovat käytössä monella eri suuryrityksellä, muun muassa Duolingolla ja The New York Timesilla. Sen tarjoamia palveluja ovat esimerkiksi tietokantapalvelut, autentikaatio, hostaus, Googlen analytiikka ja koneoppiminen. (Firebase s.a.a)

Firebasen Realtime Analytics tarjoaa ylläpitäjälleen tietoa verkkosivun tai mobiilisovelluksen käyttäjien käytöksestä, ja antaa reaaliaikaista dataa esimerkiksi käyttäjämääristä. Se myös mahdollistaa analytiikkadatan visualisoimisen ja datan validoimisen. (Firebase s.a.b.)

### **3.4 Scrum**

Scrum on ketterän kehityksen viitekehys, jota seuraamalla ihmiset, tiimit ja yritykset voivat saada aikaan ratkaisuja monitahoisiin ongelmiin. Scrumin viitekehys auttaa tiimiä työskentelemään yhdessä. Scrumissa Scrum Master pitää huolen, että tuoteomistajan tilaamat ominaisuudet ovat osana Product Backlogia, josta Scrum Team valitsee aina kehityssprintin aikana tehtävät ratkaisut, joita yleensä valitaan viisi ja lopullisia tuotteita pyritään tuottamaan ainakin kolme. Tämän jälkeen tiimi ja sidosryhmien jäsenet tarkastelevat tuotoksia, joiden pohjalta tehdään suunnitelmat ja muutokset seuraavaa sprinttiä varten. (GeeksForGeeks 5.12.2023; Scrum.org s.a.; Scrum Guides 2020.)

Yksi sprintti voi olla yhdestä viikosta useaan viikkoon, ei yleensä kuitenkaan neljää viikkoa kauempaa. Seuraava sprintti alkaa aina heti edellisen päätyttyä, ja siihen sisältyy sprintin suunnittelu, päivittäiset tapaamiset eli scrumit, sprintin katselmointi ja sprintin retrospektiivi. Sprintin aikana ei tehdä sellaisia muutoksia, jotka voisivat vaarantaa kyseisen sprintin päämäärän, eli tietyn osa-alueen valmistumisen. Tärkeintä on, ettei työn laatu huonone, ja että Product Backlogia voidaan tarkastella ja muuttaa tarvittaessa. Sprinttiin yleensä valitut tehtävät eivät kuitenkaan muutu sprintin aikana. (GeeksForGeeks 5.12.2023; Scrum Guides 2020.)

## 4 Verkkosivujen toteuttamisprosessi

Länsituulen koirahoitopalvelujen verkkosivujen toteutusprosessi lähti tarpeesta saada toimivat verkkosivut, sillä aiempia verkkosivuja ei ollut olemassa. Verkkosivut ovat tärkeä osa yrityksen markkinointia, joten tarve niille oli suuri. Prosessi aloitettiin miettimällä, millaisia ominaisuuksia verkkosivulle halutaan, ja kuinka ne voisi toteuttaa. Sivujen ulkoasu suunniteltiin tekemällä alustavat sivukartat jokaisesta sivusta, joiden pohjalta sivujen rakentaminen aloitettiin. Alkuperäisiä suunnitelmia muokattiin toteutusprosessin aikana saadusta palautteesta ja yrittäjän tarjoamista sivuston sisällöistä. Sivujen rakentaminen käytiin läpi vaihe vaiheelta Scrumin sprinttiajattelua noudattaen, ja lopuksi verkkosivut otettiin käyttöön.

### 4.1 Syyt verkkosivujen toteutukselle

Kuten aiemmin on todettu, Länsituulen koirahoitopalveluilla ei ole aiemmin ollut verkkosivuja, jossa se pystyisi esittelemään tarjoamiaan palveluita ja tavoittamaan mahdollisimman monta asiakasta, niin nykyisiä kuin tuleviakin. Verkkosivut ovat nykypäivänä yritysten yksi tärkeimmistä markkinoinnin kanavista, jonka takia tarve toimiville ja helppokäyttöisille verkkosivuille oli helposti tunnistettavissa. Verkkosivujen tarkoituksena on houkutella asiakkaita tarjoamalla tietoa yrityksen tarjoamista palveluista ja esittelemällä yritystä ja yrittäjää tarkemmin.

Verkkosivut tehtiin täydentämään yrityksen muuta sosiaalisen median, lähinnä Instagramin, markkinointia. Tämä muu sosiaalisen median markkinointi on kuitenkin hyvin pientä ja toteutettu pienessä koossa, joten hyvin suunnitellut ja toteutetut verkkosivut täydentäisivät Instagramissa tapahtuvaa markkinointia sujuvasti ja luonnollisesti. Tarkoituksena oli, että verkkosivut olisivat yrityksen tärkein ja suurin markkinointikanava, jonka takia niiden tulisikin erottua edukseen. Nykypäivänä lähes jokaisella yrityksellä on verkkosivut, ja tärkeintä on erottua massasta: henkilökuvaukset tulisivat olla henkilökohtaisia ja personoituja, sivujen tulisi puhutella nimenomaan asiakkaita, kuvien ja mahdollisten videoiden tulisi olla itseotettuja eikä ilmaisipalveluiden tuottamia, ja sivut tulisivat aina olla ajan tasalla (Forbes Agency Council 21.4.2017). Länsituulen koirahoitopalvelujen tapauksessa tarkoituksena oli luoda verkkosivut, joita yrittäjä voisi niiden julkaisun jälkeen hallinnoida ja muokata tarpeensa mukaan.

### 4.2 Verkkosivujen suunnittelu

Verkkosivujen suunnittelu aloitettiin miettimällä yhdessä yrittäjän kanssa, mitä sisältöä hän haluaisi sivulleen, sekä käytiin läpi jo olemassa olevat sosiaalisen median sisällöt ja pohdittiin kohderyhmää, johon luettiin lopulta 30–60 – vuotiaat, kaikkien sukupuolien koiranomistajat. Kohderyhmään päädyttiin miettimällä yrityksen tämänhetkisiä asiakkaita, jotka olivat verkkosivujen suurin

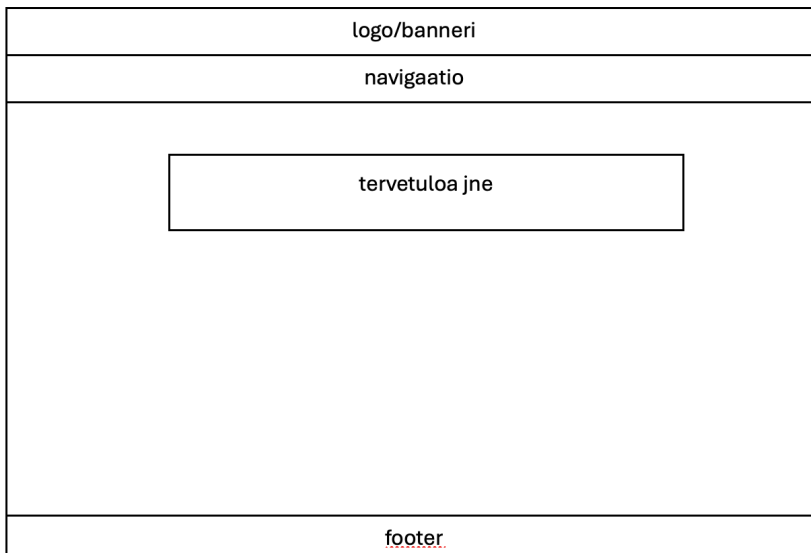
kohderyhmä. Näiden keskustelujen pohjalta päätettiin sivulle luoda yhteensä kolme erilaista sivua: etusivu, minäsivu ja hinnasto. Lisäksi verkkosivuilta tulisi olla pääsy yrittäjän ajanvaraukseen, joka on toteutettu kolmannen osapuolen verkkopalvelua käyttäen. Tässä vaiheessa pohdittiin myös, millaiset verkkosivut palvelisivat aiemmin kartoitettua kohderyhmää parhaiten. Lopulta päädyttiin selkeään ulkoasuun, jossa toistuvat samat värit ja fontit. Sivujen navigoinnista suunniteltiin yksinkertainen, ja päätettiin, että kuvat ja muu sisältö pyritään luomaan mahdollisimman yhteneväisiksi.

Verkkosivusta haluttiin myös tehdä responsiivinen. Responsiivinen verkkosivu toimii kaikilla laitteilla ja näyttökoilla, eli sen sisältö skaalautuu siististi riippuen siitä, millaisella ja minkä kokoisella näyttöllä käyttäjä sivuja katsoo (Suomen Digimarkkinointi s.a.a). Mobiililaitteiden yleistyessä responsiivisuus on entistä tärkeämpää, ja verkkosivut haluttiinkin tehdä toimivaksi sekä tietokoneen, tabletin että puhelimen näytöillä. Erillistä mobiilisovellusta ei kuitenkaan ollut tarkoitus tehdä.

Verkkosivut päätettiin toteuttaa Reactilla, lähinnä tekijän oman mieltymyksen takia. React on laaja kirjasto monipuolisia toiminnallisuuksia, ja se renderöityy ja latautuu nopeasti, ja sen mukana tulee hakukoneoptimointi. Sitä käytetään laajalti erilaisten verkkosovellusten tekoon, ja sen avulla on helppo toteuttaa skaalautuva, responsiivinen verkkosivu (Hutsulyak 1.8.2024). Nämä kaikki olivat tärkeitä seikkoja, joiden perusteella toteutustavaksi valittiin React. Reactin lisäksi toteutukseen valittiin Material UI-komponenttikirjasto hyvin samanlaisista syistä: kirjasto oli tekijälle entuudestaan tuttu, ja sen käytöstä on tehty hyvin helppoa. Se tarjoaa valmiita komponentteja, joiden toiminta koettiin hyödylliseksi juuri tähän projektiin.

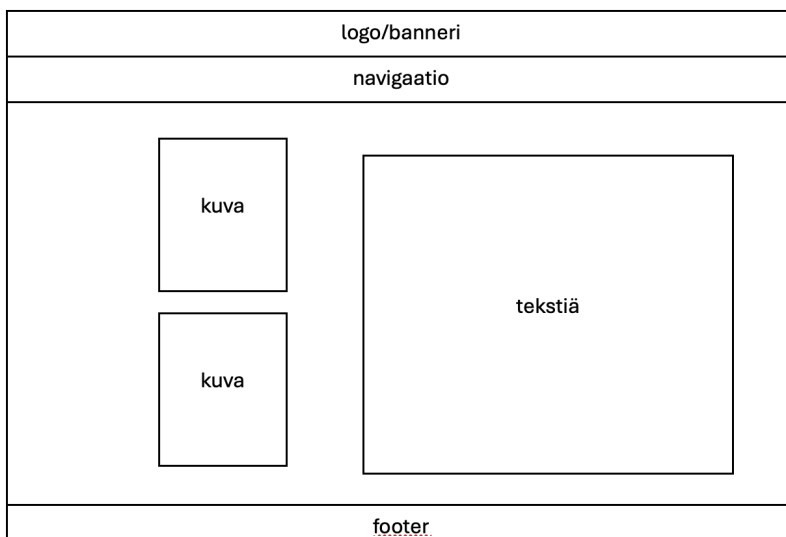
Alustavan keskustelun jälkeen sivujen suunnittelu jatkui sivujen asettelun suunnittelulla. Yhdessä yrittäjän kanssa oli jo alustavasti keskusteltu kunkin sivun sisällöstä, joten oli luontevaa suunnitella seuraavaksi, kuinka sisältö tulisi asetella kullekin sivulle. Aluksi luotiin sivukarttoja, ja tehtiin verkkosivuista ensimmäinen vedos, jossa oli paikallaan osia lopullisesta verkkosivusta: navigointi, värimaailma ja logo. Nämä suunnitelmat loivat myös pohjan sivuston koodaukselle, kun oli tieto siitä, millainen mistäkin sivusta haluttiin. Tässä vaiheessa luonnosteltiin myös sivukarttaa, eli luetteloa kaikista sivuston sivuista, joskin sivuston sivukartta on hyvin pieni ja kevyt johtuen sivujen vähäisestä määrästä (One.com s.a.). Asettelun suunnittelussa päätettiin, että sivuston selkeyden vuoksi yrityksen logo, sivuston navigaatio sekä alatunniste (footer) sijaitsevat jokaisella sivulla samassa kohtaa. Tämä helpotti paitsi asettelun suunnittelua, myös sivuston rakentamista projektin myöhäisemmässä vaiheessa. Näiden suunnitelmien perusteella yrittäjä antoi omia ideoitaan, miten verkkosivua tulisi jatkokehittää, ja sivukarttoja paranneltiin ja jatkoideoita kirjattiin ylös myöhempää kehitystä varten. Hyvin suunniteltu sivun asettelu auttaa sivuston käyttäjää sivujen välisessä navigoinnissa, ja selkeästi tehty asettelu ja visuaalinen ilme parantavat käyttäjäkokemusta (Romano

s.a.) Kuvassa 2 näkyy etusivun asettelun suunnitelma, joka on hyvin karkea, mutta josta tulee esille sivun tärkeimmät sisällöt: logo, navigaatio, sivun sisältö lyhyesti sekä alatunniste.

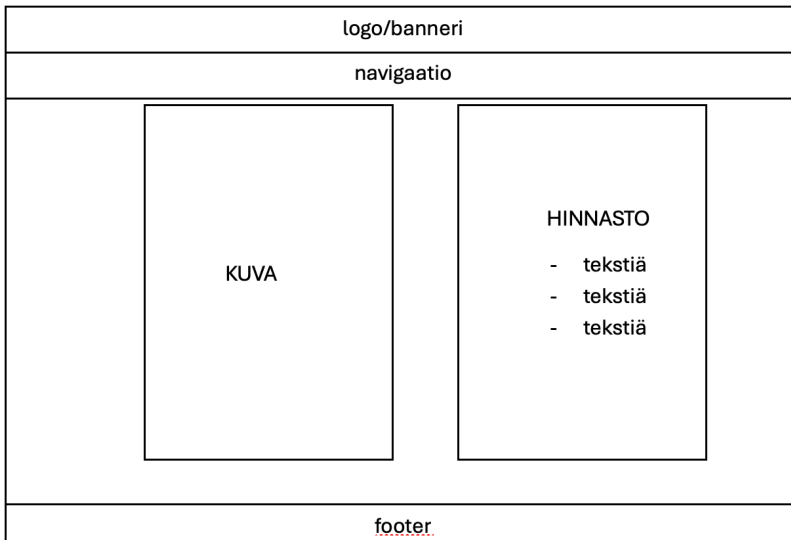


Kuva 2. Etusivun sivusuunnitelma

Koska logo, navigaatio ja alatunniste päätettiin säilyttää samoilla paikoilla jokaisella sivulla, oli toisten sivujen asettelun suunnittelu etusivun asettelun päättämisen jälkeen hyvin suoraviivaista. Kuvissa 3 ja 4 näkyy minäsivun ja hinnaston alkuperäiset suunnitelmat. Suunnitelmat hyväksyttiin yrittäjällä, jonka jälkeen lähdettiin rakentamaan varsinaista sivustoa.



Kuva 3. Minäsivun sivusuunnitelma



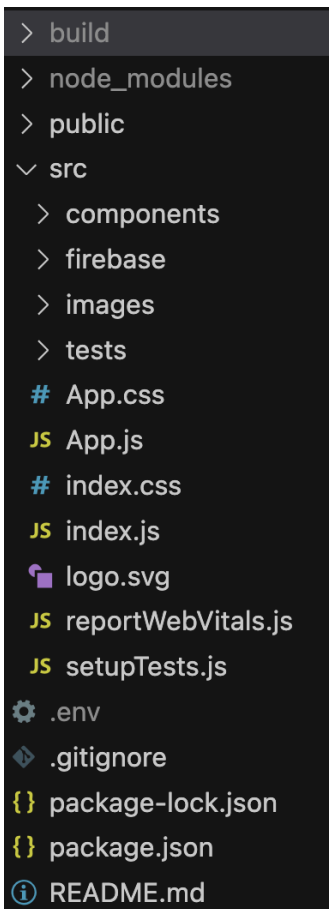
Kuva 4. Hinnaston sivusuunnitelma

### 4.3 Verkkosivujen toteutus

Verkkosivusto toteutettiin Scrumia noudattaen, joskin täydellistä Scrumin kiertoa ei saatu aikaiseksi, koska sivuston tekijöitä oli vain yksi. Scrumia pyrittiin noudattamaan luomalla viikon mittaisia sprinttejä, joiden päätteeksi katselmointiin aikaan saatua työtä ja pohdittiin, miten työn tekemistä voisi tehostaa seuraavaan sprinttiin. Sivuston toteutus kesti yhteensä kolme sprinttiä, ja työt oli jaoteltu niille tasaisesti niin, että kahden ensimmäisen sprintin aikana tehtiin varsinaista sivustoa, ja kolmannen sprintin aikana luotiin testit. Sivuston julkaisu ei kuulunut varsinaisiin sprintteihin, vaan se toteutettiin erikseen sivuston valmistumisen, palautteen keruun ja muokkausten jälkeen, lähinnä koska julkaisuun ja siihen vaadittaviin toimenpiteisiin ei kulunut kokonaista viikkoa, vaan selkeästi lyhyempi aika.

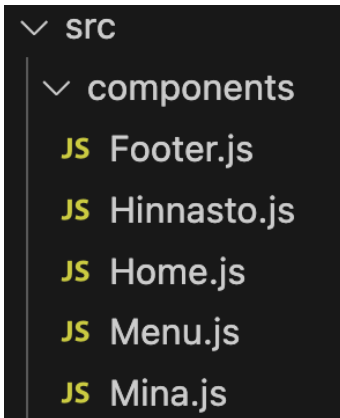
Sivusto rakennettiin Visual Studio Code-ohjelmaa käyttäen, ja komentotulkkina hyödynnettiin MacOS:n omaa komentotulkkia eli terminaalia. Molempien ohjelmien valintaan vaikutti ainoastaan oma mieltymys ja niiden helppokäyttöisyys, koska tekijällä oli niistä omakohtaista kokemusta. Visual Studio Codessa valittiin React-sovelluksen rakenne niin, että samankaltaiset tiedostot olivat aina yhden kansion alla: komponenteille, kuville, testeille ja Firebaseelle päätettiin tehdä omat kansiot, jotta rakenne olisi selkeä ja helppo navigoida koodatessa. Kuvassa 5 on esitelty React-sovelluksen rakenne kansiotasolla, josta näkee kaikki projektin kansiot, sekä kansiota jääneet tiedostot. Kansiottomat tiedostot liittyvät lähinnä sovelluksen hallintaan ja käynnistämiseen, eivätkä kuulu mihinkään muuhun osa-alueeseen: esimerkiksi App.js sisältää käyttöliittymän rakenteen, ja vastaava CSS-tiedosto taas käyttöliittymän tyyllittelyyn. Nämä tiedostot ovat tiedostoja, jotka luodaan automaattisesti, kun Create React App-komento ajetaan, pois lukien env-tiedosto. Kyseinen tiedosto

sisältää kehitysvaiheen ympäristömuuttujia, joka yleensä jätetään julkaisematta, kun projekti on julkaisuvalmis (dotenvx s.a.).



Kuva 5. React-sovelluksen rakenne koodieditorissa

Sivuston toteuttaminen alkoi suunnitteluvaiheessa luotujen sivusuunnitelmien tarkastelulla, joiden pohjalta sivustoa alettiin rakentaa osanen kerrallaan. Jokaiselle sivulle luotiin oma komponentti Visual Studio Codessa, ja ne laitettiin src-kansion components-kansioon. Komponentteja tuli yhteensä viisi, koska alavalikko päätettiin toteuttaa erillisenä komponenttina eikä koodata sitä jokaiselle sivulle erikseen. Komponentit voi nähdä alla kuvassa 6. Kuvasta selviää, että jokaiselle sivulle tehtiin yksi komponentti, eikä komponenteilla ole alakomponentteja – rakennehierarkia on siis hyvin yksinkertainen, mutta selkeä.



Kuva 6. Komponenttihierarkia

Pienemmälle ruudulle suunniteltu hampurilaisvalikko päätettiin jättää pois, koska sivuston navigaatio oli muutenkin lyhyt ja kompakti, ja skaalautui siististi pienemmälläkin ruudulla. Valikkoon tehtiin kuitenkin värinmuutos riippuen siitä, onko kyseinen sivu aktiivinen vai ei. Muutos tehtiin koodamalla valikkoon ehtolause:

```
<NavLink to="/" style={{ isActive }} => {
  return isActive ? { color: '#8ca781' } : {};}>
  Koti
</NavLink>{" "}
```

Yllä olevasta koodinäytteestä näkee, että kun kyseinen kotisivun linkki on aktiivinen, on sen väri sivuston teemaan kuuluva vihreä (#8ca781), eikä musta niin kuin oletuksena on. Tämä lisää käytettävyyttä, kun aktiivisen sivun linkki on selkeästi erivärinen kuin muut linkit.

Kuvien näkyvyyttä rajattiin niin, että kaikista pienemmillä ruuduilla, eli lähinnä mobiililaitteilla katseltaessa, kuvat katoavat, jotta ne eivät vie turhaa tilaa tärkeämmältä sisällöltä eikä sisältö mene allekkain, joka yleensä johtaa sivun turhaan vieritykseen ylös ja alas. Keskikokoisilla ja suuremmilla näytöillä kuvat näkyvät, mutta ovat responsiivisia, eli vaihtavat kokoa riippuen näytön koosta. Kuvien skaalautuvuus toteutettiin koodin avulla:

```
<img id='kuva' src={kuva} alt='Kuva' height={'auto'} width={'100%'} />
```

Koodissa kuvan korkeus säätyy automaattisesti samalla kun sen leveys pysyy samankokoisena kuin ruudukko (grid), johon kuva oli aseteltu: kun näyttökoko pienenee, pienenee myös ruudukko, ja kuva vastaa tähän pienentymällä.

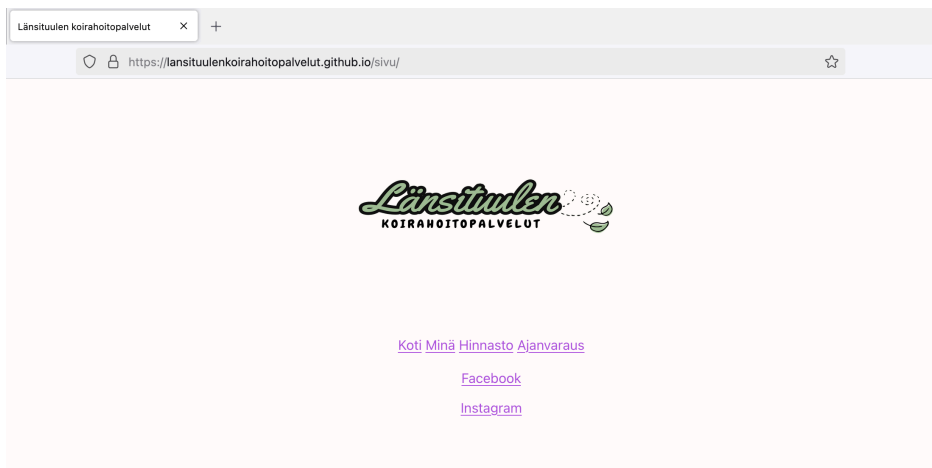
Ylipäättään verkkosivuston käyttöliittymä ei lopuksi juurikaan eroa tehdyistä sivusuunnitelmista, vaan sivusto saatiin hyvin vastaamaan tehtyjä suunnitelmia. Sivuston ensimmäisen version valmistuttua käyttöliittymään tehtiin pieniä muutoksia yrittäjän kommenttien pohjalta: tekstien paikkoja muutettiin, ja asettelua siistittiin. Esimerkiksi sivut, jotka on toteutettu ruudukkotekniikalla (minäsivu ja hinnasto), saatiin väljemmiksi muuttamalla ruudukon välistystä:

```
<Grid container spacing={3}>
```

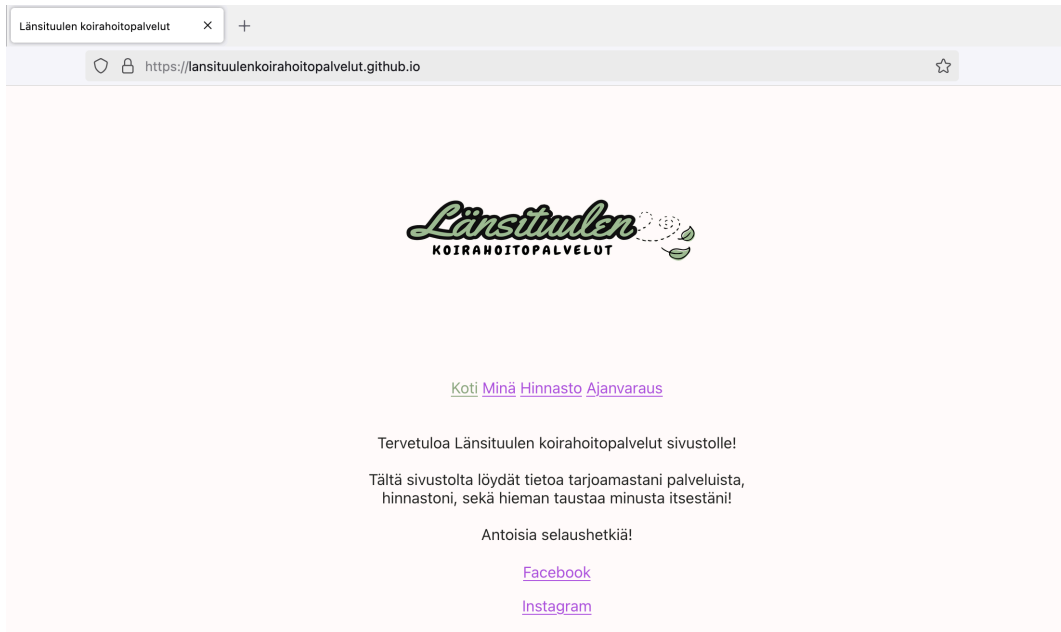
```
<Grid item xs={4} data-testid="grid-item">
```

Tässä tuli kuitenkin ongelmaksi se, ettei välistystä voinut lisätä loputtomiin, tai ruudukko olisi mennyt päävalikon päälle, tehden valikon linkeistä toimimattomia. Yllä olevalla välistyksellä saatiin kuitenkin sekä valikko toimimaan että tarpeeksi tilaa ruudukon tekstin ja kuvan välille. Sama ruudukkoväli pidettiin molemmissa sivuissa, jotka ruudukkotekniikkaa käyttävät, jotta sivut olisivat yhtenäiset ja siistit.

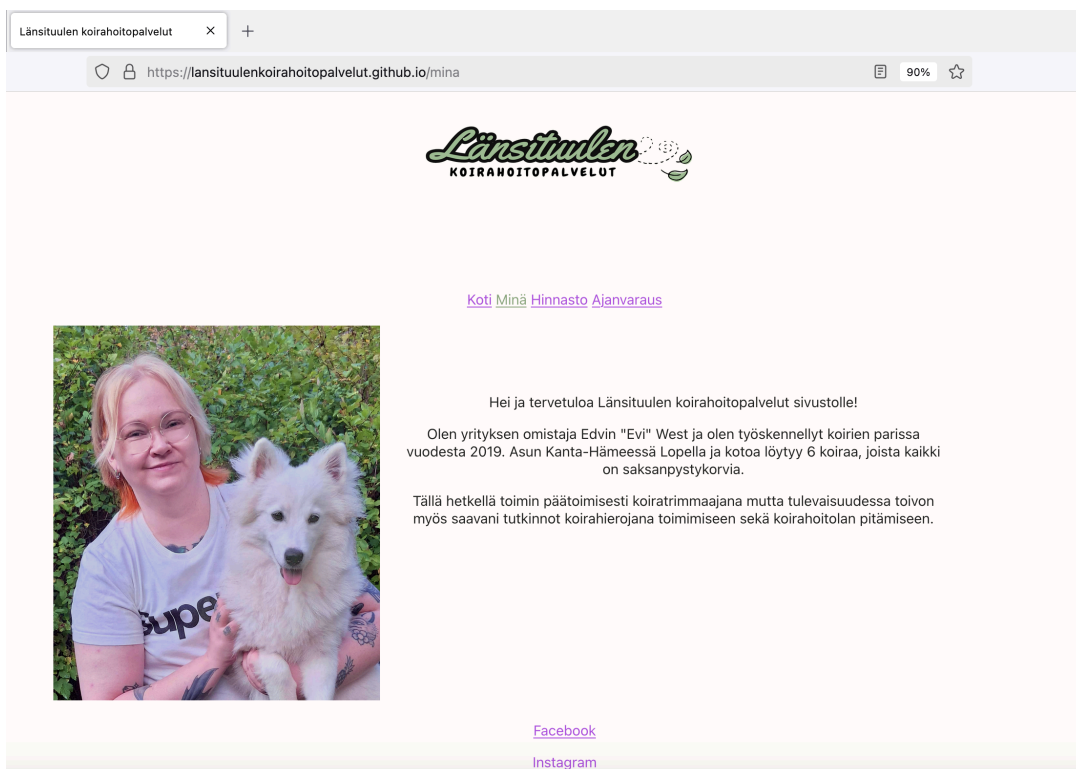
Sivujen kuva- ja tekstisisältö saatiin suoraan yrittäjältä, joten niihin ei tarvinnut ensimmäisen version valmistuttua tehdä muutoksia. Loppujen lopuksi sivuston ensimmäiseksi valmistunut versio oli jo hyvin lähellä lopullista versiota. Kyseisen lopullisen version voi nähdä kuvissa 7–10. Lopullinen tuotos esiteltiin yrittäjälle ennen testien luontia. Testien toimivuuden varmistamisen jälkeen verkkosivut saatiin virallisesti valmiiksi, ja ne julkaistiin palvelimelle kaikkien saataville.



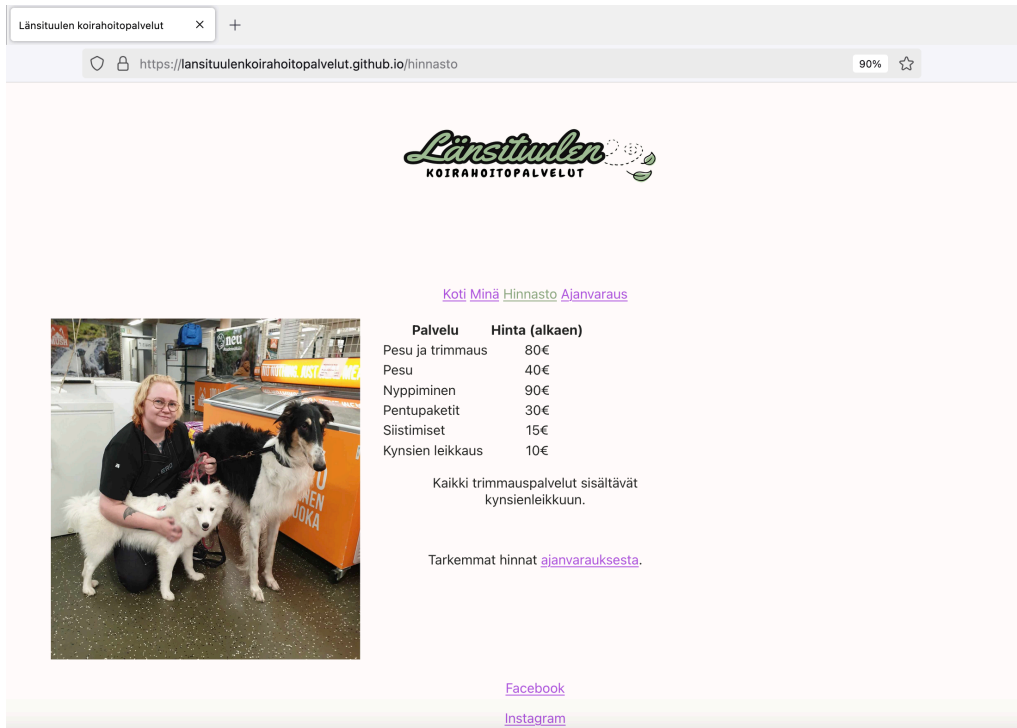
Kuva 7. Lopullinen pääsivu



Kuva 8. Lopullinen kotisivu



Kuva 9. Lopullinen minäsivu



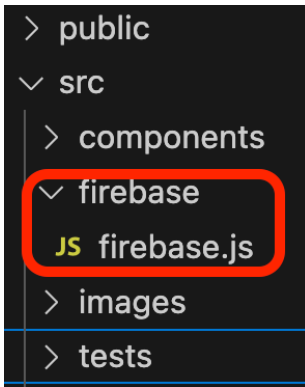
Kuva 10. Lopullinen hinnastosivu

Verkkosivustolle toteutettiin Firebase Analyticsin avulla seuranta sivuston kävijämäärästä. Tämä tehtiin lisäämällä Firebasen automaattisesti luoma koodinpätkä osaksi projektia, ja alustamalla Firebasen käyttö kahdella rivillä koodia:

```
const app = initializeApp(firebaseConfig);
```

```
const analytics = getAnalytics(app);
```

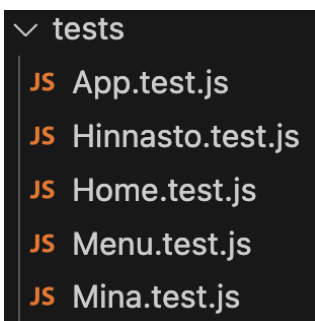
Koodissa ensin luodaan sovellus ja konfiguroidaan Firebase, jonka jälkeen sovelluksen avulla haetaan analytiikkadataa. Firebase-tiedostolle tehtiin oma kansionsa, jotta se ei sotkeutuisi komponenttien kanssa, ja olisi helpompi löytää. Firebase-tiedoston sijainnin React-projektin rakenteessa voi nähdä alla kuvassa 11.



Kuva 11. Firebase-kansion ja Firebase.js-tiedoston sijainti React-projektissa

Analytiikan toimintaan saaminen vei jonkin verran aikaa, sillä sitä ei saatu toimimaan kehitysvaiheessa monesta yrityksestä huolimatta. Analytiikka alkoi kuitenkin toimimaan, kun sivusto julkaistiin, joten vaikka kehitysvaiheen analytiikkaan ei päästyäkään käsiksi, oli tärkeintä, että analytiikan keräys saatiin toimimaan lopullisessa, julkaistussa versiossa.

Sivuston jokaiselle sivulle laadittiin niiden toimintaa testaavia testejä. Testit toteutettiin Jestillä, joka on nimenomaan JavaScript-testaukseen kehitetty viitekehys. Sen tarkoituksena on olla yksinkertainen, nopea ja turvallinen, ja sitä voi Reactin lisäksi käyttää muun muassa Noden, Angularin ja Vuen sekä monen muun eri JavaScript-projektin kanssa. Jestillä on miljoonia käyttäjiä, joista osa on kaikkien tuntemia suuryrityksiä kuten Spotify ja Instagram. (Jest s.a.) Jest valikoitu testikirjastoksi, koska se tulee Create React App:n mukana, jonka vuoksi sen käyttö todettiin kaikista yksinkertaisemmaksi. Jestin avulla luotiin testejä, jotka muun muassa testaavat, renderöityykö sivut ylipäätään, renderöityykö niiden sisällöt oikein, ja johtavatko sivuilla olevat linkit oikeisiin paikkoihin. Myös esimerkiksi taulukon rakennetta ja kuvien latautumista varten tehtiin testit, jotta voitiin varmistaa, että kaikki sisältö latautuu oikein, kun sivut renderöityvät. Lopulta testejä tehtiin yhteensä 11 kappaletta, ja ne testaavat laajasti verkkosivuston toimintaa eri näkökulmista. Erilaiset testitiedostot voi nähdä kuvassa 12.



Kuva 12. Jestillä toteutetut testitiedostot

Kuvasta 12 näkee, että jokaiselle sivukomponentille on tehty oma testitiedostonsa, sekä lisäksi App.js-tiedostolle, jolle tosin luotiin ainoastaan yksi testi: logon renderöityminen oikein.

```
test('renders the logo', () => {render(<App />);  
  
  const image = screen.getByAltText(/Logo/i);  
  
  expect(image).toBeInTheDocument();  
  
  expect(image).toHaveAttribute('src', logo);  
  
  expect(image).toHaveAttribute('alt', 'Logo');});
```

Testikoodissa renderöidään ensiksi App.js-tiedoston App-funktio, etsitään funktiossa oleva kuva, tässä tapauksessa logo, ja testataan, löytyykö logo ja sille annetut attribuutit. Muidenkin sivujen kuvien renderöityminen ja löytyminen sivulta testattiin samalla tavoin, ja testit luotiin samankaltaista ideaa noudattaen.

Ennen sivuston julkaisua toteutettiin kohdeyleisötestausta hyvin pienessä mittakaavassa, lähinnä yrittäjän ja sivuston tekijän läheisten toimesta, mutta kuitenkin niin, että testaajat olivat osa aiemmin rajattua kohdeyleisöä. Tällä testauksessa pystyttiin kuitenkin hiomaan sivuston ulkonäköä selkeämmäksi ja todentamaan helppokäyttöisyyttä, kun nähtiin, kuinka eri-ikäiset käyttäjät navigoivat sivulla luontevasti. Kohdeyleisötestauksen jälkeen tehtiin pieniä muutoksia sivuston ulkonäköön, muttei juurikaan navigointiin tai linkkien paikkoihin.

#### 4.4 Projektin käyttöönotto

Verkkosivujen valmistuttua ja yrittäjän ne hyväksytyä sivusto koottiin Create React App:n deploy-komennolla. Komento kokoaa React-projektin varmasti oikein ja optimoi sen, jotta sivusto julkaisun jälkeen toimii mahdollisimman hyvin. Komento pitää sisällään Create React App:n build-komennon, kunhan se on lisätty osaksi package.json-tiedoston scripts-otsikkoa, joten sitä ei tarvinnut erikseen ajaa. (Create React App 12.1.2022.) Alla on ote package.json-tiedostosta, josta näkee, miten build-komento on osa predeploy-komentoa, joka ajetaan aina ennen deploy-komentoa:

```
"scripts": {  
  
  "predeploy": "npm run build",  
  
  "deploy": "gh-pages -d build",
```

Kokoamisen jälkeen valmiit sivut siirrettiin GitHubin kautta GitHub Pageksi. GitHub Pages on GitHubin ylläpitämä palvelu, jonka avulla GitHubin käyttäjä voi julkaista staattisen verkkosivun

suoraan omasta koodisäilöstään (repository). Se on optimoitu vastaanottamaan HTML:ää, CSS:ää ja Javascriptia, joten se soveltuu loistavasti React-sovellusten julkaisuun. (GitHub Docs s.a.)

Sivustoa varten luotiin oma GitHub-käyttäjätunnus, johon jaettiin oikeudet sekä yrittäjälle että sivuston tekijälle. Verkkosivun koodia voi muokata suoraan GitHubissa, joka mahdollistaa sen, ettei yrittäjä sivuston päivittämiseksi tarvitse erillistä koodieditoria tai osaamista koodin siirtämisestä editorista GitHubiin. Koska sivusto on tehty GitHub Pagesina, koodin muuttaminen GitHubin editorissa ja sen lähettäminen eteenpäin (commit) tekee muokkaukset suoraan sivustolle ainoastaan pienellä viiveellä. Yrittäjää ohjeistettiin koodin muuttamiseen suoraan GitHubissa, ja lisäksi hänelle tehtiin kuvalliset ohjeet, jotka ovat tämän opinnäytetyön liitteenä (liite 1). Ohjeet ja päivittämisen osaaminen mahdollistavat sivuston ylläpidon myös yrittäjän toimesta itsenäisesti, jolloin sivusto pysyy aina ajantasaisena.

Myös Firebaseen tehtiin sivustoa varten erillinen käyttäjätunnus, joka jaettiin yrittäjälle. Firebase Analyticsin toimintaa käytiin yhdessä yrittäjän kanssa läpi, ja häntä ohjeistettiin, miten hän löytää analytiikkadatan ja kuinka sitä luetaan. Analytiikkadatasta yrittäjä näkee, kuinka paljon sivustolla vierailaan ja kuinka vierailijat käyttäytyvät sivustolla.

## 5 Yhteenveto

Tämän opinnäytetyön lopputuloksena syntyi yksityisyrittäjälle funktionaaliset verkkosivut, jossa esitellään yrityksen toimintaa, sen tarjoamia palveluja ja ohjataan asiakas yrittäjän ajanvaraukseen. Verkkosivut ovat responsiiviset, eli ne skaalautuvat tietokoneen, tabletin ja älypuhelimien näyttökoon mukaan riippuen siitä, millä laitteella sivuja tarkastelee.

Verkkosivut toteutettiin lähes alkuperäisten suunnitelmien mukaisesti, ja yrittäjän hyväksymiin suunnitelmiin tehtiin vain pieniä muutoksia sivujen toteutuksen aikana. Näillä muutoksilla saatiin aikaan siistimpää ulkoasua ja parempia toiminnallisuuksia, jotta sivusto olisi mahdollisimman selkeästi ja asiakasystävällisesti toteutettu. Tämän opinnäytetyön lopputulos eli Länsituulen koirahoitopalveluiden verkkosivut ovat nähtävissä osoitteessa <https://lansituulenkoirahoitopalvelut.github.io/sivu/>.

Yrittäjä oli aktiivisesti mukana opinnäytetyöprosessin aikana tarjoten mielipiteitä ja korjausehdotuksia varsinkin sivuston toteuttamisen ja viimeistelyn aikana. Yrittäjällä ei kuitenkaan ollut varsinaista aikataulutoivetta sivuston valmistumiselle, mikä mahdollisti opinnäytetyön työstämisen usean kuukauden ajan tekijän oman aikataulun puitteissa. Opinnäytetyön tekijä ja yrittäjä tunsivat toisensa entuudestaan, mikä mahdollisti tiiviin yhteydenpidon ja rennon työskentelyilmapiirin, sekä hyvin vapaan aikataulun työn valmistumiselle.

Opinnäytetyön tekoon käytettiin jonkin verran tekoälyä, mutta ainoastaan koodaamisen apuna sivuston koodin tarkistamisessa. Sivuston tekemisen ja kirjallisen raportin kirjoittamisen aikana ei kohdattu muita ongelmia tai työvaiheita, jotka vaikeuttaisivat työn eettisyyden arviointia, vaan työ on koottu hyviä eettisiä periaatteita noudattaen.

Työn tekijällä oli ennen kotisivuprojektin aloittamista perustiedot Reactin kanssa työskentelemisestä, ja takanaan ainoastaan yksi hyvin yksinkertainen React-sivusto sekä osaamista lähinnä HTML:stä ja JavaScriptistä. Opinnäytetyön teko syvensi paitsi React-osaamista, myös osaamista Material UI:n ja Firebasen käytön tiimoilta, sillä vaikka työkalut olivat pintapuolisesti tuttuja, ei niiden käyttö syvällisemmin ja isommassa projektissa ollut; voi hyvin todeta, että tämän opinnäytetyöprojektin aikana osaaminen sekä Reactista, Material UI:sta että Firebasesta syveni paitsi niiden käytön oppimisen myös niiden syvemmän toiminnan ymmärtämisen tiimoilta. Koska työkalut olivat vain pintapuolisesti tuttuja ennen projektin aloittamista, kohdattiin työn aikana useita ongelmia ja virheitä. Näitä virheitä selviteltiin dokumentaatioiden, StackOverflow-foorumien sekä tekoälyn avulla, ja ne olivat oppimisen ja toimintojen ymmärtämisen kannalta kaikista opettavaisimpia. Virheiden ja ongelmien selvittely vaati pitkäjänteisyyttä ja sinnikkyyttä, varsinkin vaikeimpien ongelmien edessä: oli koodin toimimattomuuksia ja Firebasen käyttöön oton hankaluuksia, joiden selvittely vei useita

tunteja työprosessista. Nämä tilanteet olivat kuitenkin oppimisen näkökulmasta hedelmällisiä, ja ne koettiin hyödyllisiksi.

Opinnäytetyöprosessi opetti myös oman työskentelyn aikataulutusta mittakaavassa, joka oli työn tekijälle uusi. Ulkoisten asioiden, kuten täysipäiväisen työnteon, vuoksi opinnäytetyön valmistuminen venyi usealla kuukaudella, ja tämä aiheutti painetta saada työ valmiiksi. Opinnäytetyön kirjallisen osuuden kirjoittaminen aiheutti myös ajoittain päänvaivaa, sillä varsinkin työn alussa ei ollut selvää, kuinka laajoja esimerkiksi teoreettisen osan luvut tulisivat olla. Nykyiseen versioon päädyttiinkin usean yrityksen ja palautteen saamisen kautta, ja lopputulos on selkeä kokonaisuus, jossa on esitelty monipuolisesti paitsi erilaisia työvälineitä verkkosivuston suunnitteluun ja toteutukseen, myös syvällisemmin React-koodausprojektia.

Tämän opinnäytetyön tuloksena syntynyt verkkosivusto on funktionaalinen ja vastaa tarpeeseen, joka prosessin alussa oli. Monia asioita voisi kuitenkin tehdä paremmin, sillä teknisesti sivusto on yksinkertainen, ja sivusto voisi sisältää monia erilaisia toimintoja, esimerkiksi käyttäjälaskuria ja videoita, jotka jäivät nyt puuttumaan. Nämä asiat eivät kuitenkaan olleet osana yrittäjän toiveita sivustoa kohtaan, joten niiden tarpeellisuutta on hyvä pohtia ennen kuin niitä lähtisi toteuttamaan. On kuitenkin mahdollista, että sivuston valmistuttua yrittäjälle tulee ideoita ja ajatuksia sivuston jatkokehittämistä kohtaan, jolloin sivustoon voisi tehdä muutoksia. Pieniä muutoksia, kuten hinnaston päivitystä, jotka eivät vaadi koodausosaamista, yrittäjä voi tehdä jatkossa itsekin, joka on sivuston jatkuvuuden ja ajantasaisuuden kannalta olennaista.

## Lähteet

Alma Media s.a. Digimarkkinointi. Luettavissa: <https://www.almamedia.fi/yrityksille/markkinoijan-tietopankki/digimarkkinointi/>. Luettu 8.9.2024.

Ben Rebah, H., Boukthir, H, Chédebois, A. 2022. Website Design and Development with HTML5 and CSS3. John Wiley & Sons, Incorporated. New Jersey. E-kirja. Luettu 4.4.2024.

Coursera 3.4.2024. What Does a Back-End Developer Do? Luettavissa: <https://www.coursera.org/articles/back-end-developer>. Luettu 8.12.2024.

Create React App 12.1.2022. Deployment. Luettavissa: <https://create-react-app.dev/docs/deployment/>. Luettu 27.12.2024.

Diggity Solutions Pvt. Ltd. 8.2.2023. 5 Stages of Web Development. LinkedIn-päivitys. Luettavissa: <https://www.linkedin.com/pulse/5-stages-web-development-diggityus/>. Luettu: 3.4.2024.

Dotenvx s.a. .env. Luettavissa: <https://dotenvx.com/docs/env-file>. Luettu 27.12.2024.

Firebase s.a.a. Firebase. Luettavissa: <https://firebase.google.com/>. Luettu 8.4.2024.

Firebase s.a.b. Google Analytics for Firebase. Luettavissa: <https://firebase.google.com/products/analytics>. Luettu 8.4.2024.

Forbes Agency Council 21.4.2017. 10 Ways To Make Your Website Stand Out From Your Competitors'. Luettavissa: <https://www.forbes.com/sites/forbesagencycouncil/2017/04/21/10-ways-to-make-your-website-stand-out-from-your-competitors/>. Luettu 13.10.2024.

GeeksForGeeks 5.12.2023. Scrum (software development). Luettavissa: <https://www.geeksforgeeks.org/scrum-software-development/>. Luettu 19.4.2024.

GitHub Docs s.a. About GitHub Pages. Luettavissa: <https://docs.github.com/en/pages/getting-started-with-github-pages/about-github-pages>. Luettu 22.12.2024.

Haltu 31.3.2023. Verkkopalvelun saavutettavuus. Haltun blogi. Luettavissa: <https://www.haltu.fi/blogi/verkkopalvelun-saavutettavuus>. Luettu 8.9.2024.

Haltu 18.10.2023. Käytettävyys ohjelmistokehityksessä: Miksi ja miten sitä tutkitaan? Haltun blogi. Luettavissa: <https://www.haltu.fi/blogi/kaytettavyys>. Luettu 8.9.2024.

Humble, J. s.a. What is the Double Diamond Design Process? The Fountain Institute's blog. Readable: <https://www.thefountaininstitute.com/blog/what-is-the-double-diamond-design-process>. Read 8.12.2024.

Hutsulyak, O. 1.8.2024. 10 Key Reasons Why You Should Use React for Web Development. Tech-Magic's blog. Readable: <https://www.techmagic.co/blog/why-we-use-react-js-in-the-development>. Read 27.10.2024.

Indeed 27.6.2024. What Are Back-End Services, and What Do They Include? Readable: <https://ca.indeed.com/career-advice/career-development/back-end-services>. Read 8.12.2024.

JAMK 27.5.2022. Tutkimuksellinen kehittämishanke opinnäytetyönä vs projektityö. Readable: <https://help.jamk.fi/opinnaytetyon-ohjaus/fi/tyoelaman-tutkiva-kehittamistoiminta/projektityo-vs-ns-toiminnallinen-tutkimuksellinen-kehittamishanke-opinnaytetyo/>. Read 19.10.2024.

Jest s.a. Jest. Readable: <https://jestjs.io/>. Read 17.11.2024.

Kundariya, H. 17.12.2021. 10 Reasons Why You Should Use React for Building Interactive User Interfaces. Shahed Nasser's blog. Readable: <https://blog.shahednasser.com/10-reasons-why-you-should-use-react-for-building-interactive-user-interfaces/>. Read: 26.3.2024.

Lakihelppi s.a. Sopimusyrittäjäsopimus. Readable: <https://lakihelppi.com/sopimusyrittajasopimus/>. Read 10.4.2024.

Material UI 2024. Material UI. Readable: <https://mui.com/material-ui/>. Read 5.4.2024.

Material UI s.a. Getting Started. Readable: <https://mui.com/material-ui/getting-started/>. Read 19.4.2024.

MDN Web Docs s.a. SPA (Single-page application). Readable: <https://developer.mozilla.org/en-US/docs/Glossary/SPA>. Read 16.7.2024.

Meio s.a. Digimarkkinointi — Mitä se on ja miksi se on tärkeää? Readable: <https://meio.fi/digimarkkinointi/>. Read 8.9.2024.

Next.js s.a. About React and Next.js. Readable: <https://nextjs.org/learn/react-foundations/what-is-react-and-nextjs>. Read 16.7.2024.

One.com s.a. Mikä on sivukartta? Readable: <https://www.one.com/fi/verkkomarkkinointi/mika-on-sivukartta>. Read 27.10.2024.

- Palvelumuotoilu Palo 8.12.2018. Palvelumuotoiluprosessin vaiheet. Palvelumuotoilu Palon blogi. Luettavissa: <https://www.palvelumuotoilupalo.fi/blogi/palvelumuotoilun-prosessin-vaiheet/>. Luettu 19.10.2024.
- Patterns s.a. Overview of React.js. Luettavissa: <https://www.patterns.dev/react/>. Luettu 3.4.2024.
- Patadiya, J. 1.4.2024. Next JS vs React : Which Framework to choose for Front end in 2024?. Radixin blogi. Luettavissa: <https://radixweb.com/blog/nextjs-vs-react>. Luettu 16.7.2024.
- Powell, Z. 29.12.2023. Angular vs React: A Detailed Side-by-Side Comparison. Kinstan blogi. Luettavissa: <https://kinsta.com/blog/angular-vs-react/>. Luettu 18.4.2024.
- Raval, N. 1.4.2024. React vs Angular: Which JS Framework to choose for Front-end Development? RadixWebin blogi. Luettavissa: <https://radixweb.com/blog/react-vs-angular>. Luettu 18.4.2024.
- React 2024a. Render and Commit. Luettavissa: <https://react.dev/learn/render-and-commit>. Luettu 27.3.2024.
- React 2024b. React. Luettavissa: <https://react.dev/>. Luettu: 26.3.2024.
- React s.a. Your First Component. Luettavissa: <https://react.dev/learn/your-first-component>. Luettu 8.12.2024.
- Romano, J. s.a. 10 best website layout ideas to get you started. Wixin blogi. Luettavissa: <https://www.wix.com/blog/website-layouts>. Luettu 27.10.2024.
- Sanoma 16.8.2022. Digitaalinen markkinointi: mitä se on ja mihin se sopii? Luettavissa: <https://media.sanoma.fi/ajankohtaista/nakemyksia-markkinoinnista/digitaalinen-markkinointi-mita-se-ja-mihin-se-sopii>. Luettu 8.9.2024.
- Sanoma s.a. Kotisivut yritykselle. Luettavissa: <https://media.sanoma.fi/mediat-palvelut/markkinoinnin-palvelut/kotisivut-yritykselle>. Luettu 8.9.2024.
- Santana Roldán, C. 2023. React 18 Design Patterns and Best Practices. Packt Publishing. Birmingham. E-kirja. Luettu: 4.4.2024.
- Scrum.org s.a. What is Scrum? Luettavissa: <https://www.scrum.org/resources/what-scrum-module>. Luettu 19.4.2024.
- Scrum Guides 2020. The 2020 Scrum Guide™. Luettavissa: <https://scrumguides.org/scrum-guide.html>. Luettu 19.4.2024.

Suomen digimarkkinointi s.a.a. 5 syytä, miksi valita responsiiviset verkkosivut. Suomen digimarkkinoinnin blogi. Luettavissa: <https://www.digimarkkinointi.fi/blogi/5-syyta-miksi-valita-responsiiviset-verkkosivut/>. Luettu 13.10.2024.

Suomen digimarkkinointi s.a.b. Verkkosivut. Luettavissa: <https://www.digimarkkinointi.fi/verkkosivut/>. Luettu 22.4.2024.

Tambe, N. 20.2.2024. Domain Vs. Hosting: Definition And Difference. Forbes Advisorin blogi. Luettavissa: <https://www.forbes.com/advisor/in/business/software/domain-vs-hosting/>. Luettu: 27.3.2024.

UXPin 5.8.2024. Double Diamond Design Process – The Best Framework for a Successful Product Design. UXPinin blogi. Luettavissa: <https://www.uxpin.com/studio/blog/double-diamond-design-process/>. Luettu 8.12.2024.

W3Schools s.a.a. JavaScript HTML DOM. Luettavissa: [https://www.w3schools.com/js/js\\_html-dom.asp](https://www.w3schools.com/js/js_html-dom.asp). Luettu: 26.3.2024.

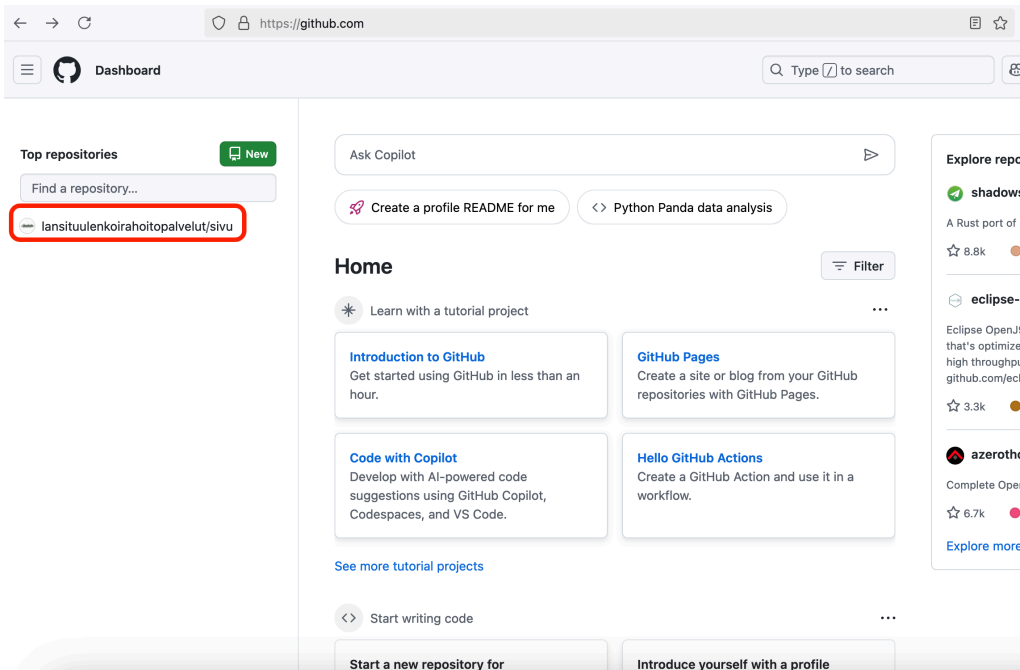
W3Schools s.a.b. What is a Front-End Developer? Luettavissa: [https://www.w3schools.com/whatis/whatis\\_frontenddev.asp](https://www.w3schools.com/whatis/whatis_frontenddev.asp). Luettu 8.12.2024.

W3Schools s.a.c. React Components. Luettavissa: [https://www.w3schools.com/react/react\\_components.asp](https://www.w3schools.com/react/react_components.asp). Luettu: 27.3.2024.

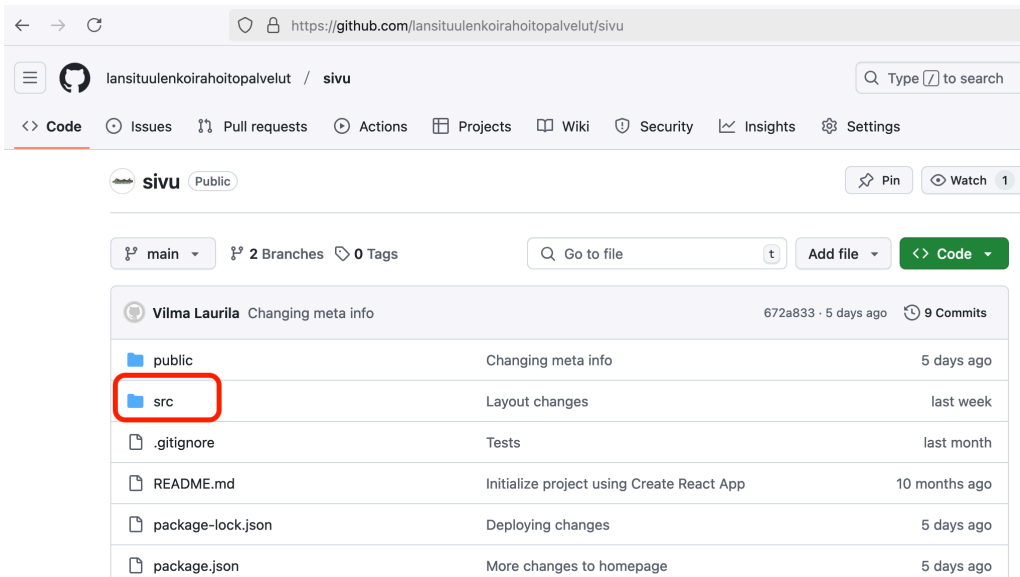
## Liitteet

### Liite 1. Ohje sivuston päivittämiseen GitHubin kautta

Kirjaudu GitHubiin osoitteessa <https://github.com/>.



Paina punaisella korostettua linkkiä, niin pääset koodisäilöön.



Klikkaa auki **src-kansio**.

The screenshot shows the GitHub interface for the repository `lansituulenkoirahoitopalvelut/sivu/tree/main/src`. The left sidebar shows the file tree with the `src` folder expanded, and the `components` subfolder highlighted. The main content area shows a commit by Vilma Laurila titled "Layout changes". Below the commit information is a table listing files and their last commit messages.

Name	Last commit message
..	
components	Layout changes
firebase	Tests
images	Tests
tests	More tests
App.css	Layout changes
App.js	Layout changes
index.css	Initialize project using Create React App
index.js	Initialize project using Create React App

Valitse valikosta **components-kansio**.

The screenshot shows the GitHub interface for the repository `lansituulenkoirahoitopalvelut/sivu/tree/main/src/components`. The left sidebar shows the file tree with the `components` folder expanded, listing files like `Footer.js`, `Hinnasto.js`, `Home.js`, `Menu.js`, and `Mina.js`. The main content area shows a commit by Vilma Laurila titled "Layout changes". Below the commit information is a table listing files and their last commit messages.

Name	Last commit message
..	
Footer.js	Tests
Hinnasto.js	Layout changes
Home.js	Tests
Menu.js	Layout changes
Mina.js	Layout changes

Avautuvasta listasta valitse sivu, mitä haluat päivittää. Tässä esimerkissä päivitetään Hinnastosi-  
vua.

The screenshot shows the GitHub web interface for the file `sivu/src/components/Hinnasto.js`. The file content is as follows:

```

1  import React from 'react';
2  import Grid from '@mui/material/Grid';
3  import Container from '@mui/material/Container';
4  import kuva from '../images/tokakuva.jpg'
5
6  function Hinnasto() {
7
8      return(
9          <div>
10             <br></br>
11             <Container>
12                 <Grid container spacing=3>
13                     <Grid item xs=4 data-testid="grid-item">
14                         <div><img id="kuva" src={kuva} alt="Kuva" height={'auto'} width={'100%'} /></div>
15                     </Grid>
16                     <Grid item xs=4 data-testid="grid-item">
17                         <div>
18                             <table>
19                                 <thead>
20                                     <tr>
21                                         <th>Palvelu</th>
22                                         <th>Hinta (alkaen)</th>
23                                     </tr>
24                                 </thead>
25                                 <tbody>
26                                     <tr>

```

The 'Edit' button in the top right corner of the file view is highlighted with a red box.

Paina oikeasta yläkulmasta Edit-nappulaa.

The screenshot shows the GitHub web interface for the file `sivu/src/components/Hinnasto.js` in the edit view. The file content is as follows:

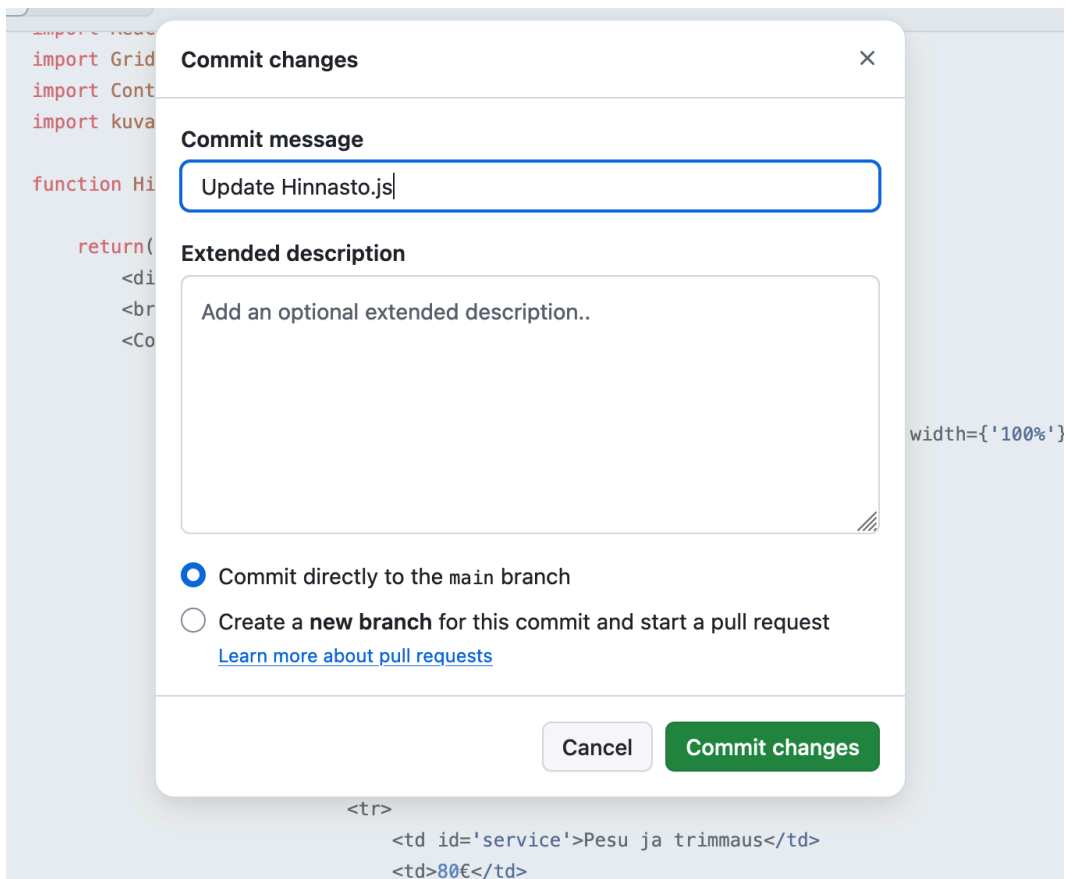
```

2  import Grid from '@mui/material/Grid';
3  import Container from '@mui/material/Container';
4  import kuva from '../images/tokakuva.jpg'
5
6  function Hinnasto() {
7
8      return(
9          <div>
10             <br></br>
11             <Container>
12                 <Grid container spacing=3>
13                     <Grid item xs=4 data-testid="grid-item">
14                         <div><img id="kuva" src={kuva} alt="Kuva" height={'auto'} width={'100%'} /></div>
15                     </Grid>
16                     <Grid item xs=4 data-testid="grid-item">
17                         <div>
18                             <table>
19                                 <thead>
20                                     <tr>
21                                         <th>Palvelu</th>
22                                         <th>Hinta (alkaen)</th>
23                                     </tr>
24                                 </thead>
25                                 <tbody>
26                                     <tr>
27                                         <td id="service">Pesu ja trimmaus</td>
28                                         <td>80</td>

```

The 'Commit changes...' button in the top right corner of the edit view is highlighted with a green box.

Tee muutos koodiin, ja paina sen jälkeen vihreää Commit changes-nappulaa oikeasta yläkulmasta.



Avautuvasta ikkunasta paina uudestaan vihreää Commit changes-nappulaa. Muita asetuksia ei tarvitse muuttaa.

Nyt koodi ja sivu on päivitetty, ja muutokset ilmestyvät sivustolle pienellä viiveellä!