

# **Verkkosivun kehittäminen WordPressillä**

**Työparina tekoäly**

LAB-ammattikorkeakoulu

Insinööri (AMK), Tieto- ja viestintätekniikka

2025

Outi Liukkonen

## Tiivistelmä

Tekijä(t)	Julkaisun laji	Valmistumisaika
Outi Liukkonen	Opinnäytetyö, AMK	2025
	Sivumäärä	
	32	
Työn nimi		
<b>Verkkosivun kehittäminen WordPressillä</b>		
Työparina tekoäly		
Tutkinto ja koulutusala		
Insinööri (AMK), Tieto- ja viestintätekniikka		
Toimeksiantajaorganisaatio (jos opinnäytetyöllä on toimeksiantaja)		
Lähiruuan Ystävät Ry		
Tiivistelmä		
<p>Opinnäytetyönä toteutettiin WordPress-verkkosivusto, joka kokoaa kaakkoissuomalaiset lähiruokaryitykset, ravintolat ja maatilamatkailukohteet yhteen karttapohjaiseen palveluun. Kevyellä, mobiiliystävällisellä verkkosivulla käyttäjä voi suodattaa kohteita kartalle ja esittelykortteihin avainsanojen avulla.</p> <p>Tavoitteena oli uudistaa aiemmin käytössä ollut verkkosivu nykyaikaiseksi ja luoda helpokäyttöinen digitaalinen alusta paikallisille lähiruokatoimijoille. Sivusto rakennettiin WordPressin, HTML:n, CSS:n ja JavaScriptin avulla sekä neljää ilmaista lisäosaa hyödyntäen. Kehitysprosessin tukena käytettiin tekoälyä, erityisesti ChatGPT:tä ja GitHub Copilotia, jotka toimivat työparina suunnittelussa, koodauksessa ja ongelmanratkaisussa.</p> <p>Projektin tuloksena syntyy toimiva verkkopalvelu, jossa on dynaamiset yrityseshittelyt ja suodattava karttakäyttöliittymä. Työn aikana kehittäjän tekninen osaaminen ja taito hyödyntää tekoälytyökaluja kasvaa merkittävästi. Kokemuksen perusteella tekoäly osoittautuu tehokkaaksi työpariksi myös silloin, kun kehittäjän ohjelmointiosaaminen on aloittelijan tasolla.</p>		
Asiasanat		
Verkkosivut, WordPress, tekoäly, ChatGPT, GitHub Copilot		

## Abstract

Author(s)	Type of Publication	Published
Outi Liukkonen	Thesis, UAS	2025
	Number of Pages	
	32	
Title of Publication		
<b>Developing a Website with WordPress</b>		
In Collaboration with Artificial Intelligence		
Degree, Field of Study		
Engineer (UAS), Information and Communications Technology		
Organisation of the client (if the thesis work is commissioned by another party)		
Lähiruuan Ystävät Ry		
<p>This thesis project involved the development of a WordPress-based website that brings together local food producers, restaurants, and agritourism destinations from Southeast Finland into one map-based service. The lightweight, mobile-friendly site allows users to filter businesses displayed on the map and in company cards using keyword tags.</p> <p>The goal was to modernize an outdated website and create an easy-to-use digital platform for local food producers and sellers. The site was built using WordPress, HTML, CSS, and JavaScript, alongside four free plugins. The development process was supported by artificial intelligence, particularly ChatGPT and GitHub Copilot, which acted as collaborative tools in design, coding, and problem-solving.</p> <p>As a result, a functional web service was created with dynamic company profiles and a filterable map interface. During the project, the developer's technical skills and ability to utilize AI tools improved significantly. The experience demonstrated that AI can be an effective partner even when the developer has beginner-level programming knowledge.</p>		
Keywords		
Website development, WordPress, Artificial intelligence, ChatGPT, GitHub Copilot		

## Sisällys

1	Johdanto.....	1
2	Verkkosivualusta ja lisäosat.....	2
2.1	WordPress, sisällönhallintajärjestelmä .....	2
2.2	All in One WP Migration, työkalu sivuston siirtoon ja varmuuskopiointiin .....	3
2.3	Astra, WordPress teema .....	4
2.4	Advanced Custom Fields, mukautettuja kenttiä ja sisältötyyppejä .....	5
2.5	GEO my WP, lisäosa geopaikannukseen .....	6
2.6	Karttakirjastot, ohjelmistokirjastot interaktiivisten karttojen luomiseen .....	7
2.7	Leaflet Map, kevyt ja pelkistetty karttakirjasto .....	8
3	Kehitysympäristöt .....	9
3.1	Local, paikallinen kehitysympäristö WordPress sivujen kehitykseen.....	9
3.2	Visual Studio Code, koodieditori .....	9
4	Tekoäly kollegana verkkosivukehityksessä.....	11
4.1	ChatGPT, keskusteleva tekoäly.....	11
4.2	GitHub Copilot, tekoäly ohjelmoinnin tueksi.....	12
4.3	Tekoäly ohjelmoijan työparina .....	13
5	Toteutus ja kehitysprosessi.....	15
5.1	Verkkosivun rakenne ja reunaehdot .....	15
5.2	Verkkosivun kehitysvaiheet ja tekoälyn rooli kehityksessä.....	15
5.2.1	Companies, mukautetun sisältötyypin luominen .....	16
5.2.2	Headerin ja etusivun asetellut.....	20
5.2.3	Karttaosion kehitys .....	21
5.2.4	Yritysten suodatus .....	22
5.2.5	Sivun viimeistely .....	23
6	Yhteenveto ja pohdinta .....	26
	Lähteet .....	29

Liite 1. ChatGPT:n luoma ensimmäinen suunnitelma verkkosivua varten

Liite 2. ChatGPT:n päivittämä suunnitelma, kun verkkosivu on saatu siirrettyä internettiin

## 1 Johdanto

Tämän opinnäytetyön tavoitteena on uudistaa Makuja kaakosta -niminen verkkosivusto. Projektissa käytetään tekoälyä sivuston rakenteen ja lähdekooditiedostojen suunnitteluun, lisäosien valitsemiseen ja asentamiseen sekä ohjelmistokoodien kirjoittamiseen ja virheiden korjaamiseen. Tavoitteena on, että tekoälyn käyttö mahdollistaa halutunlaisen sivuston toteuttamisen siitäkin huolimatta, että ohjelmoijalla itsellään ei ole vahvaa ammattiosaamista tämällyyppisistä ohjelmointiprojekteista. Työn edetessä seurataan myös, miten työn toteuttajan ymmärrys tekoälyn hyödyntämisestä kehittyy ja muodostuuko tekoälystä ohjelmoijalle luonteva työpari myös tulevaisuuden projekteihin.

Makuja kaakosta on verkkosivu, joka kokoaa yhteen kaakkoissuomalaisia lähiruokatuottajia, myyjiä, ravintoloita ja maatilamatkailukohteita. Verkkosivu on luotu vuonna 2013, ja sen taustalla oli tuolloin edistyksellinen ajatus yhdistää lähiruoka ja matkailu navigointimahdollisuuteen sekä tarjota käyttäjälle hakutoimintoja hyödyntäen mahdollisimman osuvia yritys-esittelyjä. Viime vuosina sivustoa ei ole kehitetty, ja se kaipasi sisällön sekä teknologian päivitystä. Sivustoa ylläpitää yhdistys nimeltä Lähiruuan Ystävät, joka toimii Kymenlaakson alueella ja edistää lähiruuan tunnettavuutta yhdistämällä lähiruuan tuottajat, jälleenmyyjät ja kuluttajat. Makuja kaakosta -verkkosivun ylläpidon lisäksi yhdistys järjestää vuosittain lähiruokaa markkinoivia tapahtumia, kuten Lähiruokamessut.

Makuja kaakosta -verkkosivu on luotu käyttäen WordPress-sisällönhallintajärjestelmää. Lähtökohtana sivun uudistukselle on, että sivu rakennetaan uudestaan olemassa olevaan verkkohotelliin käyttäen edelleen WordPressiä. Toteutuksessa painotetaan ylläpidon helppoutta, jotta sisällönhallinta onnistuu myös henkilöltä, jolla ei ole teknistä taustaa tai aiempaa kokemusta verkkosivujen kehittämisestä. Sivuston keskeinen toiminnallisuus on interaktiivinen kartta, jonka avulla käyttäjä voi paikantaa alueellisia lähiruokayrityksiä ja suodattaa niitä avainsanojen perusteella. Koska verkkopalvelu tuotetaan yhdistykselle, asettaa tämä verkkosivun teknisille ratkaisuille erityisiä vaatimuksia.

Merkittävin tekninen vaatimus on, että kaikki käytetyt työkalut, teemat ja lisäosat ovat käytettävissä ilman kustannuksia. Esimerkiksi Googlen karttapalvelu jätetään toteutuksesta pois, sillä sen käyttö edellyttää maksukorttitietojen syöttämistä Googlen kehittäjätiliin. Yhdistystoiminnassa ei ole mielekäästä käyttää verkkosivuston teknisiin ratkaisuihin yksittäisen henkilön maksukorttitietoja. Tällainen toimintatapa aiheuttaisi hallinnollisia haasteita kyseisen henkilön poistuessa yhdistyksen toiminnasta. Lisäksi on riski, että kerran verkkosivuun liitettyltä kortilta voidaan veloittaa maksuja automaattisesti. Näistä syistä verkkosivuston karttasovellukseksi valitaan avoimen lähdekoodin Leaflet Map -karttakirjasto, joka ei edellytä rekisteröitymistä eikä maksutietojen syöttämistä.

## 2 Verkkosivualusta ja lisäosat

### 2.1 WordPress, sisällönhallintajärjestelmä

Verkkosivujen ammattimainen kehitys hyödyntää olemassa olevia rakenteellisia ratkaisuja kuten kehityskehykset (frameworks) ja sisällönhallintajärjestelmät (CMS). Verkkosivut voidaan myös toteuttaa kirjoittamalla kaikki lähdekooditiedostot alusta asti itse, mutta tällainen lähestymistapa sopii lähinnä oppimistarkoitukseen. Kehitystavan valinnassa on otettava huomioon verkkosivun käyttäjät, kehittämiseen ja ylläpitoon käytössä oleva budjetti ja skaalautumisen tarve. Kehityskehykset ovat joustavia ja skaalautuvia, mutta vaativat ohjelmointiosaamista ja enemmän aikaa kehittämiseen. Sisällönhallintajärjestelmät ovat helposti lähestyttäviä, nopeita ottaa käyttöön eivätkä vaadi syvällistä ohjelmointiosaamista. Sisällönhallintajärjestelmät sopivat erityisesti pienille ja keskisuurille yrityksille, joiden vaatimukset verkkosivulle eivät ole erityisiä, vaan toteutettavissa sisällönhallintajärjestelmästä ja sen lisäosista valmiiksi löytyvillä työkaluilla. (Garderes 2024.)

WordPressin sisällönhallintajärjestelmä on yksi suosituimmista verkkosivualustoista maailmassa. WordPressia voidaan käyttää verkkosivujen, blogien ja verkkokauppojen luomiseen. WordPress sai alkunsa blogialustasta nimeltä b2/Cafelog, jonka Michel Valdrighi kehitti vuonna 2001. Sen tavoitteena oli tarjota käyttäjille helppo tapa julkaista omia ajatuksiaan verkossa. Valdrighin versiosta puuttui kuitenkin käyttäjille oleellisia ominaisuuksia kuten helppo asennettavuus, valmiiden ulkoasumallien muokattavuus ja mahdollisuus lisäosien asennukseen ja käyttöön. (Hossen 2023.) Näihin tarpeisiin vastauksena Mike Little ja Matt Mullenweg loivat oman versionsa b2/cafelogista ja näin WordPress sai virallisesti alkunsa (WordPress a).

WordPress on suunniteltu niin, että sitä voi käyttää sujuvasti tarvitsematta kirjoittaa yhtään riviä ohjelmistokoodia. Siitä huolimatta WordPress on monipuolinen alusta ja täynnä mahdollisuuksia myös ohjelmoijille. WordPress perustuu avoimeen lähdekoodiin. Se on rakennettu PHP-ohjelmointikielellä, ja se käyttää MySQL-tietokantaa. WordPressin kehittäjille on alusta asti ollut tärkeää, että käyttäjillä on vapaus

- käyttää ohjelmaa mihin tahansa tarkoitukseen
- tutkia, miten ohjelma toimii, ja muokata sitä omiin tarpeisiin sopivaksi
- levittää ohjelman kopioita edelleen
- jakaa muokattuja versioita muiden kanssa.

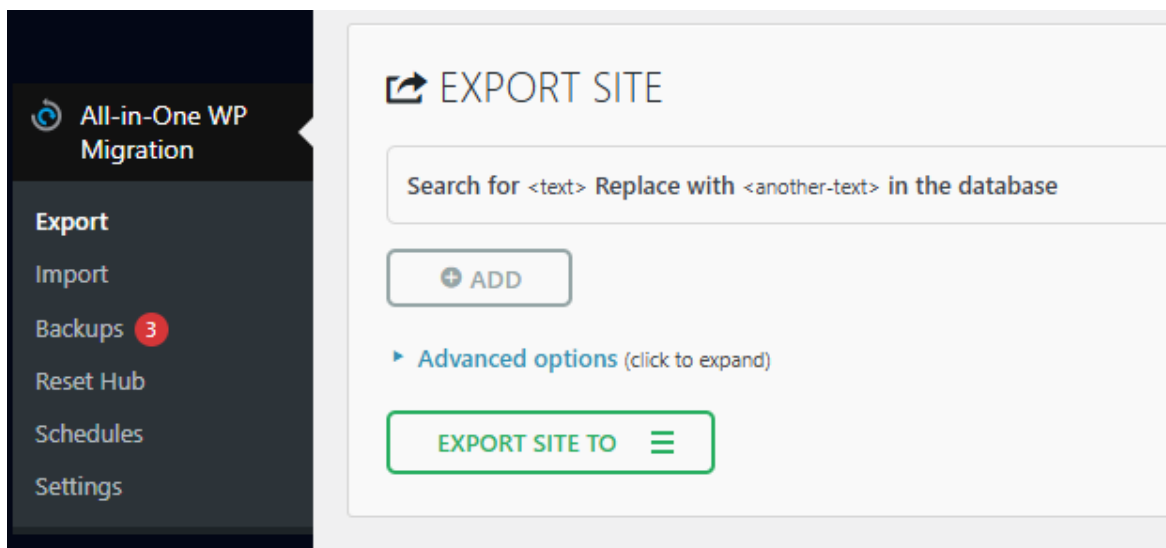
Näinhän ensimmäinenkin WordPresskin sai alkunsa yhdestä b2/Cafelogin kehityshaarasta. (WordPress a.)

WordPressia käyttää yli 43 % kaikista internetistä olevista verkkosivuista (WordPress a). Sitä käyttävät suuret mediatilat, kuten The New York Times Company ja New York Post, merkittävät julkishallinnon organisaatiot, kuten Nasa ja The White House, yliopistot, kulttuuritoimijat, verkkokaupat, vapaaehtoistoimijat, ravintolat ja monet muut aina pieniin yrityksiin ja henkilökohtaisiin blogeihin asti. (WordPress e.) WordPressin suosio perustuu yhtä paljon sen helppokäyttöisyyteen ja monipuolisiin ominaisuuksiin kuin sen ympärille vuosikymmenten aikana kehittyneeseen yhteisöön, joka tekee intohimoista kehitystyötä ja jakaa ideoitaan muidenkin käytettäväksi (WordPress a).

## 2.2 All in One WP Migration, työkalu sivuston siirtoon ja varmuuskopiointiin

All in One WP Migration on ServMaskin kehittämä migraatiotyökalu WordPressille. Se mahdollistaa verkkosivuston varmuuskopioinnin ja siirtämisen suoraan WordPressin hallintapaneelista. Ilman tätä lisäosaa varmuuskopiointi ja siirrot vaatisivat manuaalista työtä webhosting hallintapaneelissa. Migraatiotyökalun avulla verkkosivuston siirtäminen paikalliseen kehitysympäristöön tai takaisin tuotantoympäristöön tapahtuu helposti ilman tarvetta erilliselle FTP-ohjelmalle. All in One WP Migration -lisäosa tehostaa kehittäjän työskentelyä vähentämällä tarpeettomia vaiheita, kuten kirjautumista web-hosting hallintapaneeliin ja erillisten varmuuskopioiden käsittelyä. (ServMask 2025.)

All in One WP Migrationin -käyttöliittymä on käyttäjäystävällinen ja intuitiivinen (ServMask 2025). Siinä on erilliset osiot viennille (export), tuonnille (import), varmuuskopioille (backups), palautukselle (reset hubs), aikataulutukselle (schedules) ja asetuksille (settings) (Kuva 1). Vientiosiossa voi korvata tietyn sanan koko tietokannassa, valita lisäominaisuuksista esimerkiksi salasanasuojauksen varmuuskopioinnille tai mediatiedostojen pois jättämisen varmuuskopiosta. Export site to -valikosta voi valita useasta eri vaihtoehdosta, miten ja minne varmuuskopioinnin haluaa tehdä, kuten esimerkiksi tiedostoon (file), FTP-palvelimelle, Dropboxiin tai Google Driveen. Lisäosa tarjoaa siis sekä kehittyneemmälle, että vasta aloittelevalle kehittäjälle paljon mahdollisuuksia. Aloittelija voi luoda ja ladata vientiosiossa tiedostomuotoisen varmuuskopion ja tuoda sen takaisin tuontiosion raahaa ja pudota -toiminnolla. Edistyneempi käyttäjä voi automatisoida varmuuskopiointia aikatauluvälilehdellä ja määrittää varmuuskopioiden säilymisen asetuksia tarkemmin asetuksissa.



Kuva 1. All in One WP Migration -valikko WordPressin hallintapaneelissa

### 2.3 Astra, WordPress teema

WordPress tarvitsee aina teeman, joka määrittelee, miten sivuston sisältö visuaalisesti esitetään selaimessa. Teema pitää sisällään sivuston värit, fontit, palstoitusasetukset, sivupalkit ja esikatselukuvat. Teema koostuu erilaisista HTML-, PHP-, CSS-, JavaScript-, JSON- ja mediatiedostoista. (WordPress 2014.) Teeman voi tehdä itse, mutta on olemassa paljon valmiita teemoja, jotka ovat pitkälle kehitettyjä ja saatavana myös ilmaisversioina. Valmiin teeman koodiakin voi muokata itse. Tätä varten kannattaa aina luoda lapsiteema olemassa olevasta teemasta, niin ei vahingossa korvaa alkuperäisen teeman ohjelmistokoodia niin, että teema ei enää toimikaan. Lapsiteema on alateema, joka sisältää kaiken pääteeman toiminnallisuuden, mutta pääteeman päivitykset eivät vaikuta lapsiteeman muotoiluun. (Vastbinder 2024.)

Kolme käytetyintä WordPress-teemaa ovat Hello Elementor, Divi ja Astra (Vastbinder 2024). Nämä eroavat toisistaan selkeästi siinä, miten tiiviisti ne liittyvät yhteen sivustonrakentajan kanssa. Sivustonrakentaja on alusta, joka mahdollistaa verkkosivujen luomisen vedä ja pudota -toimintoja hyödyntäen. Sivustorakentajassa on pitkälle suunniteltuja ja kehitettyjä valmiita malleja ja muokattavia elementtejä, jotka nopeuttavat verkkosivujen kehittämistä. (Elementor.) Hello Elementor on Elementor-sivustonrakentajan oma teema, joka on kehitetty toimimaan yhteen nimenomaan kyseisen sivustonrakentajan kanssa (Elementor 2024). Divi puolestaan on teema ja sivuston rakentaja yhdessä (Elegant themes 2024). Kolmanneksi suosituin teema on Astra, joka ei suoraan yhdisty yhteenkään sivustonrakentajaan. Se on päinvastoin suunniteltu toimimaan hyvin yhteen monen sivustonrakentajan

kanssa kuten Spectra, Elementor ja Beaver Builder, mutta sitä voi hyvin käyttää myös ilman sivustonrakentajaa. (WordPress c.)

Astra on suunniteltu latautumaan nopeasti ja se on rakenteeltaan kevyt. Astrassa on monipuolisesti muokkausmahdollisuuksia ilman ohjelmointitarvetta, mutta se on myös erittäin käyttökelpoinen pohja niille, jotka haluavat tehdä verkkosivun kehitystyötä ohjelmistokoodin tasolla. Astra on mobiililystävällinen ja sen selkeä koodi ja sisäänrakennetut Schema.org-merkinnät edistävät hakukoneoptimointia. Lisäksi Astra on hyvin yhteensopiva WooCommerce:n kanssa, joka on erittäin käytetty verkkokauppaliisäosa. Astrasta on olemassa ilmainen ja maksullinen versio. (WordPress c.) Ilmaisversioon voi itse ohjelmoimalla lisätä monenlaisia toiminnallisuuksia, mutta työskentelyä voi myös nopeuttaa ostamalla Astra Pro -version lisenssin. Maksullinen versio tarjoaa laajemmat ulkoasun hallintamahdollisuudet kuten enemmän vaihtoehtoja sivupohjien taustoihin ja leveyksiin, väreihin ja typografisiin asetuksiin. Se mahdollistaa myös syvemmän integraation WooCommerce:n kanssa, mikä lisää verkkokauppoihin yksilöllisyyttä ja monipuolisuutta. (Astra.)

## 2.4 Advanced Custom Fields, mukautettuja kenttiä ja sisältötyyppejä

Kuten kappaleessa 2.1 todetaan, WordPress perustuu avoimeen lähdekoodiin ja tämä mahdollistaa käyttäjien omien versioiden tekemisen. Käyttäjistä on muodostunut laaja ja aktiivinen verkosto, jossa kehitetään niin kokonaisia teemoja kuin lisäosiaakin, joilla on hyvin monenlaisia käyttötarkoituksia. Yksi pitkälle kehitetty ja monikäyttöinen lisäosa on WP Enginen omistama Advanced Custom Fields (myöhemmin ACF), joka mahdollistaa mukautettujen kenttien lisäämisen sivuille ja artikkeleihin, sekä WordPressin hallintapaneeliin kehittäjän ja ylläpitäjän käytettäväksi. Ilmaisessa versiossa käytettävissä on yli 30 erilaista kenttätyyppiä ja maksullisessa vieläkin enemmän, esimerkiksi toistuvat kentät ja joustavat sisältörakenteet. Kehittäjän työtä helpottaa myös se, että ACF on kehitytty käyttäjäystävälliseksi ja intuitiiviseksi ja se noudattaa saavutettavuusstandardeja. Käyttöliittymä toimii myös ruudunlukijoiden kanssa. (WordPress b.)

ACF on siis varsin monipuolinen työkalu WordPressin kehitykseen. Jopa ilmaisella versiolla voi luoda kokonaisia sisältötyyppejä (Custom Post Types), mikä helpottaa ja nopeuttaa kehittäjän työtä. WordPressiin ei tarvitse asentaa erillisiä lisäosia esimerkiksi kurssien, reseptien, tiiminjäsenten, asiakastarinoiden tai tapahtumakalenterin hallintaan ja esittämiseen, vaan nämä voidaan toteuttaa luomalla ACF:llä mukautettuja sisältötyyppejä. Mukautettu sisältötyyppi on omanlaisensa sisältömuoto, samaan tapaan kuin artikkelit tai sivut, mutta ACF:n avulla sen tietokentät voi määritellä itse. Koodin tasolla kyse on erillisen objektin luomisesta. Mukautetuille sisältötyypeille voidaan luoda myös omia taksonomioita, jotka mahdollistavat luokittelun ja lajittelun omien tágien ja kategorioiden perusteella. (Advanced

Custon Fields 2023.) Esimerkiksi jos halutaan luoda verkkosivu, jolla esitellään paikallisia ravintoloita, voidaan luoda mukautettu sisältötyyppi ravintolat. Tälle lisätään nimi- ja esitelykenttien lisäksi omat kentät kuten osoite, puhelinnumero ja aukioloajat. Näin ravintoloiden tietoja syötettäessä, ei luodakaan uutta sivua tai artikkelia, vaan uusi sisältötyyppinimeltä ravintola suoraan WordPressin hallintapaneelistä. Ravintoloille voidaan määritellä oma taksonomia, kuten ravintolatyyppejä, jonka mukaan ravintoloita voidaan sitten suodattaa tai hakea sivustolla. Lisäksi tätä ravintolaobjektia voidaan käyttää uudelleen sivustolla niin monta kertaa kuin tarpeellista.

## 2.5 GEO my WP, lisäosa geopaikannukseen

Geopaikannus on yksinkertaisimmillaan sijaintitietojen muuttamista kirjoitetusta muodosta koordinaateiksi ja takaisin. Verkkosivustoilla geopaikannusta käytetään muun muassa paikallisten yritysten, käyttäjien tai tapahtumien esittämiseen kartalla. Alkuperäinen sijaintitieto voidaan syöttää joko maantieteellisinä koordinaatteina tai tekstimuotoisena osoitteena. Kirjaimin ja numeroin kirjoitetun osoitteen muuntaminen koordinaateiksi tunnetaan nimellä eteenpäin geokoodaus (forward geocoding). Vastaavasti, kun tunnetaan koordinaatit, mutta halutaan palauttaa osoite luettavassa muodossa, käytetään käänteistä geokoodausta (reverse geocoding). Molemmat menetelmät toteutetaan usein geokoodaus API:n avulla, joka toimii verkkopalveluna. Se vastaanottaa käyttäjän pyynnön (esim. osoitteen), etsii vastaavan sijainnin taustatietokannastaan ja palauttaa koordinaatit tai osoitteen käyttäjälle. (Ranktracker 2023.)

Aiemmin esitelty Advanced Custom Field on ratkaisu moneen Wordpress-sivun haasteeseen, mutta geopaikannusta se ei tee. Siihen tarvitaan erillinen lisäosa, joka on WordPressille kehitetty GEO my WP. Se on Eyal Fitoussin kehittämä WordPressin geopaikannus-, kartoitus- ja lähihakutoimintoja mahdollistava lisäosa. GEO my WP tukee sekä Google Mapsiä että Open StreetMapsiä (Leaflet). Lisäosan ilmaisversio mahdollistaa esimerkiksi sijaintilomakkeen lisäämisen artikkelieditoriin, useiden sijaintien näyttämisen samalla kartalla (Mushup-kartta) ja yksittäisen sijainnin näyttämisen lyhytkoodilla. GEO my WP -lisäosan ilmaisversiolla on mahdollista näyttää myös käyttäjien sijainti. Laajempaa käyttäjien geopaikannuksen hyödyntämistä varten voidaan käyttää lisäosaa nimeltä BuddyPress Members Locator. GEO my WP:n maksullinen versio mahdollistaa esimerkiksi useamman sijainnin lisäämisen yhdelle käyttäjälle ja sijaintien suodattamisen usealla mukautetulla kentällä yhtä aikaa. (WordPress d.)

## 2.6 Karttakirjastot, ohjelmistokirjastot interaktiivisten karttojen luomiseen

Interaktiivisen kartan esittämiseen verkkosivulla tarvitaan JavaScript-karttakirjasto (Mapping framework). Ilman JavaScript-kirjastoa kartta on vain staattinen kuva, joka ei liiku eikä muokkautu käyttäjän toimintojen perusteella. Karttakirjasto on vuorovaikutuksessa käyttäjän kanssa ja lataa uusia kartan osia näkyviin, zoomaa tai toimii käyttäjän klikkausten perusteella. Karttakirjastot on myös optimoitu toimimaan eri selaimien ja laitteiden kanssa. Tällä hetkellä tunnetuin ja käytetyin karttakirjasto on Google Maps, joka tarjoaa monipuolisia maksullisia sisäänrakennettuja ominaisuuksia esimerkiksi reitityksen, geokoodauksen ja Street View:n. Google Mapsille vaihtoehtoja tarjoavat myös maksullinen Mapbox sekä avoimen lähdekoodin ilmaiset karttakirjastot OpenLayers ja Leaflet Map. (Cimbura, 2016.)

Maksullisista karttakirjastoista Mapbox on yhdistelmä avointa lähdekoodia ja kaupallisuutta. Mapboxin erityisyys on karttojen laaja visuaalinen muokattavuus ja Mapboxilla halutaankin keskittyä luomaan räätälöityjä karttaratkaisuja eri toimialoille. (Mapbox b.) Autoteollisuudessa esimerkiksi BMW ja Toyota, media-alalta CNN, The Washington Post ja The New York Times, matkailun alalta Booking.com ja liikunta- ja hyvinvointialalta Suunto käyttävät Mapboxia (Mapbox a.) Google Maps ei ole visuaalisesti niin muokattavissa. Se on suljettu, kaupallinen palvelu, jonka vahvuuksia on esimerkiksi pitkälle kehitetyt reititysominaisuudet ja reittien optimointityökalut (Google). Mapboxissakin on reititysominaisuuksia hyvin saatavilla, mutta ne eivät ole ihan Google Mapsin luokkaa. Mapbox kuitenkin mahdollistaa avoimella lähdekoodilla laajan kehittäjä- ja käyttäjäyhteisön, mikä taas edistää aktiivista ja jatkuvaa kehittymistä moneen suuntaan myös harrastajien toimesta (Mapbox b). Molemmissa karttakirjastoissa hinnoittelu perustuu käyttömääriin, palveluita voi käyttää pienimuotoisesti ilmaiseksi ja käytön lisääntyessä palveluille kertyy käyttömaksua (Google, Mapbox b). Mapboxin ilmaisen käytön voi aloittaa ilman luottokorttitietojen antamista (Mapbox b), mutta Google Maps vaatii nämä jo Google Maps Platform tilin tekemisen yhteydessä (google).

Puhtaasti avoimeen lähdekoodiin perustuvat OpenLayers ja Leaflet Map ovat käytettyjä ja erittäin toimivia ratkaisuja interaktiivisten karttojen luomiseen verkkosivuille. Näistä OpenLayers on laajempi kirjasto, joka tarjoaa kehittyneempiä ominaisuuksia kuten kerroksellisia karttoja, suurten datamäärien hallinnan, tuen erilaisille paikkatietostandardeille ja sisäänrakennettuja kustomointimahdollisuuksia. Tämä voi kuormittaa selaimen muistia ja prosessointitehoa. Huomattavasti kevyempi vaihtoehto on Leaflet Map, jossa karttakirjaston perustoiminnot saa toteutettua yksinkertaisesti ja intuitiivisesti, kustomointiin on tarjolla lisäosia. (KokaTic 2024.) Leaflet Mapista tarkemmin seuraavassa kappaleessa.

## 2.7 Leaflet Map, kevyt ja pelkistetty karttakirjasto

Käytetyimmistä karttakirjastoista Leaflet Map on selkeästi kevyin ja sen suunnittelussa on erityisesti kiinnitetty huomiota suorituskykyyn, käytettävyyteen sekä mobiiliyhteensopivuuteen. Tämän karttakirjaston arkkitehtuurin kehityksessä on keskitytty siihen, miten saadaan toimiva ja käytettävä kirjasto mahdollisimman vähällä ohjelmistokoodilla aikaiseksi. Leaflet Mapin sivulla todetaankin, että Leaflet *doesn't try to do everything for everyone. Instead it focuses on making the basic things work perfectly*. Tämä strategia on tuottanut Leaflet Mapille tunnettuutta ja sillä on valtavasti käyttäjiä. Leaflet Mapia käyttävät esimerkiksi Euroopan komissio, Etsy, Financial Times, ja USA Today (Leaflet a).

Leaflet Map on kooltaan vain noin 42 KB. Kuten edellä jo todettiin, sen keveys perustuu pelkistettyyn ydinrakenteeseen. Leaflet Map:illa ei ole omia karttalaattoja, vaan se hyödyntää ulkoisia palveluita. Sen kanssa voidaan käyttää avoimen lähdekoodin OpenStreetMap-laattoja tai maksullisia vaihtoehtoja, kuten Google Mapsia tai Mapboxia. Suljettujen lähdekoodien karttalaattojen käyttö vaatii API-avainta, kun taas avoimen lähdekoodin kirjaston käyttö onnistuu yksinkertaisesti määrittämällä karttakerrokseksi OpenStreetMapin JSON-tiedoston. (Leaflet a.)

Vaikka Leaflet on rakenteeltaan yksinkertainen, sitä voidaan kustomoida erillisillä lisäosilla käyttäjän tarpeiden mukaan. Avoin lähdekoodi mahdollista sen, että lisäosia voi myös tehdä itse. (Leaflet a.) Muissa karttakirjastoissa ominaisuudet ovat sisäänrakennettuja ja suljetuissa järjestelmissä kuten Google Maps, lisäosia ei voi tehdä itse (Cimbura, 2016). Leaflet Mapille on saatavilla lisäosia, jotka tarjoavat esimerkiksi piirtotyökaluja, markkereiden muokkaus- ja klusterointimahdollisuuksia, reititystoimintoja sekä geokoodaukseen yhdistyvän osoitehaun. (Leaflet b.)

### 3 Kehitysympäristöt

#### 3.1 Local, paikallinen kehitysympäristö WordPress sivujen kehitykseen

WordPress-sivuston kehittäminen on tehokasta ja turvallista paikallisessa kehitysympäristössä. Kun kehitysympäristön asennetaan omalle tietokoneelle, työskentely nopeutuu, koska päivityksiä ei tarvitse siirtää internetin välityksellä toiselle palvelimelle. Muokkaukset voi tehdä turvallisesti, sillä internetissä oleva versio säilyy ennallaan paikallisen kehityksen aikana. Lisäksi virheellistä sisältöä tai koodia ei päädy vahingossa verkkoon. Paikallinen kehitysympäristö antaa mahdollisuuden työskennellä myös ilman internetyhteyttä. (WP Engine 2024.)

Local on WP Enginen ilmainen tietokoneelle asennettava kehitysympäristö, jossa voi kehittää WordPress-verkkosivuja ilman webhosting-tilaa ja domainia. Sivut sijaitsevat käyttäjän koneella ja avautuvat verkkoselaimen samalla tavalla kuin verkossa olevat sivut, mutta eivät ole muiden nähtävissä ja käytettävissä. Local kuitenkin mahdollistaa väliaikaisen sivuston esikatselulinkin jakamisen oman tietokoneen ulkopuolelle Live Links -ominaisuuden avulla. Local on helppo asentaa ja käyttää, esimerkiksi WordPressin saa asennettua kehitysympäristöön yhdellä klikkauksella. Verkkosivujen kehittäminen on nopeaa ja sujuvaa, koska muutokset näkyvät verkkosivulla saman tien eikä esimerkiksi nettiyhteyden hitaus hidasta kehittämistä. Localin tukemat käyttöjärjestelmät ovat Windows, macOS ja Linux. (WP Engine b.)

WP Engine toteaa itse sivuillaan, että se on *The #1 local WordPress® development tool*. Se tarjoaa monipuolisesti ominaisuuksia verkkosivujen kehittäjien käyttöön. Local mahdollistaa esimerkiksi varmuuskopioinnin pilvipalveluihin kuten Google Driveen tai Dropboxiin. (WP Engine b.) Localin omistaja, WP Engine, tarjoaa myös itse hostingpalvelua ja Localissa on sisäänrakennettuna push ja pull -toiminnot, joiden avulla WordPress sivuja voi siirtää paikallisen koneen ja webhosting palvelimen välillä ilman erillistä FTP-ohjelmaa. Myös erilaisten sivuversioiden luominen ja testaaminen on Localin avulla helppoa, sillä se mahdollistaa olemassa olevan sivuston kloonaamisen ja erilaisten versioiden luomisen. (WP Engine a.)

#### 3.2 Visual Studio Code, koodieditori

WordPressissä on paljon ominaisuuksia valmiina ja lisäosien kautta vieläkin enemmän. Siitä huolimatta verkkosivuille voi olla tarpeellista lisätä toiminnallisuuksia koodaamalla. Paikallisia lähdekooditiedostoja voi muokata koodieditorissa. Visual Studio Code, myöhemmin VS Code, on Microsoftin omistama ilmainen avoimen lähdekoodin koodieditori. VS Code on saatavilla Windowsille, macOS:lle sekä Linuxille. Peruseditorin käyttö ei kuormita

tietokoneen järjestelmää tai vaikuta merkittävästi suoritustehoon. Sovellus tukee ohjelmointikieliä kuten Python, JavaScript, C++, Java ja Go. VS Code selkeyttää lähdekoodin syntaksia esittämällä eri rakenteet väreihin ja tyyleihin (engl. syntax highlighting), mikä helpottaa ohjelmakoodin kirjoittamista ja auttaa vähentämään virheitä. Sovelluksessa on myös sisäänrakennettu koodin täydennys, mikä nopeuttaa lähdekoodin kirjoittamista. Sisäänrakennettu ominaisuus on myös GIT-tuki, joka mahdollistaa commitit, branchien hallinnan ja muutosten tarkastelun suoraan editorissa. VS Codeen on asennettavissa paljon lisäosia, joilla käyttäjä voi muokata sovellusta omiin tarpeisiin sopivaksi. (Microsoft Software, 2025.) Näistä lisäosista yhtä, GitHub Copilota, tarkastellaan syvemmin tässä opinnäytetyössä myöhemmin.

VS Code on ollut suosituin kehitysympäristö Stack Overflown vuosittaisissa kehittäjä tutkimuksissa viimeiset 4 vuotta. Tutkimuksissa on eritelty vuosittain ainakin ammattikehittäjät ja välillä myös vasta opettelevat ja muut kehittäjät. VS Code on saanut kaikkien keskuudessa yli 70 % kaikista äänistä näiden neljän viimeisen vuoden aikana. (Stack Overflow 2021, Stack Overflow 2022, Stack Overflow 2023, Stack Overflow 2024.) Stack Overflow on ohjelmointiin ja ohjelmistokehitykseen keskittynyt keskustelufoorumi, jonka ympärille on kehittynyt laaja verkkoyhteisö. Esimerkiksi vuonna 2024 Stack Overflow -kehittäjä tutkimuksen kysymykseen *mitä kehitysympäristöä käytät säännöllisesti...?*, vastasi yhteensä 58 121 käyttäjää. Heistä 74 % oli ammatikseen ohjelmistokehittämistä tekeviä, ja heistä 74 % vastasi käyttävänsä VS Codea. Vastauksissa oli myös ohjelmointia opettelevia 6 300 käyttäjää, ja heistä 77,6 % käytti VS Codea (Stack Overflow 2024).

## 4 Tekoäly kollegana verkkosivukehityksessä

### 4.1 ChatGPT, keskusteleva tekoäly

Tekoälyä on odotettu pitkään ohjelmistokehitykseen avuksi. Kun ChatGPT julkaistiin yleiseen käyttöön vuonna 2022, saivat ohjelmoijat kaivatun työkalun. ChatGPT on OpenAI:n kehittämä kielimalli, joka on suunniteltu käymään luonnollisia keskusteluja, vastaamaan kysymyksiin, kirjoittamaan tekstiä ja avustamaan monenlaisissa tehtävissä. (OpenAI 2022.)

ChatGPT on koulutettu vahvistusoppimisen menetelmällä, missä ihmiset kouluttavat tekoälyä esittämällä sekä ihmiskäyttäjää että tekoälyavustajaa. Kielimalliakin käytettiin keskustelun vastausten luomiseen, mutta tekoälyavustajaa näyttelevä ihminen paransi kielimallin antamia vastauksia. Sitten tätä paranneltua keskustelua käytettiin jälleen kielimallin koulutukseen ja samaa kuviota toistettiin uudelleen. Toisessa vaiheessa kielimalli laitettiin luomaan rinnakkaisia vastauksia ja ihmiskäyttäjä vertaili ja arvioi niitä ja laittoi ne paremmuusjärjestykseen. (OpenAI 2022.) Vahvistusoppimisen lisäksi kielimalli rakentaa osaamistaan myös valvomattoman oppimisen (Unsupervised learning) avulla. Lukiessaan valtavia määriä dataa, se oppii miten lauseet muodostuvat, mitkä sanat liittyvät toisiinsa ja miten mistäkin aiheesta yleensä puhutaan. Tämän lisäksi ChatGPT skaalaa päättelyään (Scaling reasoning) ja perustelee vastauksiaan ajatusketjuilla. Tämä on erityisen hyödyllinen ominaisuus loogisissa, matemaattisissa ja muissa tieteellisissä ongelmissa, kuten esimerkiksi ohjelmoinnissa. (OpenAI 2025.)

Tekoälyyn kohdistuu myös kritiikkiä ja sen käytössä on omat haasteensa. Niin ChatGPT kuin muutkin kielimallit antavat välillä uskottavalta vaikuttavia, mutta sisällöltään virheellisiä tai järjettömiä vastauksia. Tätä on vaikea korjata, sillä kun kielimallia koulutetaan vahvistusoppimisen menetelmällä, oikean ja väärän vastauksen tulkinta jää ihmiselle. Jos kouluttaja ei itse tiedä vastausta, hän arvioi vastauksen oikeellisuutta sen perusteella miten uskottavalle se näyttää. Ja mahdollisesti jopa korjaa mallin antamaa parempaa vastausta huonomaksi omien tietojensa perusteella. Välillä kielimalli voi antaa myös epäsopivia tai puolueellisia vastauksia, vaikka sitä onkin opetettu kieltäytymään epäasiallisista pyynnöistä. Toisaalta liian pitkälle viety varovaisuus vastauksen antamisessa karsii huonojen vastausten lisäksi myös oikeita ja tarpeellisia vastauksia pois, mikä voi myös olla ongelmallista. Vaikka ChatGPT on toimintatavaltaan keskusteleva, sen olisi hyvä esittää tarkentavia kysymyksiä. Tällä hetkellä se arvaa ja täydentää puuttuvat tiedot itse ja antaa vastauksen yhden pyynnön perusteella kysymättä lisätietoja käyttäjältä. (OpenAI 2022.)

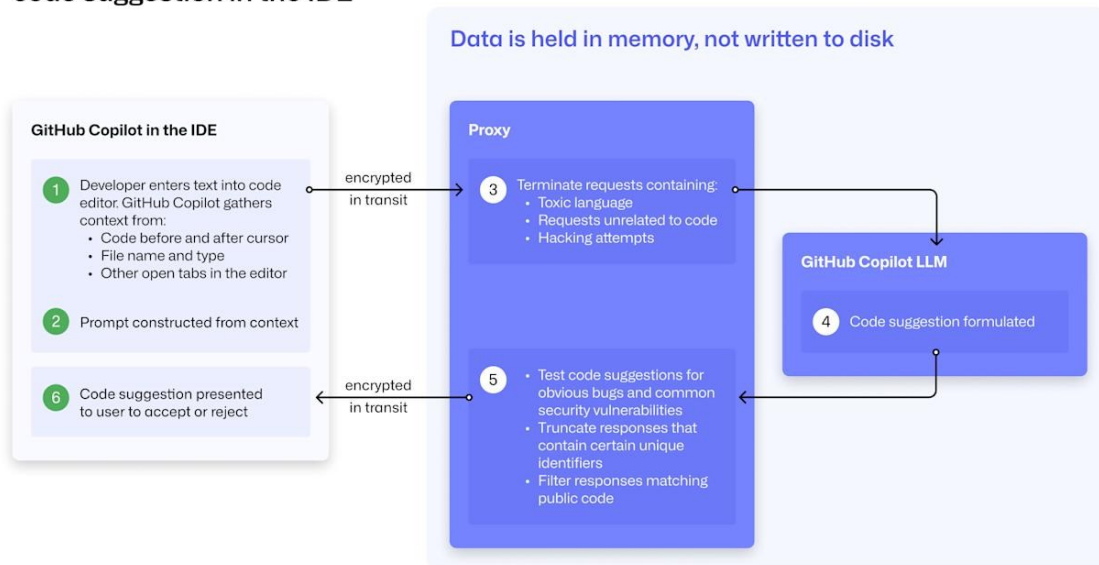
## 4.2 GitHub Copilot, tekoäly ohjelmoinnin tueksi

ChatGPT:n lisäksi vuonna 2022 julkaistiin yleiseen käyttöön toinenkin tekoälysovellus, josta on erityisesti hyötyä ohjelmoijille. GitHub Copilot (myöhemmin Copilot) on koodintäydennystyökalu, joka luo ja ehdottaa ohjelmistokoodia käyttäjän kirjoittaman koodin, kommentin tai tekstin perusteella sekä tarkastaa ja tarkastelee kirjoitettuja algoritmeja kokonaisuutena. (Dakheel ym. 2023, 1.) Copilot on saanut valtavasti käyttäjiä heti julkaisunsa jälkeen. Vuonna 2023 sillä oli Stack Overflown kehittäjätkimuksessa 22 078 käyttäjää ja 2024 vuonna 19 038 käyttäjää. Vuonna 2024 Copilotia enemmän käytettiin ainoastaan ChatGPT:tä. (Stack Overflow 2024.)

Copilot toimii lisäosana useassa eri kehitysympäristössä. Sen asennusdokumentaatiosta löytyy asennusohjeet seuraaviin kehitysympäristöihin: Azure Data Studio, Eclipse, JetBrains IDEs, Vim/Neovim, Visual Studio, VS Code ja Xcode (GitHub 2025). GitHub on kehittänyt Copilotin yhdessä OpenAI:n kanssa ja alkuperäinen Copilot käytti tekoälymallinaan OpenAI Codexia. OpenAI Codex on luotu kouluttamalla sitä erityisen laajalla lähdekoodiaineistolla. (Friedman 2021.) 2024 GitHub ilmoitti, että Copilotin käyttäjät voivat jatkossa valita useasta eri tekoälymallista käyttöönsä sopivimman (Warren 2024). Copilot toimii monen ohjelmointikielen kanssa, mutta erityisen hyvin se ymmärtää seuraavia kieliä: Python, JavaScript, TypeScript, Ruby ja Go. (Friedman 2021.) 2023 GitHub lisäsi Copilotiin chat-ominaisuuden, mikä mahdollistaa käyttäjän ja Copilotin keskustelemisen luonnollisen kielen avulla. Näin käyttäjä voi selittää omin sanoin mitä haluaa ohjelmakoodilla saavuttaa ja Copilot sekä kirjoittaa lähdekoodin että selittää, mitä lähdekoodissa tapahtuu. (Dohmke 2023.)

Käyttäjä voi antaa syötteen Copilotille VS Coden koodieditorissa tai keskusteluikkunassa, mutta Copilotia voi käyttää myös GitHubin kautta verkossa. Saatuaan käyttäjän syötteen Copilot kerää kontekstia koodieditorista kursoria ennen ja sen jälkeen olevasta lähdekoodista, sekä muista auki olevista välilehdistä, tiedostojen nimistä ja keskusteluikkunan aikaisemmista keskusteluista. (Kuva 2.) Kerätystä kontekstista muodostuu lopullinen syöte yhdessä käyttäjän syötteen kanssa. Copilot testaa muodostettua syötettä välityspalvelimella (proxy) suodattaen pois hakkerointiin viittaavat tai muuten turhat, vahingolliset tai laittomat osat. Tämän jälkeen syöte siirtyy GitHubin palvelimelle, jossa muodostetaan ehdotus koodille. Koodiehdotus palaa välityspalvelimen kautta takaisin käyttäjälle. Copilotin kouluttamiseen käytetystä lähdekoodista on voinut päätyä tunnistetietoja kuten salasanoja, API-avaimia, tekijänoikeuksilla suojattua tai vahingollista materiaalia koodiehdotukseen. Näiden varalta koodiehdotus tarkistetaan välityspalvelimella, ennen kuin se palautetaan käyttäjälle. Lopuksi käyttäjä joko hyväksyy tai hylkää ehdotuksen. (GitHub a.)

## The life cycle of a GitHub Copilot code suggestion in the IDE



Kuva 2. GitHub Copilotin tiedonkäsittely (GitHub a)

### 4.3 Tekoöly ohjelmoijan työparina

Mielenkiinto työparina toimivaa tekoölyä kohtaan on suurta ja sen käytettävyyttä on myös tutkittu ja seurattu alusta asti. Esimerkiksi Kanadassa Dakhel ym. (2023, 7) tekivät tutkimuksen, jossa arvioitiin tekoölyn tuottaman koodin laatua ja sitä kautta luotettavuutta ohjelmoijan työparina. Tutkimuksessa huomattiin, että GitHub Copilot tuottaa toimivaa ja optimoitua koodia kun kyseessä on selkeästi määritelty suhteellisen tyypillinen algoritminen ongelma. Mitä selkeämpi syöte, sitä varmemmin ratkaisu oli toimiva. Tekoöly ei kuitenkaan ymmärrä koodin tarkoitusta yhtä laajasti kuin ihminen, ja tämä on tekoölyavusteisen ohjelmoinnin suurin haaste. Ihmisen on osattava kertoa hyvin laajasti ohjelmointiprojektista syötteessä, eikä tekoöly siltikään välttämättä ymmärrä kaikkia reunaehtoja. Tästä syystä tutkimus suosittelee ohjelmoijia suhtautumaan varauksella tekoölyn tuottamiin ratkaisuihin. Tutkijat painottavat myös, ettei aloittelevan ohjelmoijan tulisi tukeutua tekoölyyn, koska sen tuottamissa algoritmeissa voi esiintyä virheitä, jotka johtavat huonolaatuisen ohjelmistokoodiin.

Samankaltaisia tuloksia esitellään myös Liun ym. (2023, 1) tutkimuksessa, jossa arvioitiin ChatGPT:n tuottamaa koodia. Tutkimuksessa keskityttiin tarkastelemaan tekoölyn luoman koodin oikeellisuutta, vaikeusastetta ja turvallisuutta. Liu ym. havaitsi, että ChatGPT pystyy tuottamaan toimivaa koodia, erityisesti yksinkertaisissa ja selkeästi määritellyissä tehtävissä. ChatGPT tuotti oikean ratkaisun noin 66–79 % tehtävistä. Tuotettu koodi oli usein

selkeää ja luettavaa, mutta tehokkuudessa oli parantamisen varaa. Monimutkaisemmissa ratkaisuisissa esiintyi enemmän virheitä, puutteita sekä epätoivottuja ohjelmointikäytäntöjä, jotka voivat heikentää ohjelman luotettavuutta ja turvallisuutta. (Liu ym. 2023, 1.) Liun ym. tutkimustulokset tukevat Dakhelin ym. näkemystä, että tekoälyllä tuotettua ohjelmistokoodia on arvioitava kriittisesti erityisesti vaativimmissa ohjelmointitehtävissä.

Yang ja Tsai (2024) tukevat tutkimuksellaan näkemystä, että tekoäly on hyödyllinen varsinkin virheiden korjaamisessa ja ohjelmistokoodin ymmärtämisen syventämisessä. Heidän tutkimusasetelmassaan osa opiskelijoista käytti tekoälyä ohjelmoinnin tukena ja osa ei. Ohjelmointitehtävien lopullisessa suorituksissa ei ollut merkittävää eroa ryhmien välillä, mutta ChatGPT:tä käyttäneet opiskelijat kokivat oppimisprosessin mielekkäämmäksi ja aikovat hyödyntää tekoälyä jatkossakin ohjelmoinnin tukena. Dohmke ym. (2023, 1-2) tarkastelevat artikkelissaan GitHub Copilotin vaikutusta ohjelmistokehitykseen ja sen tuottavuuteen. Sen mukaan Copilotin käyttäjät hyväksyvät keskimäärin 30 % tekoälyn ehdottamasta koodista. Sen lisäksi, että tekoäly vaikutti tuottavuuteen, se lisäsi käyttäjien tyytyväisyyttä. Dohmken ym. lähestymistapa tekoälyn hyödyntämiseen eroaa Dakhelin ym., Liun ym. sekä osittain myös Yangin ja Tsain näkemyksistä. Dohmke ym. (2023, 1-2) korostavat artikkelissaan oppimisprosessia. Aloittelevat ohjelmoijat hyväksyvät enemmän tekoälyn ehdotuksia, mutta myös oppivat samalla. Tämä nopeuttaa sekä työnalla olevan projektin edistymistä että ohjelmoijan taitojen kehittymistä.

Dohmken ym. (2023, 1) artikkelin mukaan tekoäly yhdistettynä nimenomaan ohjelmistokehitykseen voi lisätä maailman bruttokansantuotetta yli 1,5 biljoonalla dollarilla vuoteen 2030 mennessä. Vertailun vuoksi BigDATAwire (2024) referoi artikkelissaan IDC:n tutkimusta, joka arvioi tekoälyn vaikuttavan kokonaisvaltaisesti maailman bruttokansantuotteeseen lisäten sitä 19,9 biljoonalla dollarilla.

## 5 Toteutus ja kehitysprosessi

### 5.1 Verkkosivun rakenne ja reunaehdot

Projektin tavoitteena oli kehittää verkkosivulle karttanäkymä, johon on merkitty kaikki verkkosivuilla esitellyt yritykset. Lisäksi käyttäjä voi suodattaa yrityksiä avainsanojen avulla. Samojen yritysten esittely näkyy myös karttanäkymän alla etusivulla. Käyttäjä voi keskittää kartan omaan sijaintiinsa tai syöttää osoitteen, jonka perusteella kartta kohdistetaan, ja tämän jälkeen hakea läheisiä yrityksiä. Kartan voi myös sulkea ja selata vain yritysesittelyjä.

Koska verkkosivuprojekti tehtiin yhdistykselle, tuli ratkaisut toteuttaa ensisijaisesti maksuttomilla työkaluilla. Yhdistyksellä ei ole luottokorttia maksullisten lisäosien käyttöä varten, mikä asetti projektille tiukat reunaehdot. Google Maps suljettiin pois vaihtoehtoista, koska sen käyttö olisi edellyttänyt henkilökohtaisiin tietoihin perustuvan tilin luomista sekä joiltain osin maksullisia ominaisuuksia. Maksuttomien lisäosien hyödyntäminen oli myös oppimisen kannalta hyvä ratkaisu, sillä se edisti kehittäjän syvällisempää ymmärrystä ohjelmoinnista verrattuna valmiisiin maksullisiin ratkaisuihin. Ainoa maksullinen työkalu oli GitHub Copilot, koska samalla oli tarkoitus opetella sen käyttöä tulevaisuuden kehitysprojekteja varten. ChatGPT:stä käytettiin ilmaisversiota, joka tarjosi tiedostoja kirjoittavan ja keskusteluihin liitetiedostoja hyväksyvän GPT-4o -mallin käyttöön aina hetkellisesti 2–3 kertaa vuorokaudessa.

### 5.2 Verkkosivun kehitysvaiheet ja tekoälyn rooli kehityksessä

Verkkosivua kehitettiin aluksi palasina kysyen ChatGPT:ltä apua yksittäisiin ongelmiin ja vertaillen tekoälyn avulla mahdollisia vaihtoehtoja. Lopullisen sivuston kehittäminen aloitettiin kuitenkin kysymällä ChatGPT:ltä, miten tekoälyn kanssa kannattaisi työskennellä, jotta tämä muistaisi koko käydyn keskustelun ja ratkaisut, joihin on päädytty ja osaisi ohjeistaa kehittäjää näiden ratkaisujen pohjalta oikeaan suuntaan. Tekoälyn vastaukset olivat seuraavat: Ota käyttöön muisti ChatGPT:ssä ja/tai tee manuaalinen muisti eli yksinkertainen dokumentti aiemmista päätöksistä ja liitä se keskusteluun. Projektissa valittiin lähestymistavaksi tehdä ChatGPT:n kanssa yhdessä suunnitelmatiedosto, jota käytetään keskustelussa tukena koko kehityksen ajan. Kuvassa 3 on käyttäjän ChatGPT:lle antama vastaus, jonka pohjalta suunnitelman kehittäminen aloitettiin. Ensimmäinen versio suunnitelmasta on liitteessä 1.

Joo, luo pohja näiden tietojen pohjalta: Verkkosivun nimi on Makuja kaakosta ja se esittelee suomalaisia lähiruuan tuottajia ja myyjiä sekä ravintoloita ja maatilamatkailukohteita jotka ovat erikoistuneet lähiruoka-asioihin nimenomaan kaakkois-suomesta. Sivusto toteutetaan WordPressillä ja Astra teemalla. Siihen käytetään vain ilmaisia lisäosia. Etusivulle on tarkoitus laittaa kartta, jossa näkyy kaikki yritykset, jotka ovat ostaneet paikan verkkosivulta. Kartan alla on pienet esittelyt yrityksistä. Kartan vasemmalla puolella on avainsanahaku, josta valitsemalla checkboxeilla avainsanoja, kartalle suodattuvat ne yritykset, joiden esittelyissä on kyseiset avainsanat. Samalla etusivun avainsanalistaan avainsanojen perässä oleva lukumäärä päivittyy. Eli aluksi avainsanan perässä näytetään kaikkien artikkelien lukumäärä, joissa kyseinen avainsana esiintyy. Kun joitakin avainsanoja on valittu, avainsanan perässä näytetään enää niiden artikkelien lukumäärä, joissa on valitut avainsanat ja kyseinen avainsana. Lisäksi sivustoon olisi tarkoitus tehdä oma "backend" osio ylläpitäjälle, joka syöttää uusia yrityksiä sivustolle, mutta ei kajoa muuten sivuston asetuksiin ja ylläpitoon."

Kuva 3. ChatGPT:lle annettu syöte suunnitelmaa varten

Sivuston kehitys aloitettiin ChatGPT:n luoman suunnitelman mukaan Local-kehitysympäristössä luomalla tyhjä WordPress-sivu, asentamalla teemaksi Astra ja poistamalla muut oletuksena tulleet teemat. ChatGPT:ltä kysyttiin neuvoja lapsiteeman toteutukseen. ChatGPT tarjoutui tekemään tarvittavat lähdekooditiedostot ja neuvoi lapsiteeman tekemisen työvaiheet. Käytännössä lapsiteemaa varten luotiin oma kansio WordPressin tiedostohakemistoon teemakansion sisälle ja sinne kaksi tiedostoa; style.css ja functions.php. Näihin molempiin ChatGPT teki valmiit koodit.

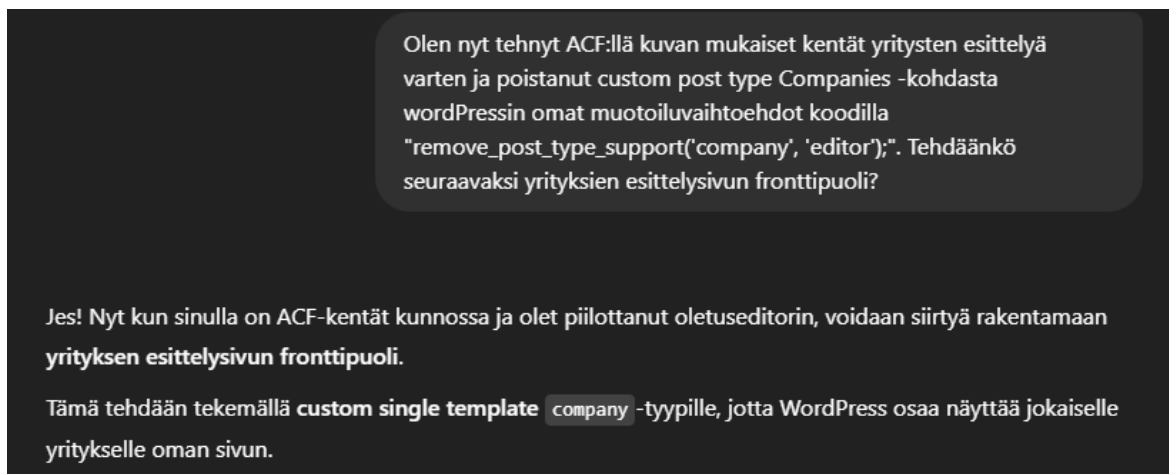
### 5.2.1 Companies, mukautetun sisältötyypin luominen

ChatGPT:n alkuperäisessä suunnitelmassa ei ollut seuraavana vaiheena yritysesittelyjen toteutuksen suunnittelua, vaan kehittäjä kysyi siitä tekoälyltä. ChatGPT lisäsi ohjeeseen yritysesittelyjen toteuttamisen seuraavaksi. ChatGPT ehdotti yritysten tietojen hallintaan Advanced Custom Fields -lisäosan (ACF) käyttöä ja listasi mahdollisia kenttiä yritysten esittelyä varten. Kehittäjä huomautti, että käytössä on ainoastaan ACF-lisäosan ilmainen versio, ja ehdotti lisäksi GEO my WP -lisäosan hyödyntämistä yrityksen sijainnin määrittelyyn kartalla.

WordPressiin asennettiin suunnitelman mukaan Advanced Custom Fields -lisäosa yritysten tietojen tallentamista varten. Sen jälkeen lisättiin Astran lapsiteeman functions.php-tiedostoon ChatGPT:n tuottama funktio, joka lisää WordPressin hallintapaneelin sivupalkkiin

uuden Companies-nimisen mukautetun sisältötyypin. Seuraavaksi luotiin ACF:llä kenttäryhmä yrityksen tietojen tallentamista varten. Kun kenttäryhmän näyttökohteeksi asetettiin edellä luotu sisältötyyppi company, saatiin yritysten tietojen syöttämiseksi luotua käyttöliittymä. Varmuuden vuoksi poistettiin vielä käytöstä WordPressin oletusmuokkaustyökalut GitHub Copilotin avulla, jotta yritysten tiedot tulevat kirjatuksi varmasti ACF:llä luotuihin kenttiin. Yritysten tietojen syöttö ja tallennus oli nyt mahdollista, joten seuraavaksi luotiin yksittäisten yritysten esittelysivuja varten tiedostot single-company.php ja company.css, joihin ChatGPT tarjosi valmiit koodiratkaisut. Nämä luodut tiedostot rekisteröitiin käyttöön Astran lapsiteeman functions.php -tiedostossa. Näin saatiin luotua erilliset sivut, joilla näkyy yrityksestä annetut tiedot.

Kehittäjä alkoi käyttää projektin aikana myös tekniikkaa, jossa hän lisäsi ChatGPT:lle kuvia erilaisista tilanteista projektin edetessä. Esimerkiksi ACF:n asetuksia ei saatu suoraan toimimaan tekoälyn ohjeilla, vaan tämä vaati kehittäjältä omaa tiedonetsintää. Kun asetukset olivat kunnossa, kehittäjä laittoi kuvan ACF:n valikosta tekoälylle analysoitavaksi, jotta suunnitelma pysyy ajan tasalla. Kuvassa 4 on kehittäjän ja ChatGPT:n keskustelua ennen single-company.php ja company.css -tiedostojen luomista. Tässä on myös hyvä esimerkki siitä, miten positiivisesti tekoäly aloittaa aina vastaamisen. Lisäksi sen antamat vastaukset tukevat kehittäjän ammatillista kasvua, koska vastauksissa käytetään alan termistöä, vaikka kehittäjä ei sitä käyttäisikään. Tässä esimerkiksi tekoäly käyttää sanaa *oletuseditori*, kun kehittäjä on todennut ylimalkaisesti *WordPressin omat muotoiluvaihtoehdot*.



Kuva 4. Ote ChatGPT:n ja kehittäjän keskustelusta

ChatGPT:n luoma koodi VS Code -koodieditorissa mukautetulle sisältötyypille Companies ja sivupalkissa luodut tiedostot single-company.php ja company.css näkyvät kuvassa 5. Siinä on myös toinen hyvä esimerkki tekoälyn käyttämästä kielestä. ChatGPT on kirjoittanut kommenttiin suoran käännöksen sanasta post type eli *postityyppi* todennäköisesti siksi, että

kehittäjäkin on käyttänyt edellisen kuvan syötteessä tätä ilmaisua. Esimerkiksi tällaisten ilmaisujen takia tekoälyn käyttäjän on oltava tarkkana. Kuitenkin pyytämällä erikseen parempaa käännöstä ja täsmällisempää ilmaisua, tekoäly monesti sellaisen tuottaa.

```

32
33 // Rekisteröidään mukautettu postityyppi "company"
34 add_action('init', 'makuja_register_company_post_type');
35 function makuja_register_company_post_type() {
36     $args = array(
37         'labels' => array(
38             'name' => 'Companies',
39             'singular_name' => 'Company',
40             'menu_name' => 'Companies',
41             'add_new' => 'Add New',
42             'add_new_item' => 'Add New Company',
43             'edit_item' => 'Edit Company',
44             'view_item' => 'View Company',
45             'all_items' => 'All Companies',
46             'search_items' => 'Search Companies',
47             'not_found' => 'No companies found.',
48         ),
49         'public' => true,
50         'has_archive' => false,
51         'rewrite' => array('slug' => 'companies'),
52         'show_in_rest' => true,
53         'supports' => array('title', 'thumbnail'),
54         'menu_position' => 5,
55         'menu_icon' => 'dashicons-store',
56     );
57     register_post_type('company', $args);
58     remove_post_type_support('company', 'editor'); // Poistetaan tekstieditori
59 }
60

```

Kuva 5. Mukautetun sisältötyypin koodi Visual Studio Codexissa

Seuraavaksi `company_tags`-toiminto lisättiin `functions.php`-tiedostoon, jotta yrityksiä voidaan myöhemmässä vaiheessa lajitella tágien avulla. Yksittäisten funktioiden lisäämisessä olemassa oleviin tiedostoihin hyödynnettiin GitHub Copilotia: sen avulla selvitettiin, mihin kohtaan tiedostoa uusi koodi kannattaa sijoittaa tai miten olemassa olevaa koodia tulee päivittää niin, että ChatGPT:n ehdottamat muutokset otetaan huomioon.

Sijaintitietokentän lisääminen kohtaan, johon yrityksen tietoja syötetään, sujui helposti tekoälyn ohjeiden mukaan. Ensin asennettiin GEO my WP-lisäosa. Sen asetuksista Post Locator kohdistettiin sisältötyyppiin Company ja näin saatiin osoitetietokentät näkyviin yrityksen tietojen syöttölomakkeelle. Kuvissa 6 ja 7 näkyy uuden yrityksen tietojen syöttämislomake alusta ja lopusta. Alussa (kuva 6) ACF:llä tehdyt kentät ja lopussa (kuva 7) GEO my WP:llä. Kun kirjoittaa yrityksen osoitteen GEO my WP:n osoiteriveille, lisäosa tallentaa osoitteen myös koordinaatteina. ChatGPT neuvoi, miten GEO my WP-lisäosalla tallennettu osoite voidaan tuoda koodin avulla erilliselle yrityksen esittelysivulle näkyviin, mutta tätä ratkaisua ei monesta yrityksestä huolimatta saatu toimimaan. Kehittäjä etsi GEO my WP:n dokumentaatiosta lyhytkoodin, jolla saatiin yrityksen sijainti kartalle näkymään yrityksivulla.

Tästä kehittelemällä ChatGPT keksi ratkaisun, jolla yrityksen osoitetietokin voidaan tuoda sivulle erikseen näkyviin. Ratkaisussa hyödynnettiin WordPressin omaa lyhytkoodia `echo do_shortcode('[gmw_single_location object="post" elements="address"]');` kun muiden yrityksen tietokenttiensisällöt haettiin yksinkertaisesti `get_field` -toiminnolla.

The screenshot shows the 'Add New Company' form in the WordPress dashboard. The left sidebar contains navigation options: Dashboard, Posts, Companies (selected), All Companies, Add New Company, Categories, Tags, Media, Pages, Comments, Astra, Appearance, and Plugins. The main content area is titled 'Add New Company' and features a text input field for 'Add title'. Below this is a section titled 'Company Information' containing a text input field for 'Company Name'. There are two 'Company Image' sections, each with the text 'No image selected' and an 'Add Image' button.

Kuva 6. WordPressin hallintapaneeli, ACF:llä tehtyjä kenttiä

The screenshot shows the 'Add New Company' form with the 'Location' section expanded. The left sidebar is the same as in the previous image. The main content area has a search bar at the top. Below it is a 'Location' section with a dropdown menu. The dropdown menu is open, showing options: Location, Address, Coordinates, Contact, and Hours of Operation. To the right of the dropdown are several text input fields: 'Street', 'Apt/Suit', 'City', 'State', 'Zipcode', and 'Country'. At the bottom of the form, there is a blue 'Save location' button and a red error message that says 'No location found'.

Kuva 7. WordPressin hallintapaneeli, GEO my WP:llä tehtyjä kenttiä

### 5.2.2 Headerin ja etusivun asettelut

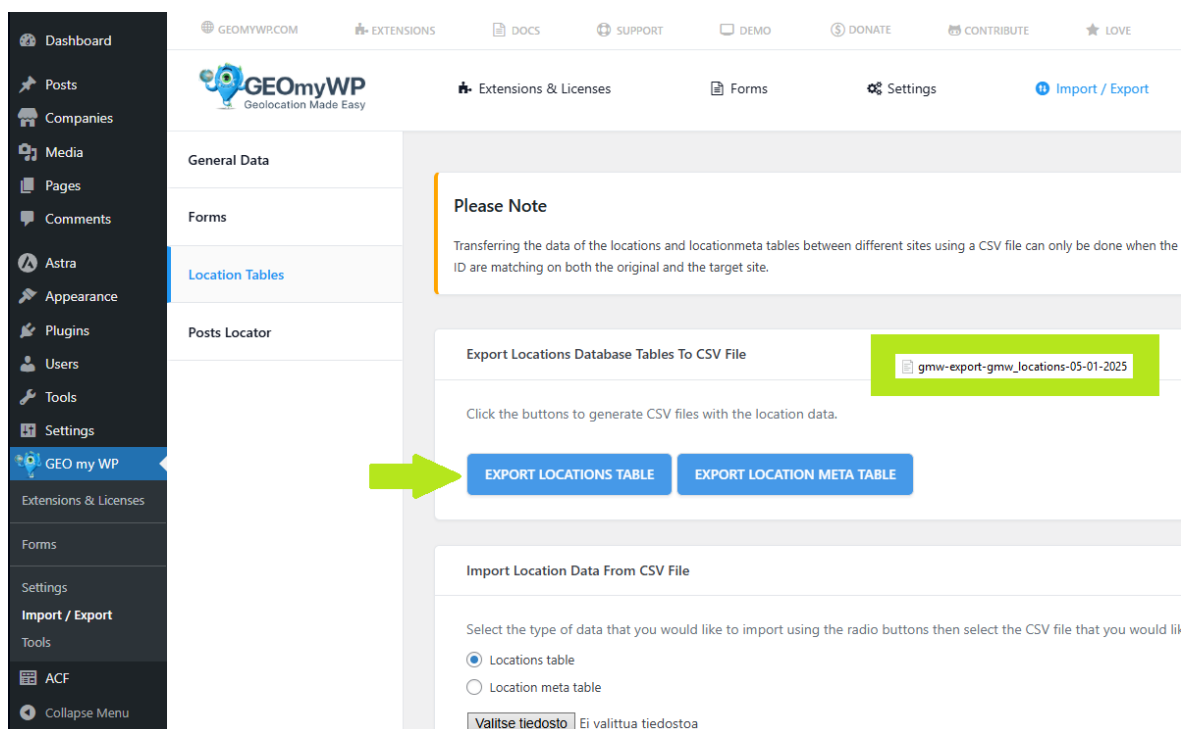
Sivuston asettelua oli jo hiukan tehty yrityssivun rakennusvaiheessa, mutta nyt oli vuorossa laajempi visuaalinen projekti. Aluksi kehitettiin etusivulle sisältöalueet luomalla front-page.php tiedosto ja sille tyylitiedosto front-page.css. Näiden sisällöt tuotti ChatGPT. Sivut lisättiin teeman functions.php-tiedostoon jo tuttuun tapaan ChatGPT:n tuottamalla koodilla. ChatGPT:n koodi nosti sisällön teeman olemassa olevan headerin päälle, joten tähän apua pyydettiin ja saatiin GitHub Copilotilta, joka kirjoitti tyylitiedostoon korjaukset, jolla sisältö saatiin laskettua alaspäin. Seuraavaksi vuorossa oli headerin toteutus. Koska headeriin tulee yritysten kategoriat, lisättiin functions.php-tiedoston kautta kategoriaominaisuus company\_category yritystietoihin ChatGPT:n ohjeilla. Headerin asettelua, logon tuontia sekä sosiaalisen median ja kielivalintojen kuvakkeita päivitettiin taas GitHub Copilotin avulla.

ChatGPT loi etusivun rakenteelle koodin. Kuitenkin sen antamassa koodissa sivupalkki ja pääsisältö tulivat allekkain eikä vierekkäin, kuten oli toivottu. Tämä saatiin korjattua, kun kehittäjä kopioi ChatGPT:lle Astran oman index.php:n koodin, jota mukailen ChatGPT muokkasi omaa koodiaan. Etusivun asettelulle tehtiin ensin rakenne ja sitten tuotiin ChatGPT:n aikaisemmin ehdottamia koodikokonaisuuksia jokaiseen osaan erikseen. Näitä asettaessa käytettiin taas GitHub Copilotia, joka ehdotti, että osioiden lähdekoodeille tehtäisiin erilliset tiedostot, jolloin sivuston koodin ylläpito olisi helpompaa.

Astra-teeman template parts -kansioon lisättiin tiedostot company-list.php ja tag-filter.php. Visuaalista muotoilua varten front-page.css -tiedostoon lisättiin myös koodia. Yritysten esitelykortteja oli tarkoitus saada useampi vierekkäin karttaosion alle. Tämä ei meinannut onnistua millään, kumpikaan tekoälysovellus ei osannut tarjota ratkaisua, jolla tämä olisi toteutunut. Ongelma johtui siitä, että company-list.php-tiedostossa luodaan kokonainen luokka nimeltä company-list ja tälle piti asettaa sääntö display: flex;. Vaikka GitHub Copilot itse ehdotti erillisen tiedoston tekemistä, se ei tajunnut company-list -luokan olemassaoloa. Myöhemmin korttien visuaalista ulkonäköä päivitettiin antamalla kuva toisen verkkosivun yrityskortin asetteluista ChatGPT:lle ja pyytämällä muotoilukoodia kuvan tyyppiseen muotoiluun. ChatGPT:n antama koodi oli hyvä pohja, mutta lopullinen yrityskortin ulkonäkö muodostui kehittäjän itse kokeilemalla ja hakemalla ratkaisuja verkkoselaimen kehittäjäikkunassa.

### 5.2.3 Karttaosion kehitys

Karttaosion rakentamiseen ChatGPT ehdotti yritystietojen hakemista WordPressin REST API:n kautta. Tämä ratkaisu olisi mahdollistanut karttamerkintöjen luomisen JavaScriptillä suoraan JSON-vastauksen perusteella, mutta sen toteuttaminen tuntui kehittäjästä liian monimutkaiselta. Koska kehittäjä oli jo kertaalleen rakentanut auttavasti toimivan karttaosion GitHub Copilotin avulla, lähdettiin karttaosiota rakentamaan toisella tavalla. Tähän projektiin tuotiin aiemmasta projektista toimiva karttaosio, ja Copilotille kerrottiin, että tähän halutaan toteuttaa vastaavanlainen ratkaisu. Copilot ehdotti map-filter.js-tiedoston luomista sekä päivityksiä front-page.php- ja front-page.css-tiedostoihin karttaosioon liittyen. Copilotin kanssa keskustellen, koodiosio ja tiedosto kerrallaan läpi käyden ja välillä selaimen konsolin virheviestejä tulkiten kartta saatiin näkyviin. Yritysten markkerit eivät näkyneet kartalla ja tämän selvittelyyn tarvittiin kehittäjän omaa osaamista. Lopulta kuten kuvassa 8 on esitetty, katsottiin, että GEO my WP tallentaa paikkatiedot gmw-export-gmv\_location -nimiseen tiedostoon ja tämän tiedon avulla saatiin markkerit näkymään. Kuvassa 9 GitHub Copilotin muokkaama koodi, joka käyttää gmw\_location -taulukon parametrien oikeita nimiä.



Kuva 8. WordPressin hallintapaneeli, GEO my WP -lisäosa, sijaintitietojen lataus

```

$query = new WP_Query($args);
$businesses = [];

if ($query->have_posts()) {
    while ($query->have_posts()) {
        $query->the_post();
        $post_id = get_the_ID();

        // Hae koordinaatit gmv_locations-taulusta
        $location = $wpdb->get_row(
            $wpdb->prepare(
                "SELECT latitude, longitude FROM {$wpdb->prefix}gmv_locations WHERE object_id = %d AND object_type = 'post'",
                $post_id
            ),
            ARRAY_A
        );

        if (!empty($location['latitude']) && !empty($location['longitude'])) {
            $businesses[] = [
                'title' => get_the_title(),
                'url' => get_permalink(),
                'coordinates' => [
                    'lat' => $location['latitude'],
                    'lng' => $location['longitude']
                ],
                'image' => get_field('logo') ? get_field('logo')['url'] : '',
                'overview' => get_field('short_description')
            ];
        }
    }
}

wp_send_json($businesses);

```

Kuva 9. Markkerit gmv\_locations-taulusta (VS Code)

Karttaosiolle haluttiin myös mahdollisuus sulkea se kokonaan pois näkyvistä ja tätä ominaisuutta lähdettiin rakentamaan GitHub Copilotin kanssa. Antamalla Copilotille kontekstiksi kaikki auki olevat tiedostovälilehdet, Copilot pystyi kehittämään koodia eteenpäin niin, että kartta saatiin piilotettua ja jälleen näkyviin. Kehitys eteni jälleen vaiheittain useiden kokeilujen kautta. Lopuksi karttaosion visuaalista ilmettä vielä muotoiltiin yhteistyössä Copilotin kanssa.

#### 5.2.4 Yritysten suodatus

Yritysten suodatussysteemin rakentaminen ajaxilla aiheutti jonkun verran haasteita. Kehittäjä oli tehnyt tämän jo aiemmin käyttäen vain GitHub Copilotia ja prosessi oli silloinkin vaikea eikä lopputulos toiminut täydellisesti. Tällä kertaa prosessiin lähdettiin ChatGPT:n kautta, jolla kuitenkin oli kokonaisvaltaisempi käsitys kontekstista, johon suodatusta oltiin luomassa. ChatGPT neuvoi luomaan tiedoston tag-filter.js ja loi siihen koodin, jonka avulla ajax toimi niin, että kun tägi valitaan, kaikki yrityskortit poistuvat näkyvistä. Tämä ei tietenkään ollut toivottu lopputulos, mutta hyvä lähtökohta. Toiminnon testaus ja virheenetsintä tapahtui GitHub Copilotin avulla maalaamalla koodin osia ja kysymällä niihin liittyviä kysymyksiä. Lopulta usean virheenetsintäkierroksen jälkeen suodatus saatiin toimimaan niin, että sivusto suodatti yritykset oikein, mutta tulosti vain yritysten nimet. Ongelma yritysten löytämisessä oli lopulta funktion `filter_companies_by_tags` parametrien käsittelyssä.

Kun suodatustoiminto saatiin alustavasti kuntoon, pakattiin koko Astran lapsiteema ja liitettiin ChatGPT:n keskusteluun. Näin ChatGPT pystyi tutkimaan tiedostoista missä projektin kanssa mennään, ja ehdottamaan ratkaisua, miten yritysesityskortit saadaan näkyviin. Aluksi näkyviin saatiin vain yritysten nimet ilman visuaalista ulkoasua, koska esitystapa oli kovakoodattu vain yrityslistan tulostavaan tiedostoon (`company-list.php`). ChatGPT ehdotti, että luodaan *template-parts*-osioon erillinen `company-card.php` ja siirretään sinne osa `company-list`-tiedostoon kirjoitetusta koodista. Samalla `company-list.php` vaihdettiin käyttämään uutta, juuri luotua `company-card.php`-tiedostoa, ja tietenkin käytettiin myös `company-card.php`-tiedostoa `functions.php`-tiedostossa korttien luomiseen suodatuksen yhteydessä. Näin saatiin aikaan uudelleenkäytettävä komponentti, jonka ansiosta sekä alkuperäinen lista että AJAX-vastauksena palautuvat kortit säilyivät visuaalisesti yhtenäisinä. ChatGPT kirjoitti koodit ja neuvoi niiden sijoittelut, ja suodatus saatiin tämän muutoksen ansiosta toimimaan kunnolla.

Seuraavaksi toteutettiin yritysten suodatus kartalle samaan tapaan. Tämä tapahtui sujuvasti ja vaivattomasti. ChatGPT ehdotti koodinmuokkauksia ja muutaman korjauskierroksen jälkeen suodatus saatiin toimimaan niin, että sekä yrityskortit että markkerit kartalla päivittyivät samaan tahtiin. Näiden päivitysten yhteydessä karttaosion piilotus, kartan keskittäminen syötettyyn osoitteeseen ja käyttäjän sijainnin käyttäminen kartan keskittämiseen hävisivät. Tämän ratkomiseen käytettiin vuorotellen ChatGPT:tä ja GitHub Copilottia ja lopulta kehittäjä päätteli, että `tag-filter.js` ja `map-filter.js` -tiedostot sisältävät jonkun verran samoja toiminnallisuuksia ja yrittävät toimia päällekkäin, joten päädyttiin korjaamaan tiedostorakennetta niin, että kaikki JavaScript koodit tallennettiin samaan `tag-filter.js` -tiedostoon. Jotta saatiin toimivat ohjelmistokoodit, käyttäjä pakkasi jälleen projektin luodut tiedostot ChatGPT:n luettavaksi ja sitä kautta tiedostojen ohjelmistokoodit saatiin uusittua toimimaan paremmin ja käyttäjän sijainti näkymään kartalle.

### 5.2.5 Sivun viimeistely

Verkkosivun kriittisimmät toiminnot oli nyt saatu toteutettua. Tässä kohtaa kehittäjä pyysi ChatGPT:tä tekemään tarkastuslistan sivun julkaisua varten ja se käytiin kohta kohdalta läpi. Listalta puuttui kuitenkin vielä tágien laskemiseen liittyvän ominaisuuden toteutus. Sivupalkin tágien perään haluttiin sulkuihin yritysten määrät eli kuinka monella yrityksellä on kyseinen tági. ChatGPT suunnitteli toteutusjärjestyksen ja selitti jokaisen vaiheen tarkemmin, kuten kuvasta 10 näkyy. Tekoälyn kirjoittama koodi tuotti suodatustuloksen oikein ensimmäisen tágin valinnalla, mutta toisen valinta ei enää muuttanut tuloksia. Tämä vaati jälleen ChatGPT:ltä monta kierrosta koodin kirjoitusta ja virheenkorjausta. Lopulta ongelma

ratkesi muokkaamalla functions.php -tiedoston ajax-handleria. Tekoäly ehdotti tätä itse, mutta vasta usean epäonnistuneen korjausyrityksen jälkeen.

**Toteutuksen rakenne:**

**1. PHP-puoli (AJAX-handlerin lisättävä):**

- Laskee kaikkien `company_tags` -termien määrän rajoittuen nykyisiin filttereihin
- Palauttaa erillisen taulukon, tyyliin:

```
php
' tag_counts' => [
  'liha' => 4,
  'kasvis' => 2,
  ...
]
```

**2. JS-puoli:**

- Lukee `tag_counts` -taulukon AJAX-vastauksesta
- Päivittää `<label>` -elementtien sisällöt, kuten:

```
html
<label><input type="checkbox" value="liha"> Liha (4)</label>
```

**Päivitän nämä sinulle vaiheittain:**

- ✓ 1. PHP-handlerin lisättävä koodin osa
- ✓ 2. `tag-filter.php`:n muuttaminen niin että `<label>` -rakenteet voidaan päivittää JS:llä
- ✓ 3. `tag-filter.js`:n lisäys joka päivittää määrät suodatuksen yhteydessä

Kuva 10. ChatGPT:n suunnitelma tágien laskemiseen

Rakenteellisena ratkaisuna luotiin vielä erilliset kategoriasivut, joissa esitellään kaikki kyseiseen kategoriaan liittyvät yritykset. Tähän pyydettiin apua ChatGPT:ltä, joka ehdotti toimintajärjestyksen ja toimivat muutokset koodiin. Näiden toteutuksen jälkeen sivun headerin alareunassa näkyi kategorioiden nimet. Nimeä painamalla päätyi erilliselle sivulle, jossa kerrotaan kategoriasta yleiskuvaus ja esitellään kaikki kategoriaan kuuluvat yritykset yrityskortteina, joista taas pääsee yrityksen omalle esittelysivulle. Uusina tiedostoina luotiin `taxonomy-company_category.php` ja `taxonomy-company_category.css`. Tässä oli

mahdollista hyödyntää jo olemassa olevaa koodia, sillä kategoriasivu hyödyntää osittain etusivun yrityskorttien css-koodeja.

ChatGPT:n luomassa tarkastuslistassa oli useampi kohta liittyen sivuston visuaalisiin ratkaisuihin. Suurin muutos oli se, että yksittäisten yritysten esittelysivu järjestettiin niin, että oikealle tuli yhteystiedot ja kartta, vasemmalle muu sisältö. Tähänkin pyydettiin jälleen koodausapua ChatGPT:ltä. Tekoälyn tuottama koodi ei suoraan asetellut sisältöä oikein, mutta toivottuun lopputulokseen päästiin, kun kehittäjä tutki verkkoselaimen kehittäjäikkunasta css-asettelujen vaikutusta lopputulokseen ja välitti nämä tiedot ChatGPT:lle. Muut visuaaliset asetellut ja responsiivisuus toteutettiin GitHub Copilotin ja kehittäjän yhteistyöllä. Ratkaistavat ongelmat olivat yksinkertaisia ja toisaalta kehittäjällä on vahva osaaminen css-koodin kirjoittamisesta, joten työskentely eteni sujuvasti ja tehokkaasti.

Viimeisenä vaiheena sivun kehityksessä oli paikallisesti kehitetyn sivun vieminen verkkoon ja olemassa olevan version korvaaminen uudella versiolla. Tämä toteutettiin All-in-One WP Migration -työkalulla, joka pakkaa kaikki sivuston tiedostot yhdeksi ladattavaksi paketiksi. Pakkauksen jälkeen kirjaututtiin internetissä olemassa olevalle verkkosivulle ja asennettiin sen migraatiotyökalulla paikallisesti kehitetty ja pakattu sivusto käyttöön verkossa olevalle sivupohjalle. Tämän jälkeen kehittäjä pyysi ChatGPT:tä tekemään tarkastuslistan siitä, mitä sivuston julkaisun jälkeen vielä kannattaa tehdä. Lisäksi kehittäjä pyysi tekoälyä päivittämään alkuperäiseen suunnitelmaan asioita, jotka ovat vielä tekemättä. Sivun kehitystyö jatkuu, vaikka tämän opinnäytetyön osalta projekti onkin valmis. Tekoälyn kanssa yhteistyössä tehty lopullinen suunnitelma on liitteenä 2.

## 6 Yhteenveto ja pohdinta

Tämä Makuja kaakosta -verkkosivun kehittäminen aloitettiin keväällä 2024. Projektin alkuvaiheessa hyödynnettiin ChatGPT:n ilmaisversiota sekä perinteisiä tiedonhaun menetelmiä, kuten Googlea ja YouTube -videoita. Näin muodostui alustava ymmärrys verkkosivuston kehitykseen tarvittavista työkaluista ja työvaiheista. Kesällä 2024 projektissa otettiin käyttöön Visual Studio Coden maksullinen lisäosa GitHub Copilot, mikä edisti sivuston kehitystä merkittävästi. GitHub Copilot on maksullinen työkalu, mutta se päätyi käyttöön myös siitä näkökulmasta, että haluttiin arvioida, onko siitä tulevaisuudessakin hinnastaan huolimatta hyötyä verkkosivujen kehittämisprojekteissa. Copilotin avulla sivustosta saatiinkin ensimmäinen versio julkaistua, joten ainakin tässä projektissa Copilotista selkeästi oli hyötyä. Kevättalvella 2025 sivuston kehittämistä jatkettiin, rakentamalla sivusto käytännössä kokonaan uudelleen alusta asti, jotta tekoälyn ja kehittäjän yhteistyötä voitaisiin dokumentoida selkeämmin. Tämä ratkaisu osoittautui hyväksi myös siitä näkökulmasta, että vuoden aikana sekä ChatGPT:n että GitHub Copilotin kielimallit sekä käyttöliittymät olivat parantuneet merkittävästi. Näin sivuston uudelleen rakennus kehitti sivustosta huomattavasti paremman version, kuin mihin aiemmalla kerralla oli päädytty.

Työskentelykieli projektin aikana vaihteli suomen ja englannin välillä. Vuonna 2024 sivuston kehityksessä käytettiin pääasiallisena kehityskielenä englantia, koska Github Copilot vaikutti antavan selkeästi parempia vastauksia englanniksi. ChatGPT:n kanssa kommunikointi suomeksi, mutta 2024 pääasiallinen tekoälykumppani oli GitHub Copilot. Vuonna 2025 siirryttiin intuitiivisesti käyttämään suomen kieltä, koska kehittäjä oli hyödyntänyt tehokkaasti ChatGPT:tä muissa yhteyksissä suomen kielellä. Myös GitHub Copilotin suomen kielen hallinta oli kehittynyt vuoden aikana valtavasti ja yhteistyö suomeksi onnistui aikaisempaa paremmin.

Koko projektissa korostui kehittäjän oma kekseliäisyys ja looginen ajattelu. Ilman perusymmärrystä ohjelmistokoodin toiminnasta projekti ei olisi ollut mahdollinen. Toisaalta tällaisessa työskentelyssä kehittäjän pitää myös osata johtaa omaa oppimistaan ja ymmärtää hyvin mitä ei osaa ja ymmärrä, jotta osaa kysyä juuri oikeanlaisia kysymyksiä tekoälyltä. Paljon törmättiin tilanteisiin, jossa tekoälyn valmiiksi kirjoittamien koodien jälkeen, kehittäjän oli luettava koodi läpi ja huomautettava, esimerkiksi että js -tiedostot eivät ole omassa kansiossaan vaan suoraan lapsiteeman juuressa. Kekseliäisyyttä taas sai käyttää esimerkiksi siinä, että keksi mistä voisi tekoälylle saada mallin, jota se voisi hyödyntää omassa ratkaisussaan. Välillä käytiin tutkimassa kilpailevien verkkosivujen ratkaisuja, välillä omia vanhoja koodeja, niin tämän kuin muidenkin projektien tiedostoista.

Vaikka kehitystyö vaati kehittäjältä melkoista valppautta tekoälyn tarjoamien ratkaisujen arvioijana, oli tekoälystä valtavasti hyötyä. Mitä yksinkertaisemmasta asiasta oli kyse, sitä hyödyllisempi tekoäly oli. Kun käyttäjällä oli perustietämys WordPressin toiminnasta, oli melko suoraviivaista aloittaa projekti tekoälyn ohjeilla. Ongelmien muuttuessa monimutkaisemmiksi, suunnitelmatiedostosta ja lapsiteeman pakkaamisesta keskustelun liitteeksi oli selkeästi apua ChatGPT:lle ja vastaava hyöty GitHub Copilotille oli auki olevien tiedostovä-lilehtien käyttämisestä kontekstina. Siitä huolimatta vastaukset olivat melko usein jollakin tavalla virheellisiä ja vaativat käyttäjältä korjaavia huomautuksia ja verkkoselaimen kehittä-jäikkunan virheviestien liittämistä keskusteluun. Työn edetessä tekoälysovellusten käyttöön muodostui selkeää rutiinia. ChatGPT:ltä kyseltiin enemmän suuriin kokonaisuuksiin vaikut-tavia asioita, kun taas pienemmät koodinosien korjaukset sujuivat näppärästi Copilotin avulla. GitHub Copilotin ominaisuus, jolla tekoälyn kehittämän koodin voi liittää suoraan yh-dellä klikkauksella osaksi lähdekooditiedostoa jo yksikseen nopeuttaa työskentelyä valta-vasti.

Tässä työssä tuli selkeästi todistettua teoriassa esitetty väite, että tekoälystä on hyötyä oh-jelmoijan työparina. Suoraviivaista työskentely ei ole tekoälynkään kanssa, mutta tekoälyä käyttämällä aloitteleva ohjelmoija oppii samalla itse paremmin lukemaan, ymmärtämään ja jopa kirjoittamaan koodia. Tämä kokemus vahvisti myös sen teorian osan, jonka mukaan tekoäly ei pysty ymmärtämään kokonaisuuksia samalla tavalla kuin ihminen. Laajojen ja varsinkin useista tiedostoista koostuvien koodikokonaisuuksien käsittely ja loogisten pää-telmien tekeminen on tekoälylle edelleen hyvin vaikeaa. Joskin tässäkin on edistytty hurjasti jo viimeisen vuoden aikana. Siitä huolimatta projektiin tarvittiin kehittäjän vahvaa loogista ja rakenteellista hallintaa.

Tekoälyn käyttö vahvisti myös sen teoriaosassa esiintuodun väitteen, että kehittäjän tyyty-väisyys lisääntyy tekoälyn hyödyntämisen myötä. Kollegana toimiva tekoäly aloittaa jokai-sen vastauksensa kannustuksella tai pienellä positiivisella palautteella, mikä kohentaa ke-hittäjän mielialaa, lisää luottamusta projektin etenemiseen sekä kehittäjän omaan osaami-seen. Tekoäly toimii myös kehittäjän muistin tukena. Kehitysprosessin aikana kehittäjän huomio kiinnittyy usein yksittäisten teknisten yksityiskohtien ratkaisemiseen, jolloin alkupe-räinen projektisuunnitelma saattaa jäädä taka-alalle. Kun kehittäjä kuvaa tekoälylle nykyi-sen tilanteensa tai tekemänsä vaiheet, tekoäly pystyy arvioimaan, missä vaiheessa projekti on, ja mitä tulisi tehdä seuraavaksi. Näin tekoäly voi auttaa kehittäjää jatkamaan kehitys-työtä suunnitelman mukaisesti ja lisätä luottamusta projektin etenemiseen.

Kokonaisuutena voidaan todeta, että työn tavoitteet tuli saavutettua. Makuja kaakosta - verkkosivu saatiin kehitettyä tekoälyn avustamana. Lisäksi huomattiin, että tekoälyn

käyttäminen työparina tässä ohjelmointiprojektissa tehosti merkittävästi kehitystyötä. Lisäksi se opetti kehittäjälle paljon niin ohjelmoinnista kuin tekoälyn kanssa työskentelystäkin. Tämä prosessi on synnyttänyt paljon uusia ideoita ja käyttötarkoituksia yhteistyölle tekoälyn kanssa. Kehittäjä tulee varmasti käyttämään tiedostoliitteiden mahdollisuutta monenlaisissa projekteissa aina puutarhasuunnittelusta sosiaalisen median julkaisusuunnitelmiin ja toisaalta hyödyntämään tekoälyä ohjelmoinnissa varsinkin front end -ratkaisujen hakemisessa ja sivustojen rakenteellisessa suunnittelussa.

## Lähteet

Advanced Custom Fields. 2023. Post Types and Taxonomies. Overview. Viitattu 1.4.2025. Saatavissa <https://www.advancedcustomfields.com/resources/post-types-and-taxonomies/>

Astra. Do More with Astra Pro. Exclusive Features to Save Time. Viitattu 16.4.2025. Saatavissa <https://wpastra.com/pro/>

BigDATAwire. 2024. IDC: AI to Contribute \$19.9T to Global Economy, Driving 3.5% of GDP by 2030. Viitattu 14.4.2025. Saatavissa [https://www.datanami.com/this-just-in/idc-ai-to-contribute-19-9t-to-global-economy-driving-3-5-of-gdp-by-2030/?utm\\_source=chatgpt.com](https://www.datanami.com/this-just-in/idc-ai-to-contribute-19-9t-to-global-economy-driving-3-5-of-gdp-by-2030/?utm_source=chatgpt.com)

Cimbura, T. 2016. Choosing a Mapping Framework. Viitattu 4.4.2025. Saatavissa [https://luminfire.com/2016/05/10/choosing-mapping-framework/?utm\\_source=chatgpt.com](https://luminfire.com/2016/05/10/choosing-mapping-framework/?utm_source=chatgpt.com)

Dakhel, A., Majdinasab, V., Nikanjam, A., Khomh, F., Desmarais, M. & Jiang, Z. 2023. GitHub Copilot AI pair programmer: Asset or Liability? Viitattu 26.4.2025. Saatavissa <https://arxiv.org/pdf/2206.15331>

Dohmke, T. 2023: Copilot transforms GitHub into the AI-powered developer platform. 2023. Viitattu 20.3.2025. Saatavissa <https://github.blog/2023-11-08-universe-2023-copilot-transforms-github-into-the-ai-powered-developer-platform/>

Dohmke, T., Iansiti, M. & Richards, G. 2023. Sea Change in Software Development: Economic and Productivity Analysis of the AI-Powered Developer Lifecycle. Viitattu 20.3.2025. Saatavissa <https://arxiv.org/pdf/2306.15033>

Elegant themes. Divi. The Essential Site Building Framework for WordPress. Viitattu 16.4.2025. Saatavissa <https://www.elegantthemes.com/gallery/divi/>

Elementor. 2024. What is the Elementor Hello theme? Viitattu 16.4.2025. Saatavissa <https://elementor.com/help/what-is-elementor-hello-theme/>

Elementor. What Is a Website Builder & How Does It Work? Viitattu 16.4.2025. Saatavissa [https://elementor.com/resources/glossary/what-is-a-website-builder-how-does-it-work/?utm\\_source=chatgpt.com](https://elementor.com/resources/glossary/what-is-a-website-builder-how-does-it-work/?utm_source=chatgpt.com)

Friedman, N. 2021. Introducing GitHub Copilot: your AI pair programmer. Viitattu 20.3.2025. Saatavissa <https://github.blog/news-insights/product-news/introducing-github-copilot-ai-pair-programmer/>

- Garderes, G. 2024. How do you build a website in 2024 — CMS vs. frameworks. Viitattu 26.4.2025. Saatavissa <https://medium.com/@g.garderes/how-do-you-build-a-website-in-2024-cms-vs-frameworks-3b1e86ca564e>
- GitHub. a. How GitHub Copilot handles data. Viitattu 28.3.2025. Saatavissa <https://resources.github.com/learn/pathways/copilot/essentials/how-github-copilot-handles-data/>
- GitHub. b. Installing the GitHub Copilot extension in your environment. Viitattu 20.3.2025. Saatavissa <https://docs.github.com/en/copilot/managing-copilot/configure-personal-settings/installing-the-github-copilot-extension-in-your-environment>
- GEO my WP. Simple Plans to Build Smarter Location and Directory Solutions. Viitattu 4.4.2025. Saatavissa <https://geomywp.com/pricing/>
- Hosen, N. 2023. From b2/Cafelog to WordPress: A Journey of Revolutionizing Blogging. Viitattu 20.3.2025. Saatavissa <https://medium.com/@nazmul.ch11/from-b2-cafelog-to-wordpress-a-journey-of-revolutionizing-blogging-35ae1d815e31>
- KokaTic. 2024. OpenLayers vs. Leaflet: A Comparative Guide. Which one can be convenient to my project? Viitattu 4.4.2025. Saatavissa <https://koka-tic.medium.com/openlayers-vs-leaflet-a-comparative-guide-6c4341e82ee8>
- Leaflet. a. Leaflet, an open-source JavaScript library for mobile-friendly interactive maps. Viitattu 6.4.2025. Saatavissa <https://leafletjs.com>
- Leaflet. b. Leaflet Plugins database. Viitattu 6.4.2025. Saatavissa <https://leafletjs.com/plugins.html>
- Mapbox. a. Customer Showcase. Mapbox equips teams of all sizes to innovate with location technologies. Viitattu 4.4.2025. Saatavissa <https://www.mapbox.com/showcase>
- Mapbox. b. Location intelligence for business. Viitattu 4.4.2025. Saatavissa <https://www.mapbox.com>
- Moradi Dakhel, A., Majdinasab, V., Nikanjam, A., Khomh, F., Desmarais, M. C. & Jiang, Z. M. 2023. GitHub Copilot AI pair programmer: Asset or Liability? arXiv. Viitattu 10.3.2025. Saatavissa <https://arxiv.org/abs/2206.15331>
- Microsoft. 2023. Understanding how GitHub Copilot works. Viitattu 5.4.2024. <https://techcommunity.microsoft.com/t5/educator-developer-blog/understandinghow-github-copilot-works/ba-p/4021562>

Microsoft Software. 2025. Visual Studio Code. Viitattu 17.3.2025. Saatavissa <https://code.visualstudio.com>

OpenAI. 2022. Introducing ChatGPT. Viitattu 15.4.2025. Saatavissa <https://openai.com/index/chatgpt/>

OpenAI. 2025. Introducing GPT-4.5. Viitattu 15.4.2025. Saatavissa <https://openai.com/index/introducing-gpt-4-5/>

Rose-Collins, F. 2025. How Geocoding API Works: A Complete Guide. Viitattu 14.4.2025. Saatavissa <https://www.ranktracker.com/blog/how-geocoding-api-works-a-complete-guide/>

ServMask. 2025. All-in-One WP Migration. Quick, easy, and reliable migration plugin for WordPress. Viitattu 25.3.2025. Saatavissa <https://servmask.com>

Stack Overflow. 2024. 2024 Developer Survey. Viitattu 17.3.2025. Saatavissa <https://survey.stackoverflow.co/2024/technology#1-integrated-development-environment>

Stack Overflow. 2023. 2023 Developer Survey. Viitattu 17.3.2025. Saatavissa <https://survey.stackoverflow.co/2023/#section-most-popular-technologies-integrated-development-environment>

Stack Overflow. 2022. 2022 Developer Survey. Viitattu 17.3.2025. Saatavissa <https://survey.stackoverflow.co/2022/#most-popular-technologies-new-collab-tools>

Stack Overflow. 2021. 2021 Developer Survey. Viitattu 17.3.2025. Saatavissa <https://survey.stackoverflow.co/2021#section-most-popular-technologies-integrated-development-environment>

Stack Overflow. 2019. 2019 Developer Survey. Viitattu 17.3.2025. Saatavissa <https://survey.stackoverflow.co/2019#development-environments-and-tools>

Vastbinder, O. 2024. The top 10 most-used WordPress Themes. Viitattu 16.4.2025. Saatavissa [https://www.dataprovider.com/blog/asset-management/the-top-ten-most-used-wordpress-themes/?utm\\_source=chatgpt.com](https://www.dataprovider.com/blog/asset-management/the-top-ten-most-used-wordpress-themes/?utm_source=chatgpt.com)

Warren, T. 2024. GitHub Copilot will support models from Anthropic, Google, and OpenAI. Viitattu 20.3.2025. Saatavissa <https://www.theverge.com/2024/10/29/24282544/github-copilot-multi-model-anthropic-google-open-ai-github-spark-announcement>

WordPress. a. About. Viitattu 21.3.2025. Saatavissa <https://wordpress.org/about/>

WordPress. b. Advanced Custom Fields (ACF®). Viitattu 25.3.2025. Saatavissa <https://wordpress.org/plugins/advanced-custom-fields/>

WordPress. c. Astra. Viitattu 16.4.2025. Saatavissa <https://fi.wordpress.org/themes/astra/>

WordPress. d. GEO my WP. Viitattu 4.4.2025. Saatavissa <https://wordpress.org/plugins/geo-my-wp/>

WordPress. e. Showcase. Viitattu 21.3.2025. Saatavissa <https://wordpress.org/showcase/>

WordPress. 2014. What Is a Theme? Viitattu 16.4.2025. Saatavissa <https://developer.wordpress.org/themes/getting-started/what-is-a-theme/>

WP Engine. a. Getting Started With Local as Your WordPress Development Environment. Viitattu 20.3.2025. Saatavissa <https://wpengine.com/resources/local-wordpress-development-environment-how-to/>

WP Engine. b. Local. Viitattu 12.3.2025. Saatavissa <https://localwp.com>

Liite 1.

ChatGPT:n luoma ensimmäinen suunnitelma verkkosivua varten

## **Verkkosivuprojekti: Makuja Kaakosta**

### **1. Projektin yleiskuvaus**

- **Nimi:** Makuja Kaakosta
- **Tarkoitus:** Esitellä ja koota yhteen kaakkoissuomalaiset lähiruoan tuottajat, myyjät, ravintolat ja maatilamatkailukohteet.
- **Kohderyhmä:** Lähiruoasta kiinnostuneet kuluttajat, ruokamatkailijat ja paikallisten yritysten etsijät.

### **2. Tekniset valinnat**

- **Alusta:** WordPress
- **Teema:** Astra (ilmainen versio)
- **Lisäosat:** Käytetään vain ilmaisia lisäosia
- **Karttapohja:** Esimerkiksi Leaflet (ilmainen avoimen lähdekoodin karttakirjasto) + ilmainen WordPress-lisäosa, joka tukee suodatuksia

### **3. Rakenne ja sisältö**

- **Pääsivut:**
  - Etusivu (kartta + yritysesittelyt + avainsanahaku)
  - Yrityssivu (yksittäinen artikkelisivu jokaiselle yritykselle)
  - Yhteystiedot
  - Tietoa projektista
  - (Mahdollinen) Blogi tai uutisosio
- **Etusivun rakenne:**
  - Kartta, jossa näkyvät kaikki ostetun paikan saaneet yritykset
  - Kartan **vasemmalla puolella:** avainsanahaku (checkboxit)
  - **Kartan alla:** pienet yritysesittelyt
  - Avainsanahaussa näkyy kunkin avainsanan perässä lukumäärä:
    - Alussa: kaikkien artikkelien määrä, joissa kyseinen avainsana esiintyy
    - Valinnan jälkeen: määrä niistä artikkeleista, joissa esiintyy *valitut avainsanat* ja kyseinen avainsana

### **4. Visuaalinen tyyli & brändi**

- **Tyyli:** Luonnonläheinen, selkeä ja helppokäyttöinen
- **Värit:** Vihreän ja ruskean eri sävyt, murretut värit
- **Typografia:** Selkeä, moderni fontti, helposti luettavaa
- **Kuvitus:** Yritysten omat kuvat + yleiskuvat luonnosta ja lähiruoasta

## 5. Toiminnallisuudet

- **Responsiivisuus:** Täysin mobiiliystävällinen
- **Kartta ja suodatus:** Toimii dynaamisesti; suodatukset vaikuttavat karttaan ja esitelyihin sekä avainsanojen lukumääriin
- **Yritysten syöttö:**
  - Ylläpitoon tehdään oma "kevyt backend-osio" jossa voidaan syöttää uusia yrityksiä ilman pääsyä muuhun WordPressin hallintaan
  - Tämä voidaan toteuttaa esim. erillisen käyttäjäroolin avulla ja rajoitetulla näkymällä WordPressin hallintapaneelissa

## 6. Aikataulu ja työvaiheet

- **Vaihe 1:** Suunnittelu (käynnissä)
- **Vaihe 2:** WordPressin asennus ja perusasetukset
- **Vaihe 3:** Kartta + avainsanahaku -toiminnon toteutus
- **Vaihe 4:** Visuaalinen viimeistely ja responsiivisuus
- **Vaihe 5:** Testaus ja julkaisu

---

Tätä suunnitelmaa päivitetään projektin edetessä.

Liite 2.

ChatGPT:n päivittämä suunnitelma, kun verkkosivu on saatu siirrettyä internettiin

## **Verkkosivuprojekti: Makuja Kaakosta**

### **1. Projektin yleiskuvaus**

- **Nimi:** Makuja Kaakosta
- **Tarkoitus:** Esitellä ja koota yhteen kaakkoissuomalaiset lähiruoan tuottajat, myyjät, ravintolat ja maatilamatkailukohteet.
- **Kohderyhmä:** Lähiruoasta kiinnostuneet kuluttajat, ruokamatkailijat ja paikallisten yritysten etsijät.

### **2. Tekniset valinnat**

- **Alusta:** WordPress
- **Teema:** Astra (ilmainen versio)
- **Lapsiteema:** Käytössä on "Astra Child" -lapsiteema, joka on luotu ja aktivoitu kehitystä varten. Se mahdollistaa muokkausten tekemisen turvallisesti ilman, että Astra-teeman päivitykset vaikuttavat niihin.
- **Lisäosat:** Käytetään vain ilmaisia lisäosia
- **Karttapohja:** Leaflet (ilmainen avoimen lähdekoodin karttakirjasto) + GEO my WP -lisäosa (muuntaa osoitteen koordinaateiksi ja sijoittaa yrityksen kartalle ilman Google Mapsin käyttöä)

### **3. Rakenne ja sisältö**

- **Pääsivut:**
  - Etusivu (kartta + yritysesittelyt + avainsanahaku)
  - Yrityssivu (yksittäinen artikkelisivu jokaiselle yritykselle)
  - Yhteystiedot
  - Tietoa projektista
  - Tietosuojaseloste ja evästeinfo (lisätty)
  - (Mahdollinen) Blogi tai uutisosio
- **Etusivun rakenne:**
  - **Ylätunnisteessa (header):**
    - Keskitetty logo (kuva: Makuja Kaakosta)
    - Kielivalinta lippuina (toteutus visuaalisesti, ei vielä toiminnallisuutta, piilotettu toistaiseksi)
    - Kategoriavalikko heti logon alla, suodattaa yrityksiä pääkategorian mukaan
  - **Varsinaisessa sisällössä:**
    - Kartta, jossa näkyvät kaikki ostetun paikan saaneet yritykset
    - Kartan **vasemmalla puolella:** avainsanahaku (checkboxit)
    - **Kartan alla:** pienet yritysesittelyt

- Avainsanahaussa näkyy kunkin avainsanan perässä lukumäärä:
  - Alussa: kaikkien artikkelien määrä, joissa kyseinen avainsana esiintyy
  - Valinnan jälkeen: määrä niistä artikkeleista, joissa esiintyy *valitut avainsanat* ja kyseinen avainsana
- **Footerin sisältö:**
  - Lyhyt kuvaus yhdistyksestä
  - Yhteystiedot
  - Sosiaalisen median linkit
  - Linkit tietosuojaselosteeseen ja evästekäytäntöihin

#### 4. Visuaalinen tyyli & brändi

- **Tyyli:** Luonnonläheinen, selkeä ja helppokäyttöinen
- **Värit:** Vihreän ja ruskean eri sävyt, murretut värit
- **Typografia:** Selkeä, moderni fontti, helposti luettavaa
- **Kuvitus:** Yritysten omat kuvat + yleiskuvat luonnosta ja lähiruoasta

#### 5. Toteutetut toiminnot

- **Responsiivisuus:** Sivut ovat mobiiliystävällisiä alusta lähtien ja mobiilinäkymiä on viimeistelty erikseen
- **Yritysten syöttö:**
  - Ylläpitoon tehty "kevyt backend-osio" jossa voidaan syöttää uusia yrityksiä ilman pääsyä muuhun WordPressin hallintaan
  - Tämä on toteutettu erillisen CPT:n ja ACF-kenttien avulla
- **Yritysesittelyt (Custom Post Type + ACF):**
  - Käytössä on CPT "Company", joka on luotu ohjelmallisesti ja poistettu oletuseditori
  - ACF:llä on määritetty kentät: yrityksen nimi, lyhyt kuvaus, osoite, logo, aukioloajat, some-linkit, yms.
  - Käytetään taksonomiaa company\_tags (tagityyppinen suodatus)
- **Etusivun rakenne:**
  - Header rakennettu logolla, kielivalinnoilla ja kategorianavigaatiolla
  - Etusivulla kartta (Leaflet), tagien suodatus vasemmassa sivupalkissa ja yritysesittelykortit kartan alla
  - Korttien ulkoasu viimeistelty CSS:llä (kuva taustalla, logo päällä, kiinteä koko)
- **Dynaaminen sisältö:**
  - AJAX-pohjainen suodatus toimii: kun käyttäjä valitsee tageja, kartalla ja kortteissa näkyvät vain valitut yritykset
  - Kartta reagoi myös osoitteeseen tai nykyiseen sijaintiin

- Avainsanahaku näyttää nyt myös dynaamisesti kunkin tagin perässä lukumäärän:
  - Ennen suodatusta: kaikkien yritysten määrä per tagi
  - Suodatuksen jälkeen: määrä yrityksiä, joilla on sekä valittu tagi että kyseinen tagi

## 6. Aikataulu ja työvaiheet

- **Vaihe 1:** Suunnittelu ✓
- **Vaihe 2:** WordPressin asennus ja perusasetukset ✓
- **Vaihe 3:** Lapsiteeman luonti ja käyttöönotto ✓
- **Vaihe 4:** Yritysrakenne ja ACF-kentät ✓
- **Vaihe 5:** Etusivun rakenne ja headerin uudistaminen ✓
- **Vaihe 6:** Kartta + tagisuodatus AJAXilla ✓
- **Vaihe 7:** Osoitteen ja nykyisen sijainnin palauttaminen toimintaan ✓
- **Vaihe 8:** Yhdistetään kartta ja kortit yhdeksi AJAX-toiminnoksi ✓
- **Vaihe 9:** Tagien lukumäärät näkyviin ja päivittyvät suodatuksen mukaan ✓
- **Vaihe 10:** Visuaalinen viimeistely ja testaus ✓
- **Vaihe 11:** Julkaisu ✓

## 7. Käyttäjätestauksen ja jatkokehityksen suunnitelma

### Kysyttävät asiat käyttäjiltä:

- Onko navigointi selkeää ja helppoa?
- Miten käyttäjä kokee mobiiliversion käytettävyyden?
- Oliko yrityksen löytäminen helppoa kartan ja suodatuksen avulla?
- Ovatko tagit selkeitä ja hyödyllisiä?
- Onko tekstin koko ja luettavuus riittävä?

### Jatkokehityslistalle:

- Kieliversioiden toteuttaminen (liput piilotettu toistaiseksi)
- Erillinen hakutoiminto
- SEO-parannukset (metaotsikot, kuvien alt-tekstit, yms.)
- Yksittäisten yritysten arviointimahdollisuus tai palautekanava
- Analytiikka- ja evästeseurannan käyttöönotto (esim. Google Analytics, Meta Pixel)
- Template parts -tiedostojen siirtäminen turvallisesti lapsiteeman alle (päivitysten suojaaminen)

---

Tätä suunnitelmaa päivitetään projektin edetessä.