

Roni Juntila

WordPress ajanvarauskalenteri

WordPress ajanvarauskalenteri

Roni Junttila
Opinnäytetyön suunnitelma
Syksy 2023
Tietojenkäsittely
Oulun ammattikorkeakoulu

SISÄLLYS

1	JOHDANTO	6
2	TYÖHÖN LIITTYVÄT KÄSITTEET	7
2.1	Lisäosat.....	7
2.2	Saavutettavuus.....	7
2.3	WordPress.....	8
2.4	PHP.....	9
2.5	CSS.....	10
2.6	Figma	11
2.7	Responsiivisuus	12
2.8	Tietoturva	12
2.9	JavaScript.....	13
2.10	SQL.....	14
2.11	REST API.....	15
3	WORDPRESS-LISÄOSAT	17
3.1	Toteutusvaihtoehdot.....	17
3.2	Oman lisäosan suunnittelu	20
3.3	Oman lisäosan koodaaminen	21
3.4	Lisäosan tyylittely	31
3.5	Saavutettavuus.....	35
3.6	Lisäosan julkaiseminen	38
4	JOHTOPÄÄTÖKSET	40
5	POHDINTA.....	41
	LÄHTEET.....	42

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tutkinto-ohjelman nimi, Tietojenkäsittely

Tekijä(t): Roni Junttila
Opinnäytetyön nimi: WordPress Ajanvarauskalenteri
Työn ohjaaja(t): Minna Kamula
Työn valmistumislukukausi ja -vuosi: Syksy 2023 Sivumäärä: 45

2023 keväällä tuli puheeksi, että eräs sivusto tarvitsee tietyn tyyppisen ajanvarauskalenterin, jota ei löydy suoraan valmiina sopivaan hintaan ja oikean tyyppisillä ominaisuuksilla. Olin miettinyt omaa opinnäytetyötä ja sen aiheita jo jonkin aikaa, joten päätin tarttua suoraan aiheeseen. Hain aiheelle hyväksynnän ja melko nopeasti sain luvan aloittaa työn.

Tavoitteena oli tehdä yksinkertainen ja helposti käytettävä ajanvarauskalenteri. Siinä oli tarkoitus olla monta ihmistä keneltä varata aikoja eri tarkoitukseen ja tarpeisiin. Sivulle tarvittaisiin erikseen WordPressin hallintapaneelista käytettävä lisäosan hallinta, jossa voisi lisätä uusia henkilöitä ja hallita heidän tietojansa. Sen lisäksi siellä tarvittaisiin tärkein eli aikojenhallinta, josta voisi tarkistella olemassa olevia aikoja henkilöittäin, lisätä uusi aika ja poistaa olemassa oleva aika. Käyttäjälle listattaisiin työntekijät erikseen, josta käyttäjä voisi valita sopivamman. Valinnan jälkeen käyttäjä voisi siirtyä tarkastelemaan vapaana olevia aikoja ja lopuksi tehdä ajanvarauksen. Ulkonäön kuuluisi olla neutraali mutta samalla tyylikkään näköinen. Lisäksi lisäosan rakentamisen aikana tulitaisiin kiinnittämään huomiota saavutettavuuteen.

Tietoperusta käsittelee hyvin pitkälti eri ohjelmointikieliä ja niiden soveltamista. Näiden lisäksi hain materiaalia tietoturvasta ja saavutettavuudesta. Tietoa haettiin paljon erilaisilta internet sivustoilta, mutta myös osan tiedoista sain e-materiaaleista.

Loppu tuloksesta tuli tyylikkään näköinen. Käytettävyys on helppo, selkeä ja ennen kaikkea suunnitelman mukainen. Hallintapaneeliin tuli kolme sivua. Ajat sivusto, josta löytyy vapaana olevat ajat, uuden henkilön lisäämissivu, ja henkilöiden hallinta. Käyttäjän näkökulmasta sivusto tarjoilee työntekijä korttien muodossa. Jokainen kortti sisältää tärkeimmät tiedot työntekijästä ja sen perusteella käyttäjä valitsee oikean henkilön hänen tarpeisiinsa ja varaa ajan. Kaiken kaikkiaan lisäosa on suunnitelman mukainen. Lisäosaan jäi kuitenkin jatkokehittävää. Osa kehityskohteista ilmeni kesken projektin. Näistä osan tein suoraan projektin aikana, mutta valitettavasti aika ei riittänyt kaikkeen.

Projektin aikana opin paljon PHP-kielestä ja eteenkin WordPress ympäristöstä ja sen eroista. Sen lisäksi pääsin kunnolla tutkimaan saavutettavuutta, sen huomioimista nettisivun rakentamisen aikana, sekä tietoturvallisuutta ja uhkien torjumista.

Asiasanat: CSS, lisäosa, rajapinta, sisällönhallintajärjestelmä, PHP,

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Business Information Systems

Author(s): Roni Junttila
Title of thesis: WordPress booking calendar
Supervisor(s): Minna Kamula
Term and year when the thesis was submitted: Fall 2023
Number of pages: 45

The goal was to create a simple and user-friendly appointment booking calendar plugin for WordPress. The plugin should have the option for having multiple employees, their information and managing that data. Information is used to present each worker and their skills to users before they start booking an appointment from available times.

In the spring of 2023, I was talking to a mate of mine, and it came up that a certain website needed a specific type of appointment booking calendar, which couldn't be readily found at an appropriate price with the right features. I had been thinking about my thesis topic for quite some time, so I decided to dive right into it.

I completed all the features agreed upon with the client and the plugin turned out great. In the admin view, the plugin consists of three main pages. The first page is for adding new employees to the database. The second page is for reviewing and managing these existing profiles and the last page for viewing already existing appointments and adding new ones to database. In the user view, each employee is presented as a card, which contains all the necessary information about the employee along with a button. When a user clicks this button, they can view all the available appointments for that employee. After selecting a suitable time slot, a form is presented for the user to fill out with their details. Upon submission, the system will notify both the employee and the user via email, confirming the booked appointment.

The result looks stylish and is safe against cyber threats. Furthermore, accessibility was one of the most important things during the plugin's development so this plugin can reach more people!

I would add an option for administrators to change colors of the plugin from WordPress control panel as well of translating it to multiple languages. This would take the plugin to the next level as it would reach more people and their needs.

During the project, I learned a lot about coding in general and especially backend as well as cybersecurity and mitigating threats. I also learned quite a lot about WordPress environment and its differences to what I used to. Additionally, I had the opportunity to delve into accessibility. I never had any real experience dealing with it, so this was a real eye opener.

Keywords: CSS, Plugin, Interface, Content-Management-System, PHP

1 JOHDANTO

Kehittämistehtävän toimeksiantaja toimii Katariinan Järvenpää, jonka tekemälle sivustolle tarvitaan räätälöity varauskalenteri. Sivusto on tehty WordPressillä, joten kyseessä lisäosan kehittämistehtävä. Yritys, joka ottaa lisäosan ensimmäisenä käyttöön on Terapihus Resus/Terapiatalo Resurssi. Se on alhaisen kynnyksen terapiatalo, joka sijaitsee Pietarsaareissa. Yritys on perustettu vuonna 2023. Siellä työskentelee tällä hetkellä kolme vakituista terapeuttia. Kukin heistä on erikoistunut eri terapian muotoihin ja heille voi varata ajan ilman lähetettä. Palveluja tarjotaan niin suomen, kuten myös ruotsin kielellä

Opinnäytetyössä on tarkoitus kehittää helppokäyttöinen WordPress ajanvarauskalenteri, joka soveltuu myös myöhemmin muiden käyttöön. Lisäosa on tarkoitus julkaista WordPressin omassa kaupassa. Varauskalenteri näyttää tietyn ajan päähän päivät, joilla on vapaita aikoja. Käyttäjä valitsee itselleen sopivan päivän, lisäosa tarjoilee vapaana olevat ajat sille päivälle. Näistä käyttäjä valitsee sopivimman, jonka jälkeen sivusto pyytää syöttämään nimen ja sähköpostiosoitteen. Kenttien täyttämisen ja varauksen lähettämisen jälkeen käyttäjä saa itselleen sähköpostiin varausvarmistuksen, jossa lukee varauksen aika, varaajan nimi ja linkki, josta ajan voi perua. Jos linkistä painaa tämä peruu ajan ja vapauttaa sen takaisin varattavaksi. Ulkonäöllisesti lisäosasta tulisi saada virkistävän modernin näköinen, joka sulautuu hyvin ympäristöön kuin ympäristöön.

Miksi kehittämistyö tehdään?

Tarkoitukseen sopivaa lisäosaa ei löytynyt WordPressin lisäosakaupasta tai ulkopuolisilta palvelun tarjoajilta. Tavoitteena on luoda tyylikäs funktionaalinen kalenteri, jota sivuston ylläpitäjä pystyy hallitsemaan, kuten esimerkiksi lisäämään tai poistamaan aikoja oman aikataulunsa mukaisesti.

Kehittämis-/tutkimuskysymykset

Mitä saavutettavuus tarkoittaa ja miten se voidaan huomioida suunnittelun kannalta?
Onko varauskalenterista helppo varata aikoja?

2 TYÖHÖN LIITTYVÄT KÄSITTEET

2.1 Lisäosat

Lisäosa tulee englannin kielen sanasta plugin, jota voidaan myös kutsua laajennukseksi. Näiden tehtävänä on tuoda sivustolle uusia ominaisuuksia ja parantaa käytettävyyttä. (One.com 2023). Laajennukset kuten migraatiotyökalu, auttavat sivuston kehittäjää rakentamaan sivua. Laajennuksia on saatavilla valtavia määriä. WordPressin omassa lisäosahakemistossa on yli 60000 ilmaista lisäosaa (WordPress 2023c).

2.2 Saavutettavuus

Sanalla saavutettavuus tarkoitetaan ihmisten eri tarpeiden huomioon ottamista verkkosivustojen ja mobiiliapplikaatioiden rakenteen suunnittelussa. Internetin saavutettavuuden piirissä ovat kaikki esteet, jotka vaikuttavat sen käyttöön. Näitä ovat kuulo, kognitiivinen neurologinen fyysistä puhetta visuaalinen. Yrittäjille on myös vahva tarve saada mahdollisimman saavutettava sivusto, jotta kaikki asiakkaat voivat käyttää sitä. Tosin monissa tapauksissa saavutettavuutta vaaditaan lailla. Kaikista tehokkain tapa luoda saavutettava sivusto on jo suunnittelu vaiheessa ottaa mahdollisimman hyvin huomioon. (Lawton 2022.)

Helppokäyttöisyys on yksi saavutettavuuden kulmakivistä. Tällä tarkoitetaan selkeää rakennetta, kuten vaikka navigaatiovalikon yksinkertaisuutta ja nimien selkeyttä.

Ymmärrettävyydellä tarkoitetaan palvelun tekstejä. Tekstien kuuluu olla helposti ymmärrettäviä. Pitkät tekstit pitää olla jaettuna lyhyisiin kappaleisiin, luettelojen kuuluu olla jäsennelty selkeästi ja linkkiteksteistä pitää käydä selväksi mitä siitä tapahtuu.

Tekninen saavutettavuus toteutuu, kun HTML-standardia ja WCAG-ohjeistusta noudatetaan, jotta palvelu toimii erilaisilla laitteilla. Tähän myös kuuluu avustavien teknologien, kuten puheohjauksen ja ruudunlukuohjaimen huomioon ottamista.

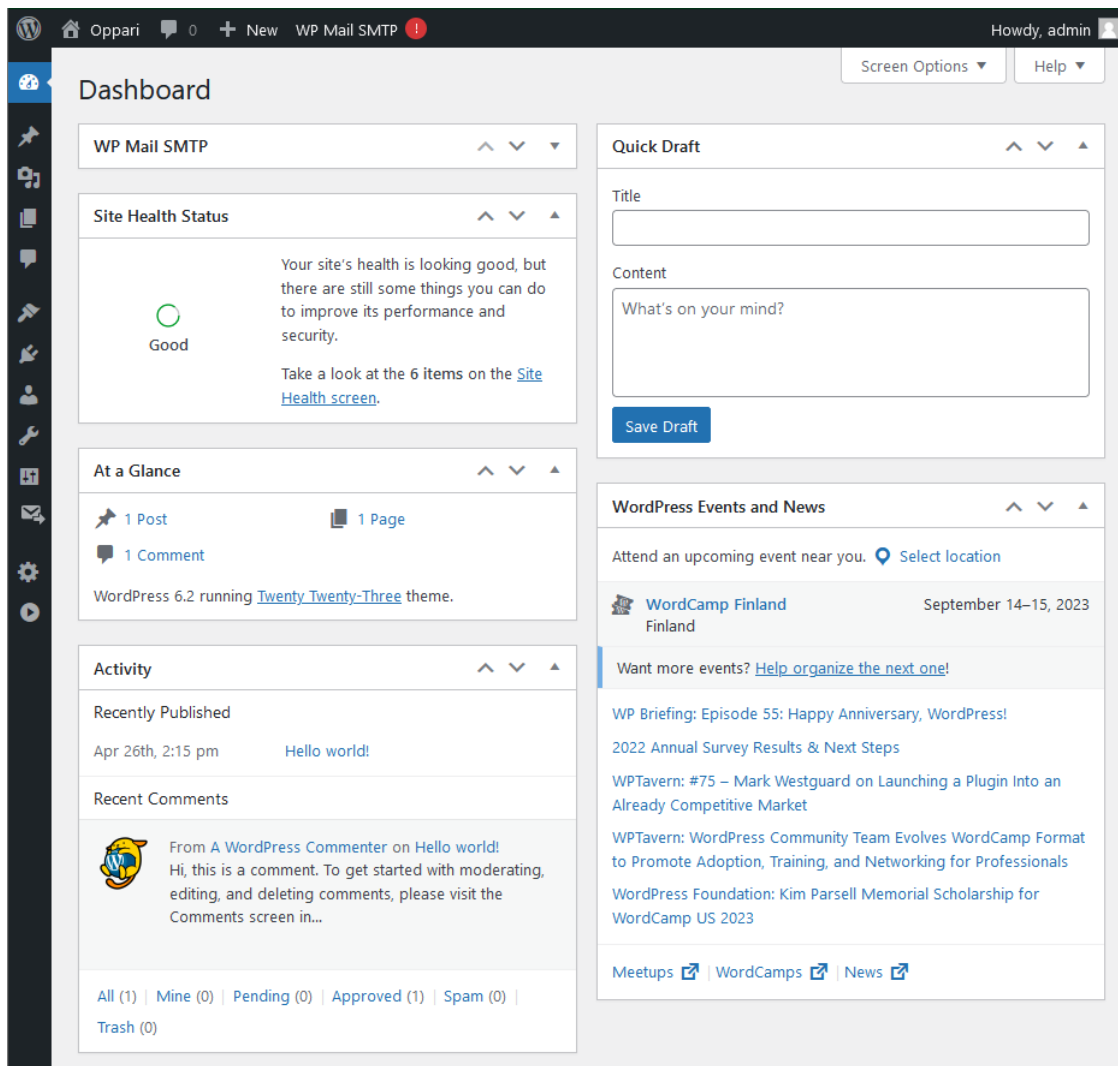
(Aluehallintovirasto 2023.)

Koko saavutettavuudesta keskustelun ja sen luomisen on aloittanut WAI. Se tulee sanoista Web Accessibility Initiative, joka perustettiin vuonna 1996. Syinä sen syntyyn olivat standardien ja ohjeiden puuttuminen, jonka vuoksi nettisivustot eivät olleet kaikkein saavutettavissa. Se myös tukee sivustojen tekijöitä, jotta yleiset hyvät käytänteet toteutuisivat. Vuonna 1997 ensimmäinen virallinen julkinen lista julkaistiin kaikkien luettavaksi. (Dardailler 2009.)

Edellä mainitut WAI:n ohjeistukset ovat vaikuttaneet niin, että EU on myös valveutunut asian suhteen. EU:n oma direktiivi verkkosivujen ja mobiili applikaatioidensaavutettavuudesta julkiselle sektorille on julkaistu vuonna 2016. Sen tarkoituksena oli yhtenäistää maiden välisiä eroja saavutettavuudessa, kuten myös parantaa yleistä saavutettavuuden tasoa unionin sisällä. Yhtenäistämisen vähentäisi myös kehittäjien epävarmuutta ja edistää yhteen toimivuutta. Yleinen käytäntö ennen direktiivin voimaan astumista oli käyttää kansainvälisiä standardeja. Tämä kuitenkin johti siihen, että eri maat käyttivät eri versioita ohjeistuksesta, joka johti myös vaatimustasoeroihin ja erilaisiin toteutustekniikoihin. (Directive (EU) 2016.)

2.3 WordPress

WordPress on avoimeen lähdekoodiin perustuva ilmainen julkaisujärjestelmä, jossa kuka tahansa voi luoda oman sivustonsa (Kuva 1). Avoin lähdekoodi mahdollistaa sen tuntuvan muokkaamisen ja kehittämisen tarpeiden mukaiseen käyttöön. (Huttunen 2021.) Sisällönhallintajärjestelmillä toimivista sivuista WordPressin markkinaosuus on yli 63 % vuonna 2023 (W3techs 2023).



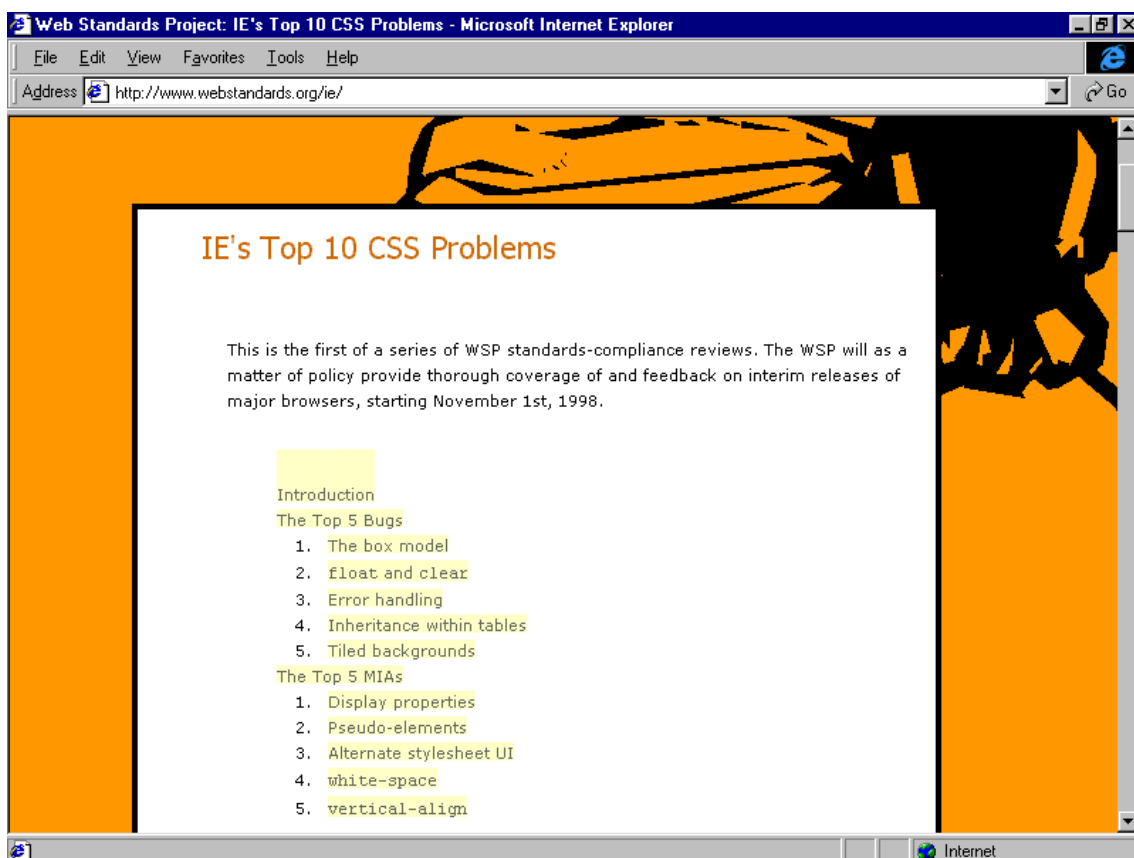
Kuva 1 WordPress hallintapaneeli (Omakuva 2023)

2.4 PHP

PHP on yleistarkoitukseen sopiva avoimenlähdekoodin kieli, joka tulee sanoista Hypertext Preprocessor. Kieli on pääasiassa tarkoitettu palvelin puolen koodaamiseen, mutta sitä voidaan käyttää myös rajoitetusti työpöytäsovelluksien tekemisessä. Siihen löytyy monia ilmaisia tulkkeja. PHP toiminta perustuu siihen, kun käyttäjä pyytää verkkosivua palvelimelta, sivu tulketaan ja suoritetaan PHP-skriptissä määritellyt toiminnot. Tämän seurauksena generoituu Hypertext Markup Language (HTML) -koodi. HTML-koodi lähetetään sitten selainohjelmalle yhdessä muiden sivun HTML-elementtien kanssa, mikä mahdollistaa sisällön saumattoman näyttämisen. Verkkosivuja, jotka sisältävät PHP-skriptiä, kutsutaan dynaamisiksi HTML-sivuiksi, koska niiden sisältö vaihtelee skriptin tulkinnan tulosten perusteella. Useimmat käyttöjärjestelmät tukevat PHP kieltä ja myös yleisimmin käytössä olevat tietokannat tukevat PHP-kieltä, kuten MySQL ja MongoDB. (Sheldon 2023.)

2.5 CSS

CSS on lyhennetty sanoista Cascading Style Sheets. Se on julkaistu vuonna 1996. Sama vuonna Internet Explorer 3 oli ensimmäinen kaupallinen selain, joka tuki CSS-tyylejä (Kuva 2). Sillä toimitettiin suurin osa väreistä, sivustojen taustat, erilaiset fontit ja tekstiominaisuudet.



Kuva 2 Internet Explorer vuonna 1998 (Bos 2016)

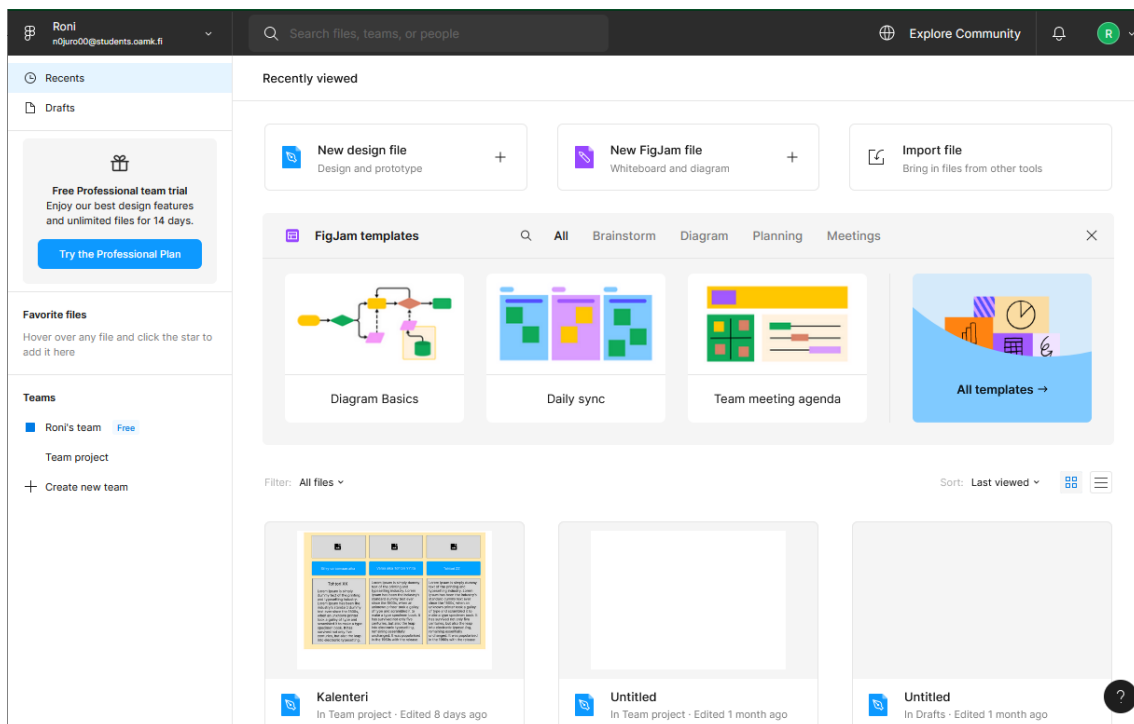
Ennen CSS-kielen julkaisua ja jokin aika sen rinnalle jokaisella selaimella oli omat tyylimäärittelynsä. Se toi omat rajoituksensa, sekä ongelmia samaa sivustoa katsoessa eri selaimilla. Kuitenkin jo kielen julkaisuvuotena se sai W3C suosituksen. Tämän ideana oli parantaa yhteensopivuutta selainten välillä. Sivustoja, jotka käyttävät jotain muuta kieltä tyyllittelyissä on hankala arvioida, mutta se liikkuu jossain alle 2 prosentin luokassa. (Bos 2016).

CSS on tyylien tekemiseen luotu kieli. CSS:ä käytetään kuvaamaan lähdedokumentin esitystapaa, eikä se yleensä muuta dokumentin kielen ilmaisemaa taustalla olevaa semantiikkaa. CSS tekemiseen ei tarvitse kuin tekstin käsittely ohjelman! (W3c 2023.)

CSS-kielen käyttämisen etuja ovat sivuston nopeampi latautuminen. Se parantaa myös yleistä käyttökokemusta, koska se parantaa yleistä luettavuutta, kuten myös käytettävyyttä ja tekee sivustosta miellyttävämmän silmälle. Kielen suhteellisen helppo syntaksi myös nopeuttaa sivuston kehittämisaikaa. Yhdellä rivillä saa jo paljon aikaan ja tagien uudelleen käyttäminen helpottaa huomattavasti muotoilua ja vähentää työmäärää. Myös myöhemmin sivuston muuttaminen on huomattavasti nopeampaa. (Devmountain 2023.)

2.6 Figma

Figma on käyttöliittymän suunnitteluun tarkoitettu ohjelma, joka ei vaadi käyttäjältä koodaamista. Sillä rakentaminen perustuu vedä ja pudota tyyppiseen ratkaisuun, joka on käyttäjältä suhteellisen helppo opetella. Figma soveltuu loistavasti nettisivustojen tai vaikkapa mobiiliapplikaatio prototyypin suunnitteluun. (Piispanen 2023.)



Kuva 3 Figman hallintaneeli (Omakuva 2023)

Figma on pilvestä toimiva, joten palvelu omaa projektia voi muokata mistä vain missä on verkko-yhteys ja samaa sivua voi muokata useampi käyttäjä yhtä aikaa (Kuva 3). Se tarjoaa myös mahdollisuuden luoda interaktiivisen käyttöliittymän. Nämä kaikki tekevät siitä erittäin joustavan ja

houkuttelevan monille yrityksille. Mikäli Figman käyttö onnistuu, mikäli sujuvasti se, aukaisee mahdollisuuksia erilaisille urapoluille, koska se on käytössä niin laajasti (Noble Desktop 2023.)

2.7 Responsiivisuus

Responsiivisuus tulee englanninkielisestä sanasta responsive. Se parantaa käyttäjän käyttökoke-
musta ja vaikuttaa mm. hakukoneoptimointiin positiivisesti. Lähes jokainen sivusto haluaa toimia
matkapuhelimilla, koska mobiililaitteita käytetään enemmän ja enemmän. (Suovesi 2022.)

Tutkimuksen mukaan vuonna 2022 matkapuhelimet olivat noin 51 % kaikista verkkosivustojen lii-
kenteestä. Vastaavasti pöytäkoneiden osuus oli noin 48 % ja tabletit vain hieman alle 2 %.
(Statcounter 2023.)

Suunnittelussa täytyy ottaa huomioon käyttäjän näyttö koon, laite tyyppin ja näytön orientaation mu-
kaan, kuten myös käyttäjän alustan mukaan. Käytännössä tämä tarkoittaa sisällön paketoitua jon-
kinlaisen joustavan ruudukon sisälle, joka skaalaa tekstit oikein. Sen lisäksi täytyy ottaa huomioon
kuvien koot. Hyvin perinteinen tapa on käyttää CSS media kyselyjen tekemisen aikana, jolla voi-
daan määritellä erilaisia ruutukokoja, mutta myös muita tapoja löytyy riippuen alustasta millä sivus-
toa tehdään. Täytyy myös ottaa huomioon, miten navigointi toimii erilaisilla laitteilla. Yleensä käy-
tössä on hampurilaismenu käytössä ja isommalla ruuduilla perinteinen navigointi yläpalkki. (Fried-
man 2018.)

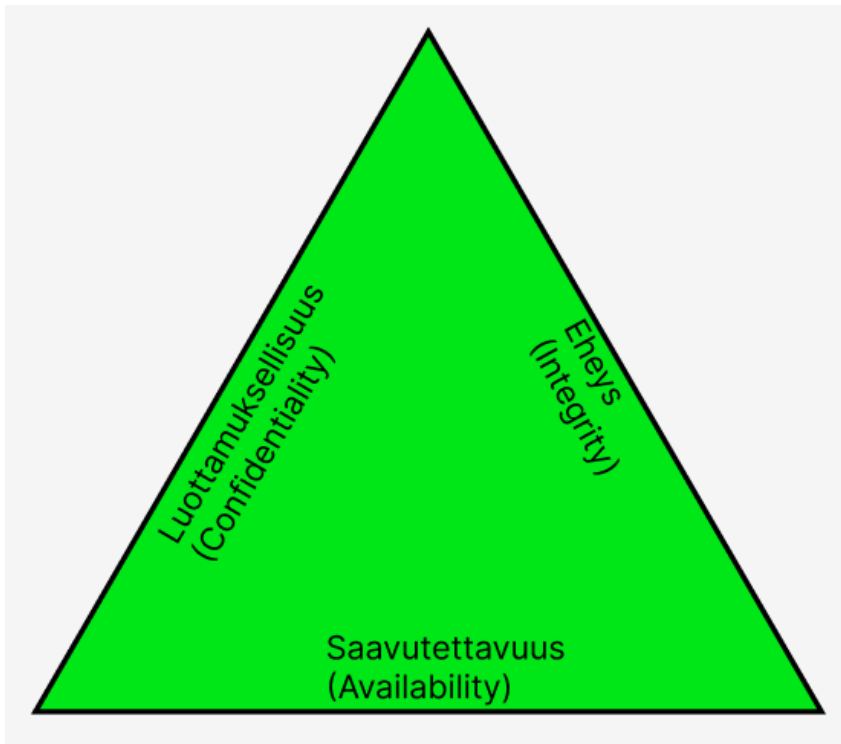
2.8 Tietoturva

Tietoturvalla tarkoitetaan teknisiä sekä hallinnollisia toimia, jotta voidaan varmistaa tietojen luotta-
muksellisuus, eheys ja käytettävyys (Kuva 4).

- Luottamuksellisuudella tarkoitetaan, että tiedot ovat vain niiden käyttöön oikeutettujen saatavilla.
- Eheydellä tarkoitetaan, että tiedot ovat muutettavissa vain siihen oikeutetut.
- Käytettävyydellä eli se, että tiedot ja tietojärjestelmät ovat hyödynnettävissä siihen oi-
keutetuilla.

Tietoturva sääntelyn piirissä ovat mm. teleyritykset, viestinnän välittäjät, yhteisötilaajat ja monet
muut. Velvoitteet ovat säädetty laissa. Niitä kuteinkin tarkennetaan kyberturvallisuuskeskuksen

määräyksillä, joka myös valvoo ja ohjaa veloitteiden noudattamista. (Kyberturvallisuuskeskus 2020.)



Kuva 4 CIA-kolmio (Omakuva 2023 mukailen Walkowski 2019)

Nykypäivänä turvallisuutta ei kuitenkaan välttämättä nähdä liiketoimintaa edistävänä asiana vaan enemmänkin vain lisäkuluna ja resurssien viejänä. Turvahankkeita on myös vaikeita perustella yrityksissä. Kuitenkin tutkimuksen mukaan 62 % kotimaisista yrityksistä epäilee joutuvansa jonkunlaisen kyberhyökkäyksen kohteeksi seuraavan kahden vuoden aikana (Loihde Trust 2023.)

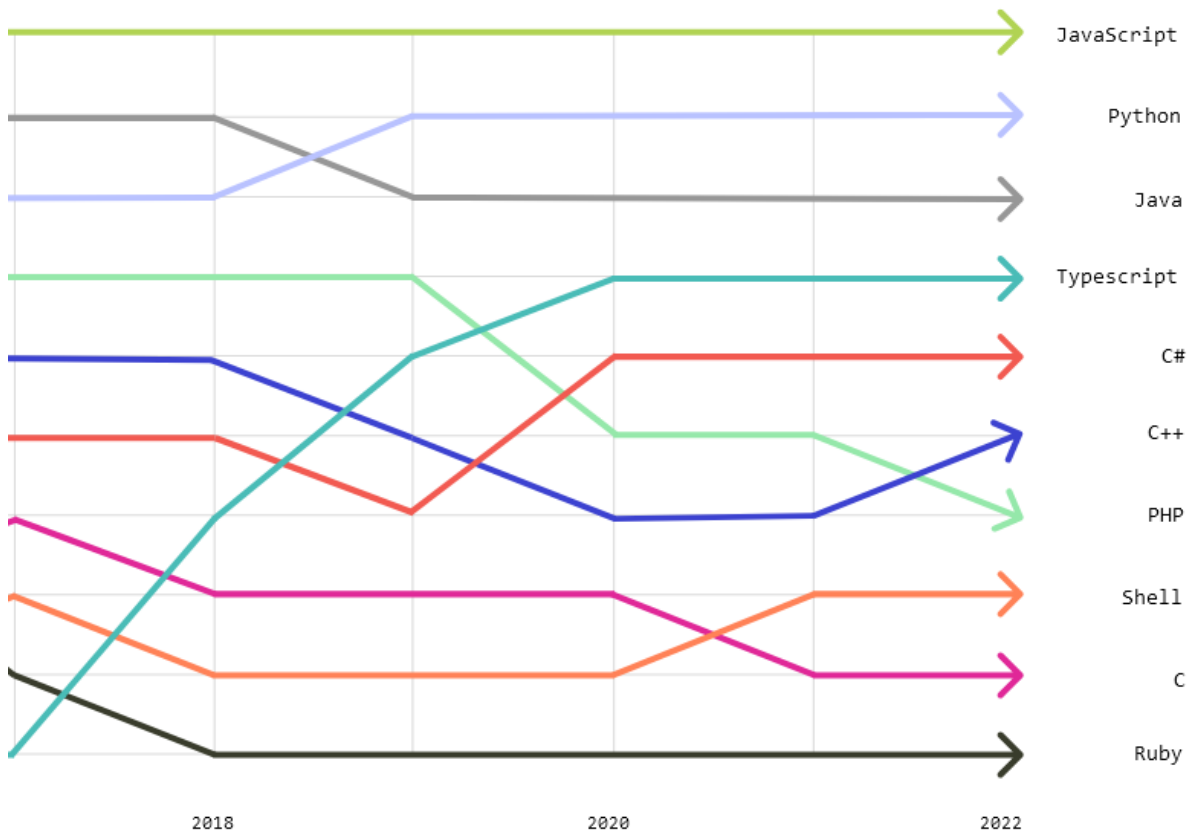
2.9 JavaScript

Vuonna 1995 Brendan Eich kehitti ensimmäisen version JavaScriptistä vain 10 päivässä. Kielelle hän antoi nimen Mocha, jota kuitenkin hyvin nopeasti alettiin kutsumaan nimellä LiveScript. Myöhemmin nimi vaihtui JavaScriptiksi. Kielen kehittämisen taustalla oli tarve päästä eroon staattisista nettisivuista, koska ne eivät kyenneet dynaamiseen käyttäytymiseen, eikä vuorovaikutukseen. Netscape oli ensimmäinen selain, joka otti JavaScriptin käyttöön. (Dickson 2022.)

JavaScript on kevyt käyttöinen ja dynaaminen ohjelmointikieli. Se tukee objekti-olio- ja funktio-ohjelmointia, joka tekee siitä joustavan moneen eri tarkoitukseen ja tyyliin ohjelmoida. Sen dynaami-

siin ominaisuuksiin kuuluu esimerkiksi funktiomuuttujat, dynaamiset komentosarjat ja reaaliaikainen objektin kokoaminen. Pääasiassa kieltä käytetään nettisivujen tekemiseen, mutta sitä voit käyttää myös selaimen ulkopuolella moniin tarkoituksiin, kuten vaikka Adobe Acrobatissa. JavaScriptistä käytetään usein lyhennettä JS (Mozilla Foundation 2023.)

Vuonna 2022 JavaScript piti edelleen ensimmäistä sijaa maailman käytetyimpänä ohjelmointikielenä (Kuva 5).



Kuva 5 JavaScriptin suosio (Github 2022)

2.10 SQL

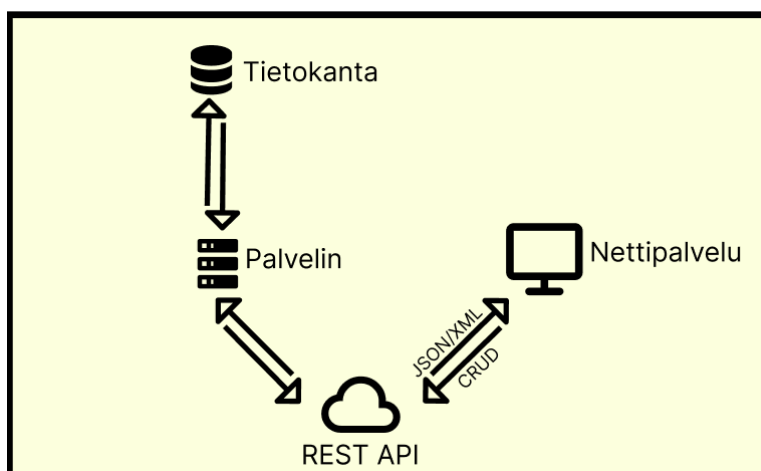
Vuonna 1970 IBM:n tutkimuslaboratoriossa työskennellyt E.F.Codd julkaisi raportin "A Relational Model of Data for Large Shared Data Banks", joka käsitteli miten data kuuluisi esittää useiden data pöytien muodossa sen sijaan että käytettäisiin osoittimia, joiden avulla navigoidaan eri entiteettien välillä. Tätä kutsuttiin relaatiomalliksi. Raportin julkaisun yhteydessä hän ehdotti DSL/ALPHA nimestä kieltä, jolla voitaisiin käsitellä relaatiotaulukoiden dataa. Pian tämän jälkeen IBM määräsi ryhmän luomaan prototyypin näiden ideoiden pohjalta. Ryhmä loi yksinkertaisimman version Coddin kielestä, jota kutsuttiin nimellä SQUARE. Myöhemmin kieltä päivitettiin, joka johti uuteen kieleen

SEQUEL. Lopulta se nimettiin uudelleen SQL:ksi. Yleisesti ymmärretään, että SQL tulee sanoista Structured Query Language. Puhekielessä siitä voidaan puhua myös sequel-nimellä. (Beaulieu 2005, 1.)

SQL on kieli, jolla hallitaan tietokantaan. Sillä voidaan luoda, käsitellä ja hakea dataa tietokannasta. Sen käyttäminen on yksinkertaista ja hyvin tehokasta. Oikealla tavalla luotu tietokanta voi sisältää valtavia määriä tietoa ja sen käsitteleminen, kuten muokkaaminen tai tiedon hakeminen on helppoa. Kieli tulee jatkossakin säilyttämään vahvan suosionsa, vaikka se onkin jo hyvin vanha! (Beaulieu 2005, preface.)

2.11 REST API

REST tulee sanoista Representational State Transfer, kun taas API muodostuu sanoista Application Programming Interface. API on yksinkertaisesti linkki tai yhteys kahden erityyppisen ohjelmiston, palvelun tai alustan välillä. REST-sovellukset tekevät pyyntöjä HTTP-protokollaa käyttäen, käsitelläkseen CRUD-toimintoja, eli luoda, lukea, päivittää ja poistaa (Kuva 6). RESTiä käyttävät lähes kaikki suurimmat verkkopalvelut, kuten Facebook ja Google käyttävät REST-sovellusliittymää. Suurin syy tähän on REST rakenne. Tämän lisäksi REST on hyvin keyyt ja joustava käyttöinen. Sillä pystyy myös käsittelemään suuria määriä toimintaa jouhevasti mikä tekee siitä erityisen houkuttelevan. Sen ansiosta REST on hyvin monipuolinen ja helposti muiden käytettävissä. Kuka tahansa voi ottaa käyttöön oman version RESTistä vakiokirjasto ominaisuuksilla ja valita itselleen sopivan kielen kuten, PHP. (Uzayr, Rooney 2016, 1.)



Kuva 6 REST API kaavio (Omakuva 2023 mukaillen Dalal 2020)

WordPressissä REST API on erittäin merkittävässä roolissa. Se helpottaa sivustojen valmistamista esimerkiksi erikoiseditorien ja sivutökalujen avulla. WordPress kehittäjille sen oman API:n käyttö ei ole mikään uusi asia, mutta ennen REST API:n aikaa se oli hyvä vain sisäisille prosesseille kuten laajennuksille. Se loi omat rajoituksensa, koska siitä ei ollut hyötyä ulkopuolisille palveluille. REST API:n mukaan tuominen antoi kehittäjille ja sivustolle mahdollisuuden olla yhteydessä ulkopuolisten palvelujen ja verkkosivustojen kanssa. Tietoja voidaan vaihtaa muiden HTTP-protokollaa tukevien verkkopalvelun kanssa. Nämä palvelut voivat olla vaikka eri koodauskielillä toteutettuja ratkaisuja tai niiden rakenne voi olla täysin erilainen. Tämä auttaa erityisesti, jos haluat käyttää kolmannen osapuolen palveluita tai sisältöä, jotka ovat yhteydessä WordPressin kanssa. (Uzayr, Rooney 2016, 6.)

3 WORDPRESS-LISÄOSAT

Tavoitteena on löytää innovoivia mutta selkeitä ratkaisuja nettisivua varten. Lisäosa tehdään tarkoitusta varten, siihen voidaan integroida toimeksiantajalle tärkeitä, joita valmiit paketit eivät tarjoa. Tähän myös saadaan asiakkaan näkökulma, jossa käyttäjillä testaaminen on oleellinen osa tutkimusta. Tietoja analysoimalla on helpompi rakentaa käyttäjäkokemusta edistävä lisäosa.

Lisäosaa suunniteltaessa täytyy muistaa saavutettavuuden täytyminen, jotta se kohtaa loppu sivuston tasoa. Osa tyyleistä ja fonteista tulee vastaamaan sivustoa ja ne on jo ennalta mietitty saavutettavuuden, kuten myös visuaalisuuden kannalta. Tämä helpottaa hieman suunnitteluvaihetta, joten voin keskittyä muihin asioihin saavutettavuudessa.

3.1 Toteutusvaihtoehdot

Ennen päätöstä lisäosan rakentamisesta useita vaihtoehto käytiin läpi. Kriteereistä tärkein oli, että sen käyttö ei maksa mitään. Muita kriteerejä oli helppokäyttöisyys ja WordPress integrointi mahdollisuus. Suhteellisen hyviäkin lisäosia, mutta kaikki kriteerit täyttävää lisäosaa ei kuitenkaan löytynyt. Useimmat tarjoaa mahdollisuuden pieni muotoiseen käyttöön rajoitetuilla ominaisuuksia. Jos haluaa enemmän ominaisuuksia ja vapautta, se tarkoittaa samalla kuukausimaksua 10–100 € välillä. Minun näkökulmastani tämä mahtava tilaisuus kehittää oma lisäosa. Muiden lisäosien tutkimisprosessi antoi ideoita rakenteeseen ja siihen mikä on funktionaalisesti miellyttävien käyttää niin puhelimella kuin pöytäkoneellakin.

1. WordPress appointment hour booking

Lisäosan rakenne on selkeä. Ominaisuuksia on paljon mm. sähköpostivahvistus, statistiikka varten on mahdollista tuoda lisäosasta tiedot .csv muodossa ja analysoida niitä Excelissä. Lisäksi lisäosalla on vaikuttavat arvostelut. (WordPress 2023a.)

Valitettavasti kuitenkin lisäosan hieman vanha ja tylsä ulkonäkö ei sovellu sivuston kanssa. Sen lisäksi rakenne on suunniteltu jokapäiväisen käytön ympärille, joka ei sovellu käyttö

kohteeseen. Käyttäjä joutuisi käymään jokaisen päivän erikseen nähdäkseen onko vapaana olevia aikoja. Sen lisäksi ajan peruminen on valitettavasti maksumuurin takana.

2. WordPress booking calendar contact form

Teknisesti suhteellisen lupaavan oloinen lisäosa. Sisään rakennuttu captcha, hyvin muokattavissa soveltuu siis yleiskäyttöön. Lisäosa on myös käännetty Suomen kielelle. (WordPress 2023b.)

Valitettavasti ei ole mahdollisuutta testata käytännössä ilman maksamista vain lisäosa tarjoilee suoraan maksumuurin.

3. Vello

Velloilta löytyy varauskalenteri kattavilla ominaisuuksilla ja lupaavan näköisellä kalenterilla. Tyylikäs ulkonäkö, helpon oloinen käyttöliittymä ja ennen kaikkea WordPress integrointi mahdollisuus tekee siitä mielenkiintoisen vaihtoehdon (Kuva 7). Vello tarjoaa ilmaisen version leikatuilla ominaisuuksilla. Mahdollisuutena on myös kokeilla 30 päivää ilmaisena maksullisia versioita, jotka tarjoavat kattavammin ominaisuuksia ja tuo lisäosaan joustavuutta (Vello.fi 2023a.)

Vello oli kaikista kiinnostavin näistä vaihtoehtoista. Valitettavasti ilmaisen puolen ominaisuudet eivät riitä tässäkään käyttötarkoitukseen.



Kuva 7 Vello kalenteri (Vello 2023b)

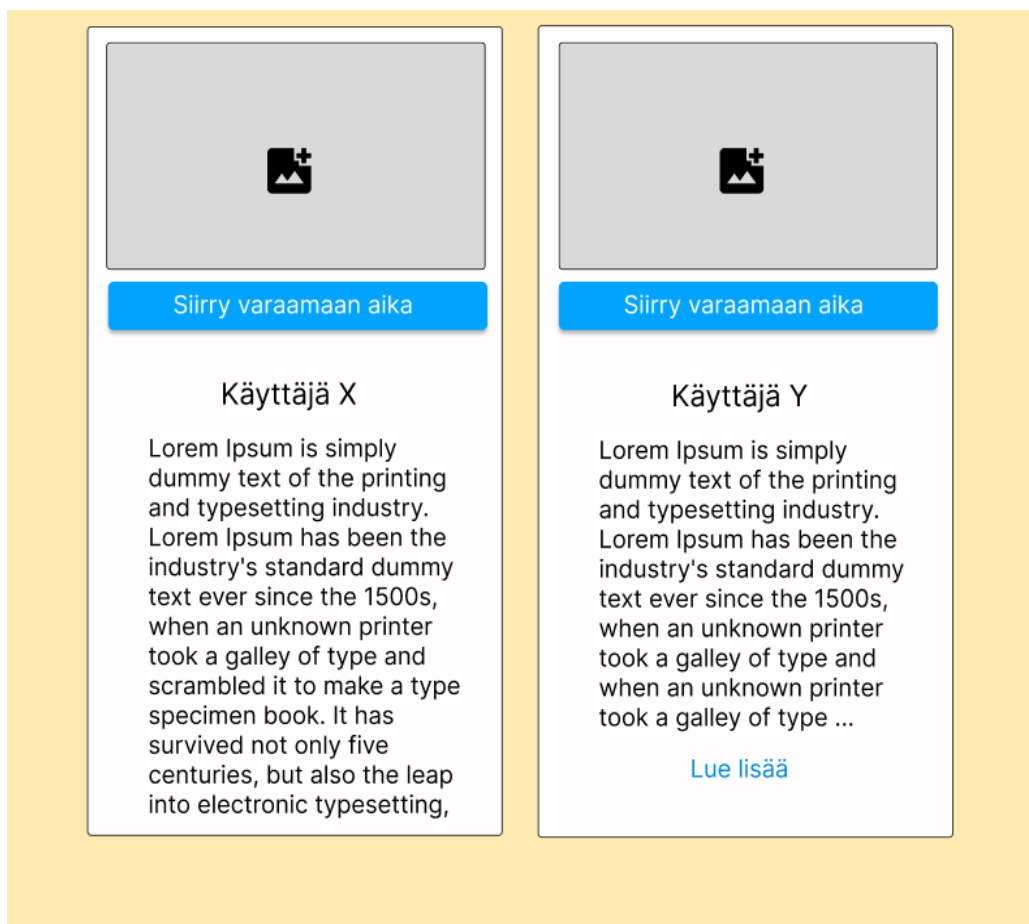
4. TIMIFY

TIMIFY tarjoaa eniten ominaisuuksia ja joustavuutta muokattavuuden suhteen. Se on helppo ottaa käyttöön ja tarjoaa esimerkiksi WordPress laajennuksen. Maksullisessa versiossa on asiakaspalvelu käytettävissä. Palveluntarjoaja on myös voittanut palkintoja ja jopa sertifioitu (TIMIFY.com 2023.)

Vaikka sitä onkin mahdollista käyttää ilmaiseksi se käytännössä tarkoittaa vain hyvin pieni muotoista yrittämistä, koska ominaisuuksia on rajoitettu hyvin paljon. TIMIFY on myös vaihtoehtoista selkeästi kallein. Aloitus hinta on 25 € kuukaudessa ja hinta voi nousta yli sataan euroon kuukaudessa riippuen siitä miten paljon ominaisuuksia haluaa käyttää. Hinnat ilmoitetaan aina ilman arvolisäveroa. TIMIFY tarjoaa myös Enterprise plus version, jonka hinta täytyy erikseen kysyä.

3.2 Oman lisäosan suunnittelu

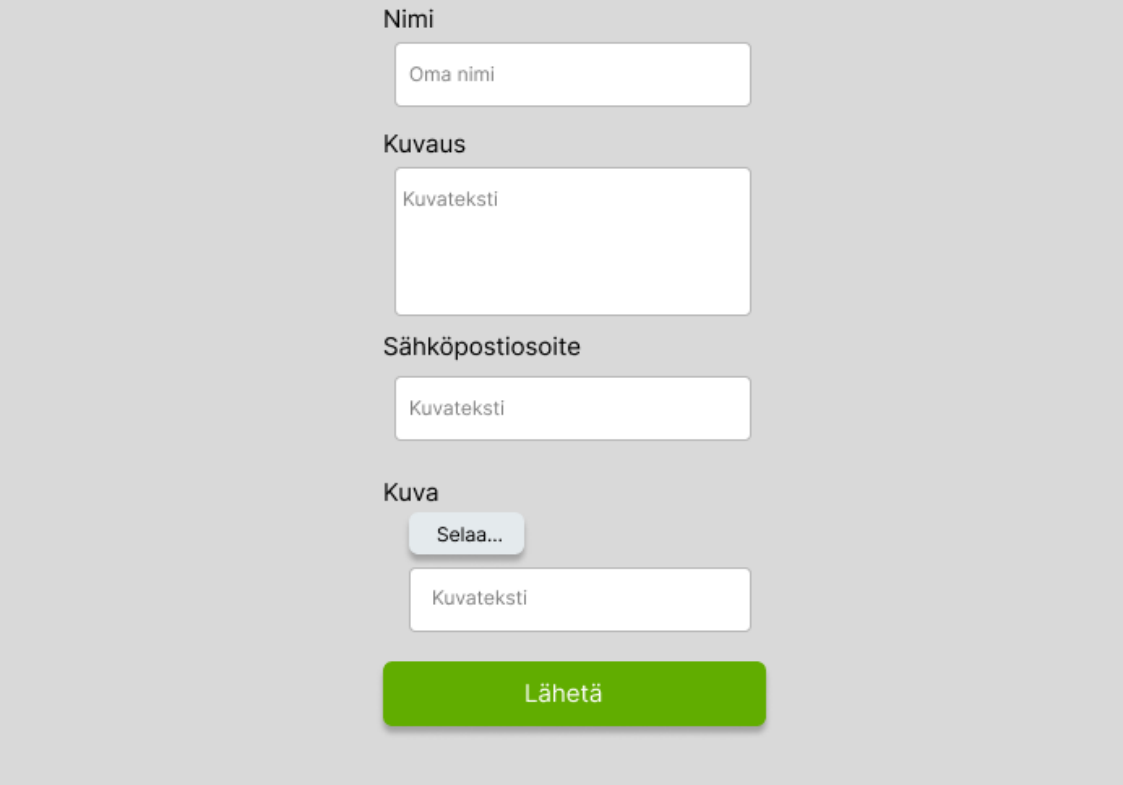
Lisäosan rakentaminen lähti käyntiin käyttöliittymän suunnittelusta Figmalla. Suunnitteluun kului yllättävän pitkä ja versioita käyttöliittymästä tehtiin useampi ennen kuin siitä tuli miellyttävä. Ideana oli luoda hyvin mobiililla ja pöytäkoneelle suunniteltu käyttöliittymä, joka skaalautuisi helposti isommillekin ruudulle. Käyttäjälle tarjotaan ensimmäisenä valinta keneltä voi varata ajan (Kuva 8). Jokaiselta varattavalta henkilöltä löytyy oma kortti, jossa on kuva, kuvausteksti ja nappula, josta voi siirtyä katselamaan vapaita aikoja. Käyttäjän valitessa itselleen sopivimman henkilön, hänelle tarjotaan vapaana olevat ajat. Ajan valinnan jälkeen käyttäjä täyttää varauslomakkeen ja saa vahvistusviestin sähköpostiin. Samalla aika poistuu varattavien joukosta. Tein lopuksi Figmasta oikeasti käytettävän. Tällä saatiin suoraan akuuttia palautetta ja se oli helpompi ymmärtää miten lisäosa tulisi käyttäytymään oikeasti.



Kuva 8 Henkilöiden esittely käyttäjälle (Omakuva 2023)

Hallintapaneelin suunnittelu aloitettiin miettimällä, mitä tietoja työntekijöistä tarvitaan. Eli nimi, kuvaus henkilöstä tai omasta osaamisesta, jonkinlainen kuva ja kuvateksti (Kuva 9). Seuraavaksi

tietenkin tarvittiin tietokannasta olevien henkilöiden tietojen tarkistaminen, kuten myös poistaminen ja tietojen olemassa olevien tietojen muokkaaminen. Tein sille erillisen sivuston, jossa on työntekijät listattuna allekkain. Jokaisen työntekijän perässä on kaksi painiketta, ”Muokkaa” ja ”Poista”. Muokkaa painike aukaisee modaali-ikkunan, jossa on kaikki tietokannasta löytyvät tiedot. Näitä voi joko muokata tai vain tarkistaa tietojen paikkaansa pitävyyden. Poista napista työntekijä poistetaan tietokannasta. Tämän jälkeen siirryttiin uusien aikojen lisäämissivulle. Sinne tuli pudotusvalikko, jossa on olemassa olevat työntekijät, ajankohdan valinta ja nappi, josta lisätään uusi aika. Samalla sivulle myös tulee valitun henkilön olemassa olevat listana. Jokaisen ajan vieressä on ”poista” painike, jota painamalla aika poistetaan tietokannasta.



The image shows a user profile form with the following fields and elements:

- Nimi**: A text input field with the placeholder text "Oma nimi".
- Kuvaus**: A larger text input field with the placeholder text "Kuvateksti".
- Sähköpostiosoite**: A text input field with the placeholder text "Kuvateksti".
- Kuva**: A button labeled "Selaa..." followed by a text input field with the placeholder text "Kuvateksti".
- Lähetä**: A prominent green button at the bottom of the form.

Kuva 9 Hallintapaneelin suunnittelu (Omakuva 2023)

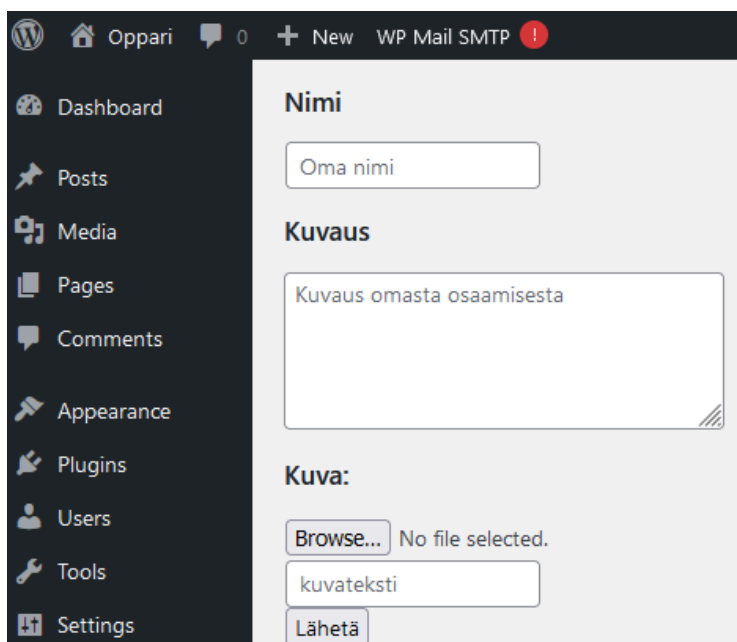
3.3 Oman lisäosan koodaaminen

Lisäosan koodaaminen alkoi WordPress perusteiden kertaamisella, koska edellisestä kerrasta WordPress-ympäristössä oli kulunut jo vuosi. Tutkin WordPressin omaa APIa ja kaivoin esille vanhoja koodausharjoituksia, joiden perusteella pääsin alkuun. Tarvitsin myös kertausta SQL-tietokannan hallitsemista ja sen koodaamisesta.

Päätin aloittaa koodaamisen kalenterin hallintapuolelta, jonka jälkeen olisi luonnollista siirtyä tekemään käyttäjän näkymää. Lisäosan hallintapanelin tekeminen alkoi lisäosan rungon rakentamisella ja lisäosan pakollisten tietojen täyttämällä. WordPress vaatii joitakin tietoja lisäosasta, kuten tekijän, lisenssin, kuvaus ja lisäosan nimi. Tein tietokantaa varten komennot, joiden avulla lisäosa luo oikeanlaiset taulut tietokantaa asennuksen yhteydessä. Seuraavana tarvittiin vain enää alivalikot, mikä oli viimeinen tehtävä ennen kuin pääsin asentamaan lisäosaa ensimmäistä kertaa.

Lisäosan oli nyt valmis, joten siirryin tekemään ensimmäistä hallintapaneelinsivua. Uuden henkilön lisääminen oli tärkeysjärjestyksessä ensimmäisenä, joten aloitin siitä. Se vaati muutaman tekstikentän ja kuva lisäämisen. Eli omanimikenttä, kuvauskenttä mikä on oma osaaminen, sähköposti, kuvan lisääminen ja kuvan kuvausteksti (Kuva 10). Sähköpostikenttä puuttuu vielä kuvasta. Kuvan lisääminen toimii tällä hetkellä vielä tietokantaan *LONGBLOB* datatyypinä, mutta se päivittyy jossain vaiheessa suoraan kuvien lataamiseen palvelimelle. Se nopeuttaa hieman sivuston toimintaan. Lähetä napista tiedot päivittyvät MySQL-tietokantaan. Laitoin myös muutaman virheenkäsittelyn alustavasti, että tietokantaan ei tule tyhjiä kenttiä vahingossa. Virheenkäsittelyä käydään todennäköisesti läpi myöhemmin, kun nähdään käyttäjän kokemukset.

Näin aluksi lisäosa on hyvin karkean näköinen, koska toiminnallisuuksien tekeminen on tärkein. Tässä vaiheessa tarvitsin vielä yhteyden tietokantaan ja tietojen lukemisen sivun lomakkeelta, ennen kuin pääsin kokeilemaan ensimmäistä kertaa, miten se toimi.

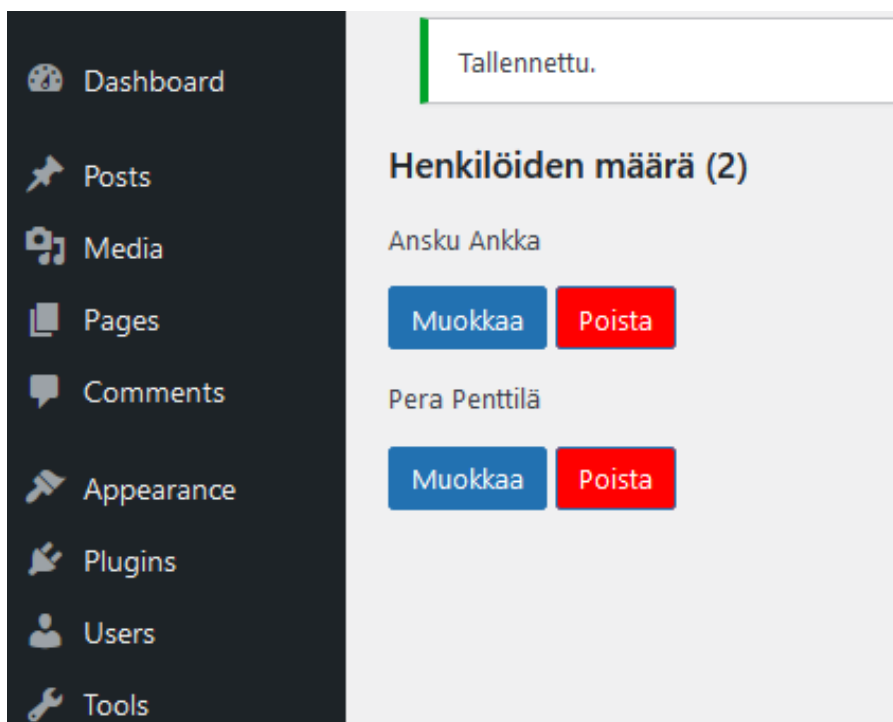


Kuva 10 WordPress hallintapaneeliin lisätty oma plugin (Omakekuva 2023)

Tässä vaiheessa testasin käyttäjien lisäämistä ja varmistin sen toimivuuden. Seuraava looginen askel oli tehdä tietokannassa olevien työntekijöiden hallintasivu. Täältä näkee olemassa olevat henkilöt ja niiden profiileja voi muokata ja myös poistaa (Kuva 11). Sivusto toteuttaa käyttäjien hallinnan tietokannassa. Sivuston latautuessa, se hakee olemassa olevat käyttäjät tietokannasta ja näyttää ne listana. Käyttäjä voi valita haluamansa käyttäjän listalta ja poistaa henkilön painamalla poista-nappulaa

Poista-nappulaa painaessa, sivusto varmistaa käyttäjän poiston, kysymällä: "Oletko varma, että haluat poistaa kyseisen henkilön?". Tässä vaiheessa käyttäjällä on mahdollisuus perua toiminto tai jatkaa poistamista. Jos käyttäjä vahvistaa poiston, sivusto suorittaa SQL-komennon, joka poistaa valitun käyttäjän tiedot tietokannasta. Toimenpide poistaa pysyvästi käyttäjän tiedot.

Tämä käyttäjien hallintajärjestelmä mahdollistaa tehokkaan ja helpon henkilöiden hallinnan tietokannasta, eikä tulevilla lisäosan käyttäjille tarvitse opetella erikseen SQL-käyttöä.



Kuva 11 Ensimmäinen version henkilöiden hallinnasta (Omakuva 2023)

Lisäksi sivustolle on toteutettu erillinen ponnahdusikkuna, joka mahdollistaa käyttäjätietojen tarkistamisen ja muokkaamisen. Käyttäjän valitessa haluamansa henkilön nimen ja napsauttaa "muokkaa"-nappulaa, sivusto avaa uuden ikkunan, jossa käyttäjä voi muokata valitun käyttäjän tietoja. Uudessa ikkunassa käyttäjälle esitetään lomake, missä näytetään valitun käyttäjän tiedot, nimi, sähköpostiosoite, puhelinnumero, kuva ja kuvan kuvaus. Nämä tiedot haetaan PHP komentoilla

tietokannasta ja täytetään HTML-kentät olemassa olevilla tiedoilla (Kuva 12). Näin käyttäjän on helppo tarkistaa mitä tietoja tietokannasta löytyy ja käyttäjä voi tehdä haluamiaan muutoksia näihin tietoihin.

Käyttäjän tehtyä tarvittavat muutokset, hänellä on kaksi vaihtoehtoa: hän voi joko sulkea ikkunan painamalla ohi ikkunasta tai tallentaa muutokset painamalla "tallenna muutokset"-nappulaa. Käyttäjä voi painaa ohi ikkunasta, jolloin modaali-ikkuna sulkeutuu eikä mitään muutoksia lähetetä tietokantaan. Jos käyttäjä sen sijaan valitsee "tallenna muutokset", sivusto tarjoaa ponnahdusikkunan, jossa käyttäjää pyydetään vahvistamaan muutosten tallentaminen. Ikkuna esittää kysymyksen: "Oletko varma, että haluat tallentaa muutokset?" Käyttäjällä on jälleen kaksi vaihtoehtoa: hän voi peruuttaa tallennuksen painamalla "peruuta" tai vahvistaa muutosten tallentamisen painamalla "tallenna". Jos käyttäjä vahvistaa tallennuksen, sivusto suorittaa SQL lauseen, joka päivittää muokatut tiedot tietokantaan. Näin varmistetaan, että käyttäjän tekemät muutokset ovat voimassa ja päivittyvät järjestelmään.

Tällainen ominaisuus tarjoaa käyttäjäkokemuksen joustavuutta ja parantaa käyttäjien tietojen hallintaa. Käyttäjät voivat tarkastella ja päivittää tietojaan helposti. Muu muokkaaminen on myös tärkeää. Tiedot saattavat päivittyä oman ammatin kehittymisen myötä, sähköpostiosoite saattaa muuttua, kuvan päivittäminen ajantasaiseksi tai yksinkertaisen virheen korjaaminen.

Muokkaa tietoja

Nimi:

Kuvaus:

Ankkalinnan parhain

Kuva: No file selected.

Kuvateksti:

Kuva 12 Henkilön muokkaaminen (Omakuva 2023)

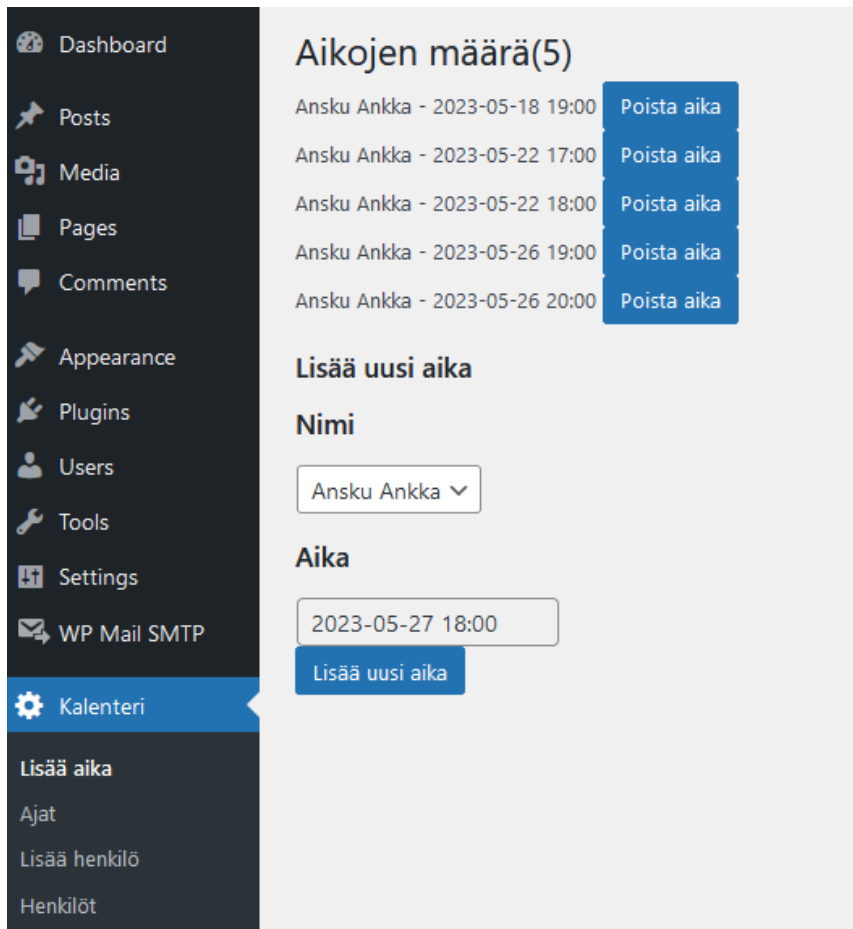
Nyt tietokannasta löytyy kaikki tarvittavat tiedot uuden ajan luomiseen eli henkilöt, lähdin hahmottelemaan miltä uuden ajan lisäyssivu näyttää toimintojen kannalta. Ensimmäisenä hain tietokannasta olemassa olevat henkilöt PHP:llä, jonka jälkeen tein koodin, joka lisää jokaisen henkilön pudotusvalikkoon. Käyttäjän valitessa pudotusvalikosta nimen, sivusto hakee tietokannasta kyseiselle nimelle olemassa olevat ajat ja listaa ne allekkain. Jokaisen ajan perään tulee "poista aika" painike (Kuva 13). Tätä painiketta painamalla suoritetaan SQL-lause, joka poistaa tietokannasta valitun ajan. Sivusto pyytää varmistamaan ennen poistoa, että varmasti valittu aika on oikea. Ennen aikojen hakemista sivusto tarkistaa myös, onko kyseinen aika jo vanhentunut. Jos aika on mennyt jo, sivusto suorittaa automaattisesti SQL-komennon, joka poistaa vanhentuneet ajat.

Lisäsin JavaScriptillä valmiiksi tehdyn kalenterin, jonka avulla pystytään valitsemaan haluttu ajan kohta. Painamalla "lisää uusi aika" PHP-koodi lukee kalenterista valitun ajan ja valitun nimen, jonka se lisää tietokantaan. Ajan lisäyksen jälkeen sivusto päivittää itsensä, jonka jälkeen uusi aika näkyy listassa. Tyyllittelyn jätin myöhempää vaihetta varten, koska funktionaalisesti se oli toimiva tässä vaiheessa. Tätä sivustoa varten koodia oli kertynyt yhteen tiedostoon jo niin paljon, että päätin jakaa sen useampaan tiedostoon luettavuuden helpottamiseksi.

Tein ajanlisäyssivulle virheidenkäsittelyä, jotta käyttäjä saa selkeämpiä viestejä, jos jotain menee pieleen esimerkiksi aikaa lisätessä tai poistaessa. Onnistuneesti lisätty aika antaa käyttäjälle WordPressin oman tyyllittelyn mukaisen tekstin "Aika lisätty", joka on merkattu vihreällä tagilla onnistumisen merkiksi. Virheen sattuessa käyttäjä saa asianmukaisen virheen punaisella tagilla ja virhe tekstillä.

Tässä vaiheessa optimoin tietokantaoperaatiot, jotta sivusto toimii nopeammin. Aion myös varmistaa, että sivusto on turvallinen ja suojattu mahdollisia tietoturvaongelmia vastaan, kuten SQL-injektioita. Ensimmäisenä puhdistin syötteet. Käytin *htmlspecialchars()*-funktiolla, jolla poistetaan tai muutetaan syötteestä kaikki ei-toivotut merkit tai tiedot. Tällä tavalla estetään esimerkiksi XSS-hyökkäykset, jolla pyritään vaikuttamaan tietokantaan. Seuraavana validoin sähköpostin kentän syötteet käyttämällä PHP:n omaa *filter_var()*. Tässä tarkistetaan, onko käyttäjä antanut kelvollisen sähköpostiosoitteen. Jos ei sähköposti ei kelpaa, sivusto antaa siitä virheilmoituksen käyttäjälle, eikä tietoja suoriteta ennen kuin oikea sähköpostiosoite on annettu. Lopuksi vielä ennen tietokantaan kyselyn tekemistä käytin WordPressin omaa *prepare()*-funktiota. Se vastaa *mysqli_real_escape_string()*, mutta se on WordPressin sisään rakennettu versio. Tässä vielä varmistetaan tietojen

puhtaus ennen kuin ne menevät tietokantaan. Näillä pienillä muutoksilla sain sivustosta huomattavasti turvallisemman käyttöä. Tietoturva kuitenkin tulisi päivittää uusien uhkien ja hyökkäysten varalta säännöllisesti.

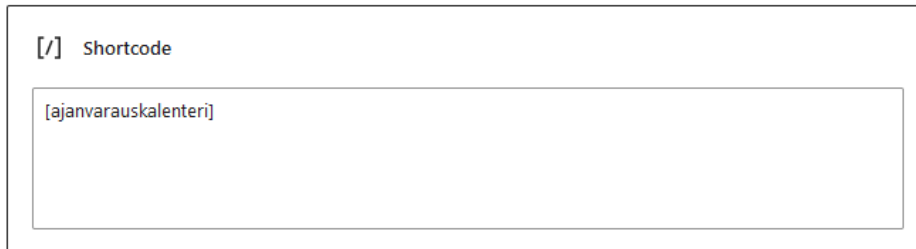


Kuva 13 Olemassa olevat ajat (Omakuva 2023)

Seuraavaksi lähdin tekemään asiakkaiden näkymää. Ensin tein PHP koodin pätkän, jossa määritellään WordPressissä lyhytkoodi. Sen avulla ajanvarauskalenterin paikka on helppo määrittellä tulevalla sivustolla (Kuva 14). Tämä on myös tuttu ja turvallinen tapa monelle WordPress-kehittäjille.

providing quality doohickeys to the public ever since. Located in Gotham City, XYZ employs over 2,000 people and does all kinds of awesome things for the Gotham community.

As a new WordPress user, you should go to [your dashboard](#) to delete this page and create new pages for your content. Have fun!

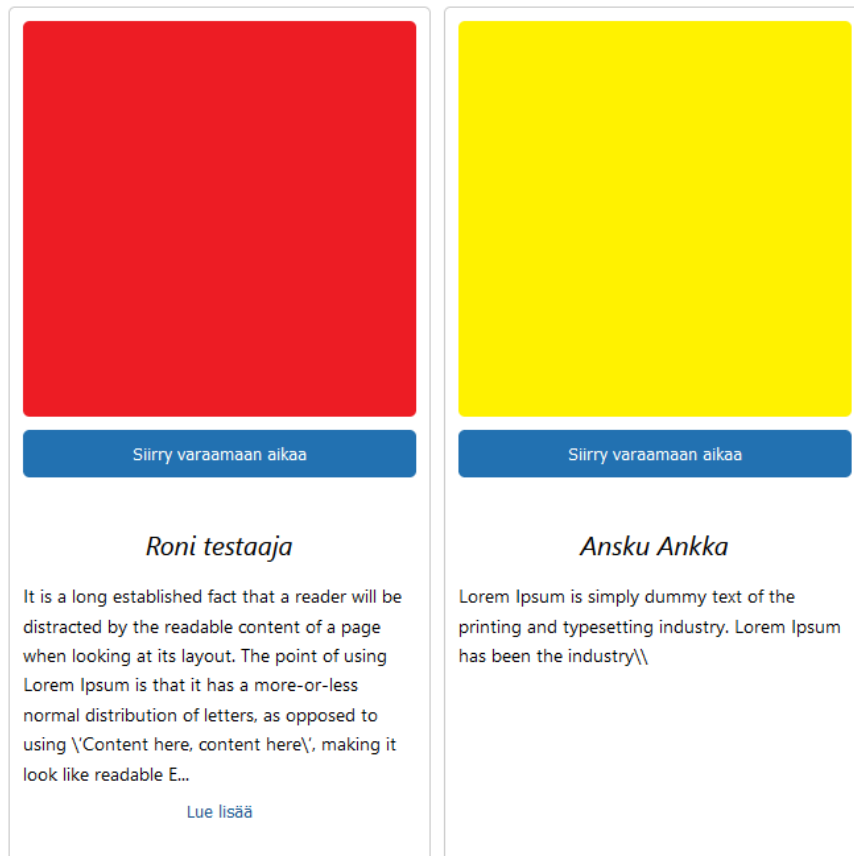


Kuva 14 WordPress lyhytkoodin (Omakekuva 2023)

Tämän jälkeen hain tietokannasta työntekijöiden tiedot. Suoritin PHP komennot, jolla sain tarvittavat tiedot. Tarvittiin työntekijän asettama kuva, kuvateksti ja työntekijän nimi käyttäjän näkymään varten, mutta myös sähköpostiosoite, jos käyttäjä päättää varata ajan. Tässä vaiheessa muotoilin CSS-tyylien avulla kortin, jonka sisälle voitaisiin purkaa tarvittavat tiedot. Jokaiselle työntekijälle tuli oman korttinsa, josta nämä tiedot löytyivät tehdyn Figma mallin mukaisesti (Kuva 15).

Lisäsin myös napin, josta pääsee tutkailemaan valitseman henkilön vapaina olevia aikoja. Tekstit rajoitettiin 300 sanaan per kortti ja jos raja ylittyy, korttiin tulee "Lue lisää" -valinta, josta pääsee lukemaan koko tekstin. Tässä vaiheessa myös tein alustavasti responsiivisuuden. Kortit skaalautuvat hyvin näyttökoon mukaan. Pienemmillä ruuduilla, kuten puhelimella, kortit tulevat allekkain. Viimeistely jää kuitenkin vielä myöhempää ajan kohtaa varten.

Nyt sivusto näytti nyt jo huomattavasti käyttäjäystävällisemmältä ja informatiivisemmalta. Käyttäjä voi helposti löytää haluamansa henkilön tiedot ja tarkastella valitun työntekijän vapaana olevia aikoja. Lisäksi "Lue lisää" -toiminto mahdollistaa tarkemman tiedon saannin kiinnostavista työntekijöistä.



Kuva 15 Käyttäjän valinta näkymä (Omakuva 2023)

Nyt sivustolla on jo käytössä tyylikäs käyttöliittymä, jota on mukava lähteä jatkamaan. Sivun esittelee erilaisia henkilöitä käyttäjälle kuvakorttien muodossa. Jokaisen henkilön kuvakortilla on esitetty kuva ja lyhyt kuvausteksti, joka on toteutettu lorem ipsum -mallitekstin avulla.

Käyttäjä voi selata eri henkilöiden kuvakortteja ja lukea niiden kuvaustekstit nähdäkseen, ovatko henkilöt potentiaalisia valintoja omiin tarpeisiinsa. Käyttäjän löytäessä sopivan henkilön, hän voi valita kyseisen henkilön napsauttamalla kuvakortin sisältä löytyvää painiketta, jossa lukee "Siirry varaamaan aika".

Sopivan henkilön kohdalta napsauttamalla "Siirry varaamaan aika" -painiketta, ruudulle aukeaa modaali-ikkuna. Ikkunassa näytetään kaikki vapaana olevat ajat valitsemalleen henkilölle. Tiedot haetaan tietokannasta PHP komennolla. Sen jälkeen JavaScript koodi saa tietokannasta haetut tiedot. Aikoja ei tule paljon per työntekijä, vapaana olevien aikojen listaus tapahtuu allekkain aika järjestyksessä.

Jos aikoja ei ole vapaana, käyttäjälle tarjotaan modaali-ikkunassa teksti "Ei vapaana olevia aikoja". Jokainen aika on klikattava painike. Käyttäjän valitessa haluamansa ajan, heidät ohjataan eteenpäin lomakkeelle täyttämään omia tietojaan (Kuva 16). Tämän lisäksi lisäsin SQL komentoon hakemaan vain tästä ajasta eteenpäin olevat ajat. Loput ajat hallintapaneeli poistaa automaattisesti.

Ansku Anka, ajanvaraaminen

[Siirry takaisin tarkastelemaan aikoja](#)

Etunimi*

Sukunimi*

Katuosoite*

Postinumero*

Puhelinnumero*

Sähköposti*

Asiointikieli Suomi Ruotsi

Lisätietoja

Lähetä

Kuva 16 Ajanvarauslomake (Omakuva 2023)

Käyttäjän löytäessä sopivan ajan, hän painaa sitä jolloin, modaali-ikkunan sisältö vaihtuu varauslomakkeeseen. Omat tiedot tarvitaan varauksen vahvistamiseen. Tässä lomakkeelle pyydetään omaa etu- ja sukunimeä, osoitetta, postinumeroa, postitoimipaikkaa, asiointikieltä sekä lisätietoja. Lisätiedot on ainoa kenttä, jota ei tarvitse täyttää. Jos jokin tarvittavista kentistä on puutteellinen tai tyhjä, sivusto pyytää tarkistamaan kyseisen kentän ja täyttämään sen oikein. Tämän jälkeen 'Lähetä' painiketta painamalla tiedot siirretään palvelimelle JavaScriptillä. Käyttäjän syöttämät tiedot

puhdistetaan PHP-funktiolla, jonka jälkeen ne ovat valmiita suoritettavaksi. Sen jälkeen aika poistetaan varattavien joukosta, jonka lisäksi käyttäjä ja työntekijä saa vahvistuksen sähköpostiinsa ajan varauksesta. Tämä tarkoittaa PHP-toimintoa, joka lähettää kaksi sähköpostivahvistusta: toisen käyttäjälle ja toisen työntekijälle. Käyttäjän sähköpostiinsa vahvistuksen, jossa kerrotaan varattu aika ja varaajan nimi. Vahvistusviestin lyhyt ja ytimekäs, jotta käyttäjä saa selkeän vahvistuksen varauksestaan.

Työntekijän sähköpostivahvistus, joka sisältää kattavammin tietoa varauksesta. Siinä on varatun ajan lisäksi varaajan yhteystiedot, mahdolliset erityistoiveet tai muut lisätiedot. Viesti auttaa työntekijää valmistautumaan ajanvaraukseen ja tarjoamaan parasta mahdollista palvelua. Sähköpostivahvistusten lähettäminen tapahtuu PHP:llä. Se oli hieman haasteellista, koska se ei toiminut localhostista kovin helposti, joten jouduin turvautumaan omaan webhotelliin sen testaamisen kanssa. Oli tärkeää varmistaa, että web-hotellilla on oikeat asetukset sähköpostien lähettämiseen, jotta viestit päätyvät varmasti oikeille vastaanottajille. Minun web-hotellini sattui olemaan sama kuin Kartariinalla, joten yhteensopivuuden kanssa ei tullut ongelmia. Testasin sähköpostin lähettämistä useiden eri toimien kanssa, jotta voin hyvin mielin varmistaa sen toimivuuden. Onneksi useimmat web-hotellit tukevat PHP:n *mail()*-funktioita, jonka avulla sähköpostiviestien lähettäminen on mahdollista (Kuva 17).

```
//Test mail
$Subject = 'Testi title';
$message = "Testattavat tiedot:\n
Aika: $Time,\n
Nimi: $fName, $lName,\n
Osoite: $street,\n
Postinumero: $postcode,\n
Puhelinnumero: $tel,\n
Sähköpostiosoite: $Email,\n
Asiointikieli: $language,\n
Lisätiedot: $info";
$headers = ['Content-Type: text/html; charset=UTF-8'];
wp_mail($Email, $Subject, $message, $headers);
}
```

?>

Kuva 17 PHP mail funktion käyttö (Omakekuva 2023)

Lopuksi, kun varaus on tehty ja sähköpostivahvistukset lähetetty, käyttäjälle näytetään kiitosviesti ikkunassa. Viesti vahvistaa vielä kerran onnistuneen ajanvarauksen ja luo positiivisen loppukokemuksen ja työntekijä saa vahvistuksen uudesta ajanvarauksesta (Kuva 18).

Ansku Ankka, ajanvaraaminen

Kiitos ajanvarauksesta.



Kuva 18 Onnistunut ajanvaraus (Omakuva 2023)

3.4 Lisäosan tyylittely

Tärkeimpänä oli viimeistellä käyttäjän näkökulma, joten aloitin työn sieltä. Työ ei ollut niin suuri mitä aluksi ajattelin, koska se oli jo hyvällä mallilla. Käyttöliittymää oli jo testattu useampaa kertaa, joten sen rakenne oli valmis. Se vaati pelkästään hienosäätöä. Jokainen kohta kuitenkin vaati huomiota. Koska sivusto, jolle lisäosa on tulossa ensimmäisenä käyttöön ei ole valmis, päätimme yhdessä jättää värimaailman neutraalin valkoiseksi (Kuva 19): Tyylittelin jokaisen kohdan erikseen tällä tavalla, jotta loppu tulos olisi mahdollisimman yhdenmukainen. Toimeksiantaja muokkaa CSS-tyylejä sopimaan heidän sivustonsa mukaiseksi ennen julkaisua. Tein myös tabletti- ja mobiililaitteille sivuston responsiiviseksi, aivan kuten meille on opetettu jo ensimmäisenä vuotena.

Ansku Ankka, ajanvaraaminen

Siirry takaisin tarkastelemaan aikoja

Etunimi*

Sukunimi*

Katuosoite*

Postinumero*

Puhelinnumero*

Sähköposti*

Asiointikieli Suomi Ruotsi

Lisätietoja

Lähetä

Kuva 19 Ajanvarauslomake (Omakuva 2023)

Hallintapaneelin tyyllittely tapahtui viimeisenä. Figmalla oli hahmoteltu miltä se kuulisi näyttää. Yleiskäytettävyys oli tärkein. Käytin hallintapaneelin puolen lomakkeissa samoja tyyliä, kuin käyttäjällä. Tämä oli helpoin ja tyylikkään ratkaisu, joka samalla varmisti toimivuuden eri laitteilla. Se samalla varmisti, että tyyllittely oli käyttäjän kannalta mieleistä ja mukava käyttää (Kuva 20). Muihin, kuten uusien aikojen listaamisen kanssa, tein täysin omat tyylit, koska se poikkeisi olemassa olevista tyyleistä niin paljon. Lopuksi tyyliä testattiin myös oikealla käyttäjällä. Oikealla käyttäjällä testaaminen varmisti, että ajan lisäystä oli helppo ja mukava käyttää.

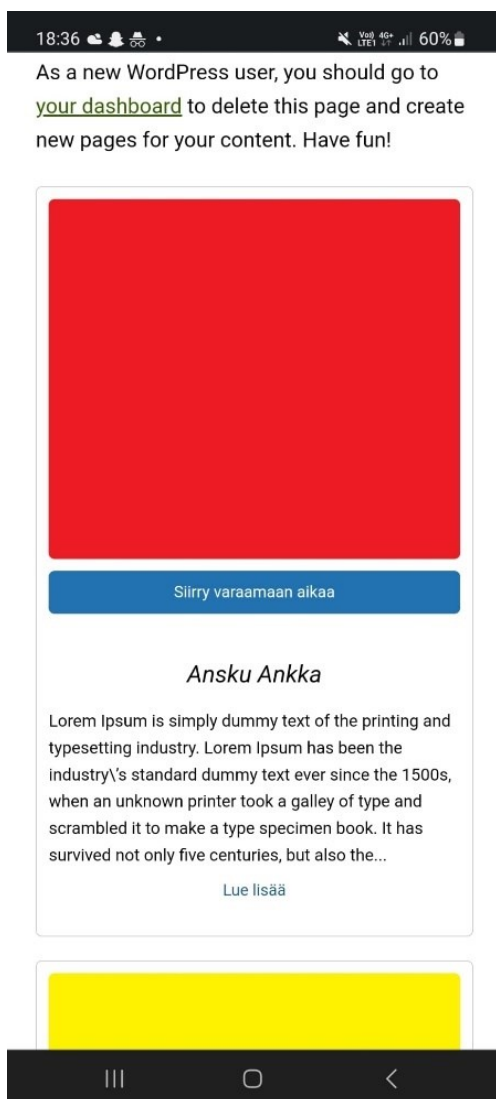
Muokkaa tietoja

Koko nimi	<input type="text" value="Ansku Ankka"/>
Kuvaus	<div style="border: 1px solid #ccc; padding: 5px; min-height: 50px;">Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry\</div>
Sähköposti	<input type="text" value="testi@gmail.com"/>
Kuva	<input type="button" value="Browse..."/> No file selected.
Kuvateksti	<input type="text" value="Kuva minusta"/>

Kuva 20 Kuva tyylielystä hallintapaneelistä (Omakuva 2023)

Viimeisenä oli responsiivisuuden koodaaminen. Se tapahtui CSS mediakyselyjen avulla, jotka tunnistavat käyttäjän ruudun kokoa ja käyttävät mukautettuja tyyliä niiden perusteella. Tämä suunnittelutapa varmistaa, että asettelu on optimoitu mobiililaitteille, mikä parantaa käyttökokemusta merkittävästi. Sivusto toimi loistavasti pöytäkoneella, mutta vaati hienosäätöä esimerkiksi puhelimella tai tabletilla käyttöä varten. Tätä varten loin erilliset komennot kutakin laitetyyppiä varten. Ensimmäisenä tein tabletilaitteiden mediakyselyt. Eli jos ruutu on suurempi kuin puhelin mutta pienempi kuin pöytätietokone nämä arvot tulevat käyttöön. Sivuston ulkoasu ei muutu tabletilaitteella paljoakaan, koska sivusto oli suunniteltu toimimaan jo valmiiksi pienemmillä laitteilla ja vastaamaan WordPressin sisällön kokoa. Etukäteen suunnittelu säästi hieman työmäärää tässä vaiheessa. Tein ainoastaan pieniä muutoksia sisällön kokoon. Testasin lopuksi sivuston käyttöä tabletilla todetakseni toimivuuden.

Viimeisenä tein mediakyselyt käyttäjille, jotka käyttävät verkkosivustoa älypuhelimella. Tiesin, että tässä on suurin työ, joten jätin sen viimeiseksi. Sivusto tunnistaa, jos laite on pienempi kuin pöytä-kone ja tabletti, jolloin nämä komennot tulevat voimaan. Tässä kortteja piti tulla yksi per rivi, koska korteissa oli niin paljon tietoa, ettei ollut järkevät yrittää kutistaa niitä pienemmiksi. Järjestämällä kortit yhteen sarakkeeseen, käyttäjät voivat vaivattomasti selata sisältöä, ja tekstiä mukautetaan sopivaksi pienemmälle näytölle, mikä tekee siitä helpommin saavutettavissa olevan ja luettavamman (Kuva 21). Lisäsin myös ponnahdus ikkunat ja modaali-ikkunat skaalautumaan koko ruudun leveydelle, jolla sain syöttökentille enemmän kirjoitustilaa.



Kuva 21 Puhelimesta kuvakaappaus (Omakuva 2023)

Luomalla nämä erilliset CSS-komennot verkkosivusto voi mukautua sulavasti eri näyttökokoihin, mikä tarjoaa paremman asettelun ja luettavuuden. Mediakyselyillä on keskeinen rooli käyttäjän näytön mittojen tunnistamisessa, jolloin verkkosivusto voi muokata ulkoasuun vastaavasti.

3.5 Saavutettavuus

Saavutettavuus oli yksi tärkeimmistä projektin kohteista. Sen viimeistely jäi projektin loppupuolelle. Tarkoituksena oli tehdä mahdollisimman monelle eri tarpeisille käyttäjille sopiva ajanvarauskalenteri, jotta ajanvaraaminen ei jää sivuston saavutettavuuden takia tekemättä. Projektin aikana olinkin jo tutustunut hyvin WAI:n ja EU:n suosituksiin ja standardeihin.

Ruudunlukijat ovat keskeinen apuväline näkövammaisille käyttäjille, jotka eivät voi nähdä näyttöä tai muuta visuaalista sisältöä. Ruudunlukijat lukevat ääneen verkkosivustojen sisältöä, jolloin näkövammaiset käyttäjät voivat kuunnella sivuston tiedot ja navigoida sen eri osien välillä. Tämä mahdollistaa heille täyden pääsyn verkkosisältöihin. Siksi pidin erittäin tärkeää huolta ruudunlukijan toiminnasta lisäosassani. Kokeilin itse käyttää ajanvarausta Windowsin oman ruudunlukijan kanssa. Tällä tavalla sain paremman käsityksen lähtötilanteesta.

Koodaamisen aloitin varmistamalla, että perusasiat ovat kunnossa joka sivulla. Ensimmäisenä tarkistin, että kuvilla on omat *alt*-tekstit. Ruudunlukija lukee *alt*-tekstin, kun se pääsee sen kohdalle, joten se auttaa ymmärtämään kuvan sisällön. Seuraavaksi siirryin tutkimaan näppäimistöllä navigointia, koska osalla ei välttämättä ole mahdollista käyttää hiirtä, joten sivulla navigointi pelkästään näppäimistön kanssa piti olla mahdollista. Ensimmäisenä tarvitsin yksittäisen kortin lukemisen ja niiden välillä liikkumisen. Tein navigoinnin nuolin näppäimien avulla. Nuolinäppäintä alaspäin painaessa ruudunlukija lukee ensimmäisen kortin kuvan, uudestaan painaessa ruudunlukija lukee vaaruspainikkeen (Kuva 22).

```
// Event listener for arrow key navigation
document.addEventListener('keydown', (event) => {
  switch (event.key) {
    case 'ArrowUp':
      event.preventDefault();
      navigateAppointmentCards(-2); // Move to the card previous card.
      break;
    case 'ArrowDown':
      event.preventDefault();
      navigateAppointmentCards(2); // Move to the card next row.
      break;
    case 'Enter':
      openModalForCurrentCard();
      break;
    default:
      break;
  }
});
```

Kuva 22 Nuolinäppäimillä liikkuminen (Omakuva 2023)

Tästä halutessaan *enteriä* painamalla pääsee tarkastelemaan vapaana olevia aikoja (Kuva 23). Jos kuitenkin käyttäjä päättää painaa uudelleen alaspäin sivu lukee kuvauskentän tekstin, jonka jälkeen alaspäin painamalla pääsee seuraavan henkilön korttiin. Käyttäjän löytäessä sopivan, hän valitsee sen painamalla *enteriä*, jonka jälkeen sivusto lukee vielä kerran henkilön nimen. Sivusto ilmoittaa milloin modaali-ikkuna on ladattu. Esimerkiksi hitaamman netin kanssa tämä ominaisuus on kätevä.

```
// Function to open the modal for the currently focused card.
function openModalForCurrentCard() {
  const activeCard = document.querySelector('.appointment-card.active');
  if (activeCard) {
    const openModalButton = activeCard.querySelector('.openModal');
    if (openModalButton) {
      openModalButton.click();
    }
  }
}
```

Kuva 23 Kortin valinta Enterillä (Omakuva 2023)

Seuraavaksi siirryin tutkimaan modaali-ikkunaa. Heti ensimmäisenä huomasin, että ruudunlukija ei siirtynyt lukemaan sitä, kun ikkuna aukaistaan. Korjasin sen käyttämällä JavaScriptin *focus()* on metodia, jota käytetään asettamaan näkyvä kohdistus tai fokus tietylle HTML-elementille verkkosivulla. Asettamalla kohdistuksen elementille, se tarkoittaa, että elementti saa huomion ja voi vastaanottaa käyttäjän syötteitä, kuten näppäimistön toiminnot tai klikkaukset Tämä on erityisen hyödyllistä käytettäessä näppäimistöä, koska käyttäjä voi olla tekemisissä heti sen jälkeen kyseisen elementin kanssa, eikä hänen tarvitse klikata tai painaa erikseen ikkunaa. Lisäsin myös *role*-attribuutti toiminnon. Modaali-ikkunan yhteydessä *role*-attribuutti kertoo, että kyseessä on modaali-ikkuna. Samalla käytin *aria-labelledby*-attribuuttia yhdistääkseni ikkunan sen otsikkoon, jolloin käyttäjille selviää, mistä modaali-ikkunassa on kyse. Ruudunlukijat ilmoittavat otsikon ääneen, kun modaali-ikkuna avataan, mikä helpottaa käyttäjien ymmärtämistä.

Tässä vaiheessa lisäsin ruudunlukijan käyttöä helpottamaan *aria-hiddenejä* paikkoihin, joita ruudunlukijan ei tarvitse lukea. Se selkeyttää sisällönrakennetta. Seuraavaksi tarvitsin navigoinnin ensimmäiseen vapaana olevaan aikaan. Nuolinäppäintä alas painamalla sivusto lukee ensimmäisen ajan kohdan. Jos aikoja ei ole sivusto ilmoittaa siitä. Sopivan ajan on löytyessä *enteriä* painamalla pääsee varauslomakkeelle.

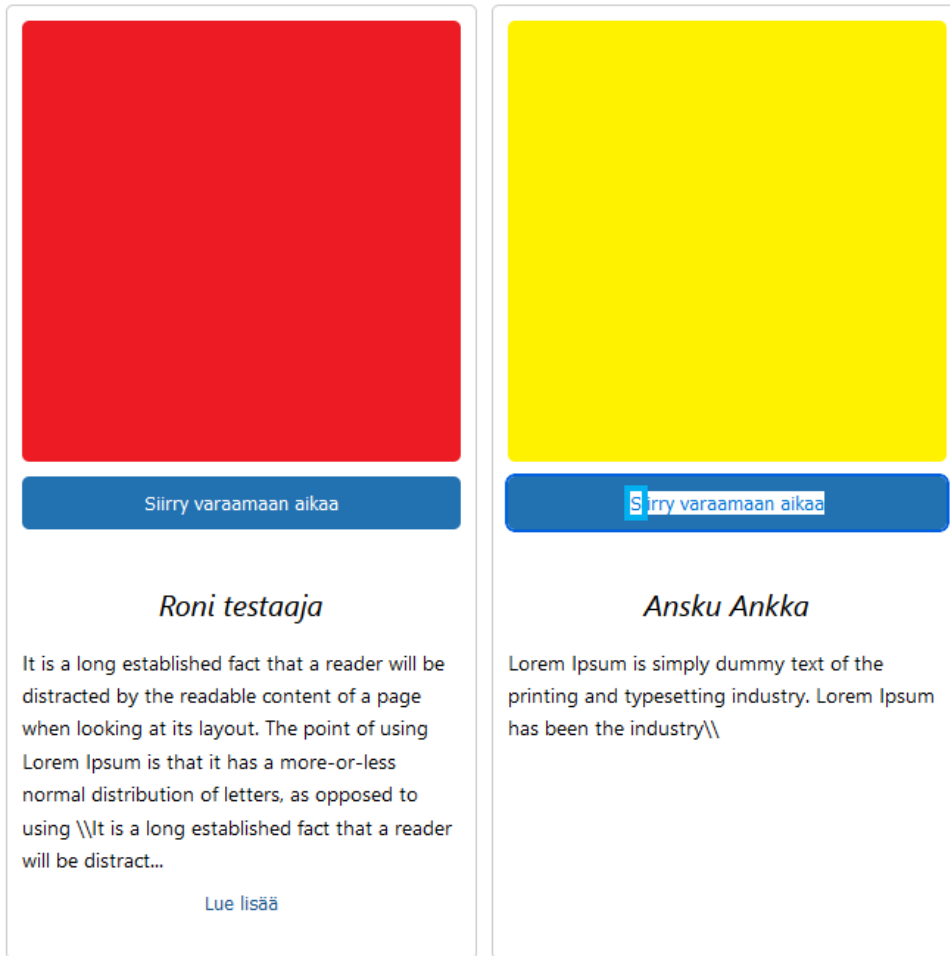
Varauslomakkeen lataamisen jälkeen, ruudunlukija ilmoittaa siitä käyttäjälle. Nuolinäppäintä painamalla alaspäin pääsee ensimmäiseen kenttään, jolloin sivusto kertoo sen nimen eli ”etunimi” ja sen että se on pakollinen kenttä. Kentän nimen lukeminen lisää saavutettavuutta ja se tapahtui suhteellisen helposti käyttämällä labelin sisällä for-attribuuttia ja syöttökentässä id samalla nimellä. Näin ruudunlukija osaa yhdistää nämä kaksi. Pakollinen kenttä on tehty *required*-attribuutilla. Tästä ei pääse alaspäin painamalla eteenpäin ennen kuin se on täytetty. Tein näin jokaisen syöttökentän kohdalla. Radio nappuloita varten jouduin käyttämään erikseen ruudunlukijaa varten *field* saraketta sitomaan kaksi nappia yhteen toistensa kanssa. Käyttäjän täytettyä kaikki pakolliset kentät hän voi edetä varaamaan ajan. Varaamisen jälkeen ruudunlukija kertoo varauksen onnistumisesta.

Tarvitsin myös, tavan, miten modaali-ikkunan voi sulkea helposti näppäimistön avulla. Modaali-ikkunan yläreunasta löytyy ruksi mistä voi sulkea ikkunan, sen käyttäminen näppäimistön avulla on hankalaa, joten tarvitsin erinlaisen ratkaisun. Päädyin toteuttamaan tämän yksinkertaisella koodin pätkällä, joka ymmärtää, kun *esc*-näppäintä on painettu, se sulkee modaali-ikkunan (Kuva 24). Sivusto ilmoittaa siitä vielä ruudunlukijalle, jolloin käyttäjän on helpompi ymmärtää missä kohdassa sivustoa hän on menossa.

```
document.addEventListener('keydown', (event) => {  
  if (event.key === 'Escape') {  
    modal.hide();  
  }  
});
```

Kuva 24 Modaali-ikkunasta poistuminen (Omakuva 2023)

Seuraava vaihe olisi ehdottomasti testata ulkopuolisilla tätä. Muiden testausten avulla sain paremman käsityksen käytettävyydestä, koska se auttaisi tunnistamaan ja korjaamaan esteet ja puutteet, joita saavutettavuus voi aiheuttaa. (Kuva 25) Näin varmistaisin, että kaikki käyttäjät voivat nauttia ajanvaraussivuston tarjoamista eduista.



Kuva 25 Ruudunlukijan testaamista (Omakuva 2023)

3.6 Lisäosan julkaiseminen

WordPressin omassa kaupassa lisäosan julkaiseminen on tärkeä vaihe kehittäjälle, joka haluaa jakaa luomansa lisäosan laajalle käyttäjäkunnalle. Siinä on kuitenkin tiettyjä vaiheita ennen kuin julkaisu onnistuu.

Ennen kuin kehittäjä voi julkaista lisäosan WordPressin omassa kaupassa tarvitaan rekisteröitytty WordPress tili. Rekisteröityminen tapahtuu WordPressin sivuilta, jonka jälkeen saa kehittäjätilin. Tili mahdollistaa lisäosien julkaisemisen ja hallinnoinnin. Ennen julkaisua on tärkeää varmistaa, että lisäosa on valmis käyttöönottoon. Tässä vaiheessa tehdään viimeiset tarvittavat testaukset ja varmistetaan, että lisäosa on toiminnallisesti ja teknisesti valmis julkaistavaksi. Tässä vaiheessa kannattaa myös tarkistaa, että lisäosa noudattaa WordPressin käytäntöjä. Lisäosan dokumentaatio on tärkeä osa julkaisemista: Lisäosasta kuuluu laatia selkeät dokumentaatiot, sekä ohjeet. Tämä sisältää ohjeet lisäosan asentamiseen, käyttöön ja konfiguraation (WordPress 2023d.)

Hyvä dokumentaatio on tärkeä, koska se auttaa käyttäjää asentamaan ja helpottaa lisäosan käyttöönottoa. Kannattaa myös varmistaa, että dokumentaatiossa löytyy lisäosan yhteensopivuustiedot. Lisäosan pakataan ZIP muotoon. Paketti sisältää kaikki lisäosan tiedostot. Paketin rakenteen kuuluu noudattaa WordPressin vaatimuksia ja lisäksi kaikkein tarvittavien tiedostojen tulee olla mukana. Lisäosan lähettäminen tarkistettavaksi lähtee siitä että, kirjaudutaan kehittäjätilille ja napsautetaan "Submit your Plugin" -linkkiä. Sivusto pyytää täyttämään tarvittavat tiedot, kuten lisäosan nimi, kuvaus, toiminnallisuudet, muut tiedot ja ZIP-paketin, minkä jälkeen lisäosa ladataan sivulle ja lähetään tarkistettavaksi WordPressin tiimille. Seuraavana odotetaan tarkistusprosessin valmistumista, jossa WordPressin tiimi käy läpi lisäosan ja varmistaa, että se noudattaa kaikkia julkaisuvaatimuksia. Tarkistusprosessi voi kestää joitakin päiviä tai viikkoja riippuen jonoista ja tiimin kuormituksesta. Lisäosan on läpäistessä tarkistusprosessin, WordPressin tiimi lähettää sähköpostitse tiedon hyväksynnästä ja lisäosa julkaistaan WordPressin omassa kaupassa, jossa käyttäjät voivat löytää ja asentaa sen sivustoilleen (WordPress 2023d.)

Tämän kirjoittamisen aikana lisäosan dokumentaatio ei ole vielä valmis, joten julkaisu WordPressin kaupassa jää myöhemmälle ajankohdalle.

4 JOHTOPÄÄTÖKSET

Kalenterin kuului sisältää useiden työntekijöiden tiedot, henkilöiden ja olemassa olevien aikojen hallinta sekä asiakkaille ajanvarauskalenteri. Työntekijöiden esittely ja ajanvarauskalenteri tulisi olla helppo sisällyttää WordPress sivulle. Ajanvaraaminen pitäisi olla mahdollisimman vaivatonta, joka auttaisi palvelun saatavuuden kanssa. Ulkonäön täytyisi olla selkeä ja suhteellisen helposti muokattavissa CSS koodilla. Saavutettavuus täytyisi olla hyvällä tasolla.

Vaihtoehtoja tutkittiin WordPressin oman kaupan sisältä sekä ulkopuolisilta palvelun tarjoajilta. Testasin myös käytännössä muiden lisäosia niin paljon kuin niitä oli mahdollista testata. Yleisesti paremmat vaihtoehdot löytyivät kaupan ulkopuolelta. Niissä oli enemmän ominaisuuksia ja mahdollisuuksia räätälöidä sitä enemmän omaa käyttötarkoitusta varten. Vaikka hyviäkin vaihtoehtoja löytyi, tässä päädyttiin rakentamaan oma ajanvarauskalenteri, koska toimeksiantajan tarpeisiin soveltuvaa ajanvarauskalenteria ei löytynyt. Vaihtoehtoista parhaimmat olivat yleensä hyvin rajoittuneet ominaisuuksien ilmaisissa versioissa. Tärkeät lisäominaisuudet, kuten monen henkilön järjestelmä ja muut ominaisuudet tarkoittivat yleensä suhteellisen kalliin kuukausimaksun maksamista. Kyseessä oli pieni sivutoiminen yritys, kulut haluttiin pitää mahdollisimman alhaalla.

Lisäosan rakentaminen onnistui hyvin minun mielestäni. Sain tehtyä yhdessä sovitut ominaisuudet mutta jatkokehittämiselle jäi kuitenkin varaa. Lisäosan ulkonäkö on tyylikäs mutta yksinkertainen, joka ennen kaikki skaalautuu jokaiselle laitetyypille. Saavutettavuus saatiin halutulle tasolla, joka mahdollistaa lisäosan käytön mahdollisimman monelle.

5 POHDINTA

Opin huomattavan paljon WordPress-ympäristöstä ja siellä työskentelystä. Esimerkiksi PHP:n kanssa tuli jotakin yllätyksiä ongelmia, kun toiminnoista ei suoranaisesti toiminut WordPressissä, kuten sessiot. Näihin WordPress oli kehittänyt omia ratkaisuja, joita tuli käyttää. Opin myös paremmin frontend ja backend yhteyden ja niiden välillä tiedon liikuttamisen.

Pääsin projektin aikana ensimmäistä kertaa työskentelemään saavutettavuuden parissa kunnolla. Tutustuin uusiin käsitteisiin ja standardeihin. Opin todella paljon uutta saavutettavuudesta, ja siitä miten se vaikuttaa sivuston rakentamisessa. Ruudunlukijan käyttäminen kunnolla oli hyvin mielenkiintoinen kokemus, joka auttoi ymmärtämään paremmin miltä sivusto näyttää käyttäjältä, joka tarvitsee teknologiaa päivittäin käyttämiseen.

Kehittämiskohtakin löytyy varmasti. Ensimmäisenä tekisin hallintapaneelin etusivun, jossa on tarkemmat asennusohjeet ja lisäosan toiminta periaatteen. Siihen voisi myöhemmässä vaiheessa lisätä lisäosan värien vaihtamisohjeet tai jotain muuta. Sen lisäksi ehdottomasti useammille kielelle kääntäminen olisi tärkeää, jotta lisäosa saavuttaisi mahdollisimman monen eri ihmisen tarpeet. Lisäosaan voisi lisätä erillisen uuden sivun, jossa olisi mahdollisuus vaihtaa otsikkojen ja muiden tekstien nimiä suoraan käyttäjän haluamaksi. Näillä lisäyksillä lisäosasta tulisi huomattavasti houkuttelevampi, koska toista samanlaista ei löydy tämän pohdinnan kirjoittamis- hetkellä WordPressin omasta kaupasta.

Näiden muutosten lisäksi tekisin hallintapaneeliin olemassa olevien aikojen hallinnan. Hieman samaan tyyliin kuin henkilöjen tietojen tarkistaminen, mutta sen sijaan työntekijä voisi vaihtaa olemassa olevan ajan ajankohdan. Nyt aika täytyy poistaa erikseen tietokannasta ja luoda uusi aika. Hoksasin tämän vasta tekovaiheessa, eikä aika valitettavasti riittänyt sen lisäämiseen jälkikädessä.

Viimeisenä lisäisin yhden sivun hallintapaneeliin, josta voisin säätää lisäosan värejä, fontteja ja niiden kokoja. Tällä tavalla käyttäjä saisi kätevästi tehtyä lisäosasta oman näköisen eikä koodien muuttaminen olisi tarpeen.

LÄHTEET

Aluehallintovirasto 2023. Yleistä saavutettavuudesta. Hakupäivä 13.4.2023.

<https://www.saavutettavuusvaatimukset.fi/yleista-saavutettavuudesta/>

Beaulieu, Alan 2005. Learning SQL. Kustannusyhtiö O'Reilly media, inc. Hakupäivä 31.5.2023

<https://learning.oreilly.com/library/view/learning-sql/0596007272/ch01.html#learningsql-CHP-1-SECT-2>

Bin Uzayr, Sufyyan & Rooney, Matthew 2016. Learning WordPress REST API. Kustannusyhtiö Packt Publishing. Hakupäivä 7.6.2023

<https://learning.oreilly.com/library/view/learning-wordpress-rest/9781786469243/?ar%2F%3Forpq=&email=%5Eu>

Bos, Bert 2016. A brief history of CSS until 2016. Hakupäivä 11.5.2023.

<https://www.w3.org/Style/CSS20/history.html>

Dalal, Anuj 2020. What is REST API. Hakupäivä 22.8.2023

<https://www.zestard.com/blog/rest-api-benefits/>

Devmountain.com 2023. What is CSS and Why Should You Use it? Hakupäivä 25.5.2023

<https://devmountain.com/blog/what-is-css-and-why-use-it/>

Dickson, Boateng 2022. A Brief History of JavaScript. Hakupäivä 21.8.2023

<https://dev.to/dboatengx/history-of-javascript-how-it-all-began-92a>

EUR-Lex.eu 2016. Directive (EU) 2016/2102 of the European Parliament and of the Council of 26 October 2016 on the accessibility of the websites and mobile applications of public sector bodies (Text with EEA relevance). Hakupäivä 25.5.2023

<https://eur-lex.europa.eu/eli/dir/2016/2102/oj>

Friedman, Vitali 2018. Hakupäivä 16.5.2023

<https://www.smashingmagazine.com/2011/01/guidelines-for-responsive-web-design/>

Github.com 2022. The top programming languages. Hakupäivä 23.8.2023

<https://octoverse.github.com/2022/top-programming-languages>

Huttunen, Kaisa 2021. Mikä on WordPress? Hakupäivä 17.4.2023.

<https://www.zoner.fi/wordpress/mika-on-wordpress/>

Kyberturvallisuuskeskus 2020. Tietoturva. Hakupäivä 20.4.2023

<https://www.kyberturvallisuuskeskus.fi/fi/toimintamme/saantely-ja-valvonta/tietoturva>

Mozilla Foundation 2023. JavaScript. Hakupäivä 24.8.2023

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

Lawton, Shawn 2022. Introduction to Web Accessibility. Hakupäivä 23.5.2023

<https://www.w3.org/WAI/fundamentals/accessibility-intro/#context>

Loihde Trust 2023. Suomalaisten organisaatioiden tietoturva 2023–2025. Hakupäivä 21.5.2023

<https://www.loihdetrust.com/uutiset/suomalaisten-organisaatioiden-tietoturva-2023-2025/>

Noble Desktop 2023. What is Figma? Hakupäivä 20.5.2023

<https://www.nobledesktop.com/learn/figma/what-is-figma>

One.com 2023. Mikä on plugin. Hakupäivä 13.4.2023.

<https://www.one.com/fi/wordpress/wordpress-plugins>

Peda.net 2022. CSS. Hakupäivä 14.4.2023.

<https://peda.net/kotka/perusopetus/kotkansaaren-koulu/kt/oppiaineet/tietotekniikka/ohjelmointi/html/css>

PHP.net 2023. What is PHP? Hakupäivä 17.4.2023.

<https://www.php.net/manual/en/intro-what-is.php>

Piispanen, Tuomas 2021. Figma – Paras työkalu käyttöliittymien suunnitteluun. Hakupäivä

13.4.2023. <https://webguru.fi/figma/>

Sheldon, Robert 2023. PHP (Hypertext Preprocessor). Hakupäivä 24.5.2023

<https://www.techtarget.com/whatis/definition/PHP-Hypertext-Preprocessor>

Statcounter 2023. Desktop vs Mobile vs Tablet Market Share Worldwide – April 2023. Hakupäivä

15.5.2023

<https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/worldwide/2022>

Suovesi, Sami 2022. Mitä tarkoittaa responsiivisuus, ja miksi se on tärkeää? Hakupäivä

18.4.2023. <https://sininharka.fi/mita-tarκοittaa-responsiivisuus-ja-miksi-se-on-tarkeaa/>

TIMIFY.com 2023. A state-of-the-art shader calendar for professional teams. Hakupäivä

21.4.2023. <https://www.timify.com/fi/features/online-calendar-for-teams/>

Vello.fi 2023a. The easy online booking tool you and your customers love. Get started now!

Hakupäivä 20.4.2023. <https://www.vello.fi/>

Vello.fi 2023b. Kuva 4. Hakupäivä 26.5.2023 <https://www.vello.fi/en/>

W3c 2009. WAI early days. Hakupäivä 22.5.2023 <https://www.w3.org/WAI/history>

W3c 2022. CSS Snapshot 2022. Hakupäivä 18.5.2023 <https://www.w3.org/TR/css-2022/>

W3Techs 2023. Usage statistic and market share of WordPress Hakupäivä 17.4.2023.

<https://w3techs.com/technologies/details/cm-wordpress>

Walkowski, Debbie 2019. What Is the CIA Triad? Hakupäivä 22.8.2023

<https://www.f5.com/labs/learning-center/what-is-the-cia-triad>

WordPress 2023a. Appointment Hour Booking – WordPress Booking Plugin. Hakupäivä 23.4.2023.

<https://wordpress.org/plugins/appointment-hour-booking/>

WordPress 2023b. Booking calendar contact form. Hakupäivä 20.4.2023.

<https://wordpress.org/plugins/booking-calendar-contact-form/>

WordPress 2023c. Lisäosat. Hakupäivä 16.4.2023.

<https://fi.wordpress.org/plugins/>

WordPress 2023d. Planning, Submitting, and Maintaining Plugins. Hakupäivä 25.7.2023

<https://developer.wordpress.org/plugins/wordpress-org/planning-submitting-and-maintaining-plugins/>