



Karelia-ammattikorkeakoulu
Tradenomi, tietojenkäsittely (AMK)

Puolueen verkkosivu-uudistuksen tekninen toteutus

Pietari Virtanen

Opinnäytetyö, toukokuu 2024

www.karelia.fi



OPINNÄYTETYÖ
Toukokuu 2024
Tietojenkäsittelyn koulutus

Tikkarinne 9
80200 JOENSUU
+358 13 260 600

Tekijä(t)
Pietari Virtanen

Nimeke
Puolueen verkkosivu-uudistuksen tekninen toteutus

Toimeksiantaja
Piraattipuolue

Tiivistelmä

Opinnäytetyön tavoitteena oli toteuttaa Suomen piraattipuolueelle uudistetut verkkosivut. Keskeisinä vaatimuksina oli luopua aiemmasta WordPress-toteutuksesta, mahdollistaa sisältöjen päivitys ilman erityistä tietoteknistä osaamista ja toteuttaa puolueen avainhenkilöille profiilit helpottamaan heidän tietojensa pitämistä ajantasaisina sivuston eri sivuilla. Tavoitteena oli lisäksi luoda avainhenkilöille käyttäjätunnukset, joiden avulla he voivat itse muokata omia tietojaan. Teknologioiden valinnassa tavoitteena oli käyttää kehityksessä ensisijaisesti vapaita avoimen lähdekoodin ratkaisuja.

Verkkosivun käyttöliittymä toteutettiin Reactilla ja sisällönhallintajärjestelmä Strapilla. Näiden väliseen kommunikaatioon käytettiin GraphQL:ää. Verkkosivun hostaus toteutettiin nginxillä Ubuntu Server -pilvipalvelimelle ja SSL-sertifikoitiin käytettiin Electronic Frontier Foundationin certbotia.

Tuloksena saatiin toimeksiantajan vaatimukset ja tavoitteet täyttävä tekninen toteutus. Muilta osin kehitysprojekti on edelleen käynnissä ja tästä saattaa myöhemmin aiheutua muutoksia tai lisäyksiä myös tekniseen toteutukseen.

Kieli
suomi

Sivuja 41
Liitteet 0
Liitesivumäärä 0

Asiasanat
verkkosivustot, verkko-ohjelmointi, verkkopalvelut, sisällönhallinta



THESIS
May 2024
Degree Programme in Business Information Technology

Tikkarinne 9
80200 JOENSUU
FINLAND
+ 358 13 260 600

Author (s)
Pietari Virtanen

Title
Technical Reimplementation of a Political Party Website

Commissioned by
Pirate Party of Finland

Abstract

The goal of this thesis was to reimplement the website for the Pirate Party of Finland. Central requirements were to abandon the previous WordPress implementation, to allow the contents to be updated without any special technical skills, and to implement profiles for the key persons in the party to make it easier to keep their information up to date across the various pages on the website. An additional goal was to create user accounts for the key persons to allow them to edit their own information. While choosing technologies for the project, a goal was to prefer free libre open-source software.

The user interface for the website was implemented using React while Strapi was used for the content management system. GraphQL was used for the communication between these. The website was hosted on a Ubuntu Server cloud server using nginx and the SSL certificates were created with Electronic Frontier Foundation's certbot.

As a result, a technical implementation meeting the commissioner's requirements and goals was produced. Non-technical aspects of the development project are still ongoing. In the future, this may cause changes or additions to the technical implementation as well.

Language
Finnish

Pages 41
Appendices 0
Pages of Appendices 0

Keywords
websites, web programming, web services, content management

Sisältö

1	Johdanto	6
1.1	Verkkosivu-uudistus poliittiselle puolueelle	6
1.2	Tarve verkkosivu-uudistukselle piraattipuolueessa	6
1.3	Parityön eriyttäminen, aiheen rajaus ja projektin työnjako	7
1.4	Projektin tavoitteet	8
1.5	Eettisyys ja turvallisuus.....	9
1.6	Tekoälyn käyttö opinnäytetyössä.....	9
2	Kehityksessä käytetyt teknologiat	10
2.1	Teknologioiden valintaperusteet	10
2.2	Strapi	11
2.3	React	12
2.4	Nginx	13
2.5	Ubuntu Server.....	14
2.6	GraphQL.....	14
2.7	Visual Studio Code	15
3	Toteutus.....	15
3.1	Viestintä ja versionhallinta	15
3.2	Arkkitehtuuri.....	16
3.2.1	Arkkitehtuurin yleiskuva	16
3.2.2	Tietokanta	18
3.3	Reititys.....	21
3.4	Header	22
3.5	Sivut.....	23
3.6	Komponentit.....	23
3.6.1	Muotoiltu teksti.....	23
3.6.2	Klikattavat painikkeet	24
3.6.3	Kalenteri	25
3.6.4	Piiriyhdistysten kartta	25
3.6.5	Henkilögalleria	27
3.6.6	Uutiset	28
3.7	Profiili.....	29
3.7.1	Julkinen profiilisivu.....	29
3.7.2	Kirjautuminen.....	31
3.7.3	Profiilin muokkaus.....	31
3.8	Sovelluspalvelimen asennus ja konfigurointi.....	36
3.8.1	Pilvipalvelin	36
3.8.2	Nginxin konfigurointi ja SSL-sertifikaattien luonti	36
3.8.3	Sovellusten asennus.....	37
4	Tulokset ja pohdinta.....	37
	Lähteet.....	39

Lyhenteet ja erityissanasto

autentikointi	Mikä tahansa menetelmä, jolla varmistetaan esimerkiksi käyttäjän henkilöllisyydestä.
backend	Verkkosivun teknisen toteutuksen osa, joka sijaitsee palvelimella ja käsittelee tietokantoja sekä palvelimen puolella tapahtuvaa toiminnallisuutta (Airfocus 2024).
frontend	Käyttäjälle näkyvä osa verkkosivun toteutuksesta (Airfocus 2024).
hostaus	Ohjelmien tai sisällön tuominen käyttäjien saataville verkkoon asentamalla ne palvelimelle.
HTML	HyperText Markup Language, merkintäkieli verkkosivun sisällön ja rakenteen esitykseen.
JWT	JSON Web Token.
komponentti	Strapin ominaisuus, jolla voidaan luoda uudelleenkäytettäviä yhdistelmiä kentistä (Strapi 2024a). Komponenteilla tarkoitetaan tässä dokumentissa aina Strapi-komponentteja; React-komponenteista käytetään termiä React-komponentti.
käänteinen proxy	Palvelin, joka toimii liikenteen välikätenä ennen lopullista palvelinta, jotta asiakasohjelmat eivät ole suoraan yhteydessä lopulliseen palvelimeen (Cloudflare 2024).
Markdown	Eräs helppolukuisesti tarkoitettu merkintäkieli, joka on muunnettavissa HTML-koodiksi (Daring Fireball 2024).
React-komponentti	Eräs Reactin ominaisuus. Yleensä HTML-näkymä tai sen osa, jota voidaan käyttää useita kertoja eri yhteyksissä (Japikse, Grossnicklaus & Dewey 2020, 542).
renderöinti	Verkkosivun koodin muuntaminen käyttäjälle näkyväksi interaktiiviseksi sisällöksi (Seobility 2024).
rich text	Muotoiltu teksti.
slug	URL-osoitteen polkuosa.
XSS	Cross-Site Scripting, hyökkäys, jossa luotetulle verkkosivulle lisätään ylimääräistä koodia (OWASP 2024).

1 Johdanto

1.1 Verkkosivu-uudistus poliittiselle puolueelle

Tämän toiminnallisen opinnäytetyön aihe on verkkosivu-uudistuksen teknisen toteutuksen suunnittelu ja toteutus Suomen piraattipuolueelle. Alun perin opinnäytetyö oli parityö toisen opiskelijan kanssa ja aiheajaukseen sisältyivät myös projektinhallinta ja uudistettujen verkkosivujen markkinointi, jotka olivat parini vastuualuetta. Tämä oli meille luonteva opinnäytetyön aihe, koska olimme molemmat puolueen toiminnassa mukana ja puolueella oli tarve verkkosivu-uudistukselle. Myöhemmin eriytimme opinnäytetyöt vastuualueittain rajatuiksi yksilötoiksi.

Verkkosivu-uudistusprojekti on joiltakin osin edelleen käynnissä tätä kirjoittaessa. Esimerkiksi osa vanhojen sivujen tekstisisällöistä on kopioimatta uusille sivuille ja sivujen ulkoasuun voi tulla käyttäjätestauksessa ilmenevän palautteen perusteella myöhemmin muutoksia. Jäljellä olevat tehtävät eivät kuitenkaan ole kehitystyön teknisen puolen kannalta erityisen mielenkiintoisia, joten ne olisivat joka tapauksessa rajautuneet opinnäytetyöni ulkopuolelle.

1.2 Tarve verkkosivu-uudistukselle piraattipuolueessa

Puolueella oli jo ennestään toimivat verkkosivut, mutta niihin liittyi joitakin ongelmia. Vanhassa toteutuksessa puolueen avainhenkilöiden, kuten ehdokkaiden ja puolueen päättävien elimien jäsenten, tietoja (mm. nimi, kuva, lyhyt kuvaus ja yhteystiedot) oli näkyvillä lukuisissa eri yhteyksissä ja tietojen muuttuessa ylläpitäjän oli manuaalisesti tehtävä muutokset joka paikkaan erikseen, mikä erityisesti vaalien alla vaati huomattavan suuren määrän työtä ihmisiltä, joilla oli jo muistakin syistä vaalien aikaan paljon tekemistä.

Lisäksi sivujen toteutukseen käytetty WordPressin versio oli jo teknisesti vanhanaikainen, eikä sen päivittäminen nykyaikaiseen versioon olisi ollut helposti

mahdollista joidenkin lisäosien yhteensopivuuden takia. WordPressillä olisi ollut myös hankalaa korjata avainhenkilöiden tietojen päivityksen työläyttä; se olisi käytännössä vaatinut lisää lisäosia. Siksi puolue halusi toteutettavan teknisesti puhtaalta pöydältä kokonaan uudet verkkosivut.

1.3 Parityön eriyttäminen, aiheen rajaus ja projektin työnjako

Eriytimme parityön kahdeksi erilliseksi yksilötyöksi, koska minulla ja parillani oli erilaiset aikataulut. Tämä oli mahdollista, koska työnjakomme oli ollut selkeä ja meillä oli eri roolit projektissa. Eriyttäminen vaikutti opinnäytetyöprosessiin, mutta varsinaiseen kehitysprojektiin sillä ei ollut vaikutusta.

Eriytyksen jälkeen opinnäytetyöni aiheajaus sisältää toteutuksen suunnittelun, sisällönhallintajärjestelmän asennuksen ja hostauksen, tietokantarakenteen toteutuksen, käyttöliittymän toteutuksen ja hostauksen sekä teknologioiden valitsemisen näihin tarkoituksiin. Projektissa minun vastuullani on käytännössä myös lisätä sivuille vanhojen sivujen sisällöt ja hoitaa joitakin ylläpitäjän tehtäviä, mutta en käsittele näitä tässä opinnäytetyössä erityisemmin. Parini vastuualueita olivat projektinhallinta, käyttäjäkokemuksen suunnittelu, käyttäjättestaus ja markkinointi.

Työnjako projektissa määräytyi molempien henkilöiden aikaisemman osaamisen perusteella. Omiin opintoihini oli kuulunut paljon webkehitystä sisältäneitä kursseja ja olin aiemmin joissakin henkilökohtaisissa projekteissani hostannut verkkosivuja pilvipalvelimille, joten osa projektin teknisen toteutuksen kannalta relevanteista teknologioista oli minulle valmiiksi tuttuja ja näin ollen tekninen toteutus oli minulle luonteva vastuualue ja samalla kiinnostava mahdollisuus syventää olemassaollutta osaamistani edelleen.

1.4 Projektin tavoitteet

Kokonaisuudessaan projektin tavoitteena oli saada valmiiksi verkkosivut, jotka piraattipuolue voi ottaa käyttöön vanhojen sivujen tilalle. Uusilla verkkosivuilla oli oltava suunnilleen samat toiminnallisuudet ja sisällöt kuin vanhoillakin. Uudistuksen yhteydessä oli myös mahdollista toteuttaa yleisiä parannuksia esimerkiksi käyttäjäkokemukseen ja tavoitteena oli selvittää puolueen jäsenistön toiveita ja mielipiteitä tähän liittyen kyselyllä. Puolueella oli myös tiettyjä erityisiä vaatimuksia uusille sivuille. Näistä teknisen toteutuksen kannalta oleelliset käsitellään tässä luvussa tarkemmin.

Puolue halusi päästä eroon vanhasta toteutuksesta WordPressin teknologisesti vanhentuneella versiolla, jonka päivittäminen nykyaikaiseen versioon oli lisäosien yhteensopivuuden vuoksi hankalaa. Lisäksi kaikkien vaatimusten toteuttaminen WordPressillä olisi edellyttänyt uusien lisäosien asentamista, mikä olisi entisestään sekavoittanut yhteensopivuusvaatimuksia. Verkkosivu-uudistus hallettiin toteuttaa ilman WordPressiä, jotta samanlaisia ongelmia ei tulisi myöhemmin uudestaan.

Jotta puolue voisi tarvittaessa antaa verkkosivujen sisällöstä huolehtimisen vastuuksi sellaisille henkilöille, joilla ei ole koodaustaitoa tai muuta merkittävää tietoteknistä osaamista, vaatimuksena oli, että sisällön on oltava luotavissa, päivitettävissä ja muokattavissa ilman tällaista osaamista. Esimerkiksi sivuston teknisen toteutuksen tai ulkoasun ei ollut tarpeellista olla yhtä helposti muokattavissa, koska puolueen ei tavallisesti tarvitse tehdä näihin muutoksia.

Vanhoilla sivuilla avainhenkilöiden tietojen päivittäminen kaikkiin eri asiayhteyksiin oli vaatinut paljon manuaalista työtä ylläpitäjältä. Uudistuksen tavoitteena oli luoda tietokanta, josta avainhenkilöiden tiedot haetaan sivuille kaikkiin asiayhteyksiin, jotta tietojen muuttuessa muutokset voidaan tehdä tietokantaan ja näin ollen samoja muutoksia ei tarvitse turhaan tehdä useita kertoja. Lisäksi puolueen toiveena oli saada luotua avainhenkilöille tunnukset, joilla he pääsevät kirjautumaan sisään ja muokkaamaan omaa profiiliaan, mikä edelleen vähentäisi

ylläpitäjien työmäärää. Tätä varten sivuille tarvittiin myös käyttöliittymä sisäänkirjautumiseen ja omien tietojen muokkaamiseen.

1.5 Eettisyys ja turvallisuus

Opinnäytetyössä toteutetaan verkkosivu, jossa henkilöillä on heistä tietoja sisältäviä profiileja. Kaikki henkilöt ovat itse päättäneet, mitä tietoja he ovat tätä tarkoitusta varten piraattipuolueelle lähettäneet ymmärtäen, että tiedot julkaistaan Internetissä piraattipuolueen verkkosivuilla. Henkilötietojen sisältö on sekä projektissa työskentelyni kannalta että opinnäytetyöni kannalta täysin tangentialista enkä käsittele sitä mitenkään. Tässä dokumentissa käytän fiktiivistä testidataa profiileihin liittyvää toiminnallisuutta havainnollistavissa kuvakaappauksissa.

Kaikkiin kehitysprojekteihin liittyy riski siitä, että kehitettyyn sovellukseen jää tietoturva-aukkoja, joita ei kehityksen tai testauksen aikana huomata. Tässä projektissa riskiä kasvattaa se, että käytän joitakin teknologioita, joista minulla ei ole aikaisempaa kokemusta. Projektiin liittyy myös käyttäjien autentikointia ja käyttäjien luomaa sisältöä verkkosivuilla, jotka ominaisuuksina ovat sellaisia, että niitä toteutettaessa on kiinnitettävä erityistä huomiota tietoturvaan. Riskejä ei ole mahdollista kokonaan eliminoida, mutta olen kehitystyön kaikissa vaiheissa pyrkinyt kykyjeni mukaan huomioimaan tietoturvan ja olemaan mahdollisimman hyvin kartalla siitä, mitä käyttämäni teknologiat konkreettisesti tekevät.

1.6 Tekoälyn käyttö opinnäytetyössä

Käytin tiedonhankinnassa ChatGPT 3.5 -tekoälyä apuna keksimään avainsanoja relevanttien julkaisujen hakemiseen ja DeepL-tekoälyä joidenkin muiden kuin englannin- tai suomenkielisten julkaisujen kääntämiseen. En käyttänyt lähteenä mitään tekoälykäännöksenä lukemaani julkaisua, mutta halusin olla niidenkin sisällöstä suunnilleen tietoinen osatakseni paremmin arvioida käyttämiäni lähteiden edustavuutta aiheista, joista lähteitä oli vain vähän.

Lisäksi käytin ChatGPT 3.5 -tekoälyä generoimaan testidataa fiktiivisistä henkilöistä. Käytin tätä dataa joidenkin verkkosivun ominaisuuksien esittelyyn tässä dokumentissa.

2 Kehityksessä käytetyt teknologiat

2.1 Teknologioiden valintaperusteet

Piraattipuolue on linjannut suosivansa avoimien ja vapaiden ohjelmistojen käyttöä johtuen niiden taloudellisuudesta, jatkokehityksen joustavuudesta ja tietoturvan paremmasta auditoinnista (Piraattipuolue 2021). Tämä linjaus vaikuttaa myös verkkosivu-uudistusprojektiin, vaikka se ei olekaan täysin sitova.

Käytännössä olen pyrkinyt valitsemaan avoimia ja vapaita ratkaisuja aina, jos se on ollut kohtuullisesti mahdollista eikä huomattavasti parempaa ja/tai minulle valmiiksi tutumpaa yksityisen lisenssin vaihtoehtoa ole ollut saatavilla. Useimmissa tapauksissa tämä on ollut helppoa, koska laadukkaita ja helppokäyttöisiä avoimia ratkaisuja webkehityksen tarpeisiin on nykyään saatavilla runsaasti. Niitä käytetäänkin laajasti myös työelämässä (GitHub 2017).

Lisäksi tavoitteena on ollut pitää verkkosivun kehityksen ja ylläpidon kustannukset mahdollisimman alhaisina. Siksi käytetyt teknologiat ovat rajautuneet avoimien ja vapaiden vaihtoehtojen lisäksi muihin ilmaisiin tai edullisiin vaihtoehtoihin.

Perustelen tässä luvussa, miksi olen päätenyt valitsemini teknologioihin. Luku toimii samalla opinnäytetyön tietoperustana.

2.2 Strapi

Strapi on avoimen lähdekoodin headless-sisällönhallintajärjestelmä eli CMS, jonka hallintapaneelin kautta voidaan muokata ja julkaista sisältöjä (Strapi 2023a; Sobri, Abas, Yassin, Ali, Tahir, Zabidi & Rizman 2022, 932). Avoimen lähdekoodin ratkaisuna se sopii hyvin puolueen arvoihin. Myös Strapin kustannukset pysyvät alhaisina, koska ohjelmistona se on ilmainen ja se voidaan hostata puolueen omalla palvelimella. Headless tarkoittaa tässä yhteydessä sitä, että sisällönhallintajärjestelmään ei kuulu valmista verkkosivun käyttöliittymää, eli verkkosivun kehittäjän on luotava se itse joillakin muilla teknologioilla ja sisällönhallintajärjestelmässä on ohjelmointirajapinta, josta sisällöt voidaan hakea käyttöliittymään riippumatta siitä, millä teknologioilla käyttöliittymä on toteutettu. Tämä toisaalta tuo kehitykseen joustavuutta, koska kehittäjien ei tarvitse olla sidottuja tiettyihin ennalta määrättyihin teknologioihin. (Strapi 2023b, 5.)

Strapissa on myös sisäänrakennettuna käyttäjänhallintajärjestelmä (Sobri ym. 2022, 932), mistä on tässä kehitysprojektissa huomattavasti hyötyä. Yksi puolueen toiveista uuden verkkosivun suhteen on, että puolueen avainhenkilöt pystyisivät itse muokkaamaan omia profiilejaan käyttäjätunnuksella ja salasanalla. Strapin valmista käyttäjänhallintajärjestelmää voi käyttää tähän tarkoitukseen.

Sisällönhallintajärjestelmältä vaaditaan tässä projektissa käyttäjäystävällisyyttä, koska myöhemmin sivuston sisältöjä saattavat päivittää sellaisetkin henkilöt, joilla ei ole erityistä teknistä osaamista. Strapin soveltuvuutta opettajien käyttöön kurssisisältöjen hallinnoimista varten selvittäneessä tutkimuksessa havaittiin, että sen hallintapaneelin käyttöliittymä oli käyttäjäystävällinen ja johdonmukainen (Bauer, Leal & Queirós 2023, 4). Toisaalta havaittiin myös haasteita ja epäintuutiivisuutta ohjelmointitehtäviä sisältävien kurssisisältöjen luomisessa. Ongelmat liittyivät mm. kokoelmien luomiseen ja niiden yhdistämiseen toisiinsa. (Bauer ym. 2023, 7.) Tämä ei ole piraattipuolueen verkkosivujen kannalta oleellisesti haitallista, koska tällaisia muutoksia ei tarvitse tehdä sivuston rutiinimaisen sisältöjen päivityksen yhteydessä. Mathias Tannerin (2020, 9–10) opinnäytetyössä, joka käsittelee verkkosivujen toteutusta sveitsiläiselle kampaailulajiseuralle, on ollut sama vaatimus sisältöjen päivittämisestä ilman teknistä

osaamista ja hän on valinnut sisällönhallintajärjestelmäksi Strapin etupäässä sen käyttäjäystävällisyyden vuoksi. Strapi todettiin käyttäjäystävälliseksi ei-teknisille käyttäjille myös vertailussa Directusin ja Supabasen kanssa, jotka ovat eräitä muita vaihtoehtoja verkkosivun backendeiksi (Sobri ym. 2022, 935).

Strapi ei ollut minulle ennestään tuttu teknologia, joten on erityisen tärkeää, että Internetistä löytyy hyödyllisiä resursseja, joista on apua kehitysprojektissa. Directusiin ja Supabaseen verrattuna kaikilla kolmella vaihtoehdolla oli olemassa hyvä virallinen dokumentaatio ja Strapi oli vaihtoehdoista suosituin GitHubissa (Sobri ym. 2022, 935).

2.3 React

React on Metan kehittämä, alun perin vuonna 2013 julkaistu, käyttöliittymien kehitykseen tarkoitettu avoimen lähdekoodin JavaScript-kirjasto (Thakkar 2020, 61). Valitsin käyttöliittymän kehitykseen Reactin, koska se oli minulle valmiiksi tuttu, sen avulla on suhteellisen helppoa luoda toimivia ja ajanmukaisia webbkäyttöliittymiä ja se on ilmaisena vapaan lisenssin kirjastona sekä puolueen arvoihin että projektin matalaan budjettiin hyvin sopiva vaihtoehto.

Myös Mathias Tanner (2020, 8) on käyttänyt Reactia luodakseen käyttöliittymän Strapi-backendille. Reactin lisäksi eräs toinen merkittävä vaihtoehto olisi ollut Next.js, joka on avoimen lähdekoodin React-ohjelmistokehys (Vercel 2023) ja näin ollen olisi myös soveltunut puolueen arvojen ja budjetin mukaiseen kehitysprojektiin. Next.js:ään Strapi-backendin kanssa olivat päätyneet omissa webkehitykseen liittyvissä opinnäytetöissään An Nguyen (2022, 8) ja Niko Pinnis (2022, 8).

Next.js:ssä on joitakin etuja puhtaaseen Reactiin verrattuna. Se esimerkiksi mahdollistaa sivun käyttöliittymän renderöinnin palvelimen puolella (Dinku 2022, 9; Vercel 2023) ja sillä kehitys on siinä mielessä helpompaa, että koodia tarvitsee kirjoittaa vähemmän (Dinku 2022, 22–23). Reactin etuja puolestaan on sivujen nopeampi latautuminen käyttäjälle (Dinku 2022, 23–27) ja sen yleisyys

(Dinku 2022, 15–19; Statista 2022), jonka vuoksi Internetistä löytyy enemmän kehittäjille hyödyllisiä resursseja kuten blogikirjoituksia ja YouTube-tutoriaaleja (Dinku 2022, 20). Näiden eroavuuksien valossa en nähnyt järkeväksi lähteä opettelemaan Next.js:ää kehitystyötä varten.

React on itsessään hyvin kevyt kirjasto ja sisältää pelkästään keskeisimmät toiminnot, jotta sillä tehtyjen sovellusten ei tarvitse turhaan sisältää suurta määrää kirjaston mukana tulevia käyttämättömiä ominaisuuksia. Tämä toisaalta tarkoittaa sitä, että kehittäjän on itse luotava tarvitsemansa ominaisuudet. (Meta Platforms 2022.)

Reactin ydintoimitoja ovat React-komponentit ja reititys. React-komponentit ovat tyypillisesti HTML-näkymiä tai niiden osia. Niiden avulla tietty käyttöliittymäelementti tai useampien elementtien kokonaisuus saadaan eristettyä omaksi React-komponentiksi, jota voidaan helposti käyttää useita kertoja eri yhteyksissä. React-komponentit voivat myös sisältää toisia React-komponentteja. (Japikse, Grossnicklaus & Dewey 2020, 542.) Reititys pitää huolen siitä, että verkkosivun käyttäjälle näytetty sisältö vastaa URL-osoitetta, johon käyttäjä on navigoinut, sekä välittää React-komponenteille URL-osoitteen kautta dataa (Japikse ym. 2020, 558).

2.4 Nginx

Nginx on avoimen lähdekoodin HTTP-palvelin ja käänteinen proxy (Liu, Xu, Wang & Zhang 2018, 116). Tämän kehitysprojektin yhteydessä sitä tarvitaan sisällönhallintajärjestelmän ja sivun käyttöliittymän tietoturvalliseen hostaamiseen.

Vertailussa Apachen ja lighttpd:n kanssa, jotka ovat eräitä muita suosittuja HTTP-palvelimia, kaikkein pienimpiä tiedostoja lähetettäessä nginx suoriutui hieman paremmin kuin lighttpd ja huomattavasti paremmin kuin Apache. 64 kilotavun kokoisissa ja suuremmissa tiedostoissa erot tasoittuivat. (Liu ym. 2018, 117–118).

Päädyin nginxiin ensisijaisesti, koska piraattipuolue on käyttänyt sitä aikaisemminkin. Lisäksi Strapiin dokumentaatiosta löytyy selkeät ohjeet sen hostaamisesta nginxillä (Strapi 2023c).

2.5 Ubuntu Server

Ubuntu Server on avoimen lähdekoodin käyttöjärjestelmä, joka on tarkoitettu erityisesti palvelimia ja datakeskuksia varten (Ubuntu 2023). Valitsin Ubuntu Serverin käyttöjärjestelmäksi palvelimelle, jonne verkkosivu ja sisällönhallintajärjestelmä hostataan, koska se on ilmainen avoimen lähdekoodin järjestelmä ja minulla on aikaisempaa kokemusta verkkosivujen hostaamisesta Ubuntu Serverillä.

Käyttöjärjestelmän valinnassa oli huomioitava, että kaikista palvelimelle tarvituista ohjelmista on olemassa käyttöjärjestelmälle soveltuvat versiot. Sen oli oltava myös asennettavissa sekä piraattipuolueen palvelimelle että omalle pilvipalvelimelleni, jota käytin kehityksen aikana. Ubuntu Server täytti molemmat vaatimukset.

2.6 GraphQL

GraphQL on kyselykieli, jolla voidaan suorittaa kyselyitä ohjelmointirajapinnassa (GraphQL 2023). Tähän kehitysprojektiin tarvittiin jonkinlainen menetelmä siihen, kuinka Strapi-sovelluksesta saadaan sisällöt haettua loppukäyttäjän selaimen. GraphQL oli luonteva valinta, koska koen sen helppokäyttöiseksi ja Strapiin on saatavilla virallinen laajennus, joka hoitaa GraphQL-kyselyiden käsittelyn Strapi-sovelluksen päässä (Strapi 2023d).

Toinen suunnilleen yhtä luonteva vaihtoehto olisi ollut REST, jonka Strapi toteuttaa joka tapauksessa automaattisesti (Strapi 2023d) ja jonka käytöstä minulla on GraphQL:ää enemmän kokemusta. GraphQL on kuitenkin RESTiä oleellisesti tehokkaampi; se palauttaa kyselyihin vastaukset 35–46 %

nopeammin ja vastaukset ovat tiedostokooltaan 21–71 % pienempiä (Oggier 2020, 27–33).

2.7 Visual Studio Code

Visual Studio Code on Microsoftin kehittämä ilmainen ja alustariippumaton koodieditori (Lardinois 2015). Microsoft ylläpitää GitHubissa Code – OSS -repositoriota, jossa pääosa Visual Studio Coden kehityksestä tapahtuu avoimen lähdekoodin projektina ja jonka koodin päälle Visual Studio Code rakentuu joillakin Microsoftin yksityisen lisenssin lisäyksillä (Microsoft 2021). Visual Studio Code ei siis määritelmällisesti ole avoimen lähdekoodin ohjelma, mutta nähdäkseni sitä voidaan kehitystyössä hyvin käyttää siitä huolimatta, koska koodieditori ei missään muodossa päädy osaksi työn lopputulosta ja käytännössä se on suunnilleen yhtä hyvin jatkokehitettävissä ja tietoturvan suhteen auditoitavissa kuin avoimen lähdekoodin ohjelmat johtuen siitä, että suurin osa koodista on avointa.

Valitsin Visual Studio Coden siksi, että olen tottunut käyttämään sitä kehitysympäristönä. Siinä on joitakin sellaisia toimintoja, kuten integroidut komentorivit ja Git-versionhallinnan käyttöliittymä, jotka tehostavat työskentelyäni ja joita monissa muissa suosituissa vaihtoehdoissa ei ole.

3 Toteutus

3.1 Viestintä ja versionhallinta

Lähtökohtaisena ajatuksena projektin viestinnässä oli, että parini vastaa viestinnästä puolueen kanssa liittyen käyttäjäkokemukseen, visuaaliseen ilmeeseen, toivottuihin ominaisuuksiin, palautteen keräämiseen testiversioista, aikatauluihin ja muihin käytännön järjestelyihin, keskustelelee näistä asioista myös minun kanssani ja muodostaa näiden keskusteluiden perusteella oman näkemyksensä, jonka mukaan projektissa edetään. Teknisissä kysymyksissä esimerkiksi

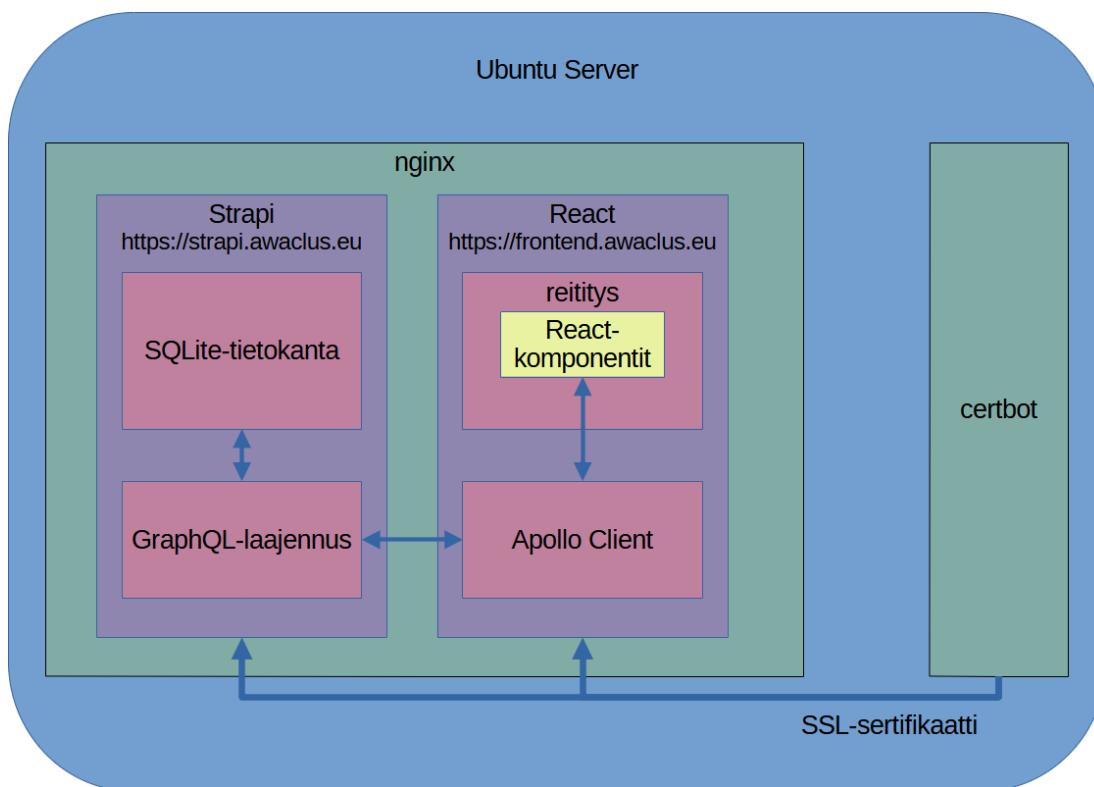
puolueen palvelimiin ja domain-nimiin liittyen olin itse suoraan yhteydessä puolueen verkkopalvelutiimiin, joka tällaisista asioista puolueessa vastaa. Lisäksi verkkosivu-uudistuksesta on puhuttu puolueen ja puoluevaltuuston tavanomaisissa kokouksissa ja projektin varrella on järjestetty muutamia erityisesti verkkosivu-uudistukseen keskittyneitä etäkokouksia aiheesta kiinnostuneille puolueaktiiveille. Näissä kokouksissa olemme molemmat olleet paikalla.

Versionhallinta projektissa tapahtui Gitillä. Projektille luotiin sitä varten Git-repositorio piraattipuolueen Gitea-instanssille.

3.2 Arkkitehtuuri

3.2.1 Arkkitehtuurin yleiskuva

Kuviossa 1 on esitetty verkkosivujen arkkitehtuuri yksinkertaistettuna kaaviona. Kaavio kuvaa arkkitehtuurin erityisesti sovelluskehityksen ja palvelimen ohjelmistojen asennuksen näkökulmasta. Verkkoliikenteen tai sovellusten ajamisen näkökulmasta kaavio olisi erilainen.



Kuvio 1. Kaavio verkkosivujen arkkitehtuurista.

Verkkosivu koostuu kahdesta erillisestä sovelluksesta: React-sovelluksesta, joka toimii verkkosivun käyttöliittymänä, ja Strapi-sovelluksesta, joka toimii sisällönhallintajärjestelmänä. Molemmat sovellukset sijaitsevat samalla Ubuntu Server -palvelimella, jossa nginx ohjaa liikenteen aladomainin perusteella oikealle sovellukselle (strapi.awaclus.eu sisällönhallintajärjestelmälle ja frontend.awaclus.eu käyttöliittymälle). Palvelimella on Electronic Frontier Foundationin certbotilla luodut SSL-sertifikaatit molemmille aladomaineille. Kehitysvaiheessa käytin omaa domainiani päästäkseni hallinnoimaan sitä itse, mutta domainit tulevat olemaan jotain muuta, kun puolue myöhemmin ottaa verkkosivut käyttöön.

Tietokantana Strapi-sovelluksessa käytän SQLiteä. Tietokanta sisältää verkkosivujen kaiken sellaisen sisällön, jota on tarkoitus pystyä helposti luomaan ja muokkaamaan. Tällaista sisältöä on muutamaa poikkeusta lukuunottamatta kaikki sisältö verkkosivuilla.

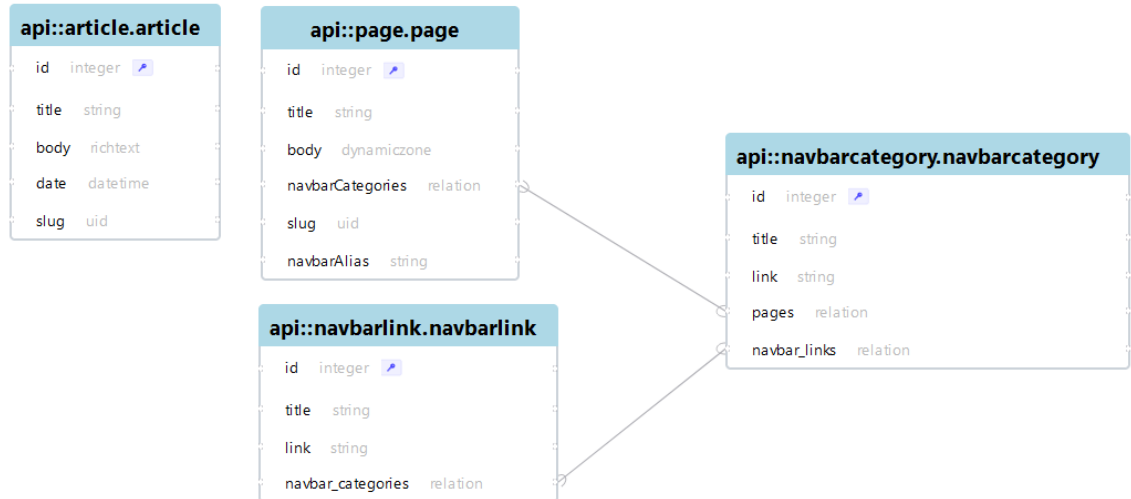
React-sovellus koostuu reitityksestä ja React-komponenteista. React-komponentit ovat käyttöliittymän käyttäjälle näkyvä osa ja sisältävät myös

käyttöliittymän toiminnallisuutta. Reititys määrittää URL-osoitteen perusteella, mitkä React-komponentit sivulla näytetään.

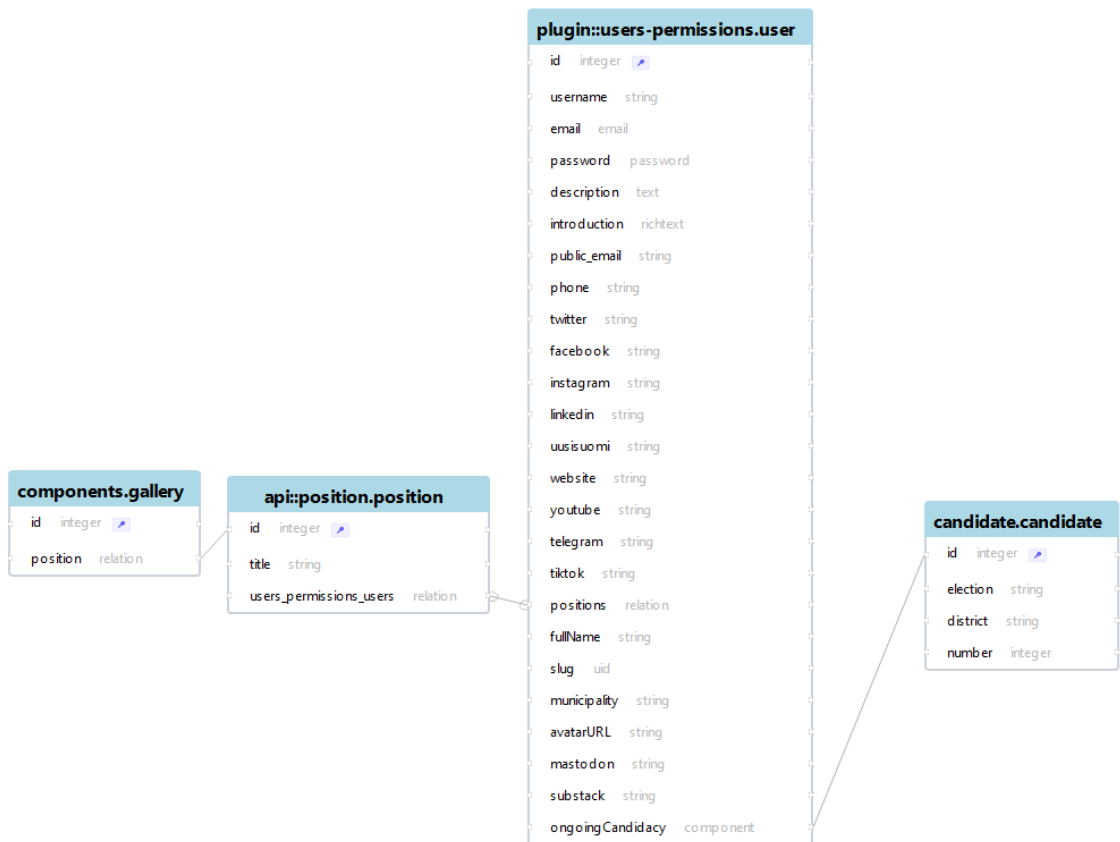
Sovellusten väliseen kommunikaatioon käytin GraphQL-kyselykieltä, mitä varten asensin Strapi-sovellukseen Strapin virallisen GraphQL-laajennuksen ja React-sovellukseen Apollo Clientin. React-komponentit tekevät GraphQL-kyselyitä, jotka Apollo Client lähettää Strapin GraphQL-laajennukselle, joka käsittelee kyselyt ja palauttaa niitä vastaavat tiedot SQLite-tietokannasta Apollo Clientille, josta React-komponentit saavat tiedot käyttöönsä.

3.2.2 Tietokanta

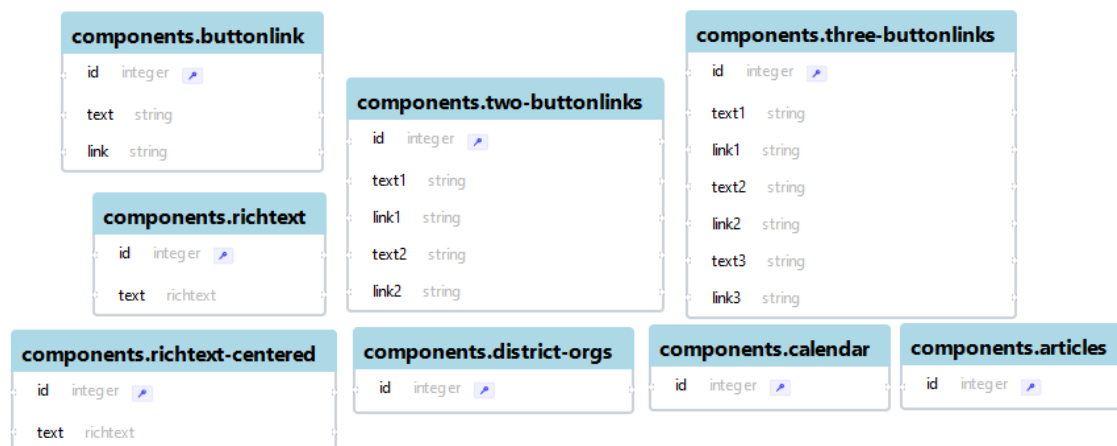
Kuvioissa 2, 3 ja 4 on esitelty ty pistetty versio Strapi-sovelluksen tietokantakaaviosta. Kaaviot on luotu Strapin Models Relation Diagram -laajennuksella ja niitä on tilan säästämiseksi ty pistetty piilottamalla esimerkiksi Strapi-sovelluksen ylläpitopaneeliin ja mediakirjastoon liittyviä ja muita automaattisesti Strapin asennuksen yhteydessä luotuja tauluja, joihin en ole tehnyt muutoksia. Myös kaavioissa näkyvistä tauluista on piilotettu automaattisesti luotuja metadataa sisältäviä kenttiä, kuten luonti- ja muokauspäivämäärät. Kentät, joiden nimet alkavat "api:." tai "plugin:." ovat kokoelmia, muut ovat komponentteja. Page-kokoelman body-kenttä on tyypiltään Dynamic Zone, joka käytännössä toimii niin kuin sillä olisi monta moneen -relaatio jokaisen "components"-alkuisen komponentin kanssa, mutta tämä relaatio ei näy kaavioissa, koska kenttä ei ole tyypiltään relaatio.



Kuvio 2. Navbarmenu-, Page-, NavbarLink- ja Article-kokoelmien tietokantakaavio.



Kuvio 3. User- ja Position-kokoelmien ja Gallery- ja Candidate-komponenttien tietokantakaavio.



Kuvio 4. Muiden komponenttien tietokantakaavio.

Strapissa on valmiiksi asennettuna käyttäjienhallintaan tarkoitettu Users & Permissions -laajennus, joka automaattisesti luo tietokantaan User-kokoelman, joka sisältää valmiiksi joitakin kenttiä, kuten Username, Email ja Password. Näitä kenttiä ei voi muokata tai poistaa, mutta kenttiä voi lisätä. (Strapi 2024b.) Käytin tätä valmista toteutusta käyttäjienhallintaan ja lisäsin kokoelmaan avainhenkilön julkiseen profiiliin liittyvät kentät, jotka näkyvät kuviossa 2 ja aiheeseen palataan tarkemmin luvuissa 3.6.5 ja 3.7. Users-kokoelmalla on monta moneen -relaatio Position-kokoelman kanssa, mikä käytännössä tarkoittaa sitä, että puolueella voi olla useita henkilöitä samassa asemassa (esim. puoluevaltuusto tai tietyn piirijärjestön hallitus) ja samalla henkilöllä voi olla useita asemia. Tätä relaatiota hyödynnetään henkilögallerioissa, jotka listaavat tietyssä asemassa olevat puolueen avainhenkilöt, mitä varten Gallery-komponentilla on monta yhteen -relaatio Position-kokoelman kanssa. Users-kokoelmalla on myös yksi yhteen -relaatio Candidate-komponentin kanssa, eli mikäli henkilö on vaaleissa ehdolla, hänen profiiliinsa voidaan lisätä ilmoitus, josta käy ilmi, missä vaaleissa, missä vaalipiirissä ja millä numerolla hän on ehdolla.

Page-kokoelma vastaa verkkosivujen sivuja, jotka sisältävät suurimman osan sivuston asiasisällöstä. Sivulla on otsikko, mielivaltainen määrä kuviossa 2 esiteltäviä "components"-alkuisia komponentteja, monta moneen -relaatio Navbar-Category-kokoelman kanssa, slug ja vapaaehtoinen navbarAlias, jota käytetään sivun otsikkona navigointivalikossa tavanomaisen otsikon sijaan, mikäli sellainen on annettu. NavbarCategory-kokoelma vastaa sivun navigointivalikon

pääkategorioita. Kategoriolla on oltava nimi, joka valikossa näkyy, ja URL-osoite, johon kategorian otsikko on linkki. Lisäksi kategoriolla on monta moneen -relaatio Page- ja NavBarLink-kokoelmien kanssa; nämä edustavat linkkejä kategorian alle avautuvassa pudotusvalikossa. NavBarLink voi olla linkki mihin tahansa URL-osoitteeseen ja sillä on oltava URL-osoitteen lisäksi otsikko.

NavBarCategoryyn lisätyt Page:t käyttävät Pagen otsikkoa tai navbarAliasta ja URL-osoitetta. Relaatio Page-kokoelmaan on periaatteessa turha, koska samoihin tuloksiin voisi päästä käyttämällä pelkästään NavBarLinkejä, mutta ylläpitäjän voi olla helpompaa lisätä kategorioiden pudotusvalikkoihin linkit suoraan relationa Page-kokoelmaan, jolloin hänen ei tarvitse huolehtia otsikoista, linkeistä ja niiden pitämisestä ajantasaisina myöhemmin mikäli Pagea muokataan.

Article-kokoelma vastaa uutisartikkeleita. Uutisilla on otsikot, tekstisisällöt, julkaisupäivämäärät ja slugit.

3.3 Reititys

Reititys on toteutettu React Router DOM -paketilla. Polku "/" on reititetty React-komponentille Homepage, jossa on sivuston etusivu.

Polku "/:slug", jossa :slug on muuttuja, on reititetty React-komponentille Page, joka hakee sisällönhallintajärjestelmän tietokannasta sivun, jonka slug on sama kuin :slug ja näyttää sen sisällön (esimerkiksi jos polku olisi "/politiikka", etsittäisiin tietokannasta sivu, jonka slug on "politiikka") tai näyttää React-komponentin NotFound, jos sellaista ei ole.

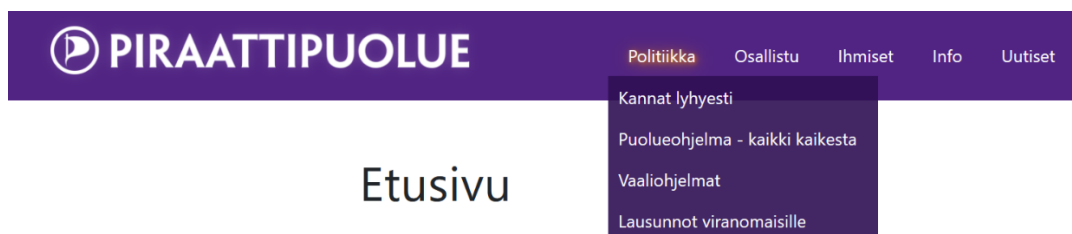
Polku "/ihmiset/:slug" on reititetty React-komponentille Profile, joka hakee tietokannasta käyttäjän, jonka slug on :slug ja näyttää hänen profiilinsa tai näyttää NotFoundin, jos sellaista ei ole. Vastaavalla tavalla polku "/uutiset/:slug" on reititetty React-komponentille Article.

Polut `"/users/login"`, `"/users/logout"` ja `"/users/editprofile"` on reititetty React-komponenteille Login, Logout ja User (tässä järjestyksessä), jotka liittyvät käyttäjän sisään- ja uloskirjautumiseen ja profiilin muokkaamiseen.

Kaikki muut polut on reititetty React-komponentille NotFound. NotFound kertoo käyttäjälle, ettei kyseisessä osoitteessa ole mitään.

3.4 Header

Sivuston jokaisen sivun yläreunassa on header eli ylätunniste. Headerissa on linkkinä etusivulle toimiva puolueen grafiikka ja navigointivalikko (kuva 1).



Kuva 1. Sivuston header, kun hiiren kursori on viety Politiikka-kategorian päälle.

Toiminnallisesti header on hyvin yksinkertainen. Kun hiiren kursorin vie jonkin navigointivalikon kategorian päälle, kyseiseen kategoriaan yhdistetyt sivut ja linkit avautuvat sen alle pudotusvalikoksi. Valikko sulkeutuu, kun kursori siirretään otsikon ja valikon päältä pois. Myös kategorian otsikko itse on linkki jollekin sivulle.

Sivuston ylläpitäjä voi helposti lisätä, poistaa ja muokata valikon kategorioita ja niihin yhdistettyjä sivuja ja linkkejä ilman koodaustaitoa, koska ne määritellään sisällönhallintajärjestelmässä. React-sovellus hakee tiedot sisällönhallintajärjestelmän tietokannasta, kun sivua ladataan. Puolueen grafiikka ei ole yhtä helpposti muokattavissa, koska se sijaitsee suoraan React-sovelluksessa, eikä sitä yleensä tarvitsekaan muokata. Se voidaan kuitenkin tarvittaessa vaihtaa ilman koodaustaitoa korvaamalla palvelimella sijaitseva kuvatiedosto toisella samannimisellä tiedostolla.

3.5 Sivut

Sivut sisältävät suurimman osan sivuston asiasisällöstä. Ylläpitäjä voi luoda, poistaa ja muokata sivuja helposti sisällönhallintajärjestelmässä. Useimmille sivuille tarvitaan pelkästään muotoiltua tekstisisältöä, kuvia ja painikkeita, mutta joillakin sivuilla on harvinaisempia ominaisuuksia; jotkin ominaisuudet on toteutettu vain yhtä sivua silmälläpitäen. Ominaisuuden tai sisällön lisääminen sivulle sisällönhallintajärjestelmässä tapahtuu konkreettisesti lisäämällä Page-kokoelman jäsenen body-kenttään uusi komponentti. Järjestelmässä on eri tyyppisiä komponentteja erilaisia sisältöjä varten. Eri tyyppiset komponentit on esitelty luvussa 3.6.

Kun sivu ladataan käyttöliittymäsovelluksessa, sovellus hakee tietokannasta kaikki sivun komponentit ja renderöi niiden mukaisen sisällön ylhäältä alaspäin siinä järjestyksessä, mihin ylläpitäjä on ne Strapi-sovelluksessa järjestänyt. Ylimmäiseksi renderöidään sivun otsikko. Jokaista eri komponentin tyyppiä varten käyttöliittymäsovelluksessa on oma toteutuksensa siihen, kuinka sen tyyppinen komponentti renderöidään.

3.6 Komponentit

3.6.1 Muotoiltu teksti

Sisällönhallintajärjestelmässä muotoiltua tekstiä vastaavat richtext- ja richtextCentered-komponentit, jotka molemmat sisältävät ylläpitäjän muokattavissa olevan richtext-tyyppisen kentän. Tekstin muotoilu tapahtuu Markdown-koodilla. Strapin tekstieditorissa on kuitenkin hiirellä klikattavat painikkeet, joiden avulla haluttu muotoilu voidaan lisätä ilman tarvetta osata Markdownia sekä esikatsetulitila, jonka saa halutessaan yhtä aikaa tekstieditorin kanssa näkyviin (Strapi 2024c). Kahdella eri komponentilla ei ole sisällönhallintajärjestelmän näkökulmasta mitään muuta eroa kuin nimi, mutta React-sovellus renderöi ne eri tavalla: richtext tasataan vasemmalle, richtextCentered keskelle. Tämä toteutus

johtuu siitä, että Markdown itsessään ei sisällä erilaisia tapoja tasata tekstiä (Markdown Land 2021).

React-sovellus käyttää React Markdown -pakettia Markdownin renderöintiin. Kaikille otsikoille lisätään HTML:ssä id, joka muodostetaan automaattisesti otsikon tasosta ja sisällöstä (esimerkiksi h2-tasoisien otsikon ”Piraattipuolueen periaateohjelma” id olisi ”h2-piraattipuolueen-periaateohjelma”), jotta niihin voidaan tarvittaessa linkittää siten, että näkymä siirtyy kyseiseen otsikkoon linkkiä klikatessa.

3.6.2 Klikattavat painikkeet

Sisällönhallintajärjestelmässä klikattavia painikkeita edustavat komponentit `buttonlink`, `twoButtonlinks` ja `threeButtonlinks`. Ne sisältävät painikkeiden määrästä riippuen 2–6 merkkijonokenttää; painiketta kohden yksi kenttä painikkeen tekstille ja yksi kenttä linkin URL-osoitteelle. Kuvassa 2 on esimerkki klikattavista painikkeista React-sovelluksessa.

Seuraa piraattipuoluetta myös muualla verkossa



Kuva 2. Richtext-komponenttiin kuuluva otsikko ja `threeButtonlinks`-komponentti renderöitynä käyttöliittymäsovelluksessa.

React-sovelluksessa painikkeet renderöidään yksinkertaisesti HTML-buttoneina, jotka toimivat linkkeinä. `twoButtonlinks` ja `threeButtonlinks` renderöidään taulukkoon, jotta painikkeet saadaan vierekkäin.

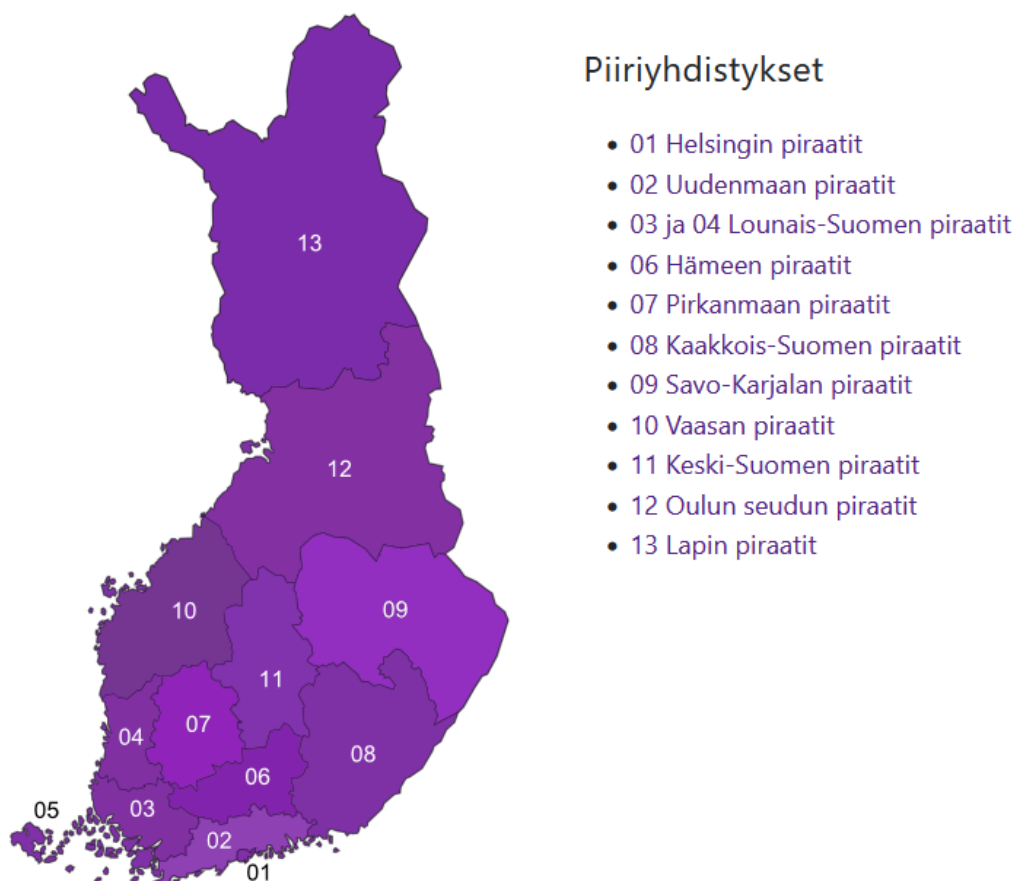
3.6.3 Kalenteri

Kalenteri on käytännössä vain yhtä sivua varten erikseen toteutettu komponentti. Ylläpitäjän on kuitenkin mahdollista lisätä useammillekin sivuille kalentereita. Calendar-komponentti ei sisällä mitään muokattavia kenttiä.

React-sovelluksessa renderöidään upotettu Google Calendar -kalenteri, johon on merkitty puolueen tapahtumia ja muita huomionarvoisia päiviä. HTML-koodi upotukseen saadaan kopioitua suoraan Google Calendarista (Google 2024).

3.6.4 Piiriyhdistysten kartta

Yhdelle sivulle tarvitaan puolueen piiriyhdistyksistä kartta, jossa eri vaalipiirit toimivat linkkeinä kyseisten alueiden piiriyhdistysten sivuille. Tätä ominaisuutta varten sisällönhallintajärjestelmässä on DistrictOrgs-komponentti. Kuva 3 havainnollistaa ominaisuutta verkkosivuilla.

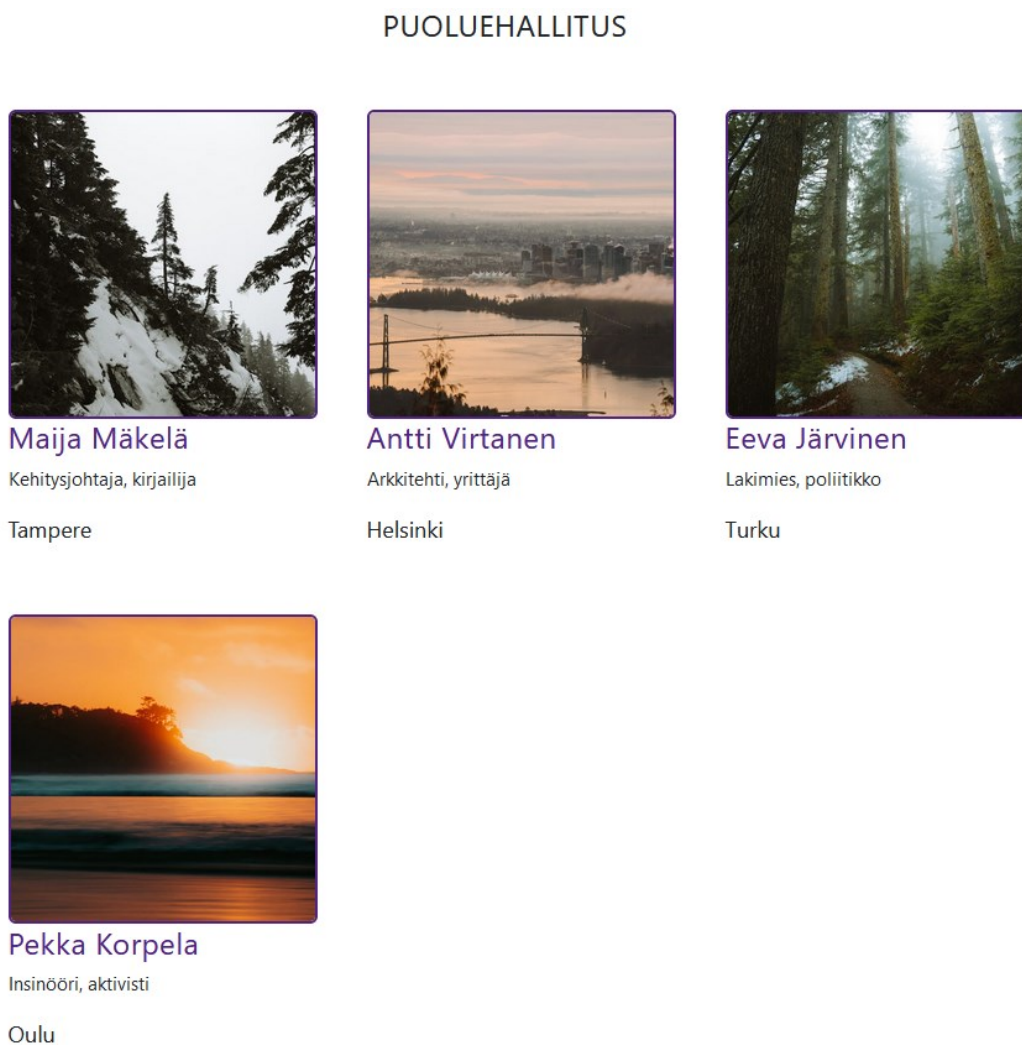


Kuva 3. DistrictOrgs-komponentti renderöitynä käyttöliittymäsovelluksessa.

DistrictOrgs-komponentissa ei ole muokattavia kenttiä. React-sovelluksessa renderöidään kuva Suomen kartasta jaettuna vaalipiireihin, jossa HTML:n map- ja area-tageilla toteutettuna suunnilleen vaalipiirien muotoiset alueet toimivat linkkeinä kunkin alueen piiriyhdistyksen sivulle. Täsmälleen oikean muotoisten alueiden koodaaminen olisi ollut erittäin työlästä, mutta suhteellisen yksinkertaiset, enintään 14-kulmaiset monikulmiot ovat käytännössä riittävän tarkka likiarvo, kunhan ne kuitenkin pysyvät kartassa näkyvien alueiden rajojen sisäpuolella. Lisäksi kartan viereen renderöidään tekstimuotoinen luettelo piiriyhdistyksistä, missä on myös linkit piiriyhdistysten sivuille.

3.6.5 Henkilögalleria

Henkilögalleria on ominaisuus, jota tarvitaan muutamilla eri sivuilla. Kuvassa 4 on esimerkki henkilögalleriasta verkkosivuilla.



Kuva 4. Gallery-komponentti renderöitynä React-sovelluksessa. Henkilöt ovat fiktiivisiä ja heidän nimensä, tittelinsä ja kotikuntansa on generoitu ChatGPT 3.5 -tekoälyllä (ChatGPT 2024). Kuvat ovat Luke Millerin ottamia ja vapaasti käytettäviä (Miller 2024a; Miller 2024b; Miller 2024c; Miller 2024d).

Sisällönhallintajärjestelmässä Gallery-komponentilla on kenttänä relaatio Position-kokoelman kanssa. Toisin sanoen ylläpitäjä voi valita, missä asemassa tai roolissa olevat henkilöt näytetään kyseisessä galleriassa. Position-kokoelmalla on relaatio User-kokoelman kanssa, joka määrittää, mitkä henkilöt ovat

kyseisessä asemassa tai roolissa. Position-kokoelmalla on myös otsikko, jonka Gallery-komponentin renderöintilogiikka renderöi gallerian yläosaan, kuten kuva 4 havainnollistaa.

Käyttöliittymäsovellus hakee GraphQL-kyselyllä tietokannasta kaikki galleriaan kuuluvat henkilöt ja galleriaan tarvittavat tiedot heistä, kun sivu, jolla Gallery-komponentti on, ladataan. Kuva lähetetään tietokannasta URL-osoitteen muodossa, joka voi olla kokonainen osoite tai pelkkä slug. React-sovellus tunnistaa, kummanlaisesta osoitteesta on kysymys ja joko renderöi kyseisessä URL-osoitteessa sijaitsevan kuvan tai, jos URL palauttaa jonkin virheilmoituksen, oletusprofiilikuvan sen sijaan. Henkilöt näytetään galleriassa siinä järjestyksessä, missä heille on lisätty relaatio kyseiseen Positioniin, ellei ylläpitäjä järjestä Positionin relaatioita eri järjestykseen, mikä tarvittaessa onnistuu Strapin ylläpityspaneelista helposti (Strapi 2024d). Henkilögalleriassa näytettävien henkilöiden määrää ei ole rajattu mitenkään; galleria näyttää aina kaikki henkilöt, joilla kyseinen Position on. Henkilön nimi ja profiilikuva toimivat linkkinä hänen julkiseen profiiliinsa, joka esitellään tarkemmin luvussa 3.7.

3.6.6 Uutiset

Articles-komponentti ja samaan ominaisuuteen liittyvä Article-kokoelma on Calendarin ja DistrictOrgsin lailla yhtä sivua varten erikseen toteutettu. Articlella on nimi, päiväys, richtext-muotoinen sisältö ja slug. Articlen sisältö renderöidään omassa React-komponentissaan, joka luvussa 3.3 kuvaillusti on reititetty omaan Articlen slugista riippuvaiseen URL-osoitteeseensa. Sisältö, otsikko ja päivämäärä haetaan tietokannasta GraphQL-kyselyllä, kun URL-osoitteeseen navigoidaan. Sisältö on richtext-muodossa ja renderöidään React Markdownilla.

Articles-komponentti näyttää listan kaikista uutisartikkeleista päivämäärän mukaan järjestettynä ja se renderöidään normaalisti osana mitä tahansa sivua, jolle se on lisätty. Kaikki artikkelit haetaan GraphQL-kyselyllä, kun sivu, jolla Articles-komponentti on, ladataan. Järjestäminen päivämäärän perusteella tapahtuu jo GraphQL-kyselyssä. Jokaisesta artikkelista renderöidään luetteloon linkki,

jonka tekstinä on päivämäärä esitettyä fi-FI-lokaalin mukaisesti numeerisessa muodossa ja artikkelin otsikko.

3.7 Profiili

3.7.1 Julkinen profiilisivu

Jokaisella puolueen avainhenkilöllä, jolla on verkkosivuilla käyttäjätunnus, on julkinen profiili, jota he pystyvät muokkaamaan ollessaan kirjautuneena sisään. Sisällönhallintajärjestelmässä profiili on toteutettu Strapin asennuksen yhteydessä automaattisesti asennetulla Users & Permissions -laajennuksella ja sen luomalla User-kokoelmalla. Sen lisäksi, että käyttäjä itse pystyy muokkaamaan tietojaan verkkosivujen käyttöliittymäsovellukseen toteutetun sisäänkirjautumisen kautta, sivuston ylläpitäjät pystyvät muokkaamaan kenen tahansa käyttäjän tietoja Strapin ylläpito-paneelista. Ylläpito-paneelin kautta onnistuu myös käyttäjän poistaminen tai uuden käyttäjän luominen; käyttäjät eivät itse pysty rekisteröitymään tai poistamaan tunnuksiaan.

Käyttöliittymäsovelluksessa profiilisivu on toteutettu omassa React-komponentissaan, joka on reititetty käyttäjän slugista riippuvaan URL-osoitteeseen, kuten on tarkemmin kuvailtu luvussa 3.3. Kun profiilisivu ladataan, käyttöliittymäsovellus hakee GraphQL-kyselyllä tietokannasta slugin perusteella käyttäjän nimen, lyhyen kuvauksen, richtext-muotoisen pidemmän kuvauksen, kotikunnan, julkisen sähköpostiosoitteen, profiilikuvan URL-osoitteen, mahdollisen vaaliehdokkuuden, ja lukuisia mahdollisia yhteystietoja kuten puhelinnumeron, kotisivun ja eri sosiaalisen median kanavia. Tieto vaaliehdokkuudesta renderöidään sivulle, jos vaaliehdokkuus löytyy. Samoin jokainen yhteystieto renderöidään vain siinä tapauksessa, että kenttiin on kirjoitettu jotain. Kuvan tilalla näytetään oletusprofiilikuva, jos sen URL palauttaa jonkin virheen. Jos URL-osoitteen muokasta käyttäjää ei löydy tietokannasta, renderöidään 404-virheilmoitus. Kuvassa 5 on esimerkki profiilisivusta.

Maija Mäkelä

Ehdokkaana: Testivaalit 2023

Vaalipiiri: Pirkanmaa

Äänestä minua numerolla 1337!

Paikkakunta: Tampere

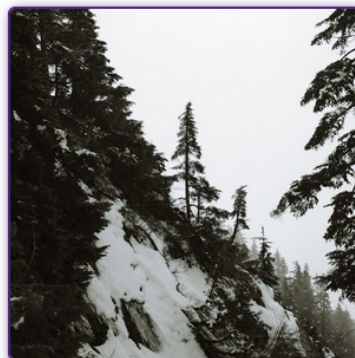
Kehitysjohdaja, kirjailija

Twitter

Facebook

TikTok

maija.makela@piraattipuolue.fi



Kuvaus

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce ultrices dolor sit amet est dapibus, id convallis justo porta. Nullam euismod leo sit amet est convallis, eu fringilla libero varius. Phasellus ullamcorper eros a sapien lobortis, vitae ultricies nisi laoreet. Mauris rutrum convallis magna, at lobortis velit mattis eget. Integer eleifend sodales mi, eget luctus metus consectetur a. Duis auctor tellus et nisi fermentum aliquam. Aliquam

- Kuva 5. Käyttäjän julkinen profiili renderöitynä käyttöliittymäsovelluksessa. Henkilö on fiktiivinen ja hänen nimensä, tittelinsä ja kotikuntansa on generoitu ChatGPT 3.5 -tekoälyllä (ChatGPT 2024). Kuva on Luke Millerin ottama ja vapaasti käytettävä (Miller 2024d).

XSS-hyökkäysten ehkäisemiseksi yhteystietojen linkkejä ei renderöidä, jos URL-osoitteet alkavat merkkijonolla "javascript:". Richtext-muotoisen pidemmän kuvauksen renderöinti on toteutettu React Markdownilla, joka huolehtii linkkien sanitoinnista (Hovlandsdal 2023).

3.7.2 Kirjautuminen

Käyttäjän autentikointi tapahtuu JWT:llä. Kun kirjautumissivu `"/users/login"` ladataan, käyttäjä siirretään profiilin muokkaussivulle `"/users/editprofile"` mikäli hän on jo kirjautunut sisään. Muussa tapauksessa renderöidään sisäänkirjautumislomake (kuva 6) ja kehoitus käyttää incognito-tilaa tietoturvasyistä.

Kirjaudu:

Kuva 6. Sisäänkirjautumislomake renderöitynä käyttöliittymäsovelluksessa.

Kun käyttäjä syöttää tiedot ja klikkaa "Kirjaudu", React-sovellus käyttää axios-pakettia lähettääkseen POST-pyyynnön backendille, joka palauttaa JWT:n, jos käyttäjän autentikointi onnistuu. Tässä tapauksessa JWT tallennetaan evästeeseen ja siirrytään osoitteeseen `"/users/editprofile"`, muussa tapauksessa käyttäjälle näytetään virheilmoitus. Eväste vanhenee kolmen tunnin kuluttua sisäänkirjautumisesta. Uloskirjautumisen käsittelee `"/users/logout"`-osoitteeseen reititetty React-komponentti, joka latautuessaan poistaa evästeen ja navigoi osoitteeseen `"/users/login"`. Käytin React-sovelluksessa React-Toastify- ja Reactstrap-paketteja kirjautumissivun ja profiilin muokkaussivun käyttöliittymän toteutukseen.

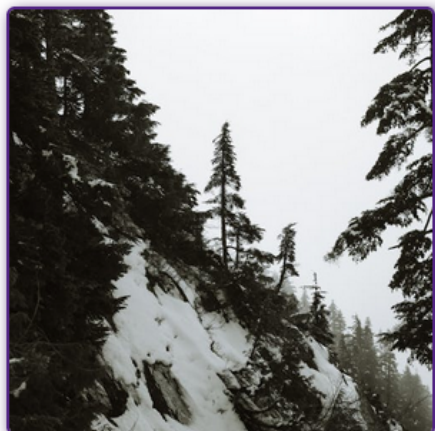
3.7.3 Profiilin muokkaus

Sisäänkirjautuneena käyttäjä voi muokata profiilinsa tietoja. Kuvassa 7 on esimerkki siitä, miltä profiilin muokkaussivu näyttää sisäänkirjautuneelle käyttäjälle.

Tervetuloa, Maija Mäkelä (maija.makela)!

[Kirjaudu ulos](#)

Täällä voit muokata profiiliäsi. Kaikki tiedot päivittyvät [julkiseen profiiliisi](#) heti, kun klikkaat "Päivitä".



[Vaihda kuva](#)

[Poista kuva](#)

Yhteystiedot

Pakolliset kentät on merkitty tähdellä.

Etu- ja sukunimi (*):

[Päivitä](#)

Julkinen e-mail:

[Päivitä](#)

Julkinen puhelin:

[Päivitä](#)

Kotisivu:

[Päivitä](#)

Kuva 7. Näkymä sisäänkirjautuneena käyttäjänä. Henkilö on fiktiivinen ja hänen nimensä on generoitu ChatGPT 3.5 -tekoälyllä (ChatGPT 2024). Kuva on Luke Millerin ottama ja vapaasti käytettävä (Miller 2024d).

Kun sivu ladataan, haetaan tietokannasta sen käyttäjän tiedot, jotka vastaavat evästeeseen tallennettua JWT:tä. Jos tämä epäonnistuu esimerkiksi koska evästettä ei ole, se on vanhentunut tai siihen tallennettu JWT ei validisti vastaa mitään käyttäjää, navigoidaan osoitteeseen `"/users/logout"`.

Sivulla on input-kenttä jokaista User-kokoelman muokattavaksi tarkoitettua kenttää kohden, lukuunottamatta avatarUrl- ja introduction-kenttiä, joiden muokkaaminen on toteutettu eri tavalla. Sivun latautuessa input-kenttiin esitätetään käyttäjän nykyiset tiedot. Kun käyttäjä klikkaa kentän vieressä olevaa ”Päivitä”-painiketta, tehdään PUT-pyyntö backendille, joka autentikoidaan evästeeseen tallennetulla JWT:llä. Jos autentikointi onnistuu, käyttäjälle näytetään ilmoitus onnistuneesta päivityksestä ja päivitetty tieto näkyy heti käyttäjän julkisessa profiilissa. Muussa tapauksessa näytetään virheilmoitus ja navigoidaan ”/users/logout”-osoitteeseen. Käyttäjä voi poistaa itseltään tietoja päivittämällä tyhjän kentän; tällöin esimerkiksi sosiaalisen median linkki poistuu käyttäjän julkisesta profiilista.

Kun käyttäjä klikkaa ”Vaihda kuva”, renderöidään modaali, josta käyttäjä voi vaihtaa profiilikuvansa. Modaali on havainnollistettu kuvassa 8.



Kuva 8. Modaali profiilikuvan vaihtamiseen, kun uusi kuva (Miller 2024a) on valittu, mutta ei vielä lähetetty.

Modaalissa on esikatselu, joka esittelee käyttäjälle, miltä hänen valitsemansa kuva näyttäisi henkilögalleriassa ennen kuin hän lähettää kuvan. Jos kuvan kuvasuhde poikkeaa neliöstä enemmän kuin 10%, käyttäjälle näytetään ilmoitus, joka kehottaa huomioimaan, että valitusta kuvasta rajautuu alueita pois. Jos käyttäjän valitsema tiedosto on kooltaan yli 1 MiB, muussa kuin image/png- tai image/jpeg-muodossa tai resoluutioltaan pienempi kuin 300*300 pikseliä, käyttäjälle näytetään virheilmoitus ja valinta tyhjennetään.

Kun käyttäjä painaa "Lähetä", kuva lähetetään autentikoidulla POST-pyyntöillä Strapi-sovelluksen mediakirjaston API:lle. Strapi-sovellus optimoi kuvan tiedostokoon, tallentaa kuvan ja palauttaa sen julkisen URL-osoitteen. Tämän jälkeen käyttäjän avatarUrl päivitetään saadulla URL-osoitteella samaan tapaan kuin muutkin kentät ja modaali sulkeutuu.

"Poista kuva"-painike profiilin muokkaussivulla avaa yksinkertaisen modaalin, jossa käyttäjää pyydetään vahvistamaan, että hän varmasti haluaa poistaa kuvan. Tässä tapauksessa käyttäjän avatarUrliksi päivitetään null, jolloin hänen profiilikuvanaan näytetään kaikissa yhteyksissä oletusprofiilikuva.

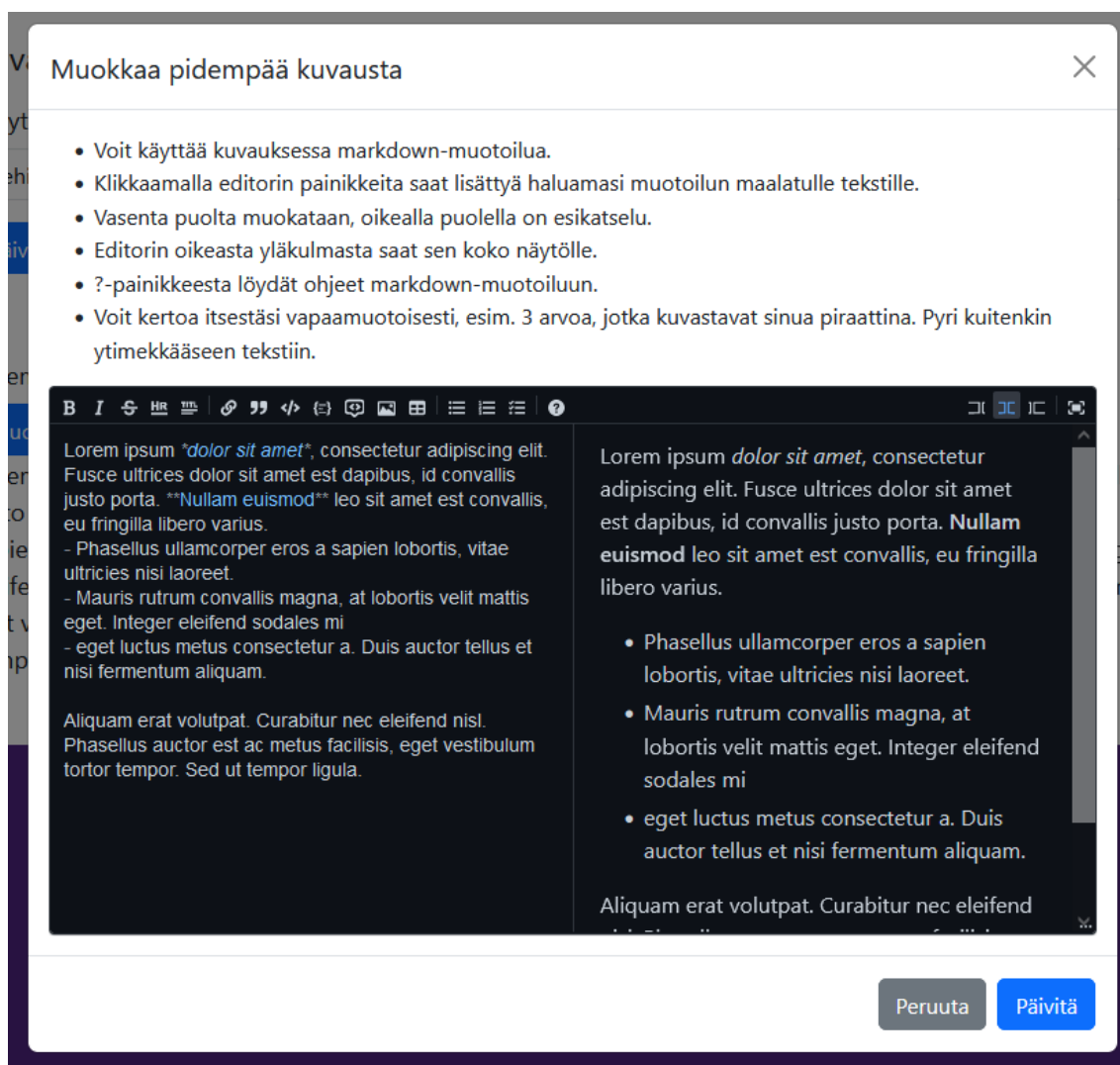
Käyttäjän introduction-kentän eli pidemmän rich text -muotoisen kuvauksen nykyinen sisältö renderöidään React Markdownilla profiilin muokkaussivulle (kuva 9). Käyttäjä voi avata modaalin muokataksaan kuvaustaan (kuva 10).

Pidempi kuvaus:

Muokkaa

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce ultrices dolor sit amet est dapibus, id convallis justo porta. Nullam euismod leo sit amet est convallis, eu fringilla libero varius. Phasellus ullamcorper eros a sapien lobortis, vitae ultricies nisi laoreet. Mauris rutrum convallis magna, at lobortis velit mattis eget. Integer eleifend sodales mi, eget luctus metus consectetur a. Duis auctor tellus et nisi fermentum aliquam. Aliquam erat volutpat. Curabitur nec eleifend nisl. Phasellus auctor est ac metus facilisis, eget vestibulum tortor tempor. Sed ut tempor ligula.

Kuva 9. Pidempi kuvaus renderöitynä profiilin muokkaussivulla.



Kuva 10. Modaali pidemmän kuvauksen muokkaamiseen.

Käytin pakettia `@uiw/react-md-editor` Markdown-editorina modaalissa. Päivitä-painike toimii tässä samalla tavalla kuin muutkin Päivitä-painikkeet profiilin muokkaussivulla ja se samalla sulkee modaalin.

3.8 Sovelluspalvelimen asennus ja konfigurointi

3.8.1 Pilvipalvelin

Palvelin on Ubuntu Server 22.04 -käyttöjärjestelmällä varustettu pilvipalvelin, jonne minulla on root-oikeudet. Palvelimella on Node.js v17.9.0, git, nginx, certbot, python3 ja pm2.

Kehitysvaiheessa olen käyttänyt palvelimena Linoden Nanode 1 GB -palvelinta, koska se on edullinen ja riittää resursseiltaan tähän tarkoitukseen. Projektin valmistuttua sivusto tullaan hostaamaan piraattipuolueen palvelimelle, joka tulee olemaan ohjelmistoiltaan samanlainen.

3.8.2 Nginxin konfigurointi ja SSL-sertifikaattien luonti

Nginxin sites-available- ja sites-enabled-kansioissa on tiedostot strapi.awaclus.eu-osoitetta ja frontend.awaclus.eu-osoitetta varten. Strapi.awaclus.eu:n tiedoston location / -blokissa on määritelty parametrit proxy_pass http://localhost:1337 ja include proxy_params eli kaikki osoitteeseen strapi.awaclus.eu/ tuleva liikenne reititetään palvelimen localhostin 1337-portille, jota Strapi-sovellus käyttää, ja pyyntöjen headereihin sisällytetään proxy_params-tiedostossa määritellyt parametrit. En ole tehnyt proxy_params-tiedostoon muutoksia. Frontend.awaclus.eu:n tiedosto käyttää nginxin virtuaalisen hostin oletuskonfiguraatiota, josta juurihakemiston sijainniksi on vaihdettu /var/www/frontend_awaclus/html ja server_nameksi frontend.awaclus.eu. Muilta osin nginx käyttää oletusasetuksia ja certbotin automaattisesti tekemiä muutoksia asetuksiin.

Toteutin SSL-sertifikaattien luonnin certbotilla seuraamalla Electronic Frontier Foundationin certbot-ohjeita nginxille Ubuntulla. Tähän sisältyi snapd:n asennus, certbotin poisto apt-getillä, certbotin asennus, certbot-komennon valmistelu, certbot-komennon ajaminen ja sertifikaattien lisääminen molemmille

aladomaineille sekä sertifikaattien automaattisen uusimisen testaaminen. (Electronic Frontier Foundation 2024.)

3.8.3 Sovellusten asennus

React-sovelluksen hostaaminen nginxillä on erittäin yksinkertaista. Toteutin tämän ajamalla paikallisella tietokoneellani React-projektin kansiossa `npm run build` ja siirtämällä sen luoman build-kansion sisällön pilvipalvelimen `/var/www/frontend_awaclus/html`-kansioon.

Toteutin Strapi-sovelluksen hostaamisen kloonaamalla git-repositorion piraattipuolueen Giteasta pilvipalvelimelle, rakentamalla Strapi-sovelluksen admin-paneelin production-ympäristöä varten ja käyttämällä pm2:a prosessin hallintaan, jotta Strapi-sovellus pysyisi käynnissä. Pm2:a varten tarvitsin `server.js`-tiedoston, jonka loin Strapin dokumentaation ohjeiden mukaisesti (Strapi 2024e). Tietokannan sisältö ei tällä menetelmällä siirry paikallisesta kehitysversiosta palvelimelle hostattuun versioon, joten käytin sen siirtämiseen Strapin transfer-ominaisuutta (Strapi 2024f).

4 Tulokset ja pohdinta

Kehitystyön tuloksena syntyi teknisesti toimiva ja helposti muokattavissa oleva verkkosivu, joka täyttää projektin vaatimukset teknisen toteutuksen osalta. Sivustolla on kaikki ominaisuudet, mitkä olivat puolueen vanhallakin sivustolla. Toteutuksessa ei käytetty WordPressiä, sisältö on helposti muokattavissa ilman erityistä tietoteknistä osaamista ja avainhenkilöiden tietojen päivittäminen onnistuu tietokannan kautta siten, että henkilöt voivat itse tehdä sen kirjautumalla sisään käyttäjätunnuksillaan. Kaikki kehitystyössä käytetyt teknologiat koodieditoria lukuunottamatta olivat vapaita avoimen lähdekoodin teknologioita.

Projektin aikataulu venyi reilusti alkuperäiseen suunnitelmaan verrattuna. Puolue oli halunnut ottaa uudet verkkosivut käyttöön ennen vuoden 2023 eduskuntavaaleja, mutta teknisen toteutuksen osalta kehitystyö valmistui vasta joulukuussa 2023. Joiltakin muilta osin, kuten käyttäjäkokemuksen suunnittelun ja käyttäjätestauksen osalta kehitystyö on tätä kirjoittaessa edelleen keskeneräinen, ja näiden edetessä myös tekniseen toteutukseen saatetaan tarvita myöhemmin muutoksia tai lisäyksiä. Näistä syistä uudistetut verkkosivut eivät ole vielä otettavissa puolueen käyttöön. Tarve uudistetuille verkkosivuilla on kuitenkin ajankohtainen myös tulevaisuudessa, kun projekti valmistuu.

Projektin toteuttaminen laajensi osaamistani webkehityksestä ja verkkopalveluiden hostaamisesta huomattavasti. Moni ominaisuus, toteutustapa ja teknologia tuli minulle tutuksi ensimmäistä kertaa tämän projektin kautta. Kaikissa aikaisemmissa webkehitysprojekteissani painopiste on ollut hyvin vahvasti frontendissä ja erillistä backendiä ei usein ole ollut ollenkaan, joten opin paljon sekä backendin toteuttamisesta että frontendin ja backendin välisestä kommunikaatiosta. Olen aikaisemminkin hostannut yksinkertaisia verkkosivuja, mutta tällaisten useammasta sovelluksesta koostuvien isompien kokonaisuuksien hostaaminen oli minulle uutta. Opin uutta myös tietoturvasta, koska käyttäjien autentikoinnin ja käyttäjien luoman sisällön julkaisemisen takia jouduin kiinnittämään siihen erityistä huomiota. Olen myös jo ehtinyt konkreettisesti hyödyntämään projektin yhteydessä hankkimaani uutta osaamista henkilökohtaisissa projekteissani.

Projektin toteutuksen aikana tietokoneeni SSD-levy hajosi sillä seurauksella, että kone ei käynnistynyt ja ongelman selvittämisessä ja korjaamisessa kesti melkein kuukausi. Tänä aikana kehitystyö ei edennyt. Tämä tapahtui vasta eduskuntavaalien jälkeen, joten se ei sinänsä ollut pääsyy projektin viivästymiselle alkuperäisestä aikataulusta, mutta se oli eräs havainnollistava kokemus siitä, miksi aikatauluihin kannattaa suunnitella tilaa ennalta-arvaamattomille haasteille ja miksi tiedostojen varmuuskopiointi on tärkeää. Piraattipuolueen Gitteassa sijainneen koodirepositorion ansiosta vahingot jäivät viivästyksen lisäksi minimaalisiksi.

Lähteet

- Airfocus. 2024. Back End (In a Website). <https://airfocus.com/glossary/what-is-a-back-end/>. 10.3.2024.
- Bauer, Y., Leal, J. P. & Queirós, R. 2023. Can a Content Management System Provide a Good User Experience to Teachers?. 4th International Computer Programming Education Conference (ICPEC 2023), 4:1–4:8.
- ChatGPT 3.5. 2024. Keksi neljä fiktiivistä suomalaista henkilöä, joilla on etu- ja sukunimi, kotikunta, ja 1 tai 2 ammattia tai titteliä. Open AI. <https://chat.openai.com>. 4.4.2024.
- Cloudflare. 2024. What is a reverse proxy?. <https://www.cloudflare.com/learning/cdn/glossary/reverse-proxy/>. 15.4.2024.
- Daring Fireball. 2024. Markdown. <https://daringfireball.net/projects/markdown/>. 2.5.2024.
- Electronic Frontier Foundation. 2024. Certbot instructions. <https://certbot.eff.org/instructions?ws=nginx&os=ubuntufocal>. 8.4.2024.
- GitHub. 2017. Open Source Survey. <https://opensourcesurvey.org/2017/>. 27.11.2022.
- Google. 2024. Add a Google calendar to your website. <https://support.google.com/calendar/answer/41207>. 4.4.2024.
- GraphQL. 2023. Introduction to GraphQL. <https://graphql.org/learn/>. 8.11.2023.
- Hovlandsdal, E. 2023. React-markdown. <https://github.com/remarkjs/react-markdown>. 15.4.2024.
- Japikse, P., Grossnicklaus, K. & Dewey, B. 2020. Building Web Applications with .NET Core 2.1 and JavaScript. 2. painos. Berkeley: Apress.
- Lardinois, F. 2015. Microsoft Launches Visual Studio Code, A Free Cross-Platform Code Editor For OS X, Linux and Windows. TechCrunch. <https://techcrunch.com/2015/04/29/microsoft-shocks-the-world-with-visual-studio-code-a-free-code-editor-for-os-x-linux-and-windows/>. 9.11.2023.
- Liu, G., Xu, J., Wang, C. & Zhang, J. 2018. A Performance Comparison of HTTP Servers in a 10G/40G Network. ICBDC '18: Proceedings of the 3rd International Conference on Big Data and Computing. Huhtikuu 2018, 115–118.
- Markdown Land. 2021. Markdown Center a Text, Image, Title, or Table. <https://markdown.land/markdown-center>. 2.4.2024.
- Meta Platforms. 2022. Design Principles. <https://reactjs.org/docs/design-principles.html>. 27.11.2022.
- Microsoft. 2021. Differences between the repository and Visual Studio Code. <https://github.com/microsoft/vscode/wiki/Differences-between-the-repository-and-Visual-Studio-Code>. 9.11.2023.
- Miller, L. 2024a. A sunset over the ocean with a beach and a tree. 16.3.2024. Pexels. <https://www.pexels.com/photo/a-sunset-over-the-ocean-with-a-beach-and-a-tree-20654356/>. 4.4.2024.

- Miller, L. 2024b. Footpath among Trees in Forest. 24.2.2024. Pexels. <https://www.pexels.com/photo/footpath-among-trees-in-forest-20386778/>. 4.4.2024.
- Miller, L. 2024c. A view of the city from a hilltop. 24.2.2024. Pexels. <https://www.pexels.com/photo/a-view-of-the-city-from-a-hilltop-20386777/>. 4.4.2024.
- Miller, L. 2024d. A black and white photo of a snow covered mountain. 24.2.2024. Pexels. <https://www.pexels.com/photo/cold-snow-wood-landscape-20386774/>. 4.4.2024.
- Nguyen, A. 2022. Building an e-commerce website using Next JS, Mantine, and Strapi. Centria-ammattikorkeakoulu. Degree Programme in Information Technology. Opinnäytetyö. <https://urn.fi/URN:NBN:fi:amk-2022053013113>.
- Oggier, C. 2020. How fast GraphQL is compared to REST APIs. Haaga-Helia ammattikorkeakoulu. Degree Programme in Business Information Technology. Opinnäytetyö. <https://urn.fi/URN:NBN:fi:amk-2020052714286>.
- OWASP. 2024. Cross Site Scripting (XSS). <https://owasp.org/www-community/attacks/xss/>. 7.4.2024.
- Pinnis, N. 2022. Modern website development with Strapi and Next.js. Lapin ammattikorkeakoulu. Tieto- ja viestintätekniikan koulutusohjelma. Opinnäytetyö. <https://urn.fi/URN:NBN:fi:amk-2022053013287>.
- Piraattipuolue. 2021. Piraattipuolueen puolueohjelma. <https://www.piraattipuolue.fi/politiikka/puolueohjelma/>. 27.11.2022.
- Seobility. 2024. Rendering. <https://www.seobility.net/en/wiki/Rendering>. 15.4.2024.
- Sobri, N. A. N., Abas, A. H., Yassin, A. I. M., Ali, S. A. M., Tahir, N. M., Zabidi, A. & Rizman, Z. I. 2022. Comparison between Headless CMS and Backend-as-a-Service Products for E-Suripreneur Backend. Mathematical Statistician and Engineering Applications. Vol. 71 No. 3s2 (2022), 928–938.
- Statista. 2022. Most used web frameworks among developers worldwide, as of 2022. <https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/>. 27.11.2022.
- Strapi. 2023a. Welcome to the Strapi Developer Docs!. <https://docs.strapi.io/dev-docs/intro>. 8.11.2023.
- Strapi. 2023b. The Guide to Headless CMS. <https://strapi.io/headless-cms-guide>. 8.11.2023.
- Strapi. 2023c. Nginx Proxying. <https://docs.strapi.io/dev-docs/deployment/nginx-proxy>. 8.11.2023.
- Strapi. 2023d. GraphQL plugin. <https://docs.strapi.io/dev-docs/plugins/graphql>. 8.11.2023.
- Strapi. 2024a. Configuring fields for content-types. <https://docs.strapi.io/user-docs/content-type-builder/configuring-fields-content-type>. 13.4.2024.
- Strapi. 2024b. List of Strapi plugins. <https://docs.strapi.io/user-docs/plugins/strapi-plugins>. 28.3.2024.
- Strapi. 2024c. Writing content. <https://docs.strapi.io/user-docs/content-manager/writing-content>. 2.4.2024.
- Strapi. 2024d. Managing relational fields. <https://docs.strapi.io/user-docs/content-manager/managing-relational-fields>. 4.4.2024.

- Strapi. 2024e. Process manager. <https://docs.strapi.io/dev-docs/deployment/process-manager>. 8.4.2024.
- Strapi. 2024f. Data transfer. <https://docs.strapi.io/dev-docs/data-management/transfer>. 8.4.2024.
- Tanner, M. 2020. Implementation of the new Swiss-Hema website using headless CMS and React.js. Haaga-Helia ammattikorkeakoulu. Degree Programme in Business Information Technology. Opinnäytetyö. <https://urn.fi/URN:NBN:fi:amk-2020052614026>.
- Thakkar, M. 2020. Building React Apps with Server-Side Rendering. Vadodara: Apress.
- Ubuntu. 2023. Ubuntu Server documentation. <https://ubuntu.com/server/docs>. 9.11.2023.
- Vercel. 2023. Next.js documentation. <https://nextjs.org/docs>. 7.11.2023.