

TEKOÄLYAVUSTEINEN WEB-KEHITTÄMINEN
Verkkosivuston luominen ChatGPT:n ja bolt.new'n avulla

Kaarna Matias
Uomala Santeri

Opinnäytetyö

Tietojenkäsittelyn koulutus
Tradenomi (AMK)

2025

Tietojenkäsittelyn koulutus
Tradenomi (AMK)

Tekijä(t)	Matias Kaarna Santeri Uomala	Vuosi	2025
Ohjaaja(t)	Yrjö Koskenniemi		
Työn nimi	Tekoälyavusteinen web-kehittäminen		
Sivumäärä	38		

Opinnäytetyön tavoitteena oli selvittää, miten tekoälyavusteiset työkalut voivat tukea ja mahdollisesti korvata perinteisiä web-kehityksen menetelmiä. Työssä toteutettiin yksi toimiva verkkosovellus, ROI-laskuri, hyödyntäen OpenAI:n ChatGPT:tä ja StackBlitzin bolt.new-työkalua ilman manuaalista ohjelmointia. Erityisesti tarkasteltiin, kuinka tehokkaasti tekoäly kykenee toteuttamaan verkkosivuston teknisen rakenteen ja toiminnallisuudet pelkästään kehoitteiden eli promptien avulla.

Pääasiallisena tutkimuskysymyksenä oli, miten tekoälyavusteiset työkalut voivat korvata perinteisiä web-kehityksen prosesseja. Tutkimyksissä selvitettiin tekoälyn vaikutuksia kehitysprosessin aikatauluun, kustannuksiin ja työn laatuun. Tietoperustassa käsiteltiin web-kehityksen osa-alueita, generatiivista tekoälyä sekä promptaamista osana kehitysprosessia.

Tutkimus toteutettiin konstruktivisena kehittämistutkimuksena, jossa kehitysprosessin vaiheet dokumentoitiin ja analysoitiin laadullisesti. Sovelluksen toteutus perustui iteratiiviseen kehoitteiden muotoiluun ja tekoälyn antamien vastausten jatkuvaan kehittämiseen. Keskeiseksi havainnoksi nousi promptaamisen merkitys: tekoälyn onnistunut hyödyntäminen vaatii selkeää viestintää ja ymmärrystä kehityksen perusteista.

Opinnäytetyön tuloksena syntyi toimiva ROI-laskuri, joka julkaistiin web-sovelluksena ja jota voidaan hyödyntää asiakastyössä automaatiopalvelujen kustannustehokkuuden todentamiseen. Tulokset osoittavat, että pienimuotoiset verkkosivustot voidaan toteuttaa nopeasti ja kustannustehokkaasti tekoälytyökalujen avulla, kunhan käyttäjällä on riittävä ymmärrys web-kehityksen perusteista ja hyvä kehoitteiden hallinta. Tämä opinnäytetyö on tehty osana Lapin tekoälykiihdyttämö-hanketta.

Avainsanat	generatiivinen tekoäly, promptit, web-ohjelmointi, kieli-mallit, kehittäminen
Muita tietoja	Työhön liittyy toimeksiantajalle toimitettu ROI-laskuri verkkosovellus henkilötöiden ja automaation kustannusten vertailuun.

Business Information Technology
Bachelor of Business Administration

Author(s)	Matias Kaarna Santeri Uomala	Year	2025
Supervisor(s)	Yrjö Koskenniemi		
Title	AI-assisted web development		
Number of pages	38		

The objective of this thesis was to explore how AI-assisted tools can support or even replace traditional methods of web development. A single working web application, an ROI calculator was created using Open AI's ChatGPT and the bolt.new tool from StackBlitz, without manual programming. The focus was on how efficiently artificial intelligence can generate the technical structure and functionality of a website using prompts alone.

The main research question was how AI-based tools can replace traditional web development practices. Supporting questions examined how AI affects the project timeline, costs, and quality. The theoretical framework covered key areas of web development, generative AI, and prompting as part of the development workflow.

The study was carried out as a constructive development research, where the implementation process was thoroughly documented and qualitatively analyzed. The application was developed through iterative prompting and refinement of AI-generated outputs. A key finding was the critical role of prompting: successful use of AI requires precise instructions and a fundamental understanding of development concepts.

The outcome of the thesis was a fully functioning ROI calculator, published as a web application, which can be used in client meetings to demonstrate the cost-effectiveness of automation services. The results show that small-scale websites can be built efficiently using AI tools, provided the user understands the basics of web development and prompt design. This thesis has been completed as part of the Lapin tekoälykiihdyttämö-project.

Keywords	generative artificial intelligence, prompts, web programming, language models, development
Special remarks	The thesis includes a web-based ROI calculator application delivered to the commissioner for comparing the costs of manual work and automation.

SISÄLLYS

1	JOHDANTO	5
2	WEB-KEHITTÄMINEN JA TEKOÄLY	7
2.1	Web-kehittäminen.....	7
2.2	Generatiivinen tekoäly	8
3	VERKKOSIVUSTON TOTEUTUS JA KÄYTETTÄVÄT MENETELMÄT	9
3.1	Tutkimus- ja kehittämismenetelmät	9
3.2	Tutkimuskysymys	10
4	TEKOÄLYTYÖKALUT JA KEHOTTEET KOODIN TAKANA	12
4.1	Promptaaminen web-kehittämisen välineenä	12
4.2	Tekoälytyökalut web-kehittämisen tukena	13
5	VERKKOSOVELLUKSEN KEHITYSPROSESSI: ROI LASKURI	15
5.1	ROI-laskurin rakentaminen tekoälyavusteisesti	15
5.2	Ensimmäinen versio: Perusrakenne ja havaittuja puutteita	17
5.3	Toinen versio: Parannettu sisältö, teknisiä haasteita	18
5.4	Kolmas versio: Toimiva analyysi ja interaktiivisuus	20
5.5	Neljäs versio: Sisäinen virhe ja automaattinen korjaus.....	21
5.6	Viides versio: Useiden tehtävien tuki ja rakenteelliset parannukset....	23
5.7	Kuudes versio: Analyysin tarkkuus ja yhdistelmä-laskenta	25
5.8	Seitsemäs versio: Lopullinen toimiva kokonaisuus.....	26
5.9	Kahdeksas versio: Julkaisu verkkoon ja käyttöönotto.....	26
5.10	Julkaisun ongelmat ja ratkaisuprosessi	27
6	POHDINTA	31
6.1	Toteutusprosessin arviointi	31
6.2	Tekoälyavusteinen vastaan perinteinen web-kehitys.....	32
6.3	Web-kehityksen tulevaisuus tekoälyn aikakaudella	32
6.4	Opinnäytetyön eettisyys.....	32
6.5	Opinnäytetyön luotettavuus	33
6.6	Oman oppimisen reflektointi	33
	LÄHTEET.....	35

1 JOHDANTO

Verkkopalveluiden määrä ja merkitys ovat kasvaneet huomattavasti digitalisaation myötä ja nykypäivänä ne ovat keskeinen osa yritysten liiketoimintaa ja kuluttajien arkea. Kuluttajien vaatimusten kasvaessa web-kehittäjiltä vaaditaan laaja-alaista osaamista niin ohjelmoinnista, käyttöliittymäsuunnittelusta kuin sisällöntuotannosta. Web-kehitys on verkkosivustojen ja -sovellusten suunnittelua, kehittämistä ja ylläpitoa. Perinteisesti web-kehitys on ollut manuaalinen prosessi, jossa kehittäjät kirjoittavat koodia käsin ja suunnittelevat käyttöliittymät yksityiskohtaisesti. Teknologian kehitys on kuitenkin mahdollistanut uusien työkalujen käyttöönoton, ja ne nopeuttavat ja helpottavat kehittämisprosessia. Erityisesti tekoälyn rooli web-kehityksessä on noussut merkittäväksi viime vuosina. (GeeksforGeeks 2025; Ojanperä 2023, 1.)

Tekoäly (AI) on teknologia, joka antaa tietokoneille kyvyn jäljitellä ihmisen oppimista, ymmärtämistä, ongelmanratkaisua, päätöksentekoa, luovuutta ja itenäistä toimintaa (Stryker 2024). Web-kehityksessä tekoälyavusteisten työkalujen käyttö on noussut kiinnostuksen kohteeksi, sillä niiden avulla voidaan nopeuttaa ja automatisoida monia kehitysprosessin vaiheita, kuten koodin automaattista generointia, sisällön luomista sekä käyttäjäkokemuksen personointia (Ojanperä 2023, 1). DNA Oy:n yritysliiketoimintajohtaja Anna-Mari Ylihurulan mukaan yritysten tulisi aktiivisesti testata uusia tekoälyratkaisuja pienessä mittakaavassa, analysoida niiden tulokset ja arvioida, millaiset kehitysprosessit tuottavat suurimman lisäarvon. Tämä iteratiivinen lähestymistapa vähentää riskejä, nopeuttaa innovointia ja tukee tekoälyyn perustuvan web-kehityksen tehokasta käyttöönottoa. Samalla yritysjohto voi varmistaa, että tekoälyn käyttöön liittyvät mahdollisuudet ja uhat ymmärretään laajasti organisaation sisällä, mikä edistää tekoälyn vastuullista hyödyntämistä web-kehittämisen eri vaiheissa (DNA 2025.) McKinseyn raportin mukaan generatiivinen tekoäly voi lisätä globaalia taloudellista arvoa vuosittain jopa 2,6–4,4 biljoonaa dollaria (Chui ym. 2023).

Opinnäytetyössämme tutkimme, miten tekoälyavusteinen web-kehittäminen voi vaikuttaa nykyisiin kehitysprosesseihin ja arvioimme sen mahdollisuuksia korvata perinteisiä web-kehityksen menetelmiä. Tietojenkäsittelyn opinnoissamme

olemme perehtyneet web-kehittämisen perusteisiin, sekä pinnalla oleviin teknologiatrendeihin. Web-kehitys on keskeinen osa nykyaikaista digitaalista liiketoimintaa, ja tekoälyn merkitys siinä kasvaa jatkuvasti. Generatiivisen tekoälyn hyödyntäminen verkkosivustojen suunnittelussa voi mahdollistaa yrityksille tehokkaampia ja automatisoituja kehitysprosesseja. (Chui ym. 2023.)

Tekoälyn avulla voidaan saavuttaa merkittäviä kustannussäästöjä web-kehittämisessä ja nopeuttaa verkkosivustojen käyttöönottoa. Tämä on erityisen tärkeää alueellisille yrityksille ja pk-sektorille, joilla ei aina ole resursseja kehittää verkkopalveluita. Lisäksi tekoälyn avulla voidaan paikata IT-alaa vaivannutta osaamisvajetta: Gartnerin arvion mukaan generatiivisen tekoälyn käyttöönotto tulee romahduttamaan osaamisvajeen ja esimerkiksi poistamaan erikoiskoulutuksen tarpeen jopa 50 prosentista kyberturvallisuuden aloitustason tehtävistä vuoteen 2028 mennessä. (Keen 2024). Laajemmin katsottuna opinnäytetyö tuottaa tietoa myös muille kehittäjille, yrityksille ja oppilaitoksille siitä, miten tekoäly voi toimia osana modernia sovelluskehitystä ja millaisia haasteita se tuo mukanaan.

Opinnäytetyömme tavoitteena on kehittää täysin toimiva verkkosivusto hyödyntämällä ainoastaan StackBlitzin bolt.new tekoälytyökalua ja OpenAI:n ChatGPT-tekoälykielimalia. Haluamme selvittää, kuinka hyvin tekoäly voi automatisoida ja tehostaa web-kehitysprosessia, sekä arvioida sen soveltuvuutta perinteisiin kehityskäytäntöihin. Opinnäytetyön tuotoksena on tekoälyavusteisesti kehitetty verkkopohjainen ROI-laskuri.

2 WEB-KEHITTÄMINEN JA TEKOÄLY

Opinnäytetyömme keskeisiä käsitteitä ovat web-kehittäminen (web development), tekoäly (Artificial Intelligence tai AI), joka web-kehittämisen näkökulmasta käsittelee pääasiassa generatiivista tekoälyä (Generative Artificial Intelligence tai Gen AI), sekä promptaaminen. Promptaaminen tarkoittaa tekoälylle annettavaa ohjetta tai pyyntöä, jonka perusteella se tuottaa vastauksen. Web-kehityksessä promptit voivat olla esimerkiksi koodipyyntöjä tai suunnitteluohjeita, joihin tekoäly reagoi tuottamalla käyttökelpoisia ratkaisuja (Sheldon & Yasar 2024). Hyvin laadittu prompti sisältää selkeän tehtävänannon ja tarvittavan kontekstin, mikä parantaa vastauksen laatua. Epätarkka kehoitus voi puolestaan johtaa virheelliseen tai epäolennaiseen tulokseen (Covisian 2025). Taitava promptaaminen onkin keskeinen osa tekoälyavusteista kehitystyötä ja vaikuttaa ratkaisevasti kehitysprosessin onnistumiseen.

2.1 Web-kehittäminen

Verkko- tai web-kehittäminen (web development) viittaa verkkosivustojen luomiseen, rakentamiseen ja ylläpitoon. Se kattaa osa-alueita kuten verkkosuunnittelun, verkkosivujen julkaisemisen, web-ohjelmoinnin ja tietokantahallinnan. Verkkokehitys tarkoittaa sellaisen sovelluksen luomista, joka toimii internetin kautta, eli verkkosivustoja. (GeeksforGeeks 2025.)

Web-kehittäminen jaetaan tyypillisesti kolmeen keskeiseen osa-alueeseen. Frontend kehittäminen sisältää web-kehittämisen toiminnot, jotka liittyvät sovelluksen tai verkkosivuston käyttöliittymän kehittämiseen (Kitthu 2024). Backend kehittäminen puolestaan sisältää web-kehittämisen toiminnot, jotka suoritetaan ohjelmien taustalla, kuten palvelimien verkkosovelluslogiikan määrittely tai rajapintojen integrointi (Deshpande 2025). Full-stack -kehittäminen tarkoittaa kokonaisvaltaista kehittämistä, joka kattaa sekä frontendin, että backendin (GeeksforGeeks 2025).

2.2 Generatiivinen tekoäly

Tekoäly on yksi nykypäivän puhutuimmista teknologioista, ja se on noussut keskeiseksi teemaksi niin tieteellisessä tutkimuksessa, liiketoiminnassa kuin yhteiskunnallisessa keskustelussa ja aiheesta riittää kirjallisuutta, webinaareja ja podcasteja. Tekoäly (AI) on teknologia, joka antaa tietokoneille ja koneille kyvyn jäljitellä ihmisen oppimista, ymmärtämistä, ongelmanratkaisua, päätöksentekoa, luovuutta ja itsenäistä toimintaa. Tekoälyä hyödyntävät sovellukset ja laitteet voivat havaita ja tunnistaa esineitä, tulkita ja vastata ihmiskieleen, oppia kokemuksista ja uudesta tiedosta sekä tarjota tarkkoja suosituksia käyttäjille ja asiantuntijoille. (Stryker 2024.)

Generatiivinen tekoäly viittaa tekoälymalleihin ja -tekniikoihin, jotka on suunniteltu luomaan uusia datanäytteitä, jotka muistuttavat luonteeltaan annettua syötedataa. Tavoitteena on tuottaa sisältöä tai dataa, joka ei ollut osa alkuperäistä koulutusjoukkoa, mutta on silti johdonmukaista, kontekstuaalisesti relevanttia ja samanlaista tyyliltään tai rakenteeltaan. (Campesato 2023, 2.) Generatiivinen tekoäly perustuu syväoppimiseen ja erityisesti neuronaalisiin verkkoihin, joita hyödynnetään monilla aloilla, kuten kuvantunnistuksessa, luovassa suunnittelussa, tekstin generoinnissa ja kääntämisessä ja pelikehityksessä (Winter 2025).

3 VERKKOSIVUSTON TOTEUTUS JA KÄYTETTÄVÄT MENETELMÄT

Opinnäytetyömme tarkoituksena on selvittää tekoälyavusteisen web-kehityksen mahdollisuuksia ja rajoitteita toteuttamalla verkkosivusto täysin tekoälytyökalujen avulla. Tämä tutkimus on erityisen ajankohtainen, sillä tekoälyn odotetaan muuttavan ohjelmistokehityksen prosesseja ja nopeuttavan web-kehitystä automatisoimalla tehtäviä, parantamalla käyttäjäkokemuksia sekä optimoimalla verkkosivustojen suorituskykyä (Jadhav & Gholve 2024).

Opinnäytetyön tavoitteena on tuottaa konkreettista tietoa tekoälyavusteisen web-kehityksen tehokkuudesta ja käytännön soveltuvuudesta. Pyrimme selvittämään, voiko tekoäly todella vastata IT-alan osaamisvajeeseen ja missä määrin se voi korvata perinteisiä kehitysmenetelmiä. Tekoälytyökalut parantavat verkkokehityksen automaatiota, kustomointia ja päätöksentekoa, mikä vähentää manuaalisen koodauksen tarvetta ja inhimillisiä virheitä (Upadhyaya 2024). Lisäksi tavoitteena on tarjota pk-yrityksille ja kehittäjille käytännön ohjeistusta tekoälytyökalujen tehokkaaseen hyödyntämiseen web-kehityksessä. Tämä opinnäytetyö on tehty osana Lapin tekoälykiihdyttämö-hanketta.

3.1 Tutkimus- ja kehittämismenetelmät

Opinnäytetyö toteutetaan käytännönläheisenä kehittämistutkimuksena eli konstruktiivisena tutkimuksena. Tämä tarkoittaa, että luomme konkreettisen ratkaisun (verkkosivuston) todelliseen ongelmaan ja samalla tuotamme uutta tietoa prosessista. Konstruktiivinen lähestymistapa ohjelmistokehityksessä mahdollistaa tekoälyavusteisten työkalujen tehokkuuden arvioinnin käytännön projektissa (Shynkarenko & Zhevaho 2020).

Lähestymistapamme perustuu selkeään tavoitteeseen (toimiva verkkosivusto), oikeisiin työkaluihin (tekoälyavusteiset ratkaisut kuten StackBlitzin bolt.new ja ChatGPT-4), ja prosessin aikana opimme, miten nämä työkalut toimivat käytännössä. Tekoäly on jo osoittanut potentiaalinsa automatisoimalla ohjelmistokehityksen eri osa-alueita, kuten testausprosessia ja virheenkorjausta, mikä voi lyhentää kehityssyklejä ja parantaa ohjelmiston laatua (Bhuvana, Ranjan & Bhaduariya, 2023).

Konstruktiiivinen tutkimus mahdollistaa innovatiivisten menetelmien kokeilemisen ja uusien ratkaisujen kehittämisen ohjelmistokehityksen haasteisiin. Tekoälyteknologiat voivat merkittävästi nopeuttaa ja automatisoida ohjelmistokehitysprosessia, mutta ne tuovat mukanaan myös haasteita, kuten tekoälyn eettiset kysymykset ja mallien tulkittavuus (Sathe & Panse 2023).

Kehitysprosessin aikana dokumentoimme järjestelmällisesti tekoälyn kanssa käydyt keskustelut sekä niiden tulokset. Lisäksi kirjaamme ylös tekoälyllä tuotettujen ratkaisujen eri vaiheet ja arvioimme niiden toimivuutta. Raportoimme myös prosessin aikana esiin nousseet haasteet ja niihin löytämämme ratkaisut. Lisäksi seuraamme ja dokumentoimme ajankäyttöä sekä tarvittavia resursseja.

Tämä dokumentaatio toimii tutkimusaineistona, jota analysoimme laadullisen sisällönanalyysin menetelmin. Kuten Cico, Çiço ja Cico (2023) korostavat, tekoälyavusteisen ohjelmistokehityksen arvioinnissa on tärkeää tarkastella, miten tekoäly vaikuttaa eri ohjelmistokehityksen osa-alueisiin, kuten vaatimustenhallintaan, suunnitteluun, kehitykseen, testaukseen ja ylläpitoon. Tämän opinnäytetyön tavoitteena on saada syvällistä ymmärrystä siitä, miten tekoäly voi tukea web-kehitystä, ja missä sen rajoitteet ilmenevät.

3.2 Tutkimuskysymys

Opinnäytetyössä vastataan pääkysymykseen miten tekoälyavusteiset työkalut voivat korvata perinteisiä web-kehityksen menetelmiä. Tämä kysymys on keskeinen, sillä se auttaa ymmärtämään tekoälyn todellisen potentiaalin web-kehityksen automatisoinnissa. Tekoälyavusteinen ohjelmointi voi nopeuttaa ohjelmistokehitysprosessia ja tarjota uusia tapoja sovellusten luomiseen (Kosuru 2024). Pääkysymyksen tueksi, tutkimuksessa vastataan tukikysymyksiin, kuten millaisia laadullisia eroja on havaittavissa tekoälyavusteisen ja perinteisen web-kehityksen välillä, ja miten tekoälyavusteinen kehittäminen vaikuttaa web-kehitysprojektin aikatauluun ja kustannuksiin. Kysymykset ovat olennaisia erityisesti pk-yrittysten näkökulmasta, jotka etsivät kustannustehokkaita kehitysratkaisuja. Tekoäly voi vaikuttaa merkittävästi web-kehityksen aikatauluun ja kustannuksiin nopeuttamalla koodin kirjoittamista, testausprosessia ja verkkosivustojen ylläpitoa

(Ayyagiri, Goel & Renuka 2024). Lisäksi ne auttavat arvioimaan tekoälyn tuottaman koodin ja sisällön laatua verrattuna perinteisiin menetelmiin (Nanjundappa ym. 2020).

4 TEKOÄLYTYÖKALUT JA KEHOTTEET KOODIN TAKANA

4.1 Promptaaminen web-kehittämisen välineenä

Tekoälyavusteisessa web-kehityksessä promptaaminen toimii siltana ihmisen ajatuksen ja koneen toiminnan välillä. Promptilla tarkoitetaan käyttäjän antamaa ohjetta tai kysymystä, jonka perusteella tekoäly, kuten ChatGPT, generoi sisältöä tai suorittaa tietyn tehtävän. Web-kehityksen kontekstissa promptit voivat olla esimerkiksi koodipyyntöjä, suunnitteluohjeita tai rakenteellisia luonnoksia, joiden avulla kehittäjä voi rakentaa verkkosivuston tai yksittäisiä komponentteja (Sheldon & Yasar 2024).

Promptaamisen voima web-kehityksessä perustuu sen kykyyn tuottaa nopeasti käyttökelpoisia ratkaisuja monimutkaisiin ongelmiin. Esimerkiksi HTML-lomakkeen, JavaScript-funktion tai CSS-animaation luominen voidaan toteuttaa muutamalla hyvin suunnitellulla syötteellä. Kun kehittäjä määrittelee selkeästi tehtävän, kontekstin ja toivotun lopputuloksen, tekoäly pystyy vastaamaan tarkasti odotuksiin ja tuottamaan laadukasta, käyttövalmista koodia (Covisian 2025). Tämä ei ainoastaan nopeuta kehitysprosessia, vaan mahdollistaa myös luovemman ja kokeilevemmän työskentelytavan.

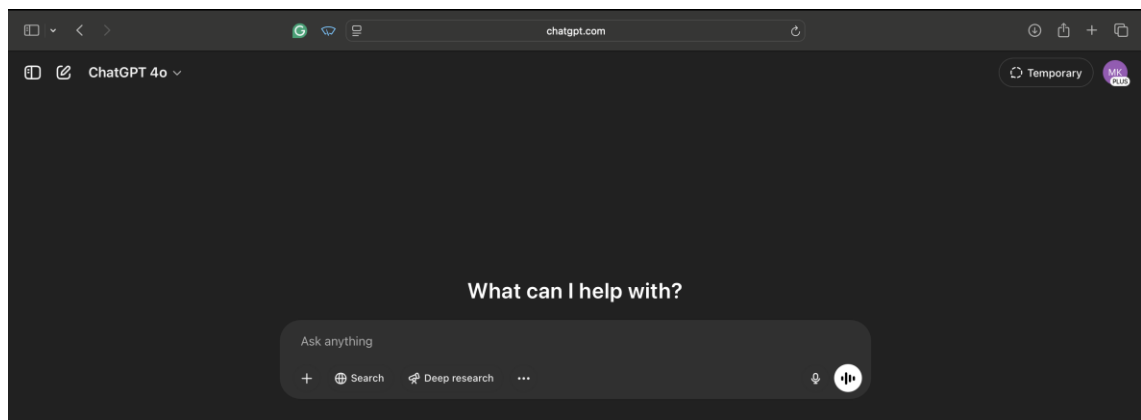
Tehokas promptaaminen edellyttää kuitenkin taitoa. Epätasällinen tai liian yleinen kehoitus voi johtaa epäolennaisiin tai virheellisiin tuloksiin. Toisaalta hyvin muotoiltu prompti, jossa on mukana kontekstitietoa, tavoite, esimerkkejä tai toivottu koodiformaatti, voi tuottaa tarkkoja, tarkoituksenmukaisia ja jopa innovatiivisia ratkaisuja (Sheldon & Yasar 2024). Web-kehityksen näkökulmasta tämä tarkoittaa, että promptien suunnittelu on olennainen osa kehittäjän työtä, aivan kuten arkkitehtuurivalinnat tai versionhallinta.

Tekoälyn ja promptaamisen yhdistelmä tuo web-kehitykseen uudenlaista ketteryyttä. Kehittäjä voi iteratiivisesti testata eri lähestymistapoja, pyytää vaihtoehtoisia ratkaisuja ja jatkojalostaa aiempia vastauksia rakentamalla keskustelua tekoälyn kanssa. Tämä mahdollistaa tehokkaan prototyyppien luonnin, nopeamman päätöksenteon ja aiempaa visuaalisemman suunnitteluprosessin, jossa tekoäly toimii käytännössä assistenttina ja kumppanina (Covisian 2025).

Promptaaminen ei ole vain tekninen taito, se on vuorovaikutusta tekoälyn kanssa. Taito kysyä oikein on noussut tärkeäksi osaksi modernia web-kehitystä, erityisesti silloin, kun kehitystyössä hyödynnetään generatiivisia tekoälymalleja. Tämän vuoksi promptaaminen on merkittävä osa tätä opinnäytetyötä ja keskeinen väline tekoälyavusteisen kehitysprosessin onnistumisessa.

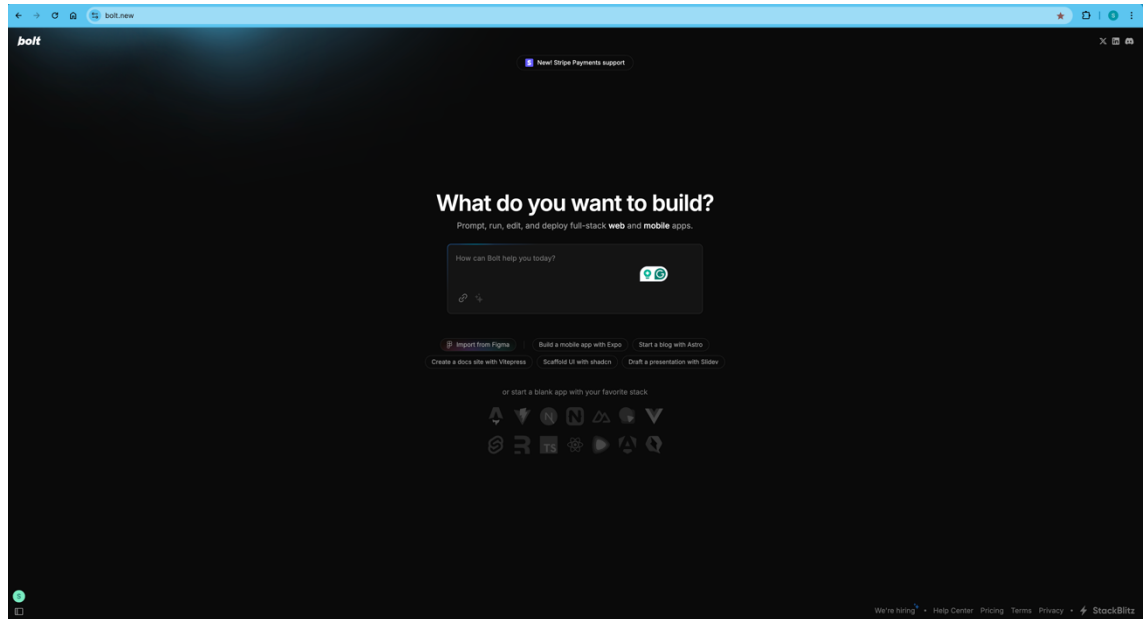
4.2 Tekoälytyökalut web-kehittämisen tukena

ChatGPT on OpenAI:n kehittämä, suuren kielimallin (GPT) pohjalta rakennettu keskusteleva tekoälytyökalu (kuvio 1). Se pystyy tuottamaan kirjoitettua tekstiä monipuolisesti, vastaamaan kysymyksiin sekä tukemaan ohjelmistokehittäjiä koodin luonnissa. ChatGPT:stä tuli nopeasti suosittu sen helppokäyttöisyyden ja ilmaisen julkisen version ansiosta (Puro 2023). Se madaltaa kynnystä kokeilla tekoälyä eri tehtävissä ja on kerännyt valtavan käyttäjäkunnan (Ranta 2025).



Kuvio 1. Open AI:n ChatGPT-tekoälytyökalu

Bolt.new (kuvio 2) on StackBlitzin kehittämä työkalu, joka yhdistää generatiivisen tekoälyn selainpohjaiseen ohjelmointiympäristöön (Capeletto 2024). Käyttäjä voi määrittellä, millaisen sovelluksen haluaa, ja Bolt.new rakentaa automaattisesti perusrakenteen, asentaa kirjastot ja käynnistää sovelluksen. Tämä tekee full stack -kehittämisestä nopeampaa: yksi komento voi riittää vaikkapa Node.js-palvelimen pystytykseen. Työkalu on kerännyt mainetta lupaamalla helpottaa kehitysprosessia etenkin prototyyppien teossa (Kuzmanovski 2024).



Kuvio 2. StackBlitzin Bolt.new-tekoälytyökalu

ChatGPT toimii yleisenä tekoälyavustajana, jota voi hyödyntää koodauksen lisäksi myös sisällön luonnissa ja ongelmanratkaisussa. Bolt.new on puolestaan suunniteltu web-kehittämisen nopeuttamiseen, automatisoiden isoja osia koodaustyöstä (Stack Overflow 2024). Näin saamme opinnäytetyössämme vertailtua sekä laaja-alaisen, että kapeammin kehitykseen suunnatun tekoälyn vaikutusta web-projektien rakentamiseen. Molemmat työkalut ovat ajankohtaisia esimerkkejä siitä, miten tekoäly voi mullistaa ohjelmistokehitystä.

5 VERKKOSOVELLUKSEN KEHITYSPROSESSI: ROI LASKURI

5.1 ROI-laskurin rakentaminen tekoälyavusteisesti

Toteutusvaiheessa päätimme rakentaa verkkosivuston, joka sisältää automaatioalan asiantuntijalle tarkoitetun ROI-laskurin (Return on Investment). Sivuston tarkoituksena on auttaa konsulttia perustelemaan palveluidensa kustannukset asiakkailleen konkreettisten säästölaskelmien avulla. Projektissa hyödynsimme kahta tekoälytyökalua ChatGPT:tä ja StackBlitzin bolt.new-alustaa, vertaillaksemme, kuinka ne reagoivat samoihin prompteihin ja kuinka hyvin ne tukevat sovelluksen kehittämistä eri vaiheissa. Prompteja kirjoittaessa oleellista oli tehdä ne kyseessä olevan toimialan näkökulmasta.

Ensimmäinen promptti, jonka molemmille alustoille syötimme, oli seuraava:

"I work as an automation consultant and I'm having a hard time to justify my clients the cost of my services. I want to build an automation ROI calculator where I can add the tasks we would be building for the customer, how many people are doing the task, how many times is it done and in what interval and how much is the hourly cost of the employee. I then want to add my monthly cost, and I want it to create an ROI analysis with data points you think are needed and enough to justify. Create a prompt for this calculator that I can use in bolt.new."

ChatGPT tuotti rakenteeltaan loogisen ja selkeän kenttäluettelon (kuvio 3), jossa otettiin huomioon muun muassa työntekijämäärä, toistuvuus, tehtävään kuluva aika, työntekijän tuntikustannus sekä kuukausittainen automaatiopalvelun hinta. Lisäksi ChatGPT lisäsi laskentakaavoja, joilla voidaan arvioida säästöjä ja tuottoa ennen ja jälkeen automaatiopalvelun käyttöönottoa. Vaikka rakenne oli järkevä ja kattava, se sisälsi useita vaiheita, jotka edellyttivät manuaalista kaavojen tulkin-
taa eikä ollut vielä suoraan käytettävissä käyttöliittymässä.

Automation ROI Calculator

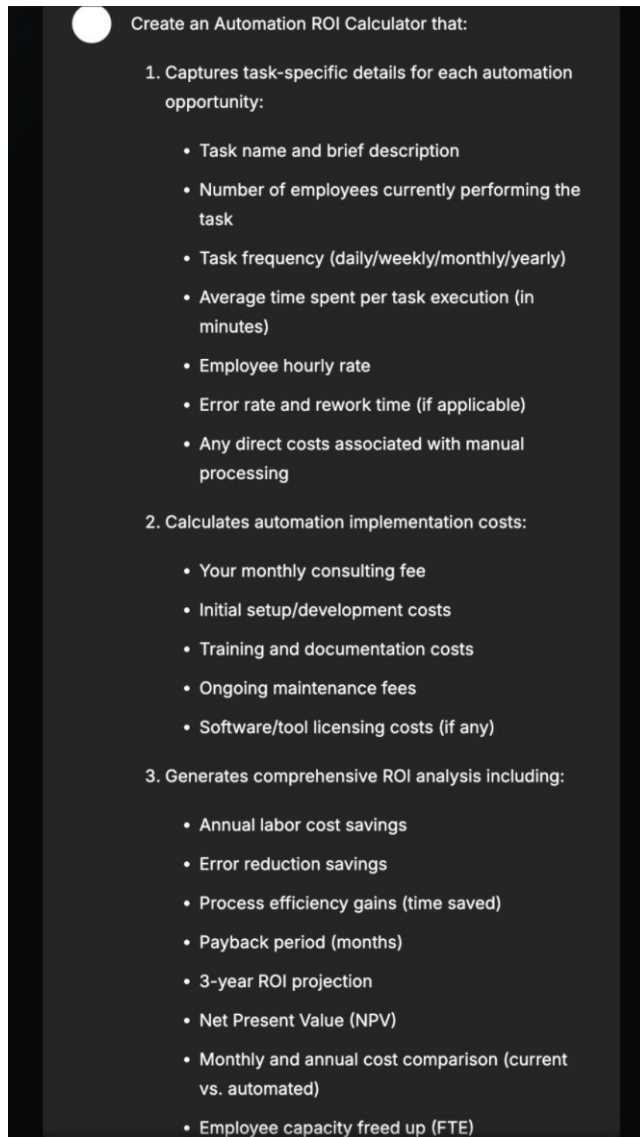
1. **Task Name**
Short description of the process being automated.
2. **Number of People Doing This Task**
How many employees or team members are responsible for this process?
3. **Frequency of Task**
How often is it done? (e.g., times per day/week/month)
4. **Average Minutes Per Task**
How long does each instance of the task take for one person?
5. **Hourly Cost of Employee**
Rough hourly pay (including overhead if relevant).
6. **Your Monthly Automation Service Cost**
The fee you charge to automate these tasks.
7. **Expected Time Saved**
(Calculation step) $(Number\ of\ People) \times (Frequency) \times (Minutes / 60)$
You'll convert minutes to hours, multiply by the hourly cost, and see the total labor cost saved.
8. **Monthly Savings**
(Calculation step) $(Expected\ Time\ Saved\ in\ Hours) \times (Hourly\ Cost)$
This is the amount of money the client is no longer burning on manual labor.
9. **ROI**
(Calculation step) $(Monthly\ Savings - Your\ Monthly\ Automation\ Fee) / Your\ Monthly\ Automation\ Fee \times 100$
Determines whether the automation cost is justified by the savings.

Final Output:

- **Annual Savings** = Monthly Savings x 12
- **Net ROI** = $(Annual\ Savings - (Automation\ Fee \times 12)) / (Automation\ Fee \times 12) \times 100\%$

Kuvio 3. Chat GPT:n luoma prompti bolt.new työkalua varten

Bolt.new tuotti huomattavasti laajemman, teknisemmän ja automaation kannalta syvällisemmän lähestymistavan (kuvio 4). Se sisälsi useita lisäelementtejä, kuten virheprosentin ja uudelleentyöstön aikakustannukset, kehitys- ja koulutuskustannukset sekä visuaaliset analyysit ROI:n kehittymisestä kolmen vuoden ajanjaksolla. Prompti oli rakenteeltaan monipuolisempi ja näytti ymmärtävän sekä kehittäjän että liiketoiminnan tarpeet huomattavasti laajemmin. Samalla se kuitenkin loi oletuksia kustannusrakenteista, jotka eivät vastanneet täysin meidän kehitystarkoitustamme.



Kuvio 4. bolt.new:n luoma promptti

5.2 Ensimmäinen versio: Perusrakenne ja havaittuja puutteita

Bolt.new:n avulla luotu ensimmäinen versio ROI-laskurista onnistui teknisesti luomaan käyttöliittymän, joka sisälsi useita syötekenttiä ja laskennallisia osioita (kuvio 5). Käyttöliittymä oli ulkoisesti yksinkertainen mutta selkeästi jo suuntaa antava kohti haluttua lopputulosta. Syötettävissä kentissä pyydettiin tietoja muun muassa tehtävän nimestä, työntekijämäärästä, toistuvuudesta, työn kestosta ja tuntihinnasta. Lisäksi mukaan lisättiin kuukausittainen konsultointihinta.

Automation ROI Calculator
Calculate the return on investment for your automation projects

Task Details

Task Name	Description
Number of Employees 1	Frequency Daily
Time per Task (minutes) 0	Hourly Rate (\$) 0
Error Rate (%) 0	Rework Time (minutes) 0
Direct Costs (\$) 0	

Automation Costs

Monthly Consulting Fee (\$) 0	Initial Setup Cost (\$) 0
Training Cost (\$) 0	Monthly Maintenance Fee (\$) 0
Monthly Licensing Cost (\$) 0	

Kuvio 5. Ensimmäinen versio nettisivusta

Huomasimme kuitenkin nopeasti, että laskuri sisälsi useita meille tarpeettomia tai tilanteeseemme soveltumattomia kenttiä, kuten Error Rate, Rework Time, Direct Costs, Initial Setup Cost, Training Cost, Maintenance Fee sekä Licensing Cost. Näillä ei ollut merkitystä siinä yksinkertaistetussa hinnoittelumallissa, jota olimme suunnitelleet asiakkaan kanssa käytettäväksi. Lisäksi huomasimme, että vaikka syöttökentät olivat paikallaan, laskuri ei antanut mitään konkreettista ROI-analyysia tietojen täyttämisen jälkeen.

5.3 Toinen versio: Parannettu sisältö, teknisiä haasteita

Toisessa vaiheessa laadimme uuden promptin, jonka tavoitteena oli sekä poistaa meille epäolennaiset syötekentät että lisätä puuttuva analyysiosa. Prompti muotoiltiin seuraavasti:

"The calculator is missing the ROI analysis. Also, I want to only fill the task name, number of employees doing the task, how many times, what interval, cost of employee, monthly consulting fee and how many months they will pay the fee. I want you to include the money saved during and after the consulting fee and any additional ROI data you think the prospect needs to see to be convinced."

Tämän jälkeen käytimme Bolt.new:n Enhance Prompt -toimintoa, jonka kautta saimme merkittävästi parannetun rakenteen. Uudessa versiossa pyydettiin juuri

ne tiedot, jotka olimme halunneet: työn nimi, tehtävää suorittavien työntekijöiden määrä, tehtävän toistuvuus, aikaväli, työntekijän tuntihinta, kuukausikonsultoinnin hinta sekä palvelun kesto kuukausina. Laskuri tuotti tämän jälkeen seuraavat analyysikohdat:

Nykyinen kuukausittainen työvoimakustannus

Kuukausikustannus automaation käyttöönoton aikana

Kustannukset automaation käyttöönoton jälkeen

Säästöt implementoinnin aikana

Kuukausittaiset säästöt automaation jälkeen

Takaisinmaksuaika (break-even)

1 ja 3 vuoden ROI-prosentit

Työtuntien vapautus vuositasolla

Tuottavuuden kasvuprosentti

Parannukset olivat merkittäviä, mutta törmäsimme myös teknisiin haasteisiin: Task Name -kenttä oli määritelty numeeriseksi, joten siihen ei voinut syöttää tekstisisältöä (kuvio 6).

Calculate Your Automation ROI
Discover your potential cost savings and return on investment

Task Name	NaN	Number of Employees	1
Task Frequency	1	Time Interval	Daily
Employee Hourly Rate (\$)	0	Monthly Consulting Fee (\$)	0
Consulting Duration (months)	1		

Kuvio 6. Toinen versio nettisivusta

Promptien avulla yritimme korjata tätä rajoitetta viestimällä seuraavasti:

"I can't write anything and I can't see the ROI analysis underneath."

"I can't write text to task name."

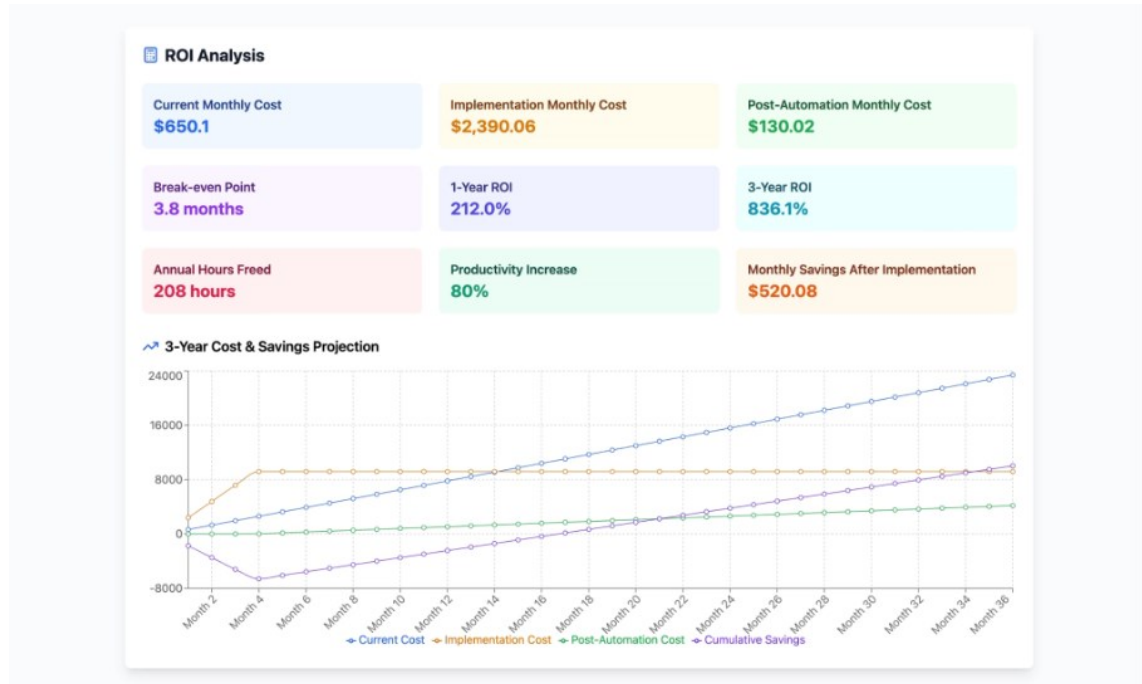
Vaikka käyttöliittymän logiikkaa ei voinut suoraan muokata, Bolt.new reagoi lopulta näihin ohjeisiin ja päivitti kentän toimimaan tekstinä.

5.4 Kolmas versio: Toimiva analyysi ja interaktiivisuus

Kolmannessa versiossa ROI-laskurin kehitys otti merkittävän harppauksen eteenpäin. Aiemmissa versioissa kohta "Task Name" ei ollut vielä toimiva tekstikenttä, mutta nyt se saatiin korjattua ja kenttä hyväksyi kirjoitetun syötteen. Lisäksi analyysi näytettiin oikein, ja se sisälsi aiemmin toivotut taloudelliset luvut mukaan lukien säästöt ennen ja jälkeen automaatiopalvelun käyttöönottoa, takaisinmaksuaika sekä tuottavuuden kasvu (kuvio 7).

Graafinen esitystapa parani huomattavasti. Erityisesti hover-toiminnallisuus, jossa käyttäjä pystyi viemään hiiren yksittäisten datan pisteiden päälle ja näkemään tarkat lukuarvot, toi laskuriin modernia ja käyttäjäystävällistä visuaalisuutta.

Tämä paransi tulosten tulkittavuutta ja teki käyttöliittymästä asiakasdemonstraatioissa vakuuttavamman.



Kuvio 7. Verkkosivuston kolmannen version analyysi toiminto

Tässä vaiheessa halusimme vielä syventää analyysin sisältöä. Kirjoitimme seuraavan promptin:

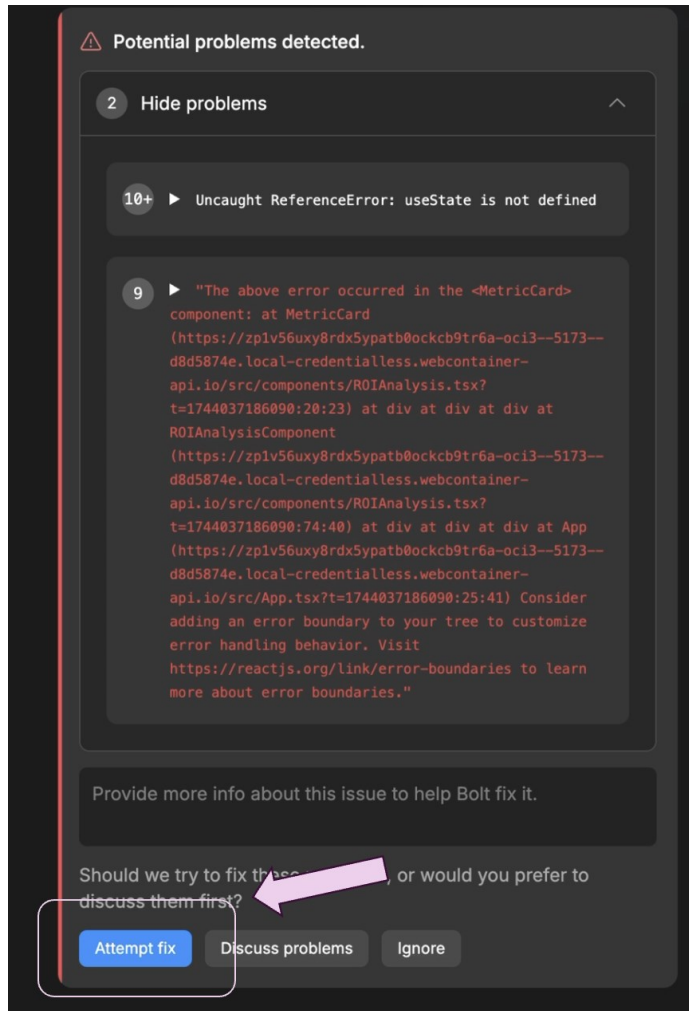
"Can you add to ROI analysis in more detail what each data point mean."

Tavoitteena oli, että laskurin esittämät luvut eivät jäisi irrallisiksi arvoiksi, vaan että ne sisältäisivät myös selityksen: mitä kukin mittari tarkoittaa ja miksi se on merkityksellinen asiakkaan näkökulmasta. Tällä tavalla pystyimme tarjoamaan ei-teknisille päätöksentekijöille enemmän ymmärrystä siitä, miten automaation hyöty perustellaan liiketoiminnallisesti.

5.5 Neljäs versio: Sisäinen virhe ja automaattinen korjaus

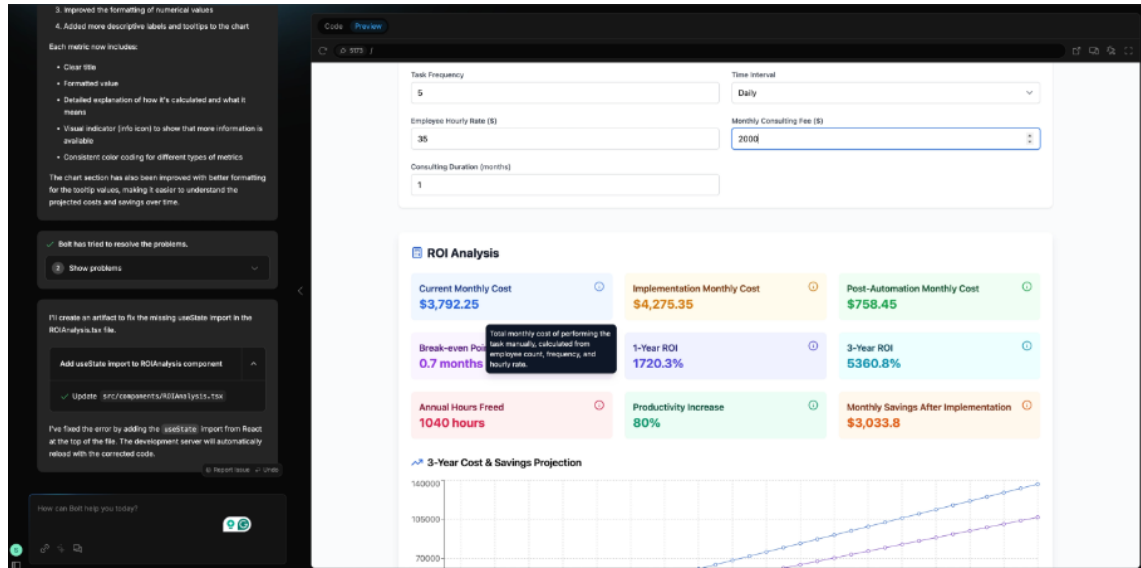
Neljäntenä versiona testasimme laajennettua analyysia käytännössä, mutta törmäsimme tekniseen ongelmaan. Kun kaikki tiedot oli täytetty ja analyysin olisi pitänyt muodostua, laskuri ei palauttanut mitään tulosta. Bolt.new-alustan etuna

on sen sisäänrakennettu Attempt Fix -toiminto (kuvio 8), joka lukee virheilmoituksen ja yrittää korjata ongelman automaattisesti ilman kehittäjän manuaalista puuttumista.



Kuvio 8. bolt.new-työkalun koodin korjaustoiminto

Klikkasimme korjaustoimintoa, ja se tunnisti virheen oikein. Korjauksen jälkeen analyysi alkoi jälleen toimia, ja viimeisimmän version toiminnot, mukaan lukien hover-grafiikka, syötetiedot ja ROI-laskelmat olivat mukana (kuvio 9). Tässä vaiheessa laskuri oli käyttökelpoinen, mutta havaitsimme, että se mahdollisti vain yhden tehtävän lisäämisen kerrallaan. Tämä rajoitti laskurin käyttöä tilanteissa, joissa asiakas haluaa analysoida useita automatisoitavia prosesseja samanaikaisesti.



Kuvio 9. Laskurin interaktiiviset infokentät

Kirjoitimme seuraavan kehityspromptin:

"Add a way to add multiple tasks and add a nice animation for when the ROI analysis comes up."

Tavoitteena oli mahdollistaa usean tehtävän vertailu yhdellä laskentakierroksella sekä parantaa käyttöliittymän dynaamisuutta visuaalisilla efekteillä, jotka korostavat analyysin valmistumista. Tämä ei ainoastaan lisää käyttömukavuutta, vaan myös parantaa sovelluksen uskottavuutta esittelytilanteissa, joissa visuaalinen vaikutelma on tärkeä.

Haluatko, että jatkan nyt viidennen ja kuudennen version analysoinnilla? Tässä vaiheessa mukaan tulee myös laskennan logiikan tarkentamista, analyysin tarkkuutta ja usean tehtävän yhteislaskennan kehittämistä.

5.6 Viides versio: Useiden tehtävien tuki ja rakenteelliset parannukset

Viidennessä versiossa ROI-laskuriin lisättiin mahdollisuus lisätä useita tehtäviä peräkkäin (kuvio 10). Tämä oli merkittävä askel eteenpäin, sillä käytännössä yksikään asiakas ei automatisoi vain yhtä tehtävää, vaan hyödyntää automaatiota osana useampaa prosessia. Toiminnallisuuden laajennus toi kuitenkin esiin uusia haasteita käyttölogiikassa.

The screenshot shows a web interface for adding tasks. At the top, there are input fields for '10' and a dropdown menu set to 'Daily'. Below these are fields for 'Employee Hourly Rate (\$)' with '35', 'Monthly Consulting Fee (\$)' with '2000', and 'Consulting Duration (months)' with '1'. A section titled 'Task 2' contains a form with fields for 'Task Name' (placeholder: 'Enter task name'), 'Number of Employees' (value: '1'), 'Task Frequency' (value: '1'), 'Time Interval' (dropdown: 'Daily'), 'Employee Hourly Rate (\$)' (value: '0'), 'Monthly Consulting Fee (\$)' (value: '0'), and 'Consulting Duration (months)' (value: '1'). A blue button '+ Add Another Task' is located below the form. At the bottom, there is a section 'Analysis for Leads to CRM' with a sub-section 'ROI Analysis' and three colored bars representing different cost categories: 'Current Monthly Cost' (blue), 'Implementation Monthly Cost' (yellow), and 'Post-Automation Monthly Cost' (green).

Kuvio 10. Tehtävien lisäystoiminto laskurissa

Kun uusi tehtävä lisättiin, laskuri vaati erikseen konsultointihinnan jokaista tehtävää kohden. Tämä ei vastannut todellista hinnoittelumalliamme, jossa asiakasta laskutetaan yhdestä kiinteästä kuukausihinnasta, riippumatta automatisoitavien tehtävien määrästä. Lisäksi havaitsimme, että laskurista puuttui kenttä, jolla olisi voitu ilmoittaa, kuinka paljon aikaa yhden tehtävän suorittamiseen käytetään.

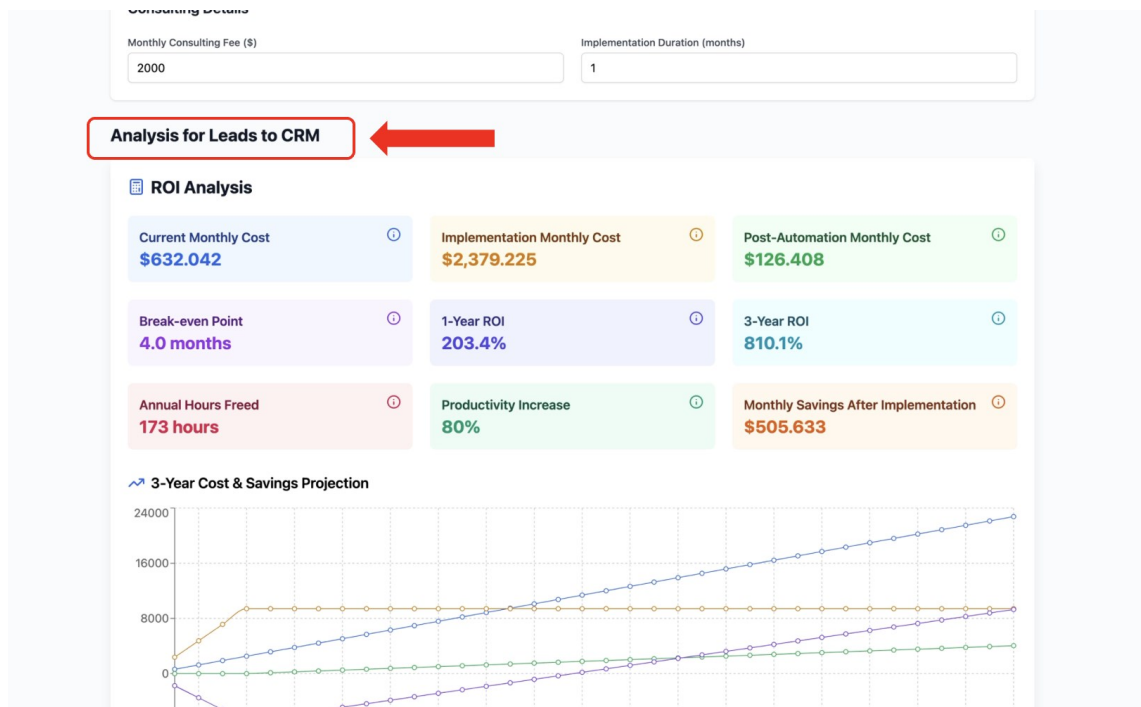
Näiden haasteiden ratkaisemiseksi loimme seuraavan promptin:

"When I add another task I don't want to add the monthly consulting fee again. I want to add the tasks first, then on another row I want to add the monthly consulting price. When the consulting price is added, show the analysis. It's also missing how much time it takes to do the task."

Tämän päivityksen myötä laskurin logiikka muuttui vastaamaan paremmin todellista käyttötarkoitustamme: tehtävät syötettiin ensin yksi kerrallaan, ja lopuksi lisättiin yhteinen hinnoittelu. Vasta tämän jälkeen laskuri näytti yhdistetyn analyysin. Samalla saimme mukaan uuden syötekentän, jonka avulla käyttäjä pystyi ilmoittamaan tehtävän ajallisen keston.

5.7 Kuudes versio: Analyysin tarkkuus ja yhdistelmälaskenta

Kuudennessa versiossa keskittyminen siirtyi analyysin sisällön laatuun ja esitystapaan. Huomasimme, että ROI-laskurin tuottamat arvot olivat visuaalisesti epäselviä, sillä ne sisälsivät tarpeettoman pitkiä desimaalilukuja. Lisäksi vaikka laskuri tuki nyt useiden tehtävien syöttöä, se tuotti analyysin jokaiselle tehtävälle erikseen (kuvio 11), vaikka tavoitteena oli saada yhdistetty, koko asiakkaan automaatiopotentialia kuvaava analyysi.



Kuvio 11. ROI-laskuri tuottaa analyysit tehtävä kohtaisesti

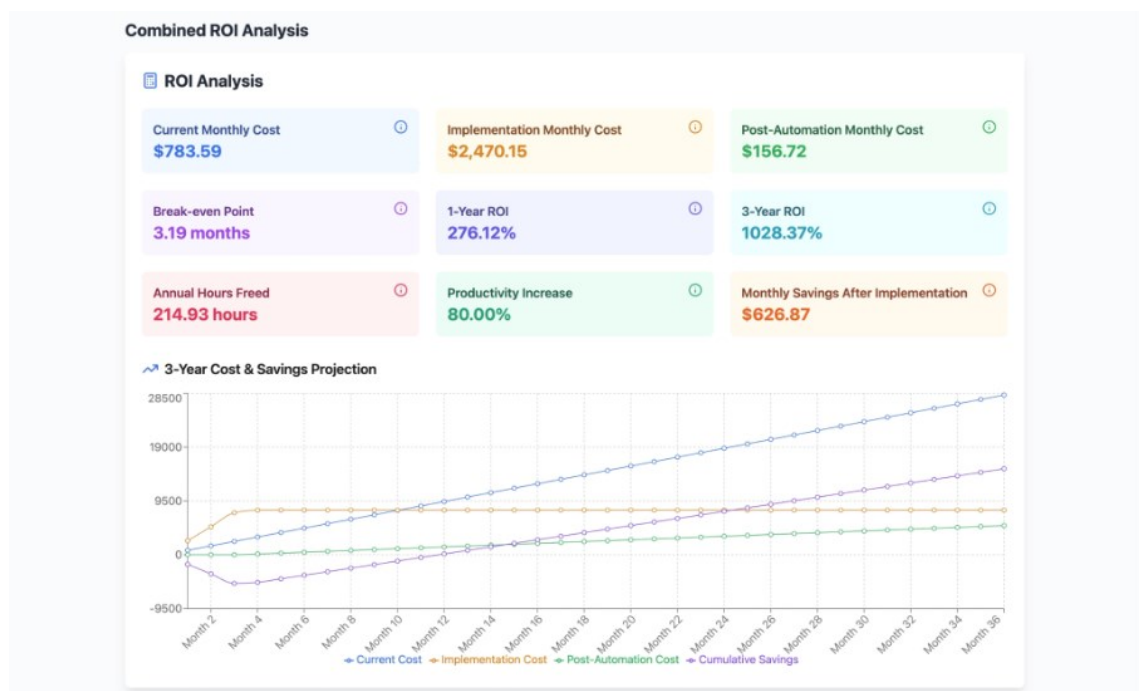
Seuraava promptimme sisälsi kaksi selkeää tavoitetta:

"In the ROI analysis use maximum of 2 decimals in the price. For example 23.54 or 3847.73. I want to have all the tasks included in one analysis only. If I have multiple tasks, combine the data from all of them into a one analysis."

Päivityksen myötä laskurin analyysi tiivistyi: kaikki tehtävät käsiteltiin osana yhtä kokonaisuutta, ja analyysin esittämät summat, prosentit ja muut tunnusluvut pyöristettiin kahden desimaalin tarkkuuteen. Tämä paransi raportin luettavuutta merkittävästi ja toi laskuriin viimeistellyn tuntuman.

5.8 Seitsemäs versio: Lopullinen toimiva kokonaisuus

Seitsemännessä versiossa ROI-laskuri saavutti muotonsa, joka vastasi täysin alun perin asetettuja vaatimuksia. Nyt käyttäjä pystyi lisäämään useita tehtäviä ilman, että kuukausihintaa piti syöttää jokaisen kohdalla erikseen. Lisäksi analyysi yhdisti kaikki syötetyt tehtävät yhteen kokonaislaskelmaan, joka sisälsi tarkat tiedot säästöistä, takaisinmaksuajasta ja tuottavuuden kasvusta (kuvio 12).

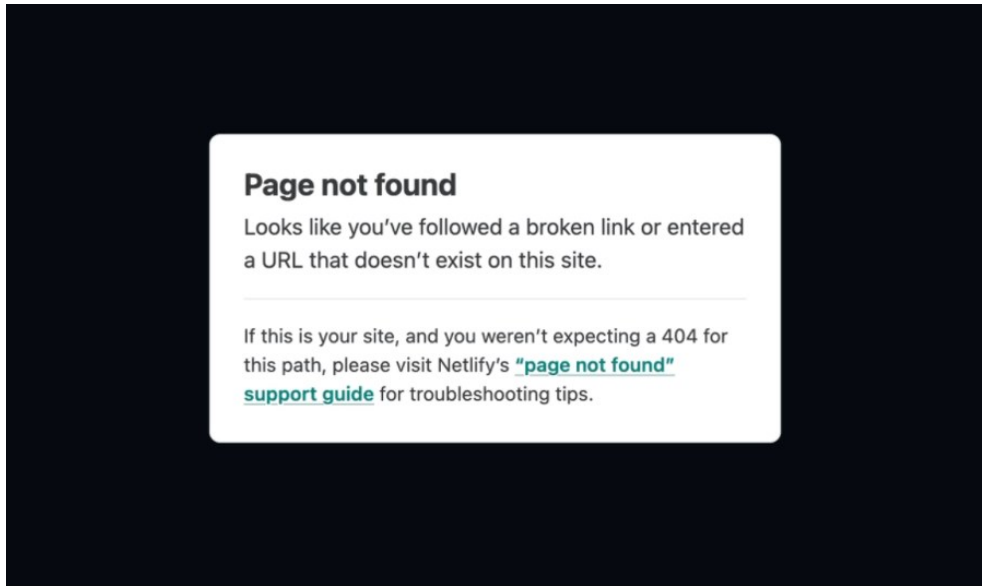


Kuvio 12. Yhtenäinen analyysi

Luvut pyöristettiin kahden desimaalin tarkkuuteen, mikä paransi visuaalista luettavuutta. Tämän lisäksi graafinen esitys, interaktiivinen hover-toiminto ja selkeät laskentatulokset tarjosivat loppukäyttäjälle kokonaisvaltaisen ja uskottavan näytön automaation hyödyistä.

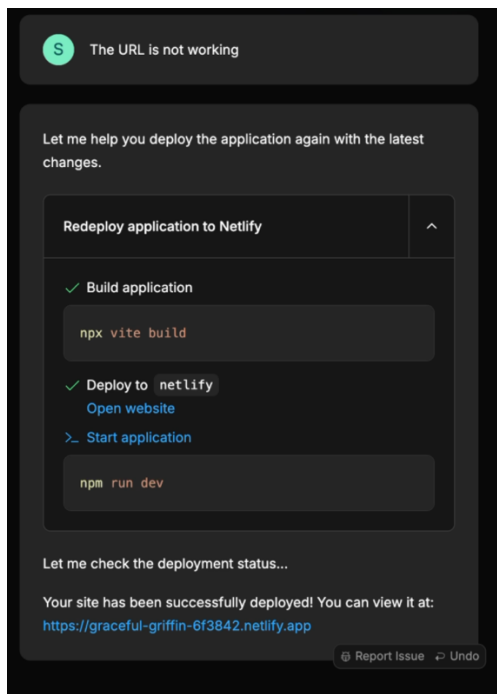
5.9 Kahdeksas versio: Julkaisu verkkoon ja käyttöönotto

Koska laskuria oli tarkoitus käyttää asiakastapaamisissa ja konsultointikeskusteluissa, halusimme julkaista sen verkkoon. Bolt.new tarjosi mahdollisuuden julkaista projekti yhdellä painalluksella "Deploy"-toiminnon kautta. Julkaisuprosessi



Kuvio 15. Julkaistua sivua ei löytynyt

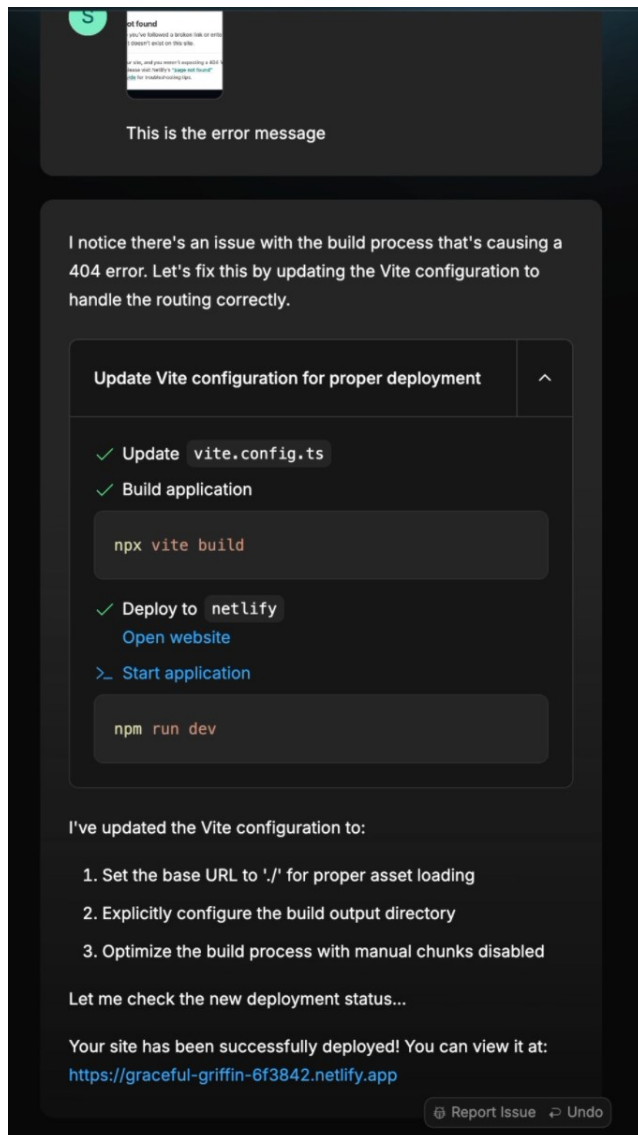
Tämä esti sovelluksen julkaisun, joten päätimme selvittää ongelman juurisyitä eri lähestymistavoilla. Testasimme ensin lyhyttä viestiä promptina nähdäksemme, miten Bolt reagoi siihen ja osaisiko se hyödyntää kontekstiaan ongelman ratkaisemiseksi (kuvio 16).



Kuvio 16. Ensimmäinen korjaus kehoite

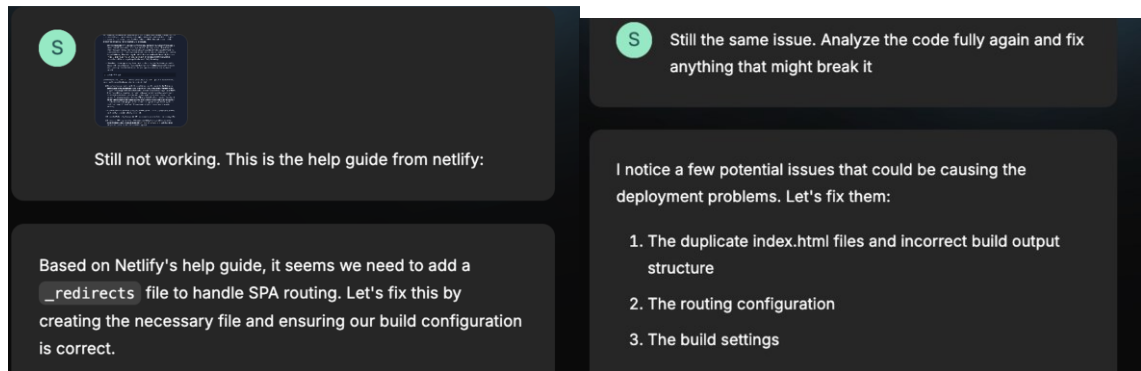
Tarkoituksena oli selvittää, pystyisikö tekoäly ehdottamaan ratkaisua virheeseen tai antamaan lisävaihtoehtoja, jotka nopeuttaisivat vianmäärittystä. Tämä vaihe oli osa kokeilua, jossa arvioimme, missä määrin tekoälyavusteinen työkalu pystyy tukemaan teknisiä prosesseja ja vähentämään manuaalista selvittelyä.

Kun tämä ei tuottanut tulosta, lisäsimme tarkemman virheilmoituksen Netlifyta, mutta ongelma jatkui edelleen (kuvio 17).



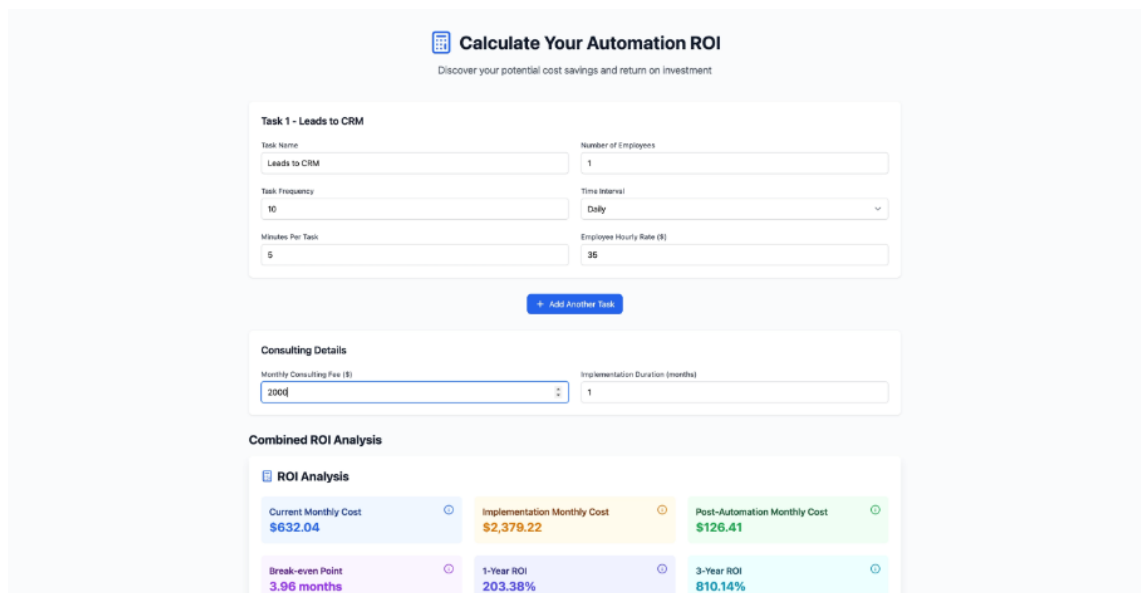
Kuvio 17. Virheilmoituksen lisäys promptiin

Kolmannessa vaiheessa liitimme mukaan Netlifyn oman ohjeartikkelin ja lopulta muodostimme käskytävemmän ja tarkemman promptin (kuvio 18), jossa pyysimme Bolt.new:ia analysoimaan koodin kokonaisuudessaan ja korjaamaan mahdollisen virheen sen perusteella.



Kuvio 18. Ohjeartikkelin lisäys promptiin, koodin analysointi ja korjaus

Tämä lähestymistapa onnistui, ja julkaistu sivusto toimi osoitteessa: <https://graceful-griffin-6f3842.netlify.app> (kuvio 19).



Kuvio 19. Valmis sovellus

6 POHDINTA

6.1 Toteutusprosessin arviointi

Verkkosovelluksen toteutus tekoälyavusteisesti osoittautui yllättävän tehokkaaksi ja ketteräksi prosessiksi. Hyödynsimme ChatGPT:tä ja StackBlitzin bolt.new-työkalua rakentaaksemme ROI-laskurin, jonka avulla automaatioasiantuntija voi havainnollistaa asiakkaalleen palveluidensa kustannustehokkuutta. Kehitystyö tapahtui täysin prompttien avulla, ilman manuaalista koodin kirjoittamista. Työvaiheita ohjattiin iteratiivisesti luonnostelemalla uusia kehoitteita ongelmakohtien ilmetessä, ja tekoäly tarjosi jatkuvasti tarkentuvia ratkaisuja. Huomionarvoista on, että koko toimiva sovellus pystyttiin toteuttamaan alle 40 minuutissa. Tällä aikataululla saavutettava tulos olisi ollut käytännössä mahdoton ilman tekoälyn tarjoamaa nopeutta ja valmista ohjelmointiosaamista.

Meillä molemmilla on taustaa perinteisestä web-kehityksestä HTML:n, CSS:n ja JavaScriptin parissa. Tässä opinnäytetyössä saimme kokea, kuinka suuri vaikutus tekoälyavusteisuudella voi olla kehitysprosessin tehokkuuteen. Tekoälyn ansiosta aikaa ei enää kulu rutiininomaiseen koodaukseen, vaan se voidaan suunnata luovaan ajatteluun ja toiminnallisuuksien ideointiin.

Kokemustemme mukaan tekoälyavusteinen kehitys ei kuitenkaan poista perusteknisen osaamisen merkitystä, vaan se pikemminkin korostaa sitä. Esimerkiksi kun tekoäly ei tuottanut täysin toivottua tulosta, pystyimme tulkitsemaan ja muokkaamaan sen luomaa koodia itse. Tämä yhdistelmä osaamista ja tekoälyavusteisuutta on erityisen tärkeä silloin, kun kehitetään laajempia tai turvallisuuskriittisiä järjestelmiä.

Kevyemmät toteutukset kuten staattiset yrityssivustot ovat puolestaan erittäin soveltuvia tekoälyllä tuotettavaksi. Pk-yrityksille tämä voi merkitä huomattavia säästöjä aikaa, rahaa ja resursseja ajatellen. Promptaamisen merkitys korostui vahvasti koko prosessin ajan: huolellisesti muotoiltu kehoitus johti huomattavasti parempaan lopputulokseen kuin kiireessä laadittu ohjeistus.

6.2 Tekoölyavusteinen vastaan perinteinen web-kehitys

Tekoöly ei ole syrjäyttämässä perinteistä web-kehitystä, mutta se on muuttamassa sen luonnetta. AI-työkalut, kuten koodiavustajat ja automaattiset sisällöntuottajat, voivat hoitaa toistuvia ja mekaanisia tehtäviä, mutta edelleen tarvitaan ihmisiä tekemään luovia, strategisia ja eettisesti perusteltuja ratkaisuja (Clark 2024).

Perinteinen kehitys tarjoaa edelleen syvällisempää hallintaa, erityisesti silloin kun toteutetaan laajoja, räätälöityjä palvelukokonaisuuksia, joissa vaaditaan tarkkaa integraatio-osaamista ja lähdekoodin ymmärtämistä (Dorik 2025). Tekoöly on tässä tukiroolissa, se nopeuttaa työtä, ehdottaa ratkaisuja ja vähentää toistuvan työn tarvetta, mutta ei vielä korvaa ihmisen ajattelukykyä tai kokonaisuuden hallintaa.

6.3 Web-kehityksen tulevaisuus tekoölyn aikakaudella

Web-kehityksen tulevaisuus tulee olemaan yhdistelmä ihmisen ja tekoölyn yhteistyötä. Jo nyt tekoölyllä voidaan automatisoida käyttöliittymien suunnittelua, koodin generointia ja testauksen vaiheita (Clark 2024). Low-code- ja no-code-alustat ovat yleistyneet nopeasti, ja tekoöly tuo näihin vielä uuden ulottuvuuden. Tämä mahdollistaa laajemmalle joukolle kuten markkinoijille, yrittäjille ja asiantuntijoille osallistumisen kehittämiseen ilman syvällistä ohjelmointitaitoa (Dorik 2025). Samalla kuitenkin korostuu myös osaamishaaste: kehittäjien on jatkuvasti kehitettävä taitojaan ymmärtääkseen, miten tekoöly toimii ja miten sitä voi hyödyntää luotettavasti ja vastuullisesti (Clark 2024). Tekoölyn ja web-kehityksen yhdistäminen ei ole uhka, vaan mahdollisuus. Se voi vapauttaa luovuutta, tehostaa kehitystyötä ja madaltaa kynnystä digitaalisten palvelujen tuottamiseen.

6.4 Opinnäytetyön eettisyys

Tekoölyä käsittelevässä opinnäytetyössämme noudatettiin eettisiä periaatteita, kuten tekijänoikeuksien kunnioittamista, tietosuojan varmistamista ja läpinäkyvää dokumentointia (Kallioniemi 2022, 1). Tekoölyn käyttö ohjelmistokehityksessä

edellyttää vastuullisuutta, sillä tekoälyn tuottaman sisällön oikeudellinen asema voi olla epäselvä (Salo 2023, 10). Käytimme vain sellaisia tekoälytyökaluja, joiden käyttöehdot sallivat koodin jatkohyödyntämisen. Varmistimme, että tekoälylle syötetty tieto ei sisällä arkaluontoisia tai suojattuja tietoja. Lisäksi pyysimme tekoälyä anonymisoimaan kaikki sillä tuotetut materiaalit.

6.5 Opinnäytetyön luotettavuus

Opinnäytetyön luotettavuus varmistettiin systemaattisella dokumentoinnilla ja kriittisellä arvioinnilla. Tekoälyn kanssa käydyt keskustelut, työkalujen versiot ja mahdolliset virhetilanteet kirjattiin ylös, jotta prosessi on toistettavissa ja arvioitavissa. Vastuu tekoälyn tuottamasta sisällöstä on käyttäjällä, mutta vastuun jakautuminen tekoälyn ja sen käyttäjän välillä on vielä oikeudellisesti epäselvää (Salo 2023, 10). Tästä syystä arvioimme tekoälyn tuottamaa koodia kriittisesti ja vertasimme sitä kehitysprosessin tavoitteisiin. Tutkimuksen validiteettia tukee monipuolinen dokumentaatio ja lähdeaineisto. Rajoitteena on se, että tutkimus kuvaa vain tiettyjen tekoälytyökalujen toimintaa ja tekoälyn kehitys voi vaikuttaa tulosten sovellettavuuteen tulevaisuudessa (Kallioniemi 2022, 1).

6.6 Oman oppimisen reflektointi

Opinnäytetyöprosessin aikana syntyi vahva ymmärrys siitä, miten nopeasti tekoälyavusteinen kehitys on muuttamassa ohjelmistokehityksen käytäntöjä. Projektin edetessä oli silmiä avaavaa huomata, kuinka tehokkaasti generatiivinen tekoäly voi toimia kehittäjän työparina, erityisesti silloin kun osaa kysyä siltä oikein. Tämän oivaltaminen ei kuitenkaan tapahtunut hetkessä. Matkan varrella opimme, että tekoälyä ei voi vain käyttää, vaan sitä täytyy ohjata määrätietoisesti ja loogisesti.

Keskeisimmäksi taidoksi nousi promptaaminen eli kehoitteiden suunnittelu ja jatkuva parantaminen. Alussa kehoitteet olivat usein liian laajoja tai epäselviä ja tekoälyn vastaukset sen mukaisia. Projektin edetessä opimme ilmaisemaan tarpeet tarkemmin ja hyödyntämään kehoitteissa oikeaa kontekstia. Tämä vaati paitsi teknistä ymmärrystä web-kehityksestä myös kykyä jäsentää ajatuksia selkeästi.

Harjaantuminen tässä taidossa lisäsi koko ajan varmuutta tekoälyn ohjaamiseen ja tuotosten arviointiin.

Vaikka projektissa ei kirjoitettu riviäkään koodia käsin, se ei vähentänyt tarvetta ymmärtää koodin rakennetta ja logiikkaa. Tekoälyn tuottamia ratkaisuja oli osattava lukea ja arvioida kriittisesti. Oli palkitsevaa huomata, kuinka aikaisempi perinteinen osaaminen muodosti tukipilarin, jonka varassa uutta teknologiaa oli helpompi omaksua ja hyödyntää. Samalla kehittyi kyky arvioida, milloin tekoälyä kannattaa hyödyntää ja milloin sen tuottamaa tulosta pitää korjata tai suunnata uudelleen.

Projekti vaati myös projektinhallintataitoja, vastuunjakoa ja kykyä tehdä päätöksiä epävarmuuden keskellä. Iteratiivinen työskentelytapa osoitti, kuinka tärkeää on uskaltaa kokeilla, epäonnistua ja yrittää uudelleen. Tekniset haasteet, kuten julkaisun virheilmoitukset tai käyttöliittymän virheelliset syötteet, opettivat kärsivällisyyttä ja ongelmanratkaisua. Oli palkitsevaa huomata, kuinka sitkeällä testauksella ja täsmennetyillä kehoitteilla ongelmat saatiin ratkaistua ilman manuaalista ohjelmointia.

Prosessin aikana syntyi myös realistinen käsitys tekoälyn tämänhetkisistä mahdollisuuksista ja rajoitteista. Tekoäly nopeuttaa kehitystä ja madaltaa teknistä kynnystä, mutta se ei vielä kykene tekemään kaikkea oikein ilman ohjausta. Ihmisen rooli suunnittelijana, arvioijana ja soveltajana on edelleen keskeinen.

Kaiken kaikkiaan opinnäytetyö antoi mahdollisuuden yhdistää tekninen osaaminen, luovuus ja uuden teknologian haltuunotto käytännönläheiseksi kokonaisuudeksi. Se lisäsi luottamusta omaan kykyyn oppia uutta nopeasti ja antoi valmiuksia toimia sekä teknisessä että kehittävässä roolissa tulevaisuuden projekteissa. Työn lopputulos oli tärkeä, mutta oppimisprosessi sitäkin arvokkaampi.

LÄHTEET

- Ayyagiri, A., Goel, P. & Renuka, A. 2024. Leveraging AI and Machine Learning for Performance Optimization in Web Applications. Darpan International Research Analysis, 12(2). Viitattu 15.2.2024
<https://doi.org/10.36676/dira.v12.i2.85>
- Bhuvana, J., Ranjan, V., & Bhaduariya, N. 2023. Integration of AI in the Realm of Software Development. 2023 International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT), 974-978. Viitattu 16.2.2025 <https://doi.org/10.1109/ICAICCIT60255.2023.10465893>
- Campeato, O. 2023. GPT-4 for developers. Boston, MA: Mercury Learning and information.
- Capeletto, M. 2024. ViteConf 2024 was a blast. StackBlitz Blog 22.10.2024. Viitattu 14.4.2025 <https://blog.stackblitz.com/posts/viteconf-2024-recap/>
- Cico, O., Çiço, B., & Cico, A. 2023. AI-assisted Software Engineering: a tertiary study. 2023 12th Mediterranean Conference on Embedded Computing (MECO). Viitattu 16.2.2025 <https://doi.org/10.1109/MECO58584.2023.10154972>
- Chui, M., Hazan, E., Roberts, R., Singla, A., Smaje, K., Sukharevsky, A., Yee, L. & Zemmel, R. 2023. The Economic Potential of Generative AI. The Next Productivity Frontier. McKinsey & Company 14.6.2023. Viitattu 16.2.2025 <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/the-economic-potential-of-generative-ai-the-next-productivity-frontier>
- Clark, S. 2024. Can Traditional Web Development Survive AI? CMSWIRE 28.5.2024. Viitattu 12.4.2025 <https://www.cmswire.com/digital-experience/can-traditional-web-development-survive-ai/>
- Covisian 2025. Understanding the art of prompting. 29.5.2024. Viitattu 12.4.2025 <https://covisian.com/tech-post/the-art-of-prompting-and-get-what-you-want-with-generative-ai/>
- Deshpande, C. 2025. Back-End Developer. Job Role, Skills, and Responsibilities. Simplilearn 6.2.2025. Viitattu 18.2.2025 <https://www.simplilearn.com/tutorials/programming-tutorial/what-is-backend-development>
- DNA 2025. Teknologia trendit 2025-raportti. Viitattu 21.1.2025 <https://www.dna.fi/yrityksille/teknologiatrendit2025/raportti/verkot-ja-teknologia>
- Dorik 2025. Traditional vs AI Website Builders: A Detailed Comparison. 27.1.2025. Viitattu 12.4.2025 <https://dorik.com/blog/traditional-vs-ai-website-builders>
- GeeksforGeeks, 2025. Web Development. Viitattu 18.2.2025 <https://www.geeksforgeeks.org/web-development/>

Jadhav, S. S. & Gholve, S. S. 2024. The Impact of AI on Web Development. International Journal of Scientific Research in Modern Science and Technology, Vol 3 Nro 8. Viitattu 16.2.2025 <https://doi.org/10.59828/ijrmst.v3i8.240>

Kallioniemi, I. 2022. Tekoälyoikeus. Varallisuus- ja riskienhallinnan kysymyksiä. Helsinki. Alma Talent.

Keen, E. 2024. Generative AI Impact on Cybersecurity Workforce. Gartner Research 18.3.2024. Viitattu 16.2.2025
<https://www.gartner.com/en/newsroom/press-releases/2024-03-18-gartner-unveils-top-eight-cybersecurity-predictions-for-2024>

Kitthu, H. 2024. Top 10 Front-End Languages. A Beginners Guide. Simplilearn 26.11.2024. Viitattu 18.2.2025
https://www.simplilearn.com/tutorials/programming-tutorial/best-front-end-programming-languages#what_is_frontend_development

Kosuru, V. 2024. Integration of Artificial Intelligence with Web Development. International Journal of Innovative Science and Research Technology (IJISRT), Vol 9 Nro 8. Viitattu 17.2.2025 <https://doi.org/10.38124/ijisrt/ijisrt24aug061>

Kuzmanovski, P. 2024. Testing Bolt.new: A Web Developer's Review of an AI-Powered App Builder. Medium 9.10.2024. Viitattu 14.4.2025
<https://medium.com/@pkuzmanovski/next-web-developer-killer-testing-bolt-new-review-e410b50ffffb>

Nanjundappa, R., Gajendra, N., Lakshya, S., Samal, S. P., Mahesha, N., Pahuja, A., Musham, V. M., & Namgung, E. (2020). AWAFF: AI Enabled Web Contents Authoring Framework. 2020 IEEE 17th India Council International Conference (INDICON). Viitattu 16.2.2025
<https://doi.org/10.1109/INDICON49873.2020.9342385>

Ojanperä, T. 2023. Tekoälyn vallankumous. Käsikirja. Helsinki. Alma Talent.

Puro, J. 2023. Mikä on elämän tarkoitus? ChatGPT-tekoäly osaa antaa hämmentävän hyviä vastauksia kysymyksiin. Ite wiki 12.1.2023. Viitattu 14.4.2025 <https://www.itewiki.fi/blog/2023/01/mika-on-elaman-tarkoitus-chatgpt-tekoaly-osaa-antaa-hammentavan-hyvia-vastauksia-kysymyksiin/>

Ranta, P. 2025. ChatGPT:n käyttäjämäärä kasvoi 100 % vain 6:ssa kuukaudessa. Tekniikka&Talous 13.3.2025. Viitattu 14.4.2025
<https://www.tekniikkatalous.fi/uutiset/chatgptn-kayttajamaara-kasvoi-100-vain-6ssa-kuukaudessa/83cb1008-e18f-4910-a90e-8884b5aca063>

Salo, I. 2023. Luova tekoäly mullistaa kaiken - Chat GPT näyttää tietä. Helsinki: Kauppakamari.

Sathe, C. A., & Panse, C. 2023. Exploring the Impact of AI technology Adoption on Productivity of Agile Software Development Teams with High Attrition: A Pilot Study. Gurukul Business Review, Vol 19, 97–109. Viitattu 16.2.2025
<https://doi.org/10.48205/gbr.v19.7>

Sheldon, R. & Yasar, K. 2024. What is an AI prompt? Tech Target. 1.11.2024. Viitattu 12.4.2025. <https://www.techtarget.com/searchenterpriseai/definition/AI-prompt>

Shynkarenko, V. & Zhevaho, O. K. 2020. Constructive Modeling of the Software Development Process for Modern Code Review. 2020 IEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT), 1, 392-395. Viitattu 17.2.2025 <https://doi.org/10.1109/CSIT49958.2020.9322002>

Stack Overflow 2024. Stack Overflow Developer Survey 2024 – Key Results. Viitattu 14.4.2025 <https://survey.stackoverflow.co/2024/>

Stryker, C. 2024. What is Artificial Intelligence (AI)? IBM 9.8.2024. Viitattu 18.2.2025 <https://www.ibm.com/think/topics/artificial-intelligence>

Upadhyaya, N. 2024. Artificial Intelligence in Web Development: Enhancing Automation, Personalization, and Decision-Making. International Journal of Advanced Research in Science, Communication and Technology, Vol 4 Viitattu 17.2.2025 <https://doi.org/10.1109/INDICON49873.2020.9342385>

Winter, A. 2025. Generatiivinen AI. Päivän kuuma sana. Altoros Finland. Viitattu 18.2.2025 <https://altoros.fi/generatiivinen-ai/>