



# Reactin oppiminen ChatGPT:n tuella

Tehokkaan kommunikoinnin salat

Ammattikorkeakoulututkinnon opinnäytetyö

Tietojenkäsittelyn koulutus

Kevät 2024

Hanna Moisala

Tietojenkäsittelyn koulutus

Tekijä Hanna Moisala

Työn nimi Reactin oppiminen ChatGPT:n avulla, tehokkaan kommunikoinnin salat

Ohjaaja Lasse Seppänen

Tiivistelmä

Vuosi 2024

---

Opinnäytetyön tavoitteena oli selvittää, kuinka ChatGPT:tä voidaan hyödyntää tehokkaasti React-ohjelmoinnin oppimisen tukena ja mitä haasteita siihen sisältyy. Työ keskittyi erityisesti ChatGPT:n kanssa tapahtuvaan kommunikointiin ja sisältää muun muassa ohjeita ja konkreettisia esimerkkejä ChatGPT:n kanssa käydyistä keskusteluista. Työn idea syntyi allekirjoittaneen tarpeesta oppia Reactia mahdollisimman tehokkaasti ja laajeni monipuolisen oppaan kehittämiseen, joka palvelee opettajia, opiskelijoita ja muita aiheesta kiinnostuneita opinnäytetyön lisäksi myös verkkosivuston muodossa. Työ antaa kattavan käsityksen ChatGPT:n ja muiden kielimallien hyödyntämisestä oppimisen tukena.

Opinnäytetyön tietoperusta koostuu tekoälyn, koneoppimisen, luonnollisen kielen käsittelyn sekä React-kehityksen keskeisistä käsitteistä, unohtamatta eettistä ja yhteiskunnallista näkökulmaa, kuten tekoälyn vaikutukset työhön ja opiskeluun. Työssä hyödynnettiin kvalitatiivista tutkimusmenetelmää, jossa aineisto kerättiin 30 päivän ajan tehdyistä päiväkirjamerkinnöistä. Aineiston analyysi keskittyi oppimisprosessiin, kohdattuihin haasteisiin sekä ChatGPT:n käytön mahdollistamiin saavutuksiin. Opinnäytetyö on toiminnallinen ja työn tuloksena ja näyttönä oppimisesta rakentui Reactin avulla yksisivuinen verkkosivu, jonne koottiin ohjeet ja esimerkit ChatGPT:n tehokkaasta hyödyntämisestä ja kommunikoinnista sen kanssa.

Johtopäätöksenä todetaan, että ChatGPT:n käyttö opiskelun tukena on arvokas lisä perinteisten oppimismenetelmien rinnalle, mutta sen tehokas hyödyntäminen vaatii taitoa kommunikoida selkeästi ja ymmärrystä työkalun rajoituksista. Tulevaisuudessa olisi hyvä kehittää tekoälypohjaisten sovellusten ja oppimisprosessien yhteistyötä sekä tutkia tarkemmin, miten tekoäly voi tukea yksilöllisiä oppimispolkuja.

Avainsanat ChatGPT, React, tekoäly, oppiminen, ohjelmointi

Sivut 83 sivua ja liitteitä 1 sivu

---

The purpose of the thesis was to find out how ChatGPT can be used effectively to support learning React programming and what challenges it contains. The study specifically focused on interactions with ChatGPT, providing guidance and concrete examples from conversations held with the AI. The idea of the work arose from the author's personal need to learn React as efficiently as possible, expanding the idea so that it resulted in a comprehensive guide for teachers, students and others interested in the topic. This guide, in addition to the thesis, is available on a website, offering a thorough understanding of leveraging ChatGPT and other language models for educational support.

The theoretical framework of the thesis is based on central concepts of artificial intelligence, machine learning, natural language processing, and React development, without forgetting the ethical and social perspective of utilizing artificial intelligence in different fields, focusing on education and especially programming. The study used a method where information was gathered from diary entries for 30 days. Analyzing this data helped understand the study process, the challenges faced and what was achieved with ChatGPT. The thesis is functional, and as evidence of the learning achieved, a single-page website was developed using React. This website gathers instructions and examples for effectively utilizing and communicating with ChatGPT.

In conclusion, the use of ChatGPT to support studying is a valuable addition to traditional learning methods, but its effective use requires the ability to communicate clearly and an understanding of the limitations of the tool. In the future, it would be good to develop cooperation between artificial intelligence-based applications and learning processes, and to investigate more closely how artificial intelligence can support individual learning paths.

Keywords ChatGPT, React, Artificial Intelligence, learning, programming

Pages 83 pages and appendices 1 page

## Sanasto

Algoritmi	Sarja ohjeita tai prosessi, jota tietokone noudattaa
ChatGPT	Tekoälymalli, jota voi käyttää verkossa käyttöliittymän kautta ja on suunniteltu ymmärtämään ja tuottamaan luonnollista kieltä
DOM	DOM eli Document Object Model on web-sivujen rakennetta kuvaava puu siitä, miten sivu on järjestetty
Embeddings	Tekniikoita, joiden avulla kielimallit voivat ymmärtää ja käsitellä sanoja ja lauseita
GDPR	General Data Protection Regulation on Euroopan unionin tietosuoja-asetus
JSX	JavaScript-laajennus, jota käytetään React-komponenttien rakenteen kuvaamiseen käyttämällä HTML:n kaltaista syntaksia suoraan JavaScript-koodissa
Prompti	Kehote tai ohje, jonka avulla kommunikoidaan suurten kielimallien kanssa
React	JavaScript-kirjasto dynaamisten käyttöliittymien kehittämiseen
Tietoaineisto	Kokoelma esimerkkejä tai dataa
UI	User Interface eli käyttöliittymä

# Sisällys

1	Johdanto .....	1
2	Tekoälyn perusteet .....	2
2.1	Tekoälyn aikakauden merkittävimmät tapahtumat.....	3
2.2	Tekoälyn määritelmä.....	4
2.2.1	Näkökulmia määritelmään .....	4
2.2.2	Määrittelyn vaikeus.....	4
2.3	Tekoälyn etiikka .....	5
2.3.1	Vaikutukset työhön ja koulutukseen.....	5
2.3.2	Yksityisyys ja vastuu.....	7
2.4	Vahva ja heikko tekoäly .....	8
2.4.1	Heikko tekoäly .....	8
2.4.2	Vahva tekoäly.....	8
2.5	Koneoppiminen ja syväoppiminen.....	9
2.5.1	Koneoppiminen .....	9
2.5.2	Syväoppiminen.....	11
2.6	Luonnollisen kielen käsittely.....	14
3	ChatGPT.....	15
3.1	Kielimallit ja GPT-historia .....	15
3.2	ChatGPT:n toimintaperiaate.....	16
3.2.1	Pluginit .....	16
3.2.2	Mukautettavat GPT:t.....	17
3.2.3	ChatGPT:n nurja puoli .....	18
3.3	Tehokas kommunikointi ja promptaus .....	19
3.3.1	Selkeät ohjeet.....	20
3.3.2	Lähteiden käyttö .....	20
3.3.3	Osittaminen .....	21
3.3.4	Aika .....	21
3.3.5	Ulkoiset työkalut .....	22
3.3.6	Testaaminen.....	22
3.3.7	Promptauksen kehitysprosessi .....	22
3.4	ChatGPT opiskelun tukena .....	23
3.4.1	ChatGPT ohjelmoinnin tukena .....	24
3.4.2	Esimerkkinä oman opiskelukurssin suunnittelu ja luominen .....	24
3.4.3	ChatGPT opettajan korvikkeena .....	25

3.4.4	Eettiset kysymykset opiskelussa.....	26
4	React.....	28
4.1	Keskeiset ominaisuudet.....	28
4.1.1	JSX.....	28
4.1.2	Komponentit.....	29
4.1.3	Tilanhallinta ja hookit.....	29
4.1.4	Propsit.....	30
4.2	Reactin vaikutukset web-kehitykseen.....	30
5	Reactin oppiminen ChatGPT:n avulla – tavoitteet ja tutkimusmenetelmät.....	32
5.1	Tutkimuksen tavoite.....	32
5.2	Kvalitatiivinen tutkimus.....	32
5.2.1	Aineistonhankintamenetelmä.....	33
5.2.2	Analyysimenetelmä.....	34
5.3	Validius ja reliabiliteetti.....	35
6	Opiskeluprosessin suunnittelu, toteutus ja dokumentointi.....	36
6.1	GPT:n luominen.....	37
6.2	Reactin perusteiden opettelu.....	43
6.2.1	Opetusmateriaalien luominen.....	44
6.2.2	Harjoitustehtävät.....	47
6.2.3	Ohjelmoinnin haasteet.....	49
6.2.4	Muut opiskelumateriaalit.....	52
6.2.5	Muut huomiot.....	52
6.3	Yksisivuinen verkkosivun luominen.....	53
6.3.1	Sovelluksen suunnittelu.....	53
6.3.2	Tekninen toteutus.....	55
6.3.3	Testaus ja julkaisu.....	62
6.4	Päiväkirjamerkintöjen analysointi ja pohdinta.....	64
6.4.1	Opiskeluprosessi yleisesti.....	64
6.4.2	Omien GPT-mallien rooli.....	64
6.4.3	Haasteet.....	65
6.4.4	Saavutukset.....	66
6.4.5	Muiden resurssien merkitys.....	67
6.4.6	Kommunikoinnin merkitys.....	68
7	Tuloksena yksisivuinen verkkosivu.....	69
8	Johtopäätökset ja pohdinta.....	74

9 Yhteenveto.....	76
Lähteet .....	77

## **Kuvat, komennot, ohjelmakoodit, taulukot ja kaavat**

Komento 1 Julkaisukomennot VS Coden terminaalissa .....	63
Kuva 1 Tekoälyn, koneoppimisen ja syväoppimisen suhde.....	9
Kuva 2 Neuroverkko .....	12
Kuva 3 Päiväkirjan pohja .....	34
Kuva 4 Luo oma GPT .....	37
Kuva 5 Pluginien asetus .....	37
Kuva 6 GPT:n ominaisuuskehotteen luominen.....	38
Kuva 7 GPT:n ominaisuuksien asetus .....	39
Kuva 8 Reactin perusteiden aikataulu.....	41
Kuva 9 Nettisivuprojektin aikataulu .....	42
Kuva 10 Ensimmäisen päivän aikataulu.....	43
Kuva 11 Reactin ydinkonseptit.....	45
Kuva 12 Ilmaiset verkkomateriaalit opiskeluun.....	46
Kuva 13 Harjoitustehtävien pyytäminen .....	47
Kuva 14 Esimerkkitiedostojen käyttö.....	48

Kuva 15 Koodin selitys ja opetus .....	50
Kuva 16 Parannusehdotukset koodiin.....	51
Kuva 17 Teknisen toteutuksen suunnitelma.....	54
Kuva 18 React Router DOM:n selitys.....	55
Kuva 19 Esimerkki tilanhallinnasta UI:ssa.....	60
Kuva 20 Step-by-step-ohjeet.....	62
Kuva 21 Etusivu.....	70
Kuva 22 Oman GPT:n luomissivu .....	71
Kuva 23 Ohjesivun kortti .....	72
Kuva 24 Tehokas kommunikointi ja ChatGPT:n hyödyntäminen opiskelussa.....	73
Ohjelmakoodi 1 Sivun rakenne ja reititys .....	56
Ohjelmakoodi 2 NavigationBar-komponentti .....	57
Ohjelmakoodi 3 Etusivun ohjelmakoodi .....	58
Ohjelmakoodi 4 Esimerkki tilanhallinnasta .....	59
Ohjelmakoodi 5 Card-komponentti etusivulla .....	60
Ohjelmakoodi 6 Komponenttien uudelleen käyttö .....	61
Ohjelmakoodi 7 Julkaisua varten tarvittavat lisäykset package.json-tiedostoon .....	63
Kaava 1 Lainanmaksukyvyyn esimerkki .....	12



## **Liitteet**

Liite 1. Aineistonhallintasuunnitelma

# 1 Johdanto

Tekoäly on jo tullut osaksi koulutusta ja työelämää. Se tuo mukanaan innovatiivisia ratkaisuja erityisesti oppimisen työkaluna. Tämä opinnäytetyö keskittyy tutkimaan, kuinka ChatGPT:tä voidaan hyödyntää React-ohjelmoinnin oppimisprosessin tehostamisessa ja millaisia haasteita sen käyttöön liittyy.

React on tunnettu JavaScript-kirjasto, jota käytetään laajasti dynaamisten web-sovellusten kehittämiseen. Tässä työssä selvitetään, miten ChatGPT voi toimia oppimisen tukena, tarjoten sekä teoreettista ymmärrystä että käytännön ohjausta React-ohjelmoinnissa.

Työ ei kuitenkaan rajoitu vain ohjelmoinnin opetteluun. Se tulee myös laajentamaan ymmärrystä tekoälyn perusteista, kone- ja syväoppimisesta, luonnollisen kielen käsittelystä sekä tekoälyn eettisistä ja yhteiskunnallisista ulottuvuuksista. Lisäksi tehdään katsaus Reactin keskeisiin ominaisuuksiin. Tavoitteena on selvittää, miten kommunikoidaan tehokkaasti suurten kielimallien kuten ChatGPT:n kanssa antaen konkreettisia esimerkkejä teorian tueksi.

Tutkimuksen käytännön osuudessa esitellään kirjoittajan henkilökohtaiset kokemukset ja oivallukset Reactin oppimisesta ChatGPT:n avustuksella. Tämä sisältää yksityiskohtaisia esimerkkejä kommunikoinnista ChatGPT:n kanssa, oman GPT:n luomisprosessin ohjeet ja konkreettisenä esimerkkinä ja näyttönä opituista asioista kehitetään yksisivuinen verkkosivu, johon kootaan tehokkaan kommunikoinnin ja ChatGPT:n hyödyntämisen periaatteita oppimisen työkaluna. Analysoimalla kokemuksia niin oppimis- kuin verkkosivuprosessista, työ pyrkii tarjoamaan arvokasta tietoa ChatGPT:n hyödyntämisen mahdollisuuksista ja haasteista ohjelmoinnin oppimisessa ja verkkosivujen kehittämistyössä.

Tutkimuskysymykset ovat:

- Miten ChatGPT voi tukea Reactin oppimisprosessia?
- Miten kommunikoidaan tehokkaasti ChatGPT:n kanssa?
- Mitä haasteita voi olla ChatGPT:n käytössä opiskelussa?

## 2 Tekoälyn perusteet

Tekoäly on kaikkialla ympärillämme ja sitä hyödynnetään monilla eri aloilla, kuten markkinoinnissa, terveydenhuollossa, liikenteessä, rahoituksessa, vakuuttamisessa, tieteessä ja koulutuksessa (Coeckelbergh, 2020/2021, ss. 12–14). Huomaamatta se on sulautunut ihmisten arkielämään, ohjaten useita palveluja, joita käytetään päivittäin. Erilaiset sovellukset, kuten älykkäät henkilökohtaiset avustajat, tekoälyä käyttävät terveyden- ja hyvinvoinnin sovellukset ja automaattiset käännöspalvelut, ovat esimerkkejä tekoälyn läsnäolosta jokapäiväisessä elämässä. (Kolari & Kallio, 2023, s. 28) Tekoälyn ymmärtämisen tärkeys on noussut lyhyessä ajassa tärkeäksi osa-alueeksi myös työelämässä. Du Plessiksen (2024) mukaan yli 80 % johtajista kokee, että työntekijöiden tulisi lisätä uusia taitoja tekoälyn tuomiin työelämän muutoksiin.

Tekoäly yhdistää useita tieteenaloja, kuten tietojenkäsittelytieteen, matematiikan, tilastotieteen ja kognitiotieteen. Sen juuret ulottuvat antiikin filosofisiin pohdintoihin älykkyydestä, mutta nykyaikaisen tekoälyn tutkimus otti ensiaskeleensa 1950-luvulla (Tarkoma, 2021, s. 95), jolloin Turingin testi esiteltiin (Coeckelbergh, 2020/2021, s. 73).

Nykypäivänä tekoälyn ala on kasvanut merkittävästi, etenkin koneoppimisen ja syväoppimisen saavuttamien läpimurtojen ansiosta. Algoritmien kehittyneisyys, lisääntyvä laskentakapasiteetti ja laajamittaisen opetusdatan saatavuus ovat olleet tässä kehityksessä kriittisiä tekijöitä. Nämä edistysaskeleet ovat mahdollistaneet monimutkaisten neuroverkkojen kehittämisen, jotka kykenevät simuloimaan ihmisaivojen toimintaa ja käsittelemään yhä monimutkaisempia ongelmia automatisoiduissa päätöksentekoprosesseissa. (Tarkoma, 2021, s. 95) Esimerkiksi ChatGPT:n perusta on syväoppimiseen pohjautuvassa neuroverkossa, joka tuottaa luonnollista kieltä vuorovaikutuksessa ihmisen kanssa (Salo, 2023, s. 43).

Teknologian nopea edistyminen siirtää työelämää uudelle aikakaudelle, muuttaen erityisesti ennustettavissa olevia ja toistuvia tehtäviä sekä datan käsittelyyn liittyviä töitä. Tekoäly ja koneoppiminen eivät ainoastaan johda uusiin läpimurtoihin, kuten itseohjautuvien ajoneuvojen kehittämiseen, vaan myös parantavat nykyisiä käytäntöjä monilla aloilla, kuten tekoälyn käyttö sairauksien tunnistamisessa terveydenhoidon alalla. (Aaltonen, 2019, ss. 146–147)

Vaikka tekoäly tarjoaa lukuisia etuja, se herättää myös samalla tärkeitä eettisiä ja yhteiskunnallisia kysymyksiä. Jos esimerkiksi edellä mainitut täysin autonomiset autot tulisivat markkinoille, olisi tarpeellista pohtia esimerkiksi sitä, miten ne käyttäytyisivät kriittisissä tilanteissa. (Coeckelbergh, 2020/2021, s. 15)

## 2.1 Tekoälyn aikakauden merkittävimmät tapahtumat

1950-luvun katsotaan olleen merkittävä ajanjakso tekoälyn syntymisessä (Salo, 2023, s. 13). Alan Turing tunnettiin tuolloin edelläkävijänä biologian, matematiikan ja tietotekniikan yhdistäjänä (Siukonen & Neittaanmäki, 2019, s. 255). Hän tutki älykkyyttä ja ajattelua sekä näiden yhdistämistä koneisiin. Erityisesti hänet muistetaan Turingin testistä. (Elements of AI, n.d.-a)

Turingin testissä ihminen keskustelelee näkymättömän keskustelukumppanin kanssa ja yrittää päätellä, onko kyseessä ihminen vai kone. Jos tuomari ei kykene luotettavasti erottamaan konetta ihmisestä, koneen katsotaan läpäisseen testin, mikä viittaa ihmisenkaltaiseen ajattelukykyyn ja älykkyyteen. (Elements of AI, n.d.-a)

Tekoälyn historia koki merkittävän hyppäyksen vuonna 1956, jolloin New Hampshiressa pidetyn historiallisen työpajan aikana John McCarthy esitteli termin ”tekoäly”. Kyseinen tapahtuma merkitsi alan virallista syntymistä. Merkittävien tapahtumien vuoksi 1950-luvun katsotaankin vakiinnuttaneen tekoälyn aseman tieteenalana. (Coeckelbergh, 2020/2021, s. 73)

Ferranti Mark I ja ENIAC edustivat 1950-luvun merkittäviä tekoälysovelluksia, jotka veivät alaa askeleen lähemmäksi modernia tekoälyä. Vaikka laitteet olivat nykyisiin tekoälysovelluksiin verrattuna alkukantaisia, olivat ne aikansa suunnannäyttäjiä. Koneet onnistuivat suorittamaan rajoitetusti monimutkaisia tehtäviä ja tekemään itsenäisiä päätöksiä. Ferranti Mark I demonstroi shakin pelaamista ja strategisten siirtojen tekemistä, kun taas ENIAC hallitsi laskentatoiminnot. (Salo, 2023, s. 14)

Tekoälyn kehitys otti merkittäviä harppauksia eteenpäin 1980-luvulla, kun neuroverkot ja syväoppimisen menetelmät kehittyivät. Tekoäly ei enää rajoittunut vain etukätehen ohjelmoituihin tehtäviin, vaan se alkoi osoittaa kykyä oppimiseen ja toimintansa parantamiseen. Neuroverkot toivat koneoppimiseen uusia ulottuvuuksia esimerkiksi puheen- ja kuvantunnistuksen saralla. Yksi aikakauden merkittävimmistä teknologisista innovaatioista

oli backpropagation-algoritmi, joka oli ratkaiseva tekijä neuroverkkojen oppimiskyvyn ja tehokkuuden parantamisessa sen painoarvojen säätämisen ansiosta. Nämä läpimurrot asettivat perustan monille nykyaikaisen tekoälyn saavutuksille. (Salo, 2023, ss. 14–15)

## 2.2 Tekoälyn määritelmä

Tekoälylle ei ole olemassa selkeää ja yksinkertaista määritelmää vaan kyseessä on asioiden laaja kokonaisuus, jota mieluummin rajataan aina tarkoitukseen sopivaksi kuin liian tiukasti tietyllä määritelmällä (Työ- ja elinkeinoministeriö, 2017, s. 15). Toisin sanoen, tekoälyn määritelmä vaihtelee sen mukaan, missä asiayhteydessä siitä puhutaan (Tiedekulma, 2021).

### 2.2.1 Näkökulmia määritelmään

Myllymäen mukaan (2021, s. 10) tekoäly tai toisin sanoen keinoäly (Artificial Intelligence) ei ole mikään yksittäinen menetelmä tai ilmiö vaan se koostuu suuresta joukosta tutkimusongelmia ja niiden ratkaisemiseksi kehitetyistä teknologioista. Tekoälystä voidaan puhua myös omana tieteenalanaan, mikä viittaa siihen, että sitä voi yrittää kuvailla, mutta tarkkaa määritelmää ei pysty antamaan. Terminä ”tekoäly” on vakiintuneempi kuin ”keinoäly”, vaikka keinoäly voisi abstraktisen luonteensa puolensa olla luontevampi käänös AI:lle (Kolari & Kallio, 2023, s. 15).

Euroopan parlamentin (2023b) mukaan tekoälyä on koneiden kyky matkia ihmismäistä älykkyyttä, kuten oppimista ja ongelmanratkaisua. Nämä järjestelmät voivat havaita ympäristönsä, analysoida tietoa, tehdä päätöksiä ja mukautua uusiin tilanteisiin itsenäisesti.

Tekoälyn määritelmät ovat vaihdelleet vuosien saatossa. Esimerkiksi Winston kuvasi vuonna 1992 tekoälyä laskennan tutkimuksena, joka mahdollistaa havaitsemisen, päättelyn ja toiminnan. Nilsson puolestaan määritteli vuonna 1998 tekoälyn olevan älykästä käyttäytymistä esineissä. (Matthias, 2021). Tekoäly voidaan nykyään kuitenkin määritellä yleisluontoisesti esimerkiksi ”Tieteenalaksi, joka tutkii keinoitekkoisia järjestelmiä ja laitteita, joiden toiminta vaikuttaa edellyttävän älykkyyttä” (Myllymäki, 2021, s. 13).

### 2.2.2 Määrittelyn vaikeus

Tekoälyn määrittelyn vaikeuteen vaikuttaa älykkyyden määrittelyn vaikeus. Miten tekoälyn voi määritellä, jos ei tarkkaan edes ymmärretä, mitä on älykkyys? Älykkyyteen liitetään joitain

sille tyypillisiä ominaisuuksia, kuten päättely- ja havainnointikyky, kyky ennustaa tulevia tapahtumia, oman toiminnan muuttaminen havaintojen perusteella sekä suunnittelu- ja suorituskyky tavoitteiden saavuttamiseksi. (Myllymäki, 2021, ss. 10–12)

Toinen määritelmän vaikeuteen liittyvä asia on ajan saatossa muuttuva käsitys älykkäästä toiminnasta. Esimerkiksi taskulaskimet ja termostaatit suorittavat tehtäviä, jotka saattavat vaikuttaa älykkäiltä, koska ne ylittävät ihmisen kyvyn nopeaan laskentaan tai jatkuvaan lämpötilan säätöön. Samoin puhelimen lukituksen avaaminen biometrisillä tunnisteilla, kuten sormenjäljellä tai kasvojentunnistuksella, voidaan kokea älykkääksi. Kun nämä teknologiat muuttuvat arkipäiväisiksi, ei niiden ajatella enää olevan tekoälyä. Tekoälytutkimus itsessään keskittyy jatkuvasti uusiin haasteisiin, siirtyen sellaisiin ongelmiin, jotka vielä odottavat ratkaisuja. (Myllymäki, 2021, ss. 13–14)

## **2.3 Tekoälyn etiikka**

Kuvitellaan skenaario, jossa itseohjautuva auto joutuu tilanteeseen, jossa sen on tehtävä päätös: ajaa lapsen yli tai törmätä seinään. Molemmissa vaihtoehdoissa uhataan ihmishenkeä. Kumpi päätös olisi eettisesti oikea? Kumman henki on arvokkaampi? Tämä on äärimmäinen esimerkki, mutta se korostaa tarvetta arvioida tekoälyn päätöksentekoprosesseja ja asettaa selkeitä eettisiä ohjeita. Eettiset kysymykset nousevat esille myös laajemmassa yhteiskunnallisessa keskustelussa, kuten tekoälyn vaikutuksesta työelämään ja koulutukseen. (Coeckelbergh, 2020/2021, ss. 15–18)

Tekoälyn laajeneva käyttö herättää huolta myös algoritmien oikeudenmukaisuudesta ja siitä, miten ne voivat tahattomasti aiheuttaa haittaa. Tekoälysovellukset voivat vahvistaa olemassa olevia vinoumia. Erityisen huolestuttavaa tämä on esimerkiksi terveydenhuollon sovelluksissa. Tämän vuoksi on keskeistä, että tekoälymallien kehityksessä kiinnitetään huomiota näiden vinoumien tunnistamiseen ja niiden vähentämiseen. (Meyer ym., 2023, ss. 8–9)

### **2.3.1 Vaikutukset työhön ja koulutukseen**

Tekoälyn kehitykseen liittyy usein myös pelko työpaikkojen menetyksestä. Tekoälyllä on monimutkainen vaikutus yhteiskuntaan, sillä se toisaalta luo uusia työpaikkoja mutta tulee myös vähentämään niitä. (Aaltonen, 2019, s. 25)

Toisaalta tekoäly avaa ovia uusille mahdollisuuksille sekä korkean että matalan taitotason tehtävissä, vaatien työntekijöitä sopeutumaan ja hankkimaan uudenlaista osaamista. Vaikka tekoälyn ajatellaan korvaavan ensisijaisesti yksinkertaisimmat työt, uusimmat tutkimukset osoittavat, että sen vaikutukset ulottuvat myös pitkälle erikoistuneisiin ammatteihin, kuten lääkäreihin ja lakimiehiin. Tämä korostaa sitä ajatusta, että tekoälyn leviäminen työelämässä on monitahoisempaa kuin aiemmin on arvioitu. (Aaltonen, 2019, s. 26)

Vaikka tekoäly ja koneet ylettyvät laajoille aloille ja muuttavat muuttumattominakin pidettyjä työelämän aloja niin vaikuttaisi siltä, että suuremmilta osin koneet eivät tule korvaamaan ihmisten töitä vaan pikemminkin yhteistyö koneiden ja ihmisten välillä lisääntyy. (Aaltonen, 2019, ss. 146–147) Esimerkiksi Tutkimuksessa " Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of AI on Knowledge Worker Productivity and Quality" havaittiin, että tekoälyn, erityisesti GPT-4:n käyttö konsulttityössä voi lisätä työn tuottavuutta ja laatua merkittävästi. Tutkimuksen mukaan tekoälyn avulla voidaan tehostaa sen osaamisalueisiin kuuluvia tehtäviä. Konsultit, jotka hyödynsivät tekoälyä tällaisissa tehtävissä, saivat tehtyä tehtäviä 12,2 % enemmän ja suoriutuivat niistä 25,1 % nopeammin verrattuna niihin, joilla ei ollut pääsyä tekoälyyn. Lisäksi heidän työnsä laatu oli yli 40 % korkeampaa verrattuna kontrolliryhmään. (Dell'Acqua ym. 2023, s. 2)

Tekoäly on muodostunut merkittäväksi tekijäksi myös koulutuksen saralla. Sen avulla voidaan tehostaa pedagogisia prosesseja, kuten opiskelijoiden yksilöllistä ohjausta sekä oppimisanalytiikkaa. Erityisesti koneoppimisen ansiosta opetussuunnitelmia voidaan räätälöidä opiskelijoiden yksilöllisiin tarpeisiin sopiviksi. Tekoälyn mahdollistamat toiminnot, kuten luonnollisen kielen käsittely ja älykkäät oppimisalustat, tuovat uuden ulottuvuuden oppimisympäristöihin. (Niemi ym., 2023, ss. 4–5)

Niemen ym. (2023, s. 6) mukaan on tärkeää syventää ymmärrystä tekoälyn eettisistä näkökohdista ja sen vaikutuksista oppimiseen. He suosittelevat tekoälyn käytön tarkastelua koulutusympäristössä, jotta voidaan varmistaa sen vastuullinen ja tehokas hyödyntäminen myös tulevaisuudessa. Mhalanga (2023) huomioi, että vaikka tekoälyä käytetään yhä laajemmin koulutusympäristöissä niin edelleen tutkinnot ovat opiskelijoiden keskuudessa korkeassa arvossa ja monet tekoälyn innovaatioihin liittyvät odotukset alalla ovat liioiteltuja. Mikä viittaa taas siihen, että tekoäly ei ole myöskään korvaamassa perinteistä koulutusprosessia.

### 2.3.2 Yksityisyys ja vastuu

Tekoälyn käyttöön liittyy laajenevassa määrin yksityisyyden kysymykset. Ihmiset eivät monesti edes tiedä, mihin ja mitä kaikkea tietoja heistä kerätään. (Coeckelbergh, 2020/2021, s. 101) Esimerkiksi Facebook kerää ja analysoi tekoälyn avulla kahden miljardin käyttäjänsä tietoa hyödyntääkseen niitä kohdennettuihin mainoksiin ja Experia luottotietoyhtiö käyttää tekoälyä kerätäkseen tietoja ja antaakseen niiden perusteella päätöksiä asuntolainoihin (Coeckelbergh, 2020/2021, ss. 97–98). Ei myöskään ajatella, että tietoja saatetaan jakaa kolmansille osapuolille eri asiayhteydessä kuin tietoja luovutettaessa (Coeckelbergh, 2020/2021, s. 101). Kun tietoja yhdistellään lukemattomista palveluista, niistä voi saada yllättävän kattavan tietopakettin ihmisestä ja hänen elämästään. Tämä herättää huolen tietomurtojen osalta. (Niemi, V., 2021, s. 115)

Yksityisyyden suoja Suomessa turvaa perustuslaki, mutta vaikka perustuslaki on voimassa, niin digitaalisessa ympäristössä yksityisyyden määrittely ei olekaan aivan yhtä suoraviivaista. Yksityisyyteen liitetään usein asioita, kuten salassa pidettävät henkilökohtaiset tiedot, yksityisyyden suoja ja kyky hallita ja suojella omia tietoja. Voidaanko siis esimerkiksi ajatella IP-osoitteen kuuluvan yksityisyyden piiriin? (Siukonen & Neittaanmäki, 2019, s. 306)

EU:n tietosuoja-asetus määrittelee, mitä ja mihin tarkoitukseen tietoja saa kerätä (Niemi, V., 2021, s. 128). Asetuksen mukaan henkilötietoja ovat kaikki sellaiset tiedot tai tietojen yhdistäminen, joista henkilö voidaan tunnistaa. Esimerkiksi sähköpostiosoite ja IP-osoite kuuluvat tällaisten henkilötietojen piiriin. (Tietosuojavaltuutetun toimisto, n.d.)

Eettiset- ja vastuukysymykset herättävät tekoälyn osalta haasteita. Jos esimerkiksi itseohjautuva auto ajaa kolarin niin kenellä on vastuu? (Siukonen & Neittaanmäki, 2019, s. 303) EU:ssa ollaan ottamassa käyttöön maailman ensimmäistä tekoälylakia, jonka tarkoituksena on suojella ja varmistaa, että tekoälyjärjestelmät ovat läpinäkyviä, turvallisia ja niitä voidaan jäljittää ja että niissä toteutuu tasa-arvo ja ympäristöystävällisyys (Euroopan parlamentti, 2023a). Valvonta- ja vastuukysymykseen liittyen ehdotuksessa painotetaan, että vastuu on aina tekoälyjärjestelmien tarjoajilla ja järjestelmät tulee suunnitella niin, että niiden valvonta onnistuu luonnollisten henkilöiden toimesta. (Euroopan komissio, 2021, ss. 32–33)



## 2.4 Vahva ja heikko tekoäly

Tietokoneohjelmat prosessoivat tietoa noudattaen vain annettuja ohjeita ja määräyksiä, ilman kykyä hahmottaa tai tulkita tietojen todellista merkitystä tai niiden asiayhteyksiä. Ne voivat jäljitellä älykkäältä vaikuttavaa toimintaa, mutta eivät omaa ihmismäistä kykyä hahmottaa ja luokitella tietoa itsenäisesti. Tekoälyä voidaan tarkastella heikon ja vahvan tekoälyn näkökulmista. Heikko tekoäly suorittaa rajattuja tehtäviä ihmisen asettamien sääntöjen mukaan, kun taas vahva tekoäly pyrkii jäljittelemään ihmisälyn laaja-alaista ymmärrystä ja päättelykykyä. (Siukonen & Neittaanmäki, 2019, ss. 44–45)

### 2.4.1 Heikko tekoäly

Heikkoa tekoälyä kuvataan ohjelmiksi, jotka kykenevät suorittamaan tiettyjä toimintoja älykkäästi algoritmien ja koneoppimisen avulla. Tällaiset ohjelmistot ovat erikoistuneet ja optimoitu suorittamaan määrättyjä tehtäviä, kuten internet-haun tehostaminen tai roskapostien suodattaminen. Heikkoa tekoälyä hyödynnetään myös tutkimusavustajana ja erilaisissa kodinkoneissa, jotka tekevät rutiininomaisia töitä ihmisen puolesta, kuten robottimurit tai automaattiset ruohonleikkurit. (Siukonen & Neittaanmäki, 2019, s. 45)

Heikon tekoälyn omaavat ohjelmat eivät ymmärrä ympäröivää maailmaa. Ihmisen voi olla hankala hahmottaa, ettei esimerkiksi generatiivinen kielimalli, kuten ChatGPT oikeasti ymmärrä tuotettuja tekstejä vaan se tekee tilastollisia analyyssejä ennustaakseen seuraavia sanoja. (Myllymäki, 2021, s. 17)

### 2.4.2 Vahva tekoäly

Vahvaa tekoälyä voidaan pitää heikon tekoälyn vastakohtana (Myllymäki, 2021, s. 15). Sen tutkimus keskittyy kehittämään ihmisen tietoisuuden kaltaisia järjestelmiä, jotka eivät vain suorita määrättyjä tehtäviä, vaan ylittävät myös ihmisen kognitiiviset kyvyt. Tavoitteena on luoda koneita, jotka omaavat tietoisuutta ja kykyä ymmärtää ympäristöään. Tällaiset järjestelmät voivat tehdä itsenäisiä päätöksiä ja kehittää omia tavoitteitaan. (Siukonen & Neittaanmäki, 2019, s. 45)

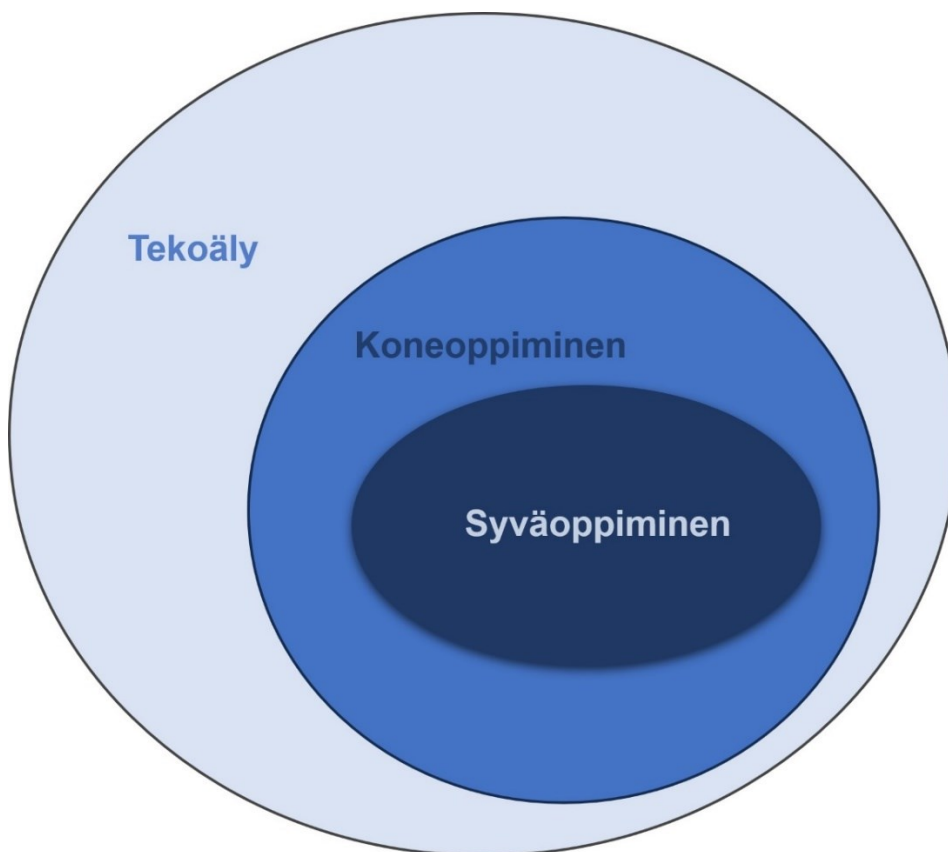
Tämä edistynyt tekoälyn muoto edellyttäisi monimutkaisia ja uudenlaisia ratkaisuja, jotka perustuvat syvälliseen ymmärrykseen ihmisaivojen toiminnasta. Vaikka vahva tekoäly on vielä pitkälti teoreettinen käsite, sen toteutuminen voisi muuttaa perustavanlaatuisesti

käsitystämme koneiden kyvystä toimia ja tehdä päätöksiä itsenäisesti. (Siukonen & Neittaanmäki, 2019, s. 45)

## 2.5 Koneoppiminen ja syväoppiminen

Myllymäen (2021, s. 20) mukaan koneoppiminen on yksi tekoälyn keskeisistä osa-alueista ja siihen kytkeytyy Kelleherin (2019/2020, s. 15) mukaan myös olennaisesti neuroverkkoihin perustuva syväoppiminen. Syväoppiminen juontaa juurensa niin tekoälytutkimuksesta kuin koneoppimisen tutkimuksesta, kuten kuva 1 esittää.

Kuva 1 Tekoälyn, koneoppimisen ja syväoppimisen suhde (mukaillen Kelleher, 2019/2020, s. 15).



### 2.5.1 Koneoppiminen

Vaikka termi "koneoppiminen" viittaa oppimiseen, ei se ole Myllymäen (2021, s. 20) mukaan ihmisen kaltaista oppimista. Koneet eivät opi itsenäisesti, vaan niiden toiminta perustuu algoritmien soveltamiseen ja säätämiseen. Kelleher (2019/2020, s. 16) täsmentää, että

algoritmien kehittäminen ja arviointi ovat keskeisiä osa-alueita koneoppimisessa ja tällaiset algoritmit on suunniteltu analysoimaan tietoaineistoja sekä oppimaan niistä toistuvia malleja.

Koneoppimisprosessi voidaan erottaa kahteen päävaiheeseen: ensimmäinen vaihe on koulutus ja toinen on soveltaminen. Koulutusvaiheessa algoritmi analysoi tietoaineistoa, valitsee parhaiten soveltuvan mallin ja tallentaa sen, jota myöhemmin käytetään ohjelmassa. Malli oppii aineiston tunnuspiirteitä, jotka ovat keskeisiä koneoppimisprosessissa. Opetuksen jälkeen malli pysyy muuttumattomana ja sitä sovelletaan uusiin aineistoihin, jolloin koneoppimismalli tuottaa ennusteita tai arvioita näistä aineistoista. (Kelleher, 2019/2020, ss. 21–22)"

Koneoppimisen keskeisimmät suuntaukset voidaan jakaa kolmeen osa-alueeseen: ohjattuun oppimiseen, ohjaamattomaan oppimiseen sekä vahvistusoppimiseen. Yleisimmin näistä on hyödynnetty ohjattua oppimista (Kelleher, 2019/2020, s. 36).

Ohjatussa oppimisessa keskeistä on, että jokaiseen tietoaineiston esimerkkiin liitetään ennakoitu ulostuloarvo eli kohdearvo, kuten Kelleher (2019/2020, s. 32) mainitsee. Tämän havainnollistamiseksi Myllymäki (2021, ss. 20–21) kuvaa tapausta, jossa ohjelmaa opetetaan tunnistamaan kuvia. Tässä prosessissa ohjelmaa koulutetaan kuvien perusteella ja se voi saada syötteenä kuvakokoelman, jossa osa kuvista on merkitty kissaksi (tunnisteella 0) ja osa koiraksi (tunnisteella 1). Ohjelman tavoitteena on oppia tunnistamaan ja luokittelemaan nämä kuvat antaen tulokseksi numeraalisia arvoja. Numerot välillä 0–0,5 luokittelevat kuvan kissaksi, kun taas numerot välillä 0,5–1 viittaavat koiraan. Koulutusdataan sisältyvät ihmisen antamat tunnisteet auttavat ohjelmaa oppimaan ja säätämään toimintaansa. Lopullisena tavoitteena on, että ohjelma pystyy itsenäisesti tunnistamaan ja luokittelemaan myös niitä kuvia, joita se ei ole aiemmin nähnyt.

Ohjaamattomassa oppimisessä keskitytään datajoukkojen klusterointiin eli samankaltaisen aineiston ryhmittelyyn. Tässä menetelmässä ei ennalta määritetä, millaisia lopullisia ryhmiä tulisi muodostua, vaan tavoitteena on tunnistaa samankaltaiset esimerkit ja järjestää ne klustereiksi niin, että klusterin sisäinen samankaltaisuus on suurempi kuin muiden klustereiden sisällä olevien esimerkkien välillä. Tämä lähestymistapa on hyödyllinen esimerkiksi asiakassegmentoinnissa, jossa yritykset pyrkivät kohdentamaan toimintaansa erilaisten asiakasryhmien tarpeiden mukaisesti. Ohjaamattomassa oppimisessä algoritmi aloittaa yleensä luomalla alustavan ryhmittelyn esimerkeistä ja parantaa sitä vaiheittain. Tavoitteena on parantaa klusterien sisäistä samankaltaisuutta ja samanaikaisesti lisätä niiden erilaisuutta verrattuna toisiinsa. (Kelleher, 2019/2020, ss. 33–34)

Vahvistusoppimisessa keskeistä on oppiminen suorassa vuorovaikutuksessa ympäristön kanssa, jossa toimija eli oppiva järjestelmä pyrkii tekemään valintoja saadun palautteen perusteella. Toisin kuin ohjatussa tai ohjaamattomassa oppimisessa, vahvistusoppimisessa ei käytetä ennalta määriteltyjä kohdearvoja tai klustereita. Jos teko johtaa positiiviseen palautteeseen, toimintatapa vahvistuu ja negatiivinen palaute heikentää sitä. Tämä jatkuva vuorovaikutus ja oppiminen ympäristöstä tekevät vahvistusoppimisesta dynaamisen ja tehokkaan menetelmän monimutkaisissa ympäristöissä, kuten tietokonepeleissä, jossa toimija oppii pelaamaan peliä menestyksekkäästi. (Kelleher, 2019/2020, s. 34)

Liun (2023, s. 1) mukaan RLHF (Reinforcement Learning from Human Feedback) eli vahvistusoppiminen ihmispalautteella on viime aikoina noussut esiin tärkeänä menetelmänä tekoälyn oppimisessa. Kyseinen menetelmä eroaa perinteisestä vahvistusoppimisesta niin, että se hyödyntää sekä automaattisesti generoitua palautetta että ihmisen antamaa palautetta. Tässä menetelmässä ihmispalautte on keskeisessä roolissa, sillä se mahdollistaa tekoälyn ohjaamisen käyttäjän kokemusten ja näkemysten mukaisesti, mikä lisää oppimisen tehokkuutta ja monipuolisuutta.

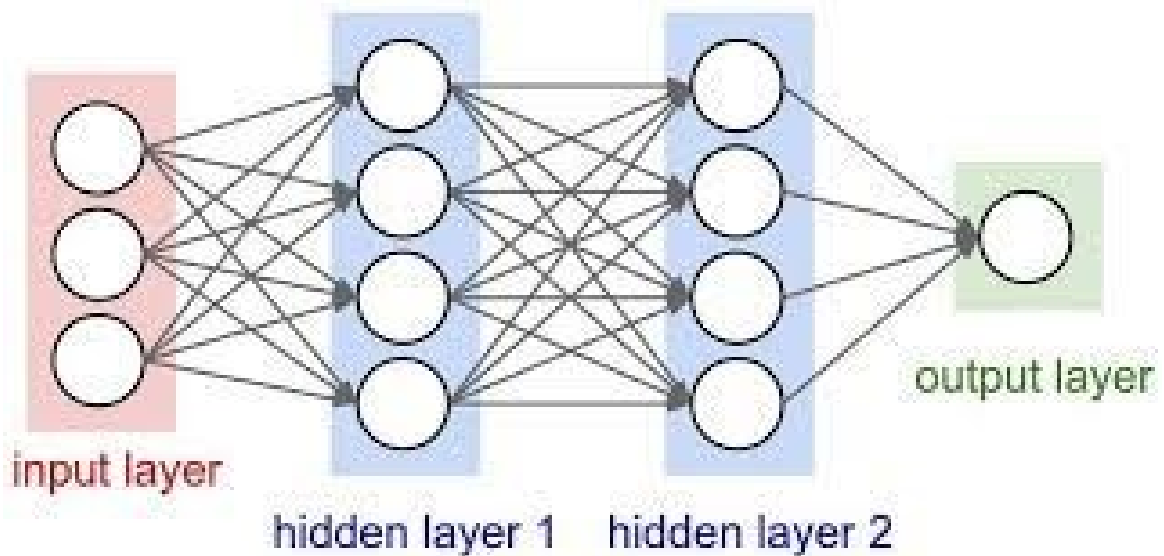
OpenAI:n ChatGPT ja DeepMindin Sparrow ovat esimerkkejä sovelluksista, jotka käyttävät RLHF menetelmää ja osoittavat tämän menetelmän avulla kehitettyjen tekoälymallien monipuolisuuden. RLHF tarjoaa monia etuja perinteiseen vahvistusoppimiseen verrattuna, mutta tuo mukanaan myös haasteita. Nämä haasteet liittyvät esimerkiksi mahdollisiin virheellisiin käyttäytymismalleihin ja ihmispalautteen keräämisen kustannuksiin. (Liu, 2023, s. 2)

### **2.5.2 Syväoppiminen**

Syväoppiminen on keskeinen osa koneoppimista, erikoistuen erityisesti syviin neuroverkkomalleihin (Kelleher, 2019/2020, s. 18). Neuroverkot jäljittelevät ihmisaivojen toimintaa yhdistämällä tietosyötteitä, painoarvoja ja vinoumia (IBM, n.d.). Syväoppimisen neuroverkot koostuvat useista kerroksista, jotka yhdessä muodostavat tiedonkäsittelyn verkoston. Jokainen kerros sisältää yksinkertaisia neuroneja, jotka ovat vertauskuvia ihmisaivojen hermosoluille. (Kelleher, 2019/2020, s. 64) Liian suoraviivaisesti ei kuitenkaan ole hyvä verrata näitä keskenään, sillä ihmisen aivot ovat paljon monimutkaisempi järjestelmä kuin keinotekoiset neuroverkot (Myllymäki, 2021, s. 24).

Neuroverkon neuroni vastaanottaa ja välittää signaaleja, jotka aktivoituvat, kun se saa riittävän vahvan ärsyksen. Kuten kuva 2 havainnollistaa, verkoissa on tyypillisesti syötekerros, useita piilokerroksia ja ulostulokerros. Piilokerrosten avulla neuroverkko pystyy käsittelemään ja mallintamaan monimutkaisia suhteita, jotka syntyvät neuronien välisestä vuorovaikutuksesta. (Kelleher, 2019/2020, ss. 65–66) Kerroksellisuuden ansiosta verkko syvenee, mikä viittaa myös termiin ”syväoppiminen”, ja pystyy oppimaan yhä monimutkaisempia asioita, mutta vaatii vastapainoksi yhä suurempia datamääriä sekä laskentatehoa (Elements of AI, n.d.-b).

Kuva 2 Neuroverkko (Johnson, 2020)



Painoarvoilla eli parametreilla on keskeinen rooli neuroverkkojen toiminnassa. Näitä painoarvoja voidaan verrata esimerkiksi yksinkertaiseen lainantajan päätöksentekoprosessiin, jossa arvioidaan lainahakemuksia. Lainantaja ottaa hakemuksissa huomioon erilaisia tekijöitä, kuten hakijan tuloja ja velkoja. Näillä tekijöillä on kullakin tietty painoarvonsa, joilla on merkitystä päätöksen kannalta. (Kelleher, 2019/2020, ss. 46–47) Lainantaja voisi esimerkiksi määritellä tulot kolme kertaa tärkeämmäksi kuin velat, kuten kaava 1 osoittaa (Kelleher, 2019/2020, s. 50).

Kaava 1 Lainanmaksukykyyn esimerkki (Kelleher, 2019/2020, s. 50)

$$\text{Lainanmaksukyky} = (\text{tulot} \times 3) + (\text{velat} \times 1)$$

Samalla ajatuksella syväoppimisen neuroverkkojen toimintaperiaate perustuu painoarvojen käyttöön. Jokainen neuroverkon neuroni laskee painotetun summan saamistaan syötteistä,

jossa kunkin syötteen merkitys määräytyy sen painoarvon perusteella. (Kelleher, 2019/2020, ss. 68–69). Nämä painoarvot ovat neuroverkon oppimisen ydin, sillä ne määrittelevät sen, että lopputulokseksi saadaan oikeat tulosteet. Painoarvoja voidaan säätää ja optimoida opetusvaiheen aikana, jolloin verkko ”oppii” tunnistamaan ja prosessoimaan tietoa tehokkaammin (Elements of AI, n.d.-b).

Kun neuroverkko syvenee, voidaan tarvita jopa miljardeja parametreja, jolloin niiden säätely luonnollisesti vaikeutuu. Esimerkiksi kuvien luokittelutehtävässä, jossa verkko käsittelee tuhansia kuvia koirista ja kissoista, on tärkeää, että verkon painoarvot säädetään oikein. Jos painoarvoja säädettäisiin vain yhden kuvan perusteella, muut kuvat saattaisivat tulla luokitelluiksi väärin. Tässä tilanteessa nousee esille backpropagation-algoritmin tärkeys, sillä se mahdollistaa painoarvojen hienosäädön, jotta kaikkien kuvien luokittelu menisi kohti oikeaa. (Myllymäki, 2021, ss. 22–23)

Backpropagation-algoritmi tekee pieniä muutoksia painoarvoihin siten, että jokaisen käsitellyn kuvan kohdalla neuroverkon antama tulos lähestyy tarkempaa luokittelua. Kun käsitellään kissojen kuvia, tavoitteena on saada tulos lähemmäksi nollaa, kun taas koirien kuvien tapauksessa tavoitteena on päästä lähelle ykköstä. Säättöprosessia toistetaan jopa miljoonia kertoja, kunnes verkko saavuttaa suuren tarkkuuden kuvien luokittelussa. Jos haluttua lopputulosta ei saavuteta, aloitetaan prosessi uudelleen antamalla painoarvoille uudet satunnaisarvot. (Myllymäki, 2021, ss. 22–23)

Painoarvojen lisäksi neuroverkoissa tarvitaan vinoumaa, joka toimii vakioterminä ja lisätään ominaisuuksien ja painoarvojen tulokseen. Sitä käytetään tuloksen tasapainottamiseen niin, että se siirtää mallin aktivointifunktiota kohti positiivista tai negatiivista suuntaa. (Turing, n.d.)

Syväoppiminen on osoittautunut merkittäväksi työkaluksi luonnollisen kielen käsittelyssä. Sen avulla voidaan kehittää monimutkaisia malleja kielen ymmärtämiseen ja tuottamiseen. Esimerkiksi, kuten kuvantunnistuksessa syväoppimisen avulla voidaan tunnistaa ja prosessoida monimutkaisia malleja niin samalla tavalla luonnollisen kielen käsittelyssä voidaan prosessoida monimutkaisia kielellisiä rakenteita. (Goodfellow, ym., 2016, ss. 99–100)

## 2.6 Luonnollisen kielen käsittely

Luonnollisen kielen käsittely (NLP) on tekoälyn osa-alue, joka keskittyy siihen, miten koneet voivat ymmärtää, käsitellä ja tuottaa ihmiskieltä tai sen kaltaisia tietoja. NLP jakautuu pääasiassa kahteen osa-alueeseen: luonnollisen kielen ymmärtämiseen (NLU) ja luonnollisen kielen generointiin (NLG). NLU keskittyy tekstin tulkintaan ja analysointiin, kun taas NLG mahdollistaa koneiden tekstin tuottamisen. NLP-teknologioita hyödynnetään monipuolisesti erilaisissa sovelluksissa, kuten automaattisissa käännöspalveluissa, tekstin luokittelussa, tiivistämisessä sekä suurten tietomassojen tiedonhaussa. Lisäksi NLP mahdollistaa vuorovaikutteiset järjestelmät, kuten edistyneen GPT-3-mallin, joka kykenee käymään keskustelua käyttäjien kanssa. (DeepLearning.AI, 2023)

Luonnollisen kielen käsittelyn (NLP) toiminta perustuu kielen osien ja niiden välisten osien ymmärtämiseen ja mallintamiseen. Prosessissa keskeistä ovat datan esikäsittely, ominaisuuksien erottelu ja mallinnus. Esikäsittelyssä tekstiä valmistellaan analyysiin ja mallintamiseen sopivaksi. Esimerkiksi tokenisaatio on osa tätä prosessia, jossa teksti jaetaan pienempiin yksiköihin, kuten sanoiksi, lauseen osiksi tai yksittäisiksi merkeiksi.

(DeepLearning.AI, 2023) Tämä vaihe on erityisen tärkeä, jotta kielimallit, kuten OpenAI:n GPT-sarja, ymmärtävät ja pystyvät analysoimaan kieltä (Salo, 2023, s. 46).

Ominaisuuksien erotteluvaiheessa NLP:ssä käytetään menetelmiä kuten upotuksia (embeddings) tekstin syvällisempään käsittelyyn. Upotukset ovat olennaisia tekoälyn kyvyssä ymmärtää ja käsitellä kieltä, auttaen sitä hahmottamaan kielellisiä yhteyksiä ja merkityksiä (Salo, 2023, s. 50). Syväoppiminen, erityisesti Transformer-arkkitehtuurit, mahdollistavat näiden upotusten tehokkaan hyödyntämisen. Mallintamisvaiheessa upotuksia käytetään hyväksi tekstiaineiston käsittelyssä, mikä mahdollistaa kielimallien ennustaa ja tuottaa ymmärrettävää sekä kontekstiin sopivaa tekstiä. (DeepLearning.AI, 2023)

## 3 ChatGPT

OpenAI:n luoma ChatGPT on generatiivinen tekoälysovellus, joka mahdollistaa vuorovaikutuksen ihmisen kanssa luonnollisella kielellä (OpenAI, 2022). Käyttäjä voi siis kirjoittaa kysymyksiä ja ChatGPT vastaa näihin kysymyksiin uskottavalla tavalla (Salo, 2023, ss. 43–44). Se perustuu GPT-3-kielimalliin, joka ilmestyi vuoden 2022 lopulla yleisön saataville. Periaatteessa ChatGPT:n voisi kuvailla sen olevan erittäin taitava chatbot, joka osaa vastaila kysymyksiin monilla eri kielillä. (Kolari & Kallio, 2023, s. 43) Ilmaisversiona on tällä hetkellä GPT-3.5 ja noin 20 euron kuukausihintaan saa Plus-version käyttöön, mikä perustuu GPT-4-malliin (Salo, 2023, s. 35). ChatGPT:ssä oli aiemmin rajoitteena sen datan ulottuminen vain 2021 syyskuuhun asti, mutta loppuvuoden 2023 tapahtuneen päivityksen myötä plusversio voi selata internetiä ja saada ajankohtaista tietoa. (OpenAI, 2024)

### 3.1 Kielimallit ja GPT-historia

Suuret kielimallit, kuten GPT-sarja (Generative Pre-trained Transformer) ovat olennainen osa tekoälyn ja luonnollisen kielen käsittelyn kehitystä (Kasneci ym., 2023). Tällaisissa generatiivisissa kielimalleissa hyödynnetään NLP:tä, jonka ansiosta ne pystyvät tunnistamaan, kääntämään, ennustamaan tai tuottamaan tekstiä tai muuta sisältöä (Elastic, n.d.). Lisäksi niissä hyödynnetään Transformer-arkkitehtuuria (Salo, 2023, s. 34), joka on tehokkaampi vaihtoehto perinteisille malleille, kuten toistuville neuroverkoille. Transformer-arkkitehtuuri esiteltiin vuonna 2017 julkaisussa ”Attention is All You Need” ja se on ensimmäinen malli, joka perustuu täysin itsehuomioon. Sen kyky ymmärtää ja käsitellä kielen monimutkaisuutta auttaa kielimalleja kehittymään. (Vaswani ym., 2017. s. 2)

Transformer-mallin kehityksestä seurasi OpenAI:n julkaisema GPT-1 vuonna 2018, mikä sisälsi 117 miljoonaa parametria (Ojanperä, 2023, s. 29). Vuonna 2019 julkistettiin GPT-1:n seuraaja GPT-2-malli, joka kykeni jo huomattavasti luonnollisempaan ja monimutkaisempiin tekstikokonaisuuksiin kuin GPT-1. Malli sisälsi 1,5 miljardia parametria ja se koulutettiin 8 miljoonan verkkosivun aineistolla. (OpenAI, 2019)

Tuolloin Salon (2023, s. 34) mukaan OpenAI päätti, ettei julkaisisi lainkaan sen koulutusmalleja väärinkäytösepäilyjen vuoksi ja vuosina 2020–2021 kehitettiin GPT-3-malli, joka pystyi jo muun muassa oppimaan ilman, että sitä erityisemmin koulutettiin. Brown ym. (2020, s. 5) täsmentää, että GPT-3-malli sisälsi jo 174 miljardia parametria. Ojanperän



(2023, s. 29) mukaan nykyinen GPT-4 sisältää biljoona parametria, joten laskentateho ja ylipäättään kielimallien kyvykkyys on noussut räjähdysmäisesti.

## 3.2 ChatGPT:n toimintaperiaate

ChatGPT tuottaa vastauksena tekstiä käyttäjän antamien syötteiden, eli ohjeiden tai toisin sanoen kehotteiden (prompti), perusteella. Promptien avulla sovellusta ohjataan suorittamaan tehtäviä halutulla tavalla ja sen ansiosta sitä voidaan hyödyntää monipuolisesti esimerkiksi sisällön luonnissa, koodin kirjoittamisessa, tekstin tiivistämisessä, keskustelutehtävissä ja luovassa kirjoittamisessa. (OpenAI, 2022) ChatGPT:n toiminta perustuu aiemmissa kappaleissa käsiteltyihin syvä- ja koneoppimiseen, neuroverkkoihin, GPT-malleihin ja luonnollisen kielen käsittelyyn. Se on koulutettu laajalla aineistolla, joka sisältää erilaisia tekstejä, kuten kirjoituksia ja artikkeleita. Näiden ansiosta ChatGPT pystyy tunnistamaan kielen rakenteita ja ennustamaan tekstiä. (Salo, 2023, s. 43)

ChatGPT ei keskity ainoastaan yksittäisten lauseiden tuottamiseen, vaan se kykenee luomaan laajempia tekstikokonaisuuksia, kuten kappaleita ja säkeistöjä. Sen tavoitteena on tuottaa yhtenäisiä ja kontekstiin sopivia vastauksia. (Salo, 2023, s. 44) ChatGPT osaa myös käsitellä jatkokysymyksiä, hylätä sopimattomia pyyntöjä, myöntää virheensä sekä korjata käyttäjän virheellisiä väitteitä (OpenAI, 2022). Eräs tärkeä ominaisuus on sen kyky muistaa ja hyödyntää aiempaa keskustelua tokenien rajoissa, mikä tekee vuorovaikutuksesta luontevaa ja johdonmukaista. Esimerkiksi käyttäjä voi pyytää ChatGPT:tä korjaamaan aiemmin tuottamaa vastausta. Näiden ominaisuuksien ansiosta ChatGPT:n kanssa käyty keskustelu muistuttaa aitoa vuoropuhelua. (Salo, 2023, s. 44) Token-rajoite näkyy konkreettisesti ChatGPT:n kanssa käydyssä vuoropuhelussa niin, että ChatGPT saattaa unohtaa aiemmin käytyjä keskusteluja sekä sen tuottama vastaus generointivaiheessa yhtäkkiä pysähtyy ja loppuu, jolloin se täytyy generoida uudestaan (Ojanperä, 2023, s. 40).

### 3.2.1 Pluginit

ChatGPT:lle kehitettiin ja julkaistiin maaliskuussa 2023 yhteensopivuus laajennoksiin eli plugineihin. Pluginit ovat eräänlaisia lisäosia, jotka on kehitetty nimenomaan kielimallien käyttöön. Nämä pluginit mahdollistavat ChatGPT:lle pääsyn muun muassa tuoreeseen tietoon sekä niiden avulla voidaan hyödyntää kolmansien osapuolten palveluja. OpenAI ryhtyi ottamaan plugineja käyttöön ChatGPT:ssä vaiheittain, jotta niiden vaikutuksia pystyttiin havainnoimaan esimerkiksi turvallisuuteen liittyen. Pluginien avulla ChatGPT voi toimia

monipuolisemmin, esimerkiksi tarjoamalla matkasuunnittelua, markkinatietoja tai auttamalla ostosten teossa, kuten Expedia, FiscalNote ja Instacart -yritysten pluginit tekevät. Lisäksi OpenAI itse ylläpitää muutamia plugineja, kuten Web Browser ja Code Interpreter. (OpenAI, 2023a)

Pluginien määrä kasvoi lyhyessä ajassa räjähdysmäisesti. Toukokuussa 2023 oli saatavilla noin 100 erilaista pluginia, kun syyskuussa 2023 niitä on ollut jo melkein 1000. Uusia lisäosia tuli keskimäärin 67 lisää viikossa. Suosituimpia ovat verkko-ostamiseen, dataan ja tutkimukseen sekä verkkosivujen työkaluihin liittyvät pluginit. Yksittäisten käyttökohteiden kärjessä ovat data-analytiikkaan, keskusteluun, matkailuun, automatisointiin, ostosassistentteihin, apuohjelmiin, ohjelmointiin, ja oppimiseen liittyvät pluginit. (Chincha, 2023)

Pluginit helpottavat ChatGPT:n käyttöä ja esimerkiksi opiskelua varten niitä on olemassa useita erilaisia. Aiemmin suosittuja plugineja olivat esimerkiksi ScholarAI, joka auttoi löytämään relevantteja artikkeleita ja esitteli niiden keskeiset sisällöt. Daily Learning taas auttoi luomaan henkilökohtaisen oppimissuunnitelman ja toimi kuten opettaja, antaen päivittäisiä tehtäviä ja opetusmateriaaleja, mutta sittemmin molemmat on sisällytetty suoraan GPT-mallien ominaisuuksiin. (whatplugin.ai, 2024) Lisäksi on olemassa promptauksen helpottamiseksi tehty PromptPerfect, joka optimoi, editoi ja tarkastaa syötetyn kehotteen mahdollisimman tehokkaaksi niin, että kielimalli ymmärtää ja osaa antaa oikeanlaisen vastauksen siihen (PromptPerfect, 2023).

Vaikka pluginit tuovat ChatGPT:lle monipuolisia toiminnallisuuksia, ne asettavat myös uusia tietoturva-, yksityisyys ja turvallisuushaasteita. Kolmannen osapuolen kehittämät pluginit eivät ole automaattisesti luotettavia. Ne voivat käynnistää kehotteen injektiohyökkäyksiä ja niiden luonnollisen kielen käyttö voi johtaa epämääräisiin ja epätarkkoihin tulkintoihin, sillä kehotteet voidaan tulkita liian laajasti tai suppeasti eikä alusta näin ollen osaa hyödyntää pluginia oikein. Lisäksi ainakin jotkut suuret kielimallit, kuten OpenAI, asettavat vain vähäisiä rajoituksia kolmannen osapuolen plugineille. Tämä voi johtaa heikkoon tarkastusprosessiin ja lisätä turvallisuusriskien kasvua. (Iqbal ym., 2023, s. 1)

### **3.2.2 Mukautettavat GPT:t**

OpenAI on kehittänyt GPT:t, jotka ovat mukautettavia ChatGPT-versioita, joita käyttäjät voivat luoda erilaisiin käyttötarkoituksiin ilman koodaustaitoja. Luomisprosessi tapahtuu

vuoropuhelulla ChatGPT:n kanssa GPT:n luomistilassa. Prosessi on suunniteltu helppokäyttöiseksi ja vuoropuhelun tarkoituksena on määrittellä GPT:lle käyttäjän haluamia ominaisuuksia. (OpenAI, 2023b)

Lisäksi GPT:t helpottavat ChatGPT:n käyttöä automatisoimalla kehoitteita ja ohjeita. Aiemmin käyttäjät kehittivät tehokkaiksi havaitsemistaan kehoitteista omia listoja, joita käyttivät ChatGPT:n kanssa kommunikointiin. Uudet GPT:t automatisoivat tämän prosessin, mikä tekee ChatGPT:n käytöstä helpompaa. Tällä hetkellä GPT:t ovat saatavilla ChatGPT Plus- ja Enterprise-käyttäjille. (OpenAI, 2023b)

Personoidut GPT:t tarjoavat etuja perinteiseen ChatGPT:hen verrattuna, erityisesti niiden kyvyssä käsitellä ajantasaista ja käyttäjäkohtaista tietoa. Toisin kuin normaali keskusteluikkuna ChatGPT:n kanssa, personoidut GPT:t voivat hyödyntää käyttäjien lataamia tiedostoja. Tämä mahdollistaa niiden tietopohjan laajentamisen ja päivittämisen uusimmilla tiedoilla, mikä myös mahdollistaa tarkempien ja yksityiskohtaisten vastausten tuottamisen. GPT:ssä on myös asetuksia, joista voi esimerkiksi valita verkkoselauksen päälle. GPT:t voidaan myös jakaa muille käyttäjille, jolloin nopeasti voi hakea tarpeisiinsa sopivan GPT:n ilman, että luo kaikkia itse. (Lutkevich, 2023)

### 3.2.3 ChatGPT:n nurja puoli

ChatGPT:n monipuolisuudesta huolimatta sen käyttöön liittyy myös haasteita. Kolarin ja Kallion (2023, ss. 44–45) mukaan, vaikka ChatGPT tuottaa uskottavalta kuulostavaa tekstiä, sen antamat vastaukset eivät aina ole virheettömiä tai edes järkevän kuuloisia. Se voi vastata kysymykseen nähden turhan laajasti ja antaa useita virheellisiä tietoja sisältäviä vastauksia. ChatGPT perustuu ihmisten kirjoittamaan dataan, mikä voi aiheuttaa epätasapainoa esimerkiksi sukupuolten tai eri ikäryhmien käsittelyssä.

ChatGPT:n ja muiden laajojen kielimallien käyttöön liittyy olennaisesti myös ympäristökysymykset. Näiden mallien koulutus ja käyttö kuluttavat merkittävästi energiaa, mikä taas johtaa korkeisiin hiilidioksidipäästöihin. Esimerkiksi BERT-mallin, joka on kooltaan pienempi kuin ChatGPT, on pelkästään koulutusvaiheen arvioitu aiheuttavan noin 21–78 tonnin CO<sub>2</sub>-päästöt. ChatGPT vaatii vielä enemmän energiaa sekä koulutuksessa että jokapäiväisessä käytössään, vastatessaan miljardeihin kysymyksiin. Tämän vuoksi olisi tärkeää kehittää vastuullisesti tekoälymalleja ottaen huomioon kestävä kehitys. (Meyer ym., 2023, ss. 5–6)

ChatGPT:n yksityisyyden suoja herättää osaltaan myös huolenaiheita. Esimerkiksi on riski, että ChatGPT jakaa tietoa käyttäjistään ilman lupaa kolmansille osapuolille, mikä on Euroopan yleisen tietosuoja-asetuksen, GDPR:n vastaista. Erityisesti on huolestuttavaa, miten laajasti ja millaisia henkilökohtaisia tietoja, kuten blogipostauksia tai tuote-arvosteluja, on mahdollisesti hyödynnetty ChatGPT:n koulutusprosessissa ilman asianomaisten suostumusta. Toinen huolenaihe aiheeseen liittyen on, miten OpenAI hallinnoi käyttäjien syöttämiä tietoja. Vaikka ChatGPT käyttää hyväkseen käyttäjien antamia kehoitteita parantaakseen toimintaansa, esimerkiksi vähentämällä virheellisiä tai harhaanjohtavia vastauksia, on silti epävarmaa, käsitelläänkö käyttäjien henkilökohtaisia tietoja asianmukaisesti ja pidetäänkö kiinni lupauksesta olla tallentamatta niitä. (Wu ym., 2023, ss. 8–10)

### 3.3 Tehokas kommunikointi ja promptaus

Kun hyödynnetään kehoitteilla toimivia suuria kielimalleja, kuten ChatGPT:tä, on tärkeää antaa niille selkeitä ja yksityiskohtaisia ohjeita, jotta saavutetaan haluttu lopputulos. Tällaisia kielimalleja voidaan ajatella älykkäinä toimijoina, jotka kuitenkin tarvitsevat tarkat tiedot tehtävästään. Esimerkiksi pyydetessä tietoa jostain tunnetusta henkilöstä, on hyödyllistä täsmentää, mitä hänestä halutaan tietää. Ollaanko kiinnostuneita esimerkiksi henkilökohtaisesta elämästä vai saavutuksista. Tekstin sävyn määrittelyllä voidaan päästä lähemmäs haluttua kontekstia esimerkiksi pyydetessä mallia generoimaan vastauksensa ammattijournalistina, tuottaa se erilaisen vastauksen kuin rennommassa sävyssä pyydetty vastaus. (Fulford & Ng, n.d.)

Toinen huomioitava seikka on, että vaikka ChatGPT toimii useilla kielillä, sen käyttö on tehokkainta englannin kielellä. Tämä johtuu siitä, että englanninkielistä koulutusdataa on saatavilla enemmän, mikä parantaa vastausten tarkkuutta ja yksityiskohtaisuutta. Suomen kielellä saadut vastaukset voivat olla enemmän virheellisiä sekä epätarkkoja johtuen suomenkielisen datan rajallisemmasta määrästä. (Kolari & Kallio, 2023, s. 44)

Vaikka ohjenuorana voidaan pitää kehoitteiden yksinkertaisuutta ja selkeyttä, tehokkaan kommunikoinnin ei tarvitse tarkoittaa lyhyitä komentoja. Päinvastoin Fulfordin ja Ng (n.d.) mielestä pidemmät viestit voivat tarjota enemmän selkeyttä ja tarvittavaa kontekstia mallille, mikä edesauttaa tarkkojen ja aiheeseen keskittyvien vastausten tuottamista.

Promptauksen kehittäminen GPT-4:n kaltaisilla suurilla kielimalleilla on monivaiheinen prosessi, jossa harkittu lähestymistapa on avainasemassa. Vaikka kokeilujen rooli on tärkeä ja jokainen käyttäjä löytää omat menetelmänsä, voidaan promptauksen peruseriaatteet jakaa kuuteen keskeiseen osa-alueeseen: selkeät ohjeet, lähteiden käyttö, osittaminen, aika, ulkoset työkalut ja testaaminen. (OpenAI, n.d.)

### 3.3.1 Selkeät ohjeet

Selkeiden, yksityiskohtaisten ja johdonmukaisten ohjeiden antaminen on ensiarvoisen tärkeää, kun kommunikoidaan suurten kielimallien, kuten GPT-4:n, kanssa. Malli ei osaa lukea käyttäjän ajatuksia, joten ohjeiden tarkkuus on avainasemassa. Tavoitteena on vähentää mallin arvailua ja näin parantaa vastausten tarkkuutta. Esimerkiksi, mikäli haluttaisiin tiivistelmä jostain tekstistä niin kannattaa pyytää lyhyitä, tiiviitä vastauksia. Jos taas vastauksissa kaivataan enemmän syvyyttä, voi pyytää mallia kirjoittamaan asiantuntijatasoisesti. Lisäksi erilaisten erottimien, kuten kolminkertaisten lainausmerkkien tai XML-tägien käyttäminen voi selkeyttää eri osien käsittelyä ja on hyödyllistä varsinkin pitkissä teksteissä, jotta malli osaa erottaa käyttäjän pyynnön muusta tekstistä. (OpenAI n.d.)

Vaikeammin selitettävissä prompteissa, mallille kannattaa syöttää ensin esimerkkejä ja vasta sen jälkeen pyytää sitä tekemään annettu tehtävä. Kyseinen keino tunnetaan nimellä ”few-shot” prompting. Mallille voi antaa esimerkiksi ensin jonkin tietyn vastaustyylin ja sen jälkeen malli aina vastaa kyseisellä tyylillä. Toinen keino on antaa mallille ohjeet vaiheittain, jolloin sen on helpompi seurata ja toteuttaa käyttäjän pyyntö askel askeleelta. (OpenAI n.d.)

Myös vastauksen pituuteen pystyy vaikuttamaan. Mallia voi pyytää esimerkiksi vastaamaan vain kahdella lauseella tai pyytää vastausta tietyllä sanamäärällä, mutta tässä on hyvä huomata, ettei malli kovin suurella todennäköisyydellä osaa luoda sanamääräisesti tarkkoja vastauksia vaan se toimii luotettavimmin lauseiden määrällä tai luettelossa luettelon merkintöjen määrällä. (OpenAI, n.d.)

### 3.3.2 Lähteiden käyttö

Kielimallit voivat joskus luoda uskottavan oloisia, mutta keksittyjä vastauksia erityisesti silloin, kun kysytään harvinaisista aiheista tai tarvitaan lähdeviittauksia. Tämän välttämiseksi mallille voidaan antaa lähdetekstejä, mikä auttaa sitä tuottamaan tarkempia ja todenmukaisempia vastauksia. (OpenAI n.d.)

Esimerkiksi, jos käyttäjä tarvitsee tietoa tietyistä dokumenteista, hän voi sisällyttää nämä dokumentit kyselyynsä käyttämällä erottimia. Mallille annetaan ohjeet viitata näihin dokumentteihin vastauksissaan, mikä lisää vastausten paikkansapitävyyttä ja mahdollistaa niiden jäljittämisen alkuperäisiin lähteisiin. (OpenAI n.d.)

### 3.3.3 Osittaminen

Kielimallien hyödyntämisessä keskeistä on monimutkaisten tehtävien jakaminen pienempiin osiin. Tämä periaate on tuttu ohjelmistosuunnittelusta, jossa laajoja järjestelmiä jaetaan erillisiksi komponenteiksi selkeyden ja hallinnan parantamiseksi. Tämä mahdollistaa myös aiempien tehtävien ratkaisujen hyödyntämisen perustana jatkotehtäville. (OpenAI, n.d.)

Pitkien keskustelujen ja laajojen dokumenttien käsittelyssä luokittelu, osittaminen ja tiivistäminen ovat keskeisiä menetelmiä, jotka auttavat mallia säilyttämään keskustelun kontekstin, pysymään johdonmukaisena ja vähentämään virheitä. Esimerkiksi, kun käsitellään asiakaspalvelukyselyitä, voidaan ensin luokitella kategoriat ”laskutus” ja ”tekninen tuki”. Tämän luokittelun pohjalta mallille voidaan antaa tarkempia ohjeita, viitaten tarpeen mukaan kuhunkin luokkaan, jolloin se kykenee antamaan yksityiskohtaisempaa tietoa. Pitkien dokumenttien, kuten kirjojen, tiivistämisessä voidaan edetä osio kerrallaan. Kukin osio tiivistetään ensin erikseen, ja nämä tiivistelmät yhdistetään sitten yhdeksi kokonaisuudeksi. (OpenAI, n.d.)

### 3.3.4 Aika

On myös tärkeää niin sanotusti ”antaa mallille aikaa ajatella” ja ohjeistaa sitä käyttämään enemmän aikaa ongelman ratkaisemiseen. Monimutkaisissa tehtävissä saattaa käydä niin, että malli kiirehtii väärään johtopäätökseen yrittäen arvata oikean vastauksen, jolloin yleensä tulos on virheellinen. Samalla tavalla kuin ihminen tarvitsee aikaa tehtävien ratkaisuun niin malli tarvitsee myös. (Fulford, n.d.)

Esimerkiksi tilanteessa, jossa pyydetään mallia arvioimaan opiskelijan ratkaisu johonkin matemaattiseen ongelmaan niin sen sijaan, että pyydetään mallia vain tarkistamaan opiskelijan vastauksen oikeellisuus, pyydetään sitä ensin itsenäisesti ratkaisemaan ongelma, jonka jälkeen se vertaa omaa ratkaisuaan opiskelijan antamaan vastaukseen ja vasta sitten arvioi opiskelijan vastauksen oikeellisuuden. (OpenAI, n.d.)

### 3.3.5 Ulkoiset työkalut

Kielimallien, kuten ChatGPT:n, tehokkuus voidaan maksimoida hyödyntämällä ulkoisia työkaluja. Niiden tarkoituksena on laajentaa kielimallien tiedonkäsittelyä ja mahdollistavat tarkempien vastausten tuottamisen. Ulkoiset työkalut, kuten tietojen hakujärjestelmät tai erilaiset kooditulkkit, mahdollistavat mallien pääsyn laajempiin tietoaaineistoihin ja niiden hyödyntämisen reaaliaikaisesti. Esimerkiksi monimutkaisen matemaattisen ongelman ratkaisemisessa, ChatGPT voisi hyödyntää ulkoista kooditulkkityökalua niin, että se lähettäisi ongelman kooditulkkityökalulle suoritettavaksi ja lopulta käyttäisi saatuja tuloksia vastauksensa muodostamiseen. (OpenAI, n.d.)

### 3.3.6 Testaaminen

Tehokkaan kommunikoinnin saavuttaminen ChatGPT:n kaltaisten kielimallien kanssa edellyttää promptien testausta ja arviointia. Tämä prosessi auttaa ymmärtämään, mitkä kehotteet toimivat parhaiten ja tuottavat halutunlaisia vastauksia. On hyvä muistaa, että jokin kehotteen pieni muutos saattaa parantaa vastauksia tietyissä tapauksissa, mutta ei välttämättä toimi yhtä hyvin kaikissa tilanteissa. Siksi on suositeltavaa testata kehoitteita laajemmin erilaisilla menetelmillä, jotta voidaan varmistaa niiden toimivuus kattavammin. (OpenAI n.d.)

Käytännössä voi kokeilla esimerkiksi vertailla mallin antamia vastauksia ennalta määrättyihin, oikeiksi tiedettyihin vastauksiin. Mallilta pyydetään vastausta, jonka jälkeen tarkistetaan, sisältääkö se tietyt tärkeät tiedot, kuten historialliset faktat. Arviointiprosessissa eritellään ja arvioidaan, onko mallin antama vastaus sisältänyt vaaditut tiedot, ja näin voidaan objektiivisesti mitata vastauksen oikeellisuutta. (OpenAI n.d.)

### 3.3.7 Promptauksen kehitysprosessi

Promptaus suurten kielimallien kanssa, kuten ChatGPT:n, on usein iteratiivinen eli jatkuva prosessi. Harvoin ensimmäisellä yrityksellä saavutetaan haluttu lopputulos vaan sen sijaan promptia tarkennetaan askel askeleelta. (Fulford & Ng, n.d.) Alkuvaiheessa olennaista on määritellä tavoite, eli mitä halutaan kehoitteella saavuttaa. Sen jälkeen muotoillaan prompti, jota testataan, säädetään ja testataan taas uudelleen eli toistetaan tätä prosessia, kunnes ollaan tyytyväisiä saavutettuun lopputulokseen. (Ojanperä, 2023, s. 43)

Säätämisvaiheessa, jos esimerkiksi mallin tuottama vastaus on liian epämääräinen tai epätarkka, kehotetta muokataan selkeämmäksi ja yksityiskohtaisemmaksi. Tämä voi sisältää lisäkontekstin antamista, ohjeiden selventämistä tai jopa vastauksen pituuden säätelyä.

(Fulford & Ng, n.d.)

Kehitysprosessin ydin on jatkuva oppiminen ja sopeutuminen. Käyttäjän tulee olla avoin uusille lähestymistavoille ja valmis kokeilemaan erilaisia strategioita mallin kanssa kommunikoinnissa. Tämä ei ainoastaan auta saavuttamaan tavoiteltua lopputulosta, vaan myös syventää ymmärrystä mallin mahdollisuuksista ja rajoitteista. Tämän prosessin aikana on tärkeää pitää mielessä, ettei ole olemassa yhtä ainoaa "täydellistä" promptia kaikille tilanteille, vaan tehokas promptaus rakentuu yksilöllisten tarpeiden ja tavoitteiden ympärille.

(Fulford & Ng, n.d.)

Vaikka promptauksen periaatteet on hyvä ymmärtää, niin ennusteiden mukaan promptien suunnittelusta tulee tarpeetonta tekoälymallien kehittyessä yhä tehokkaammiksi. Mallit tulevat ymmärtämään paremmin kontekstia ja kykenevät automatisoimaan sopivien kehoitteiden luomisen. (Ojanperä, 2023, s. 43–44)

### 3.4 ChatGPT opiskelun tukena

Oppiminen on tiedon ja taitojen karttumista, joka tapahtuu kokemusten kautta. Se voi tarkoittaa uusien tietojen muistamista ja hyödyntämistä tai olemassa olevien käsitysten syventämistä. Oppiminen ilmenee käyttäytymisen muutoksena, joka voi johtua uusien taitojen omaksumisesta tai näkemyksen laajentumisesta. (Opara ym., 2023, s. 34)

ChatGPT tarjoaa opiskelijoille monipuolisia mahdollisuuksia oppimisen tueksi. Se voi toimia henkilökohtaisena opinto-ohjaajana luoden erilaisia harjoitustehtäviä ja antaen reaaliaikaista palautetta. Siltä voi saada nopeasti vastauksia mieltä askarruttaviin kysymyksiin. Lisäksi se toimii tehokkaana apuna tutkimuksessa ja akateemisessa kirjoittamisessa esimerkiksi auttaen opiskelijoita luonnosteilemaan, muotoilemaan ja tiivistämään akateemisia tekstejä.

(Meyer ym., 2023, s. 6) Tämä voi auttaa hahmottamaan tekstien pääkohdat nopeasti.

ChatGPT voi tukea opiskelijoiden tutkimustaitojen kehittymistä tarjoamalla pääsyn laajaan tietomäärään ja esimerkiksi nostaa esille uusia, odottamattomia näkökulmia. (Kasneci ym., 2023) ChatGPT:n avulla voidaan myös kehittää opiskelijanäkökulman lisäksi opettajien toimintaa. Sen avulla voidaan helpottaa esimerkiksi kurssien ja oppituntien suunnittelua.

(Meyer ym., 2023, s. 7)



### 3.4.1 ChatGPT ohjelmoinnin tukena

ChatGPT lupaa helpottaa ohjelmointia monella tavalla, se voi esimerkiksi opastaa koodauksen alkeissa, auttaa uuden koodin luonnissa, selventää ja korjata jo olemassa olevaa koodia, sekä optimoida siitä tehokkaampaa. ChatGPT kykenee auttamaan monien ohjelmointikielten, kuten Pythonin, Javan ja C++:n, kanssa. Opetuksen saralla ChatGPT voi toimia interaktiivisena ohjaajana, vastaten kysymyksiin sopivasta ohjelmointikielestä, koodin syntaksista sekä tarjoamalla esimerkkikoodeja selityksineen. ChatGPT:llä voidaan myös tuottaa ohjelmointioppaita ja tehtäviä aiheeseen liittyen. (Meyer ym., 2023, ss. 7–8)

Aloittelevaa ohjelmoijaa ajatellen ChatGPT helpottaa niin sanotun tyhjän paperin syndrooman kanssa eli opiskelijan ei tarvitse tyhjästä aloittaa vaan ChatGPT voi generoida koodipohjan, jolle opiskelija voi helpommin lähteä työstämään omaa koodiaan. (Becker ym., 2023, s. 504)

ChatGPT voi auttaa myös taitotasosta riippumatta uuden koodin kirjoittamisessa mutta sen hyödyllisyys riippuu kuitenkin käyttäjän antamasta kehoitteesta ja koodin monimutkaisuudesta. ChatGPT onnistuu todennäköisemmin kirjoittamaan tarkasti ja virheettömästi pienempiä koodilohkoja kun taas suurempien kokonaisuuksien generointi ei välttämättä onnistu niin helposti. (Meyer ym., 2023, ss. 7–8)

ChatGPT voi myös auttaa parantamaan olemassa olevaa koodia, esimerkiksi selittämällä koodin toimintaa ja lisäämällä ja korjaamalla koodin kommentteja. Se auttaa myös yksinkertaistamaan tai vaikkapa kääntämään koodia eri ohjelmointikielille. Lisäksi se on hyödyllinen apu virheiden korjauksessa. Varsinkin virheilmoitukset ovat tunnetusti esteenä opiskelijoiden edistymiselle, joten oppiminen tehostuu, kun näitä esteitä pystytään taklaamaan tekoälyn avulla (Becker ym., 2023, s. 504). On kuitenkin tärkeää muistaa, että ChatGPT saattaa esittää virheellistä koodia oikeana ja se ei välttämättä pysty ennakoimaan erikoistapauksia, jotka saattavat rikkoa koodin toiminnallisuuden. ChatGPT:n kirjoittama koodi voi tarjota hyödyllisen lähtökohdan, mutta ei ole suositeltavaa luottaa suoraan ChatGPT:n tuottamaan koodiin varsinkaan tärkeiden sovellusten osalta. (Meyer ym., 2023, s. 8)

### 3.4.2 Esimerkkinä oman opiskelukurssin suunnittelu ja luominen

Malekosin (2023) mukaan oman kurssin suunnittelu ChatGPT:n avulla on hyvä aloittaa ideoinnista, jossa tekoäly auttaa tunnistamaan keskeiset teemat. Tämä vaihe voi sisältää

esimerkiksi listan käsiteltävien aiheiden otsikoista. Seuraavaksi tulisi määritellä kohderyhmä tarkasti, jotta kurssin sisältö vastaa kohderyhmän tarpeita ja edistää oppimista. Tässä vaiheessa voidaan hyödyntää ChatGPT:tä kohderyhmän tarpeiden ymmärtämiseen ja kurssin sisällön räätälöimiseen.

ChatGPT:tä voidaan käyttää hyödyksi kurssin rungon suunnitteluun. Tekoälyä ohjataan tarkoilla kehoitteilla, jotka määrittelevät kurssin tavoitteet. Esimerkiksi, jos tarkoituksena olisi luoda verkkokurssi niin ChatGPT:tä voitaisiin pyytää kehittämään verkkokurssi tietylle kohderyhmälle, antaen myös tarkentavia tietoja opiskeltavasta aiheesta. Sen jälkeen seuraa sisällön tuotanto, jossa ChatGPT voi auttaa luomaan monipuolisia oppimistehtäviä ja aktiviteetteja. Tässä vaiheessa voidaan käyttää kehoitteita, jotka ohjaavat ChatGPT:tä suunnittelemaan oppimisaktiviteetteja, jotka tukevat kurssin tavoitteita ja teemoja. Esimerkiksi, ChatGPT:ltä voidaan pyytää ehdotuksia erilaisten oppimismateriaalien, kuten tietovisojen, tenttien ja käytännön esimerkkien, luomiseksi. Esimerkiksi ChatGPT:tä voisi pyytää generoimaan monivalintakysymykset opetellun aiheen X perusteella. Kurssin luomisen yhteydessä tulisi muistaa, että tekoälyllä luotu sisältö voi sisältää virheellistä tietoa, joten ihmisen tekemä tarkistus on välttämätöntä (Malekos, 2023)

Oman kurssin voi luoda myös mukautetun GPT:n avulla, kuten Davisin (2023) esimerkki "DIVAA", Digital Interactive Virtual AP Assistant, osoittaa. Kyseinen GPT auttaa opiskelijoita AP Computer Science Principles -kurssilla ja toimii opetuksen apuvälineenä, tarjoten räätälöityä tukea ja ohjausta. Mukautetun GPT:n voi luoda vain Plus-jäsenet ChatGPT:n keskusteluikkunasta valitsemalla "Create a GPT". Tässä prosessissa määritellään vuorovaikutuksessa GPT:n kanssa sen ulkoasu, toimintatapa ja vuorovaikutustyyli. Lisäksi on mahdollista ladata GPT:lle erityistä tietoa, kuten opetussuunnitelman standardeja tai sanastoa tiedostomuodossa. Kun GPT on valmis, se testataan varmistaakseen, että se toimii odotetulla tavalla ja vastaa asetettuihin tavoitteisiin.

### **3.4.3 ChatGPT opettajan korvikkeena**

Opettamisen ajatellaan usein tarkoittavan henkilön tietojen tai taitojen välittämistä toiselle henkilölle. Se voi tarkoittaa kokemusten jakamista tai tiedon siirtämistä, kuten luennoilla tapahtuva oppiminen ja opettaminen. Opettamisen voisi ajatella olevan sekä taidetta että tiedettä. (Opara ym., 2023, s. 33) Opettamiseen tarvitaan luonnollisesti oppimateriaaleja, joiden tuottaminen vie huomattavasti aikaa ja vaivaa sekä vaatii erityistä asiantuntemusta. Tekoälyn avulla aukeaa uudenlaisia mahdollisuuksia opetustyöhön, kuten aiemmissa

kappaleissa käytiin läpi esimerkiksi opetusmateriaalien tuottamisen osalta. (Becker ym., 2023, s. 503) Kielimalleja pystytään jo käyttämään hyväksi opetuksen tietyillä osa-alueilla opettajien ajan säästämiseksi. (Kasneci ym., 2023)

Vaikka ChatGPT:tä voidaan käyttää apuna opetuksessa, sen käyttö edellyttää Ausatin ym., (2023, s. 16103) mukaan suunnitelmallisuutta. Koska oppiminen tapahtuu vuorovaikutuksessa ja sosiaalinen kanssakäyminen on siinä avain asemassa niin kestää aikansa ennen kuin koneet yltävät lähellekään samaa tasoa (Niemi, H., 2021, s. 143). Opettajien asiantuntemus ja henkilökohtainen läsnäolo ovat siis Ausatin ym. (2023, ss. 16103–16104) mukaan edelleen korvaamattomia, ja teknologian käyttö opetuksessa tulisi nähdä mahdollisuutena ja työkaluna parantaa oppimista ja tehostaa opetusta, mutta se ei voi täysin korvata opettajan roolia.

#### **3.4.4 Eettiset kysymykset opiskelussa**

Akateemisessa maailmassa on noussut esille huoli tekoälyohjelmien, kuten ChatGPT:n, käytöstä tieteellisissä julkaisuissa. ChatGPT:n kyky tuottaa uskottavalta kuulostavaa tekstiä tuo mukanaan haasteita, sillä se voi lisätä viittauksia tutkimuksiin, joita ei todellisuudessa ole olemassa. Tämä johtaa siihen, että tieteellisiin julkaisuihin saattaa eksyä ChatGPT:n tuottamaa tekstiä, jota ei tunnisteta valheelliseksi ja näin ollen voi vaikuttaa kirjallisuuden luotettavuuteen. (Thorp, 2023, s. 313) Lisäksi on haastavaa määritellä, missä menee raja alkuperäisen työn ja plagioinnin välillä, kun ChatGPT tai muut tekoälymallit tuottavat tekstiä. Tämä on erityisen vaikeaa tilanteissa, joissa kirjoittaja on ohjannut tekoälyä tekstintuotannossa. Tieteelliset julkaisut, kuten Science ja Nature, ovat vastanneet tähän haasteeseen asettamalla ohjeita ja rajoituksia tekoälyn tuottaman tekstin käytölle, muun muassa sen, ettei ChatGPT:n tai muiden generatiivisten tekoälytyökalujen avulla luotuja materiaaleja saa käyttää julkaisuissa tai toimia artikkelien kirjoittajana. (Meyer ym., 2023, s. 3)

Myös opiskelijat saattavat nojautua liikaa ChatGPT:hen, mikä saattaa heikentää heidän kykyään kehittää kriittisiä ajattelun taitoja, kuten itsenäistä ajattelua ja kirjoittamistaitoa. Huolta herättää myös osaltaan opiskelijoiden epärehellinen tekoälyn käyttö, esimerkiksi lunttaamiseen kotitehtävissä ja kokeissa tai kirjoittamaan raportteja ja esseitä ilman asianmukaista lähdeviittausta, mikä taas voi johtaa plagiointiin. (Meyer ym., 2023, ss. 6)

Tämän vuoksi on tärkeää, että sekä opettajat että opiskelijat harjoittavat kriittistä ajattelua ja tarkistavat saamansa tiedot luotettavista lähteistä. Opiskelijoiden on ymmärrettävä, miten tekoäly toimii ja tunnistettava sen rajoitukset, koska tekoälyn käyttö yleistyy ja kehittyy jatkuvasti. Opettajilla on tärkeä rooli ohjata ja opettaa opiskelijoille tekoälyn käyttöä niin, että sitä käytetään oikeaoppisesti ja eettisesti. Kriittisen ajattelun kehittäminen ja tekoälyn eettisen ja vastuullisen käytön edistäminen ovat avainasemassa opiskelijoiden varustamisessa tarvittavilla tiedoilla ja taidoilla nopeasti muuttuvassa teknologisessä maailmassa. (Mhlanga, 2023)

## 4 React

React on Facebookin ja erityisesti insinööri Jordan Walken vuonna 2011 kehittämä avoimen lähdekoodin JavaScript-kirjasto. Se on suunniteltu käyttöliittymien ja UI-komponenttien rakentamiseen. Sitä ylläpitää yhteisö, johon kuuluu Facebookin lisäksi myös yrityksiä ja yksittäisiä kehittäjiä. Reactin avulla voidaan kehittää esimerkiksi yksisivuisia verkkosivuja tai mobiilisovelluksia. Sen toiminta keskittyy pääasiassa tilanhallintaan ja tämän tilan esittämiseen DOM:ssa eli käyttöliittymän rakenteessa, joka kuvaa web-sivun elementtejä. Sen käyttö ei aina yksinään riitä monimutkaisempien sovellusten kehittämiseen ja sen vuoksi React-sovelluksissa tarvitaan usein lisäkirjastoja esimerkiksi reititykseen ja muihin asiakaspään toiminnallisuuksiin. (Indla & Puranik, 2021, ss. 1137–1138)

### 4.1 Keskeiset ominaisuudet

Reactin vahvuus piilee sen komponenttipohjaisessa rakenteessa. Tämä rakenne mahdollistaa itsenäisten komponenttien luomisen, jotka hallitsevat omaa tilaansa. Nämä komponentit voidaan koota yhteen ja rakentaa niiden avulla monimutkaisia käyttöliittymiä. (Indla & Puranik, 2021, s. 1138)

Yksi Reactin merkittävimmistä ominaisuuksista on virtuaalinen DOM (Virtual DOM), joka tehostaa käyttöliittymän päivityksiä muutosten yhteydessä ilman, että koko näkymää täytyy päivittää joka kerta. Eli, kun komponentin tilassa tapahtuu muutoksia esimerkiksi käyttäjän painaessa nappia, niin virtuaalinen DOM vertailee ja päivittää ainoastaan ne osat oikeasta, hitaammin toimivasta DOM:ista, jotka todella muuttuvat. Tämän ansiosta varsinkin monimutkaisimmissa sovelluksissa tarvitaan vähemmän laskentatehoa, mikä taas johtaa suorituskyvyn paranemiseen. (Indla & Puranik, 2021, s. 1138)

#### 4.1.1 JSX

Reactin tehokkuutta lisää sen kyky käyttää JSX:ää (JavaScript eXtension), joka on JavaScriptin syntaksilaajennus. JSX mahdollistaa HTML-kaltaisen merkinnän kirjoittamisen suoraan JavaScript-koodiin, minkä ansiosta sillä voidaan muokata DOM:ia helposti. (Indla & Puranik, 2021, s. 1138). Vaikka JSX:n käyttö ei ole pakollista, useimmat kehittäjät hyödyntävät sitä sen helppokäyttöisyyden vuoksi. JSX on samantapaista kuin HTML, mutta se on myös hieman tiukempaa syntaksiltaan eli esimerkiksi kaikki tagit, kuten `<br />`, tulee sulkea asianmukaisesti ja mikäli komponentti palauttaa useita JSX-elementtejä, täytyy ne

kääriä yhteiseen vanhempielementtiin, kuten `<div>` tai tyhjäan fragmenttiin `<>...</>`. (React, n.d.-f)

JSX mahdollistaa myös tietojen näyttämisen suoraan JavaScriptista käyttämällä aaltosulkeita, mikä tekee koodista dynaamisempaa. Esimerkiksi jos on objekti nimeltä "person", jonka "name"-ominaisuus sisältää nimen niin sen voisi kirjoittaa JSX:nä `{person.name}`, mikä näyttäisi käyttäjän nimen suoraan käyttöliittymässä. (React, n.d.-a)

#### 4.1.2 Komponentit

Reactissa sovellusten kehittäminen perustuu siis näiden uudelleenkäytettävien komponenttien luomiseen ja hallintaan. Komponenttien voidaan ajatella olevan kuten rakennuspalikoita esimerkiksi legoja, joita yhdistelemällä muodostetaan lopulta koko sovelluksen käyttöliittymä (Herbert, 2023). Komponenttien rakentaminen alkaa ihannetapauksessa pienempien osien luomisesta kohti laajempaa ja monimutkaista hierarkiaa. Ensin luodaan tarvittavia ja uudelleenkäytettäviä pieniä komponentteja kuten nappuloita ja pudotusvalikoita. Jokaisella komponentilla on oma logiikkansa, jotta uudelleenkäyttö onnistuu ja niitä yhdistelemällä suuremmaksi kokonaisuudeksi saadaan lopputulokseksi yksi juurikomponentti, jolla on useita alakomponentteja. (Indla & Puranik, 2021, s. 1138)

#### 4.1.3 Tilanhallinta ja hookit

Reactissa komponentit tarvitsevat usein kyvyn muistaa käyttäjän syöttämiä tietoja sekä muuttaa UI-näkymää jonkin tapahtuman yhteydessä, kuten käyttäjän kirjoittaessa lomaketta tai painaessa nappulaa. Tämä komponenttikohtainen "muisti" tunnetaan Reactissa tilana "state". Tilan avulla komponentit voivat säilyttää ja hallita tietoja, jotka voivat muuttua ajan kuluessa. React uudelleenrenderöi komponentin, kun sen tila muuttuu. `setState`-funktio on keskeinen työkalu tilan muuttamiseen, sillä sen avulla voidaan muuttujat päivittää, minkä takia tapahtuu renderöinti. Tämä mahdollistaa tilan muutosten hallinnan ja käyttöliittymän päivittämisen reaaliaikaisesti. (React, n.d.-d)

Reactin hookit tulivat uutena ominaisuutena versiossa 16.8. tarjoten tavan hallita tilaa ja muita ominaisuuksia funktionaalisissa komponenteissa ilman monimutkaisia luokkakomponentteja (Ravikiran, 2022). Hookit alkavat use-etuliitteellä, kuten `useState`, `useContext` tai `useEffect`. Huomioitavaa niiden käytössä on, että niitä tulee kutsua

ainoastaan komponenttien tai toisten hookien alussa eikä niitä saa sijoittaa silmukoihin tai ehtolauseisiin. (React, n.d.-e) Välillä tulee eteen tilanteita, kun tila pitää jakaa komponenttien kesken eli jos halutaan esimerkiksi vaihtaa kahden komponentin tila samanaikaisesti. Siihen on olemassa tekniikka, jota kutsutaan tilan nostamiseksi ”liftin state up”, jossa tila siirretään yksittäisistä komponenteista ylempään komponenttiin, joka hallitsee tätä tilaa ja välittää sen alikomponenteilleen propsien kautta. (React, n.d.-c)

#### 4.1.4 Propsit

Propsit (properties eli ominaisuudet) ovat Reactissa keskeisiä tietojen siirtämisen välineitä vanhempikomponentista lapsikomponentille. Ne auttavat React-komponentteja vastaanottamaan tietoja ylemmiltä tasoilta ja muistuttavat toiminnaltaan hieman HTML:n attribuutteja, mutta koska niiden avulla voidaan välittää kaikenlaisia JavaScript-arvoja, kuten objekteja, taulukoita ja funktioita. Näiden tekijöiden ansiosta komponenttien joustavuus paranee ja mahdollistetaan erilaisten arvojen uudelleenkäyttö. (React, n.d.-b)

Vaikka propsit voivat vastaanottaa erilaisia arvoja, ne ovat itsessään muuttumattomia. Tämä tarkoittaa sitä, ettei komponentti voi itse muuttaa saamiaan props-arvoja vaan tarvitsee tilanhallintaa kyseiseen toimintaan. Kun komponentissa pitää päivittää propseja, esimerkiksi käyttäjän toiminnan seurauksena, sen on pyydettävä nämä uudet tiedot yläkomponentilta, jossa sen tilaa muutetaan. Tämä auttaa pitämään React-sovellukset järjestyksessä ja ennustettavina. Muuttumattomat propsit auttavat varmistamaan, että komponenttien käyttäytyminen pysyy yhdenmukaisena. (React, n.d.-b)

## 4.2 Reactin vaikutukset web-kehitykseen

Reactin vaikutus web-kehitykseen on ollut merkittävä. Ennen Reactia web-sovellukset koostuivat lähinnä staattisista HTML-sivuista, joiden dynaamisuutta lisättiin PHP:lla. Reactin myötä kehittäjät saivat käyttöönsä tehokkaamman tavan rakentaa interaktiivisia yksisivuisia sovelluksia (SPA eli single-page application), jotka tarjoavat käyttäjille nopeampia ja sujuvampia vuorovaikutuskokemuksia. (Indla & Puranik, 2021, s. 1138)

Reactin komponenttipohjainen lähestymistapa, yhdistettynä tehokkaaseen tilanhallintaan ja moderniin käyttöliittymäarkkitehtuuriin, on tarjonnut kehittäjille joustavampia ja tehokkaampia työkaluja. Lisäksi sen oppimiskynnys on matala, erityisesti JavaScript-kokemusta omaaville, mutta se on hyvä valinta myös aloittelijoille. Aktiivisen yhteisön vuoksi erilaisia valmiita

komponentteja on helposti saatavilla omiin projekteihin, eli kaikkea ei tarvitse suunnitella itse. Reactin suosio jatkaa kasvuaan ja sen soveltuvuus erilaisiin web-kehitysprojekteihin on jo vakiintunut ja sitä voidaan käyttää esimerkiksi blogeissa, yritysten verkkosivujen kehitykseen, verkko-opetussivuihin, henkilökohtaisiin portfolioihin tai erilaisiin foorumeihin. Lisäksi sen arkkitehtuuri on helposti hyödynnettävissä myös mobiilisovelluksille React Nativen avulla. (Indla & Puranik, 2021, s. 1138–1139)



## 5 Reactin oppiminen ChatGPT:n avulla – tavoitteet ja tutkimusmenetelmät

Tässä osiossa esitellään tutkimuksen tavoite sekä tutkimus-, aineistonhankinta- ja analyysimenetelmät, joiden avulla tarkastellaan ChatGPT:n vaikutusta Reactin oppimisprosessissa. Tutkimusmenetelmänä käytetään kvalitatiivisesta tutkimuksesta tuttua päiväkirjaa. Konkreettisenä tavoitteena ja oppimisen tuloksena on tarkoitus syntyä yksinkertainen Reactilla toteutettu yksisivuinen verkkosivu, jonne kootaan ChatGPT:n tehokkaan hyödyntämisen-, kommunikoinnin- sekä oman GPT:n luomiseen liittyvät ohjeet esimerkkeineen.

### 5.1 Tutkimuksen tavoite

Tutkimuksen tavoitteena on tutkimuskysymyksiin vastaten syvällisesti ymmärtää ja analysoida ChatGPT:n vaikutuksia React-ohjelmoinnin oppimisprosessiin. Prosessin vaiheita dokumentoidaan päiväkirjamerkintöjen avulla. Päiväkirjassa seurataan Reactin opiskelua ChatGPT:n avulla päivittäin 30 päivän ajan, kaksi tuntia päivässä. Tavoitteena on selvittää ja tunnistaa päiväkirjamerkintöjen avulla keskeisiä teemoja, joita opiskeluprosessin aikana nousee. Erityistä huomiota kiinnitetään ChatGPT:n käytön haasteisiin, mahdollisuuksiin ja sen rooliin opiskelumateriaalin ja -menetelmien kehittäjänä.

Tavoitteena on tuottaa analyysien ja kokemusten perusteella yhteenvedona konkreettisia ohjeita ja esimerkkejä, miten ChatGPT:tä voidaan hyödyntää mahdollisimman tehokkaasti opiskelun tukena. Näistä koostetaan aineisto verkkosivulle, joka luodaan ja dokumentoidaan projektin edetessä.

### 5.2 Kvalitatiivinen tutkimus

Kvalitatiivinen tutkimusmenetelmä keskittyy syvälliseen ymmärrykseen ja ihmisten kokemusten tulkintaan. Sen avulla voidaan tutkia ja ymmärtää ilmiöitä, jotka eivät ole kvantitatiivisesti mitattavissa, kuten henkilökohtaisia kokemuksia ja näkemyksiä. Tutkimuksessa käytetään empiirisiä aineistoja, kuten päiväkirjoja, haastatteluja, keskusteluja, tekstejä tai kuvia ja tiloja. Laadullinen tutkimus vastaakin usein kysymyksiin, miten ja mitä, kun taas kvantitatiivinen tutkimus vastaa kysymykseen miksi. (Juhila, n.d-a.) Tässä työssä kvalitatiivinen lähestymistapa on valittu, koska se mahdollistaa henkilökohtaisten

kokemusten, tunteiden ja näkemysten syvällisen tarkastelun ChatGPT:n käytöstä Reactin opiskelussa.

### 5.2.1 Aineistonhankintamenetelmä

Päiväkirjamenetelmä aineistonhankintamenetelmänä sisältää autoetnografisia piirteitä, joissa tutkija hyödyntää henkilökohtaisia kokemuksiaan tiedonlähteenä. Tämä menetelmä keskittyy syvälliseen itseanalyysiin ja henkilökohtaiseen näkökulmaan, yhdistäen yksilölliset havainnot ja kokemukset teoreettiseen tietoon. Näin ollen menetelmä edellyttää tutkijalta laajaa teoreettista ymmärrystä ja kykyä vertailla omia kokemuksiaan aikaisempaan tutkimustietoon, syventäen ymmärrystä tutkittavasta aiheesta. (Kukkurainen, 2019)

Päiväkirjamerkinnot mahdollistavat reaaliaikaisen seurannan ja syvällisen ymmärryksen oppimisprosessista. Ne antavat ainutlaatuisen mahdollisuuden tallentaa kokemuksia, ajatuksia ja oivalluksia heti niiden ilmetessä, mikä on arvokasta oppimisen analysoinnissa. (UX 24/7, 2019) Päiväkirja voi olla menetelmänä strukturoitua tai avointa. Strukturoidussa päiväkirjassa käytetään ennalta määrättyä rakennetta tai kysymyksiä, jotka ohjaavat päiväkirjan kirjoittamista. Tämä auttaa fokusoimaan tiettyihin teemoihin tai aiheisiin ja tarjoaa systemaattisen lähestymistavan aineistonkeruuseen. Toisaalta avoimessa päiväkirjassa ei ole ennalta määriteltyä rakennetta, mikä antaa kirjoittajalle suuremman vapauden ilmaista ajatuksia ja kokemuksia. Tämä mahdollistaa monipuolisemman ja spontaanin aineiston tuottamisen, mutta voi samalla olla haastavampi analysoida sekä riippuen tutkittavista se voi myös antaa suppeammat tulokset. Molemmissa lähestymistavoissa päiväkirjamerkinnot antavat syvällistä ymmärrystä ja tietoa tutkittavasta aiheesta, mutta niiden käyttö riippuu tutkimuksen tavoitteista ja kontekstista. Päiväkirjamenetelmällä voidaan lisäksi vähentää muistamattomuudesta johtuvia ongelmia, esimerkiksi haastatteluihin verrattuna. (Palojoki, 1997, ss. 129–134)

Tässä opinnäytetyössä hyödynnetään puolistrukturoitua päiväkirjapohjaa, joka on suunniteltu keräämään yksityiskohtaista tietoa oppimiskokemuksista. Kyseinen menetelmä tarjoaa kattavan tavan seurata ja analysoida oppimisprosessia, mahdollistaen havaintojen tekemisen siitä, miten oppija on vuorovaikutuksessa ChatGPT:n kanssa ja tunnistaa oppimisen haasteet sekä saavutukset. Puolistrukturoitu lähestymistapa yhdistää tarkasti kohdennetut kysymykset, jotka keskittyvät päivän tavoitteeseen, opiskeluun käytetyn ajan, ChatGPT:n hyötyihin ja haittoihin niin ajallisesti kuin laadullisesti sekä osion vapaamuotoisista muista ajatuksista. Kuva 3 esittelee päiväkirjan pohjan.

## Kuva 3 Päiväkirjan pohja

**Päivämäärä:** [pp/kk/vvvv]

**Opiskeluaika (Kesto):** [Esimerkiksi 2 tuntia]

### 1. Päivän tavoitteet

- Lyhyt kuvaus päivän React-opiskelun tavoitteista.

### 2. Käytetyt materiaalit

- Mainitse tärkeimmät käytetyt materiaalit (ChatGPT, kurssimateriaalit, muut resurssit).

### 3. Opiskelun haasteet ja saavutukset:

- Kuvaile kohdatut haasteet ja kuinka ChatGPT auttoi näissä haasteissa.
- Kuvaile saavutuksia ja kuinka ChatGPT auttoi saavuttamaan nämä.

### 4. Ajankäytön arviointi

- Arvioi, kuinka tehokkaasti käytit opiskeluaikaasi ja ChatGPT:n roolia siinä.

### 5. Laadun arviointi

- Kuinka arvioisit ChatGPT:n antamien vastausten tarkkuutta ja niiden vaikutusta oppimistuloksiisi?

### 6. Muita ajatuksia

- [Kirjoita vapaamuotoisesti päivän yleiset ajatukset ja tunteet]

Strukturoitu osio varmistaa, että päiväkirjan kirjoittaminen pysyy johdonmukaisena, ja että jokaisessa merkinnässä käsitellään tietyt keskeiset aiheet tai kysymykset. Toisaalta avoin osio tarjoaa mahdollisuuden syvempään itsetutkiskeluun sekä mahdollisuuden nostaa esille ennalta arvaamattomia aiheita.

### 5.2.2 Analyysimenetelmä

Analyysin tarkoituksena on tuoda tutkimukselle lisää informaatioarvoa. Laadullisessa analyysissä on kyse aineiston muokkaamisesta teoreettiseen muotoon. Tämä prosessi ei noudata yleispätevää kaavaa, vaan tavoitteena on tuottaa aineistosta syvällisempiä ja merkityksellisempiä tulkintoja. Avainasemassa ovat teorian ja oman ajattelun yhdistäminen. (Günther ym., n.d.)

Aineiston tarkastelua ja tulkintaa varten tarvitaan analyysimenetelmiä. Laadullisessa tutkimuksessa on monia erilaisia menetelmiä aineiston analysointiin. Tutkijan on valittava

menetelmä, joka sopii parhaiten aineistonsa ja tutkimuskysymystensä käsittelyyn. Analyysimenetelmä viittaa siihen, miten tutkija käsittelee ja analysoi aineistoa. Erilaisia menetelmiä ovat esimerkiksi teemoittelu ja tyypittely. (Günther ym., n.d.)

Juhilan (n.d.-b) mukaan teemoittelussa tunnistetaan aineistosta tutkimusongelman kannalta keskeiset teemat. Teemoittelun tavoitteena on tuoda esiin aineiston sekä keskeisiä aiheita, että myös sellaisia aiheita, jotka eivät välttämättä ole suoraan ilmeisiä. Tärkeää on, että teemat syntyvät aineiston pohjalta eivätkä tutkijan ennakko-oletuksista. Tämä edellyttää aineiston syvällistä ymmärtämistä ja sen tarkastelua useista näkökulmista.

Tässä työssä hyödynnetään teemoittelua analyysimenetelmänä. Tämä menetelmä auttaa tunnistamaan päiväkirjamerkinnoista keskeiset teemat, jotka liittyvät oppimisprosessiin ja käyttökokemuksiin ChatGPT:n kanssa. Teemoittelun avulla voidaan erottaa ja analysoida näitä kokemuksia syvällisesti, yhdistäen ne teoreettiseen tietoon ja muodostaen tulkintoja oppimisprosessista. Tämä auttaa ymmärtämään, miten ChatGPT on vaikuttanut Reactin oppimiseen.

### **5.3 Validius ja reliabiliteetti**

Validius ja reliabiliteetti ovat keskeisiä laadullisen tutkimuksen luotettavuuden arvioinnin kannalta. Reliabiliteetti tarkoittaa tutkimuksen toistettavuutta ja huolellisuutta, mikä varmistaa, etteivät tulokset ole sattumanvaraisia. Tämä on erityisen tärkeää aineiston keruussa ja analyysissa. Validius puolestaan viittaa tutkimustulosten luotettavuuteen eli mitataanko tutkimuksessa todella sitä, mitä on tarkoituskin mitata. Validius voidaan jakaa kahteen osaan, joista sisäinen validius keskittyy tutkimusprosessin luotettavuuteen, kun taas ulkoinen validius tarkastelee tutkimustulosten soveltuvuutta laajempaan kontekstiin. (Mykkänen, n.d.)

Päiväkirjamerkintöjen käytöllä Reactin opiskelussa ChatGPT:n avulla pyritään varmistamaan tutkimuksen reliabiliteetti ja validius. Reliabiliteetti tarkoittaa tässä yhteydessä, että kerätyt tiedot ovat johdonmukaisia ja tarkasti dokumentoituja, mikä mahdollistaa oppimisprosessin yksityiskohtaisten vaiheiden, ajatusten, haasteiden ja oivallusten tallentamisen. Validiteetin varmistaminen tässä tutkimuksessa keskittyy siihen, että tulokset ovat merkityksellisiä Reactin opiskelua koskien. Se tarkoittaa, että tutkimuksessa mitataan ja analysoidaan nimenomaan niitä ilmiöitä, joita on tarkoituskin tutkia. Toisin sanoen, varmistetaan, että tutkimuksen tulokset ovat sovellettavissa ja relevantteja Reactin opiskelun kannalta.

## 6 Opiskeluprosessin suunnittelu, toteutus ja dokumentointi

Tässä luvussa keskitytään dokumentoimaan Reactin opiskelua ChatGPT:n tuella. Tarkastellaan konkreettisten esimerkkien avulla, miten ChatGPT:tä voidaan hyödyntää oppimisprosessissa. Prosessin pohjana hyödynnetään aiemmin luotua kattavaa tietoperustaa, johon yhdistetään henkilökohtaisia kokemuksia, jotka liittyvät vuorovaikutukseen ChatGPT:n kanssa. Kokemukset sisältävät niin oman GPT:n rakentamisen, jonka avulla Reactia opetellaan kurssityylisesti kuin erilaisten kommunikointitapojen testauksen. Lisäksi käytetään tarvittaessa opiskelun tukena kolmansien osapuolten ilmaisia verkkokursseja, kuten Helsingin yliopiston ja Houston Inc. järjestämää Fullstack-kurssia osoitteessa: <https://fullstackopen.com/> sekä Reactin omaa dokumentaatiota osoitteessa <https://react.dev/>.

Opiskeluprosessin alussa luodaan oma GPT. Se on mahdollista luoda tällä hetkellä ainoastaan ChatGPT:n maksullista versiota (GPT-4) käyttäen. Lisäksi hyödynnetään Prompt Perfect -pluginia. Tarkoituksena on luoda GPT:n avulla aikataulutettu kurssipohja ja saada opiskelumateriaalit kullekin päivälle aikataulun mukaisesti. Aikataulu ei ole täysin sitova vaan voi muuttua opiskeluprosessin edetessä. Prosessissa käytetään ja testataan myös ChatGPT:n keskusteluikkunoita. Tarkoituksena on testata miten ChatGPT:n kanssa kommunikoidessa saadaan siitä mahdollisimman tehokas opiskelun työväline ja lisäksi sivutaan sitä, mihin kaikkeen ChatGPT opettajan roolissa kykenee. Voisiko se esimerkiksi syrjäyttää oikeat opettajat tai verkkokurssit. Tarkastellaan esimerkiksi, miten ChatGPT kykenee luomaan erilaisia oppimismateriaaleja, kuten harjoitustehtäviä ja tehtäväkokonaisuuksia. Pyydetään ChatGPT:tä selittämään vaikeiksi koettuja asioita ja koodeja, joita matkan varrella ilmenee. Lisäksi tehdään pienimuotoinen vertailu ChatGPT:n tuottaman kurssimateriaalin ja verkosta löytyvien materiaalien välillä. Kolmansien osapuolten materiaaleja ei opinnäytetyössä julkaista vaan vertailu tapahtuu kirjoittajan kokemuksiin ja havaintoihin perustuen analyysin avulla. Oppimisprosessin tuloksena syntyy konkreettinen verkkosivu, jonne kootaan ChatGPT:n tehokkaan hyödyntämisen ja kommunikoinnin salat osana opiskeluprosessia.

Opiskeluprosessin kulku on jaettu kahteen päävaiheeseen. Ensimmäisessä vaiheessa keskitytään Reactin opiskeluun ja sen perusteiden hallintaan. Tämä vaihe on olennainen, koska se antaa tarvittavan teknisen osaamisen, jota tarvitaan myöhemmin verkkosivun luontiin. Toisessa vaiheessa ChatGPT toimii apuvälineenä ja tietolähteenä, kun aletaan luoda konkreettista sivua.

## 6.1 GPT:n luominen

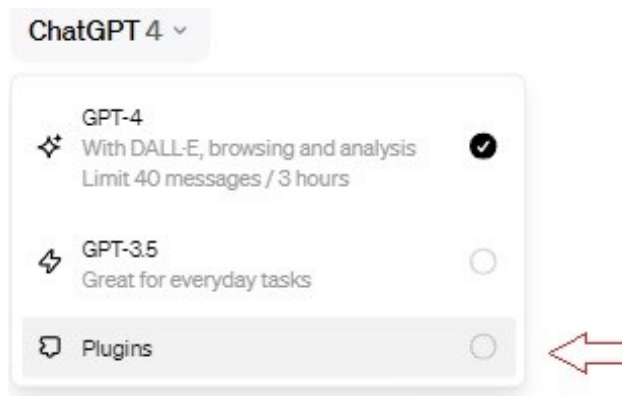
Oppimisprosessi saa alkunsa oman GPT:n luomisesta. Ensin kirjaututaan osoitteeseen <https://chat.openai.com/>. Tunnusten puuttuessa luodaan käyttäjätunnus sekä aktivoidaan Plus-jäsenyys. Kirjautumisen jälkeen avautuu uusi keskusteluikkuna ja vasempaan yläreunaan ilmestyy Explore-valikko, jota klikkaamalla avautuu omat GPT:t. Klikkaamalla "Create a GPT" pääsee luomaan oman, personoidun GPT:n, kuten kuva 4 esittää.

Kuva 4 Luo oma GPT



Avataan toinen ChatGPT:n ikkuna ja haetaan plugineista Prompt Perfect sekä asetetaan se päälle, kuten kuva 5 esittää.

Kuva 5 Pluginien asetus



Tämän jälkeen aloitetaan kirjoitus Chat-ikkunaan sanalla "Perfect", jotta Prompt Perfect aktivoituu. Tarkoituksena on saada ulos sellainen kehote eli promptti, jonka avulla kurssityylinen GPT:n luonti onnistuu. Lausetta jatketaan esimerkiksi "Olemme luomassa kurssia, jonka tarkoituksena on ensin oppia Reactin perusteet ja sen jälkeen luoda nettisivu,

jolle kootaan oppimisen tulokset. Kurssin tulisi kestää yhteensä 60 tuntia, jaettuna kahteen pääteemaan eli tekniseen opiskeluun ja nettisivun rakentamiseen.”

Kuva 6 GPT:n ominaisuuskehotteen luominen

**You**  
 Perfect Olemme luomassa kurssia, jonka tarkoituksena on ensin oppia Reactin perusteet ja sen jälkeen luoda nettisivu, jolle kootaan oppimisen tulokset. Kurssin tulisi kestää yhteensä 60 tuntia, jaettuna kahteen pääteemaan eli tekniseen opiskeluun ja nettisivun rakentamiseen.

**ChatGPT**

Used Prompt Perfect ^

REQUEST TO PROMPT PERFECT

```
{
  "text": "We are creating a course aimed at first learning the basics of React and then building a website to compile the learning outcomes. The course should last a total of 60 hours, divided into two main themes: technical study and website construction."
}
```

RESPONSE FROM PROMPT PERFECT

```
{
  "rephrased": {
    "text": "We are developing a course that has two main objectives: firstly, to provide a comprehensive introduction to the fundamentals of React, and secondly, to guide students in building a website that demonstrates their understanding of the concepts learned. The entire course is expected to span 60 hours, with a clear division between two primary themes: technical study and website construction. Can you please provide a detailed overview of the content and structure you would recommend for each theme, ensuring that all key aspects are covered within the allotted timeframe?"
  }
}
```

It sounds like an exciting and comprehensive course you're planning! To structure a 60-hour course divided between learning React basics and building a website, here's a suggested breakdown:

### Technical Study of React (30 Hours)

#### 1. Introduction to React (4 Hours)

- Overview of React and its ecosystem.
- Setting up the development environment.
- Understanding JSX.

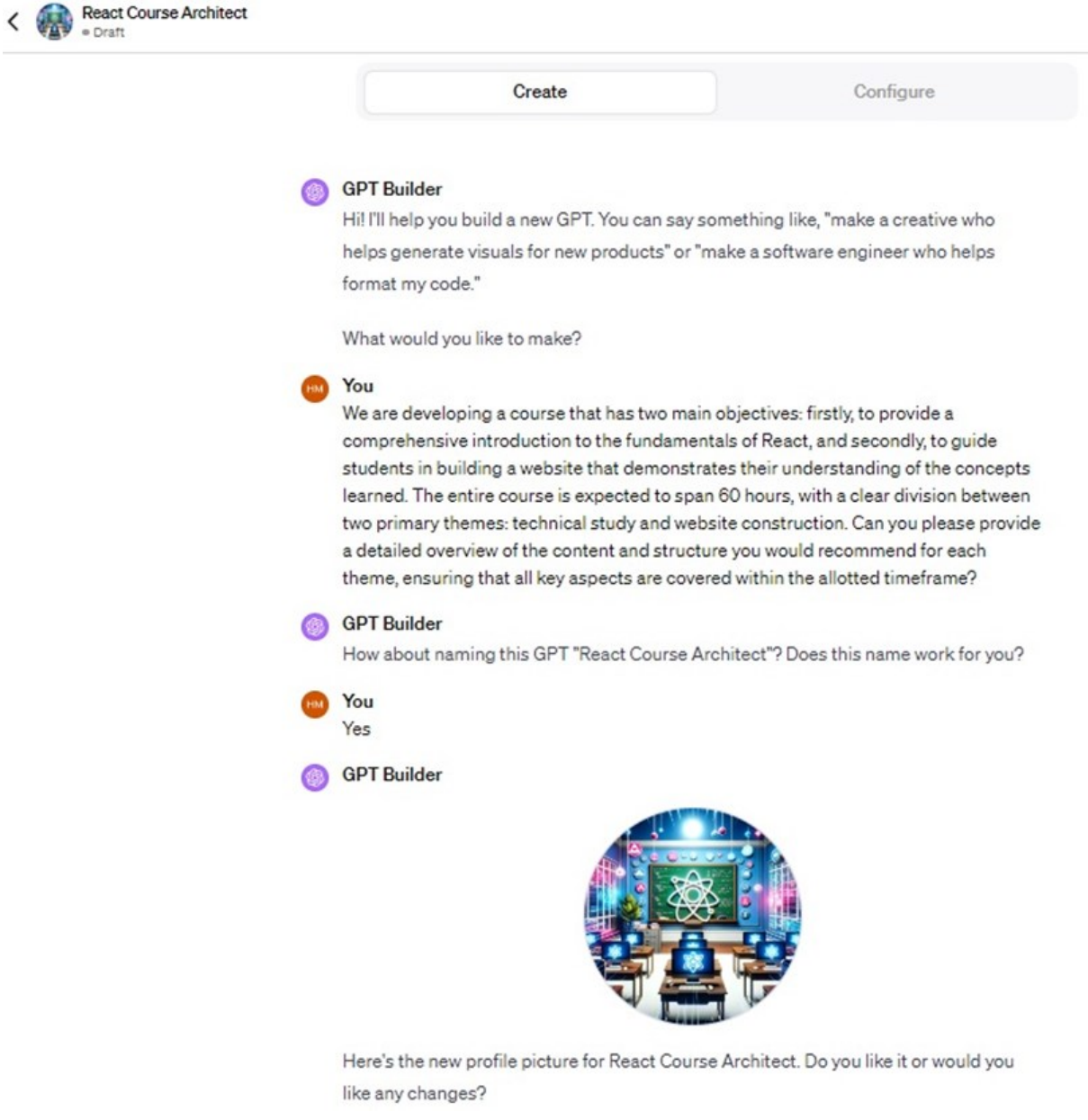
#### 2. React Fundamentals (8 Hours)

- Components and Props.
- State and Lifecycle.
- Handling Events.



Vaikka kehoitus kirjoitettiin suomen kielellä, kuten kuva 6 havainnollistaa, antaa Prompt Perfect englanniksi uuden, paremmin muotoillun kehotteen. Huomioidaan että ChatGPT antoi samalla kurssia varten alustavan aikataulurungon. Otetaan runko talteen ja käytetään sitä myöhemmässä vaiheessa alustamaan GPT:n aikataulua. Seuraavaksi kopioidaan Prompt Perfectin antama kehote ja syötetään se GPT:lle kuten kuva 7 näyttää.

Kuva 7 GPT:n ominaisuuksien asetus



The screenshot shows the 'React Course Architect' GPT interface. At the top, there are 'Create' and 'Configure' buttons. The chat history includes:

- GPT Builder:** Hi! I'll help you build a new GPT. You can say something like, "make a creative who helps generate visuals for new products" or "make a software engineer who helps format my code."
- You:** We are developing a course that has two main objectives: firstly, to provide a comprehensive introduction to the fundamentals of React, and secondly, to guide students in building a website that demonstrates their understanding of the concepts learned. The entire course is expected to span 60 hours, with a clear division between two primary themes: technical study and website construction. Can you please provide a detailed overview of the content and structure you would recommend for each theme, ensuring that all key aspects are covered within the allotted timeframe?
- GPT Builder:** How about naming this GPT "React Course Architect"? Does this name work for you?
- You:** Yes
- GPT Builder:** Here's the new profile picture for React Course Architect. Do you like it or would you like any changes?

The profile picture is a circular image of a classroom with a green chalkboard featuring a React logo, desks with laptops, and colorful balloons.

GPT kysyy tarkentavia kysymyksiä GPT:n ominaisuuksiin ja henkilökohtaisiin mieltymyksiin perustuen. Kysymyksiä tulee esimerkiksi käyttötapauksiin liittyen, kuten millainen rooli GPT:llä on, suunnitteleeko se tunteja vai keskittyykö enemmän käsitteiden selittämiseen.



Sekä käyttäjän oppimistyyliin liittyen, onko käyttäjällä tiettyjä opetusmetodeja, joita tulee korostaa tai välttää ja mikä on opiskelijan lähtötaso, onko jo osaamista esimerkiksi Reactissa tarvittavasta JavaScriptistä. Mikäli viimeisintä ei kysytä automaattisesti niin se on hyvä tarkentaa GPT:lle. Ominaisuuksia ja tarvittavia taustatietoja voi myöhemmin lisätä ja yksityiskohtaistaa.

Kyseisen GPT:n ominaisuuksiin halutaan, että pohjana käytetään toisessa keskustelussa luotua runkoa kurssille ja, että opetus painottuu esimerkkeihin ja niiden kautta selittämiseen. Kerrotaan myös tarkentavasti, että aikataulun mukaisesti tulisi opiskella 2 tuntia päivässä 60 päivän ajan ja, että GPT optimoisi ajankäytön sekä suunnittelee kurssin sen perusteella. Aikataulu näyttää nyt Reactin opiskelun kannalta kuvan (kuva 8) mukaiselta. Eli yhteensä opiskellaan teknistä puolta 36 tuntia jaettuna erilaisiin teemoihin, suurimmat teemat ovat jaettuna kahdelle eri päivälle.

## Kuva 8 Reactin perusteiden aikataulu



### React Course Architect

Ymmärretty, tässä on päivitetty aikataulu, jossa jokaiselle päivälle on varattu 2 tuntia opiskeluaikaa. Jotkut aiheet on jaettu useammalle päivälle, jotta pysytään tässä aikarajassa.

### Tekninen Opiskelu Reactista (2 tuntia päivässä)

- Päivä 1: Johdanto Reactiin (2 tuntia)**
  - Reactin peruskäsitteet ja käyttötarkoitus.
- Päivä 2: Ympäristön Pystytys (2 tuntia)**
  - React-ympäristön asentaminen ja peruskonfigurointi.
- Päivä 3-4: JSX:n Perusteet (4 tuntia)**
  - JSX:n syntaksi ja käyttö React-komponenteissa.
- Päivä 5-6: Komponentit ja Propsit (4 tuntia)**
  - Komponenttien luominen ja propsien käyttö.
- Päivä 7-8: Tilan Hallinta (4 tuntia)**
  - Tila Reactissa ja sen hallinnan perusteet.
- Päivä 9-10: Tapahtumankäsittely (4 tuntia)**
  - Tapahtumien käsittely ja tilan päivittäminen.
- Päivä 11-12: Ehdollinen Renderöinti ja Listat (4 tuntia)**
  - Ehdollinen renderöinti ja listaelementtien käsittely.
- Päivä 13-14: Reitityksen Perusteet (4 tuntia)**
  - Perusteet sivujen välillä navigoimiseen React-sovelluksessa.
- Päivä 15-16: React Sovelluksen Rakenne (4 tuntia)**
  - Hyvien käytäntöjen soveltaminen ja sovelluksen rakenteen suunnittelu.
- Päivä 17-18: Kertaus ja Syventävät Harjoitukset (4 tuntia)**
  - Opitun kertaaminen ja syventävät harjoitukset.

### Yhteensä: 36 tuntia

Tämä aikataulu tarjoaa selkeän ja hallitun tavan edetä Reactin opiskelussa, keskittyen 2 tuntiin päivässä. Voit säätää järjestystä tai keskittyä enemmän tiettyihin aiheisiin tarpeen mukaan.

ja verkkosivujen tekemiselle jää tällöin 24 tuntia ja teemat jaoteltu jokaisessa osiossa kahdelle päivälle, kuten kuva 9 havainnollistaa.

## Kuva 9 Nettisivuprojektin aikataulu

### Nettisivuprojekti (2 tuntia päivässä)

Päivä 19-20: Projektin Suunnittelu ja Ideointi (4 tuntia)

1. **Sivuston Konsepti ja Tavoitteet**
  - Määritä sivuston päämäärä ja kohderyhmä.
2. **Sisällön Rakenne**
  - Suunnittele sivuston sisällön ja rakenteen alustava hahmotelma.

Päivä 21-22: Runko ja Komponenttien Kehitys (4 tuntia)

1. **Komponenttien Suunnittelu**
  - Luo komponenttien rakenne ja hierarkia.
2. **Rakenteen Pystytys**
  - Rakenna sivuston perusrunko React-komponentteja käyttäen.

Päivä 23-24: Sisällön Lisäys ja Toiminnallisuus (4 tuntia)

1. **Sisällön Integrointi**
  - Lisää tekstiä, kuvia ja muita mediaelementtejä.
2. **Toiminnallisuuden Toteutus**
  - Lisää interaktiivisuutta, kuten navigointi, lomakkeet tai muu toiminnallisuus.

Päivä 25-26: Tyylittely ja Käyttöliittymä (4 tuntia)

1. **CSS ja Responsiivisuus**
  - Sovella tyylejä ja varmista sivuston responsiivisuus.
2. **Käyttäjäkokemuksen Hiominen**
  - Paranna käyttäjäkokemusta pienillä viilauksilla.

Päivä 27-28: Testaus ja Optimointi (4 tuntia)

1. **Sivuston Testaus**
  - Testaa toiminnallisuudet ja varmista sivuston toimivuus eri laitteilla.
2. **Suorituskyvyn Optimointi**
  - Tarkista latausajat ja mahdolliset ongelmat.

Päivä 29-30: Viimeistely ja Julkaisu (4 tuntia)

1. **Viimeistely ja Dokumentaatio**
  - Tarkista kaikki yksityiskohdat ja kirjoita lyhyt dokumentaatio.
1. **Julkaisu**
  - Valitse julkaisualusta (esim. GitHub Pages, Netlify) ja julkaise sivusto.

Tallennetaan aikataulu. GPT:iden parhaina puolina pidetään niiden mahdollisuutta vastaanottaa ja käsitellä tiedostoja. Syötetään GPT:lle tiedostona äsken saatu aikataulu. Tarkastetaan, että GPT:lle on valittu kaikki tarvittavat ominaisuudet ja hyväksytään GPT:n luonti. GPT generoi vastauksena ensimmäisen päivän sisällön, kuten kuva 10 esittää.

## Kuva 10 Ensimmäisen päivän aikataulu



You

Nyt, kun aikataulu on rakennettu niin aloitetaan ensimmäisen päivän opinnot!



React Course Architect

Mahtavaa! Aloitetaan siis ensimmäisen päivän opinnot, jossa keskitymme Reactin perusteisiin.

### Päivä 1: Johdanto Reactiin (2 tuntia)

Tavoitteena on ymmärtää, mikä React on ja miksi se on tärkeä työkalu web-sovellusten kehityksessä.

Aiheet:

#### 1. Mikä on React?

- React on JavaScript-kirjasto dynaamisten käyttöliittymien luomiseen.
- Kehittäjät ympäri maailmaa käyttävät Reactia tehokkaiden ja interaktiivisten web-sovellusten rakentamiseen.

#### 2. Reactin Historia ja Tausta

- Reactin kehitti Facebook.
- Se julkaistiin vuonna 2013 ja siitä on tullut yksi suosituimmista käyttöliittymäkirjastoista.

#### 3. Miksi Reactia Käytetään?

- Sen komponenttipohjainen rakenne tekee sovelluksen koodista modulaarista ja helposti ylläpidettävää.
- Virtuaalinen DOM (Document Object Model) parantaa suorituskykyä, koska muutokset tehdään ensin virtuaaliseen DOM:iin ennen varsinaista DOM:iin päivittämistä.
- React on joustava ja yhteensopiva monien muiden kirjastojen ja kehyksien kanssa.

Harjoitus:

- **Tutki ja Lue:** Etsi lisätietoa Reactista ja sen käytöstä. Voit käyttää virallisia React-dokumentaatioita tai muita luotettavia lähteitä.
- **Muistiinpanojen Tekeminen:** Kirjaa ylös avainkohdat Reactista, mukaan lukien sen perusominaisuudet ja miksi se on suosittu valinta web-kehityksessä.

Käytä näitä kahta tuntia tutustumiseen ja ymmärryksen syventämiseen Reactista. Kun olet valmis, voimme jatkaa seuraavana päivänä ympäristön pystytyksellä. Jos sinulla on kysymyksiä tai tarvitset apua, olen täällä auttamassa!

## 6.2 Reactin perusteiden opettelu

Aikataulutuksen ansiosta Reactin perusteiden aihealueen laajuus aukenee ja näin ollen myös mahdollisuus aiheiden syvällisempään tutkimiseen. Jotta Reactin oppimisen lisäksi opitaan kommunikoimaan ChatGPT:n kanssa tehokkaasti niin hyödynnetään aluksi pluginia nimeltä Prompt Perfect. Sen avulla saa konkreettisia esimerkkejä, miten tehostaa omia prompteja. Pyydetään Prompt Perfectin avulla GPT:tä laatimaan ensimmäisen päivän

aiheesta kattavat materiaalit ja tehtävät, jolloin saadaan uusi, tarkennettu, päivän teemoihin sopiva aikataulu. Aikataulun mukaan päivän opiskeltavia aiheina ovat: ”Johdatus Reactiin”, ”Reactin synty ja evoluutio”, ”Reactin ydinkonseptit”, ”Miksi React?”, ”Käytännön harjoitus: Yksinkertainen React-projekti” ja ”Kysymykset ja vastaukset, sekä yhteenveto”.

### **6.2.1 Opetusmateriaalien luominen**

Tehokkaan kommunikoinnin yksi osa-alue on osittaminen, joten pyydetään GPT:tä osio kerrallaan opettamaan opetettava aihe, jolloin syvennyttään ja rajataan aihe käsiteltäviin osa-alueisiin. Omalla aktiivisuudella ja kysymysten asettelulla on suuri merkitys vastauksiin. GPT ei automaattisesti luo monisivuisia tieto- ja opetuskokonaisuuksia vaan se antaa ensin yleisellä tasolla aiheet, joihin pääsee syventymään tarkentavien lisäkysymysten avulla. Esimerkiksi kuva 11 esittää, miten saadaan jo otsikkotasoa tarkempaa tietoa Reactin ydinkonseptista.

Kuva 11 Reactin ydinkonseptit

### 3. Reactin Ydinkonseptit

#### JSX (JavaScript XML)

- **Mikä on JSX?**

- JSX on syntaksi, joka yhdistää HTML:n kaltaisen merkintätavan JavaScriptiin. Se tekee komponenttien kirjoittamisesta helpompaa ja koodista luettavampaa.

- **JSX:n Käyttö:**

- Voit kirjoittaa JSX:ää kuin HTML:ää, mutta sen avulla voit myös hyödyntää JavaScriptin voimaa. Esimerkiksi, voit käyttää JavaScript-lausekkeita `{ }` sisällä JSX:ssä.

```
jsx Copy code
function WelcomeComponent() {
  return <h1>Tervetuloa, {user.name}</h1>;
}
```

#### Komponentit ja Propsit

- **Komponentit:**

- React-sovellukset rakennetaan komponenteista. Komponentti on itsenäinen ja uudelleenkäytettävä koodiyksikkö, joka hallitsee oman näkymänsä.
- Komponentteja on kahta tyyppiä: toiminnallisia ja luokkapohjaisia.

- **Propsit (Properties):**

- Propsit ovat tapa siirtää tietoa vanhemmilta komponenteilta lapsikomponenteille. Ne ovat kuin funktion argumentteja ja auttavat tekemään komponenteista joustavampia ja uudelleenkäytettävämpiä.

```
jsx Copy code
function Greeting(props) {
  return <h1>Tervehdys, {props.name}</h1>;
}
```

#### Tila (State) ja Elämänkierto (Lifecycle)


- **Tila:**


- Tila on tieto, jota komponentti hallitsee  ka voi muuttua ajan kuluessa. Tila tekee komponenteista dynaamisia.

Seuraavaksi voitaisiin pyytää GPT:tä selittämään tarkemmin JSX:stä. Esimerkiksi kertomaan mitä on JSX, miten sitä käytetään sekä pyytää luomaan niistä erilaisia esimerkkejä. Toisaalta voitaisiin haluta tietää, mitä eroa on JSX:llä ja HTML:llä.

Opetusmateriaalina halutaan käyttää oppimisen tehostamiseksi myös muita kuin pelkästään GPT:tä, joten kysytään kuvan (kuva 12) mukaisesti, mitä materiaaleja olisi tällä hetkellä saatavilla ilmaiseksi verkossa opiskeluun.

Kuva 12 Ilmaiset verkkomateriaalit opiskeluun

 **You**  
mitä ilmaisia kursseja tai materiaaleja on olemassa Reactin opiskeluun tällä hetkellä? Millaisia suomenkielisiä sivustoja?

 **ChatGPT**  
Reactin opiskeluun on saatavilla runsaasti ilmaisia kursseja ja materiaaleja eri kielillä, mukaan lukien suomeksi. Tässä joitakin suosittuja suomenkielisiä verkkosivustoja ja resursseja, jotka tarjoavat ilmaista React-opetusta:

1. **Koodikoulu:** Koodikoulu on suomenkielinen sivusto, joka tarjoaa ilmaisia oppaita ja kursseja, mukaan lukien Reactin perusteet. Voit aloittaa oppimisen osoitteessa [Koodikoulu React-opas](#).
2. **CodeAcademy:** CodeAcademy tarjoaa suomenkielisen React-kurssin, joka sisältää interaktiivisia tehtäviä ja ohjattua oppimista. Voit tutustua siihen osoitteessa [CodeAcademy React](#).
3. **Full Stack Open:** Full Stack Open on Helsingin yliopiston ja muita yhteistyökumppaneita ylläpitämä ilmainen kurssi, joka opettaa mm. Reactia ja modernia web-kehitystä. Kurssi on saatavilla suomeksi. Voit tarkistaa sen osoitteesta [Full Stack Open](#).
4. **W3Schools:** W3Schools tarjoaa monia suomeksi käännettyjä React-oppaita ja esimerkkejä. Voit tutustua niihin osoitteessa [W3Schools React-opetus](#).
5. **React.fi:** React.fi on yhteisöllinen sivusto, joka tarjoaa suomenkielistä tietoa ja resursseja Reactista. Sivusto sisältää oppaita ja artikkeleita, jotka voivat auttaa oppimisessa. Voit tutustua siihen osoitteessa [React.fi](#).
6. **YouTube:** YouTube on runsaasti suomenkielisiä React-opetusvideoita eri tasoille oppijoille. Etsi "React opetus" tai vastaavaa hakusanaa löytääksesi videoita, jotka sopivat tarpeisiisi.

Nämä ovat hyviä resursseja aloittelijoille ja niille, jotka haluavat syventää tietämystään Reactista suomeksi. Muista myös tarkistaa Reactin virallinen dokumentaatio (<https://reactjs.org/>), joka on saatavilla englanniksi ja tarjoaa kattavan kuvauksen kirjaston käytöstä ja ominaisuuksista.

Saadaan kattava listaus erilaisia opiskelukanavia esimerkiksi Helsingin yliopiston ja yhteistyökumppaneiden järjestämä Full Stack Open ja CodeAcademy sekä W3Schools. Listassa nähdään myös konkreettinen esimerkki, miksi ChatGPT:n kanssa kommunikoidessa täytyy olla kriittinen. Listan kohdassa kuusi ChatGPT antaa esimerkisivuksi React.fi, mutta kyseistä sivua tuollaisessa muodossa ei ole olemassa.



## 6.2.2 Harjoitustehtävät

Päivän opeteltavien aiheiden käsittelyn jälkeen pyydetään seuraavaksi erilaisia harjoitustehtäviä oppimisen tueksi. Koska GPT omaa sille aiemmin syötetyn tiedoston aikataulusta, voidaan sitä suoraan pyytää tiedostoa hyödyntäen tekemään päivän aiheisiin liittyen tehtäväpaketin esimerkiksi kehoitteella ”Tee nyt 6 erilaista tehtävänantoa päivän 10 aiheisiin liittyen”.

Kuva 13 Harjoitustehtävien pyytäminen



### Functional React Course Architect

Tässä on kuusi tehtävänantoa päivän 10 aiheesta, Tapahtumankäsittely Reactissa:

1. **Perus Klikkaustapahtuman Käsittely:** Luo yksinkertainen React-komponentti, joka renderöi painikkeen. Kun painiketta klikataan, sen tulisi näyttää hälytys mukautetulla viestillä.
2. **Lomakkeen Käsittely:** Kehitä lomake, jossa on syöttökenttä ja lähetä-painike Reactissa. Toteuta tapahtumankäsittely syötetyn datan kaappaamiseksi ja näyttämiseksi, kun lomake lähetetään.
3. **Asynkroninen Datan Haku:** Käytä Fetch API:ta ja kirjoita React-komponentti, joka hakee tietoja API-päätepisteestä painikkeen klikkauksen jälkeen ja näyttää haetut tiedot.
4. **Dynaamiset Tapahtumakuuntelijat:** Luo lista kohteita Reactissa. Lisää toiminnallisuus, joka sallii jokaisen kohteen omaavan oman klikkaustapahtuman kuuntelijan, joka näyttää kohteen tiedot klikattaessa.
5. **Ehdollinen Renderöinti Tapahtuman Perusteella:** Rakenna React-komponentti, jossa on vaihtopainike. Toteuta tapahtumankäsittely siten, että painikkeen klikkaaminen vaihtelee tekstikappaleen näkyvyyttä.
6. **Tapahtumankäsittely Tilan Päivityksellä:** Suunnittele laskurikomponentti Reactiin. Sisällytä lisää- ja vähennä-painikkeet ja käytä tapahtumankäsittelijöitä laskurin arvon päivittämiseen ja näyttämiseen.


Näiden tehtävien tarkoituksena on parantaa ymmärrystäsi siitä, miten käsitellä erilaisia tapahtumia Reactissa, keskittyen erityisesti siihen, miten tapahtumat voivat vaikuttaa tilaan ja laukaista käyttöliittymän päivityksiä.


Kuten kuva 13 havainnollistaa, eivät tehtävänannot ole kovin yksityiskohtaisia. Tilanne voidaan korjata joko muuttamalla kysymyksenasettelua yksityiskohtaisemmaksi ja pyytää tietyn tyylistä ja laajuista harjoitusta tai tässä tapauksessa ajan säästämiseksi kokeilla syöttää GPT:lle tiedostolla erilaisia esimerkkejä tehtävänannoista. Esimerkkitehtäviä



saadaan ilmaisilta verkkokursseilta/dokumentaatiosta ja niiden pohjalta pyydetään ChatGPT:tä luomaan yksityiskohtaisempia tehtävänantoja. Esimerkiksi tehtävänannon kohdasta kolme ”Asynkroninen Datan Haku: Käytä Fetch API:ta ja kirjoita React-komponentti, joka hakee tietoja API-päätepisteestä painikkeen klikkauksen jälkeen ja näyttää haetut tiedot” saadaan esimerkkitehdoston avulla yksityiskohtaisempi pyytämällä GPT:tä ottamaan huomioon sille syötetyn tiedoston tehtävänannot ja luomaan sen pohjalta tehtävä uudestaan.

#### Kuva 14 Esimerkkitehdosten käyttö

 **You**  
Tee kohdan 3 tehtävänannosta yksityiskohtaisempi. Ota yksityiskohtaisuuteen mallia sinulle syötetystä tiedostosta "esimerkkitehdavananto"

 **Functional React Course Architect**  
Tehtävä 3: **Asynkroninen Datan Haku ja Näyttö Reactissa**

Tavoite: Luo React-komponentti, joka hakee tietoja API-päätepisteestä painikkeen klikkauksen jälkeen ja näyttää nämä tiedot käyttöliittymässä.

Vaiheet:

- Komponentin Luonti:** Luo uusi React-komponentti, esim. `DataFetcher``.
- Tilan Alustus:** Käytä `useState``-koukkaa tilan alustamiseen. Tarvitset vähintään kaksi tilamuuttujaa: yhden datan tallentamiseen ja toisen lataustilan indikointiin.
- Fetch-API:n Käyttö:** Kirjoita funktio, joka käyttää Fetch-API:ta datan hakemiseen. Voit esimerkiksi hakea JSON-muotoista dataa jostakin julkisesta API:sta.
- Tapahtumankäsittelijä:** Lisää painike, joka aktivoi datan haun. Tapahtumankäsittelijän tulee kutsua datan haku -funktiota ja päivittää tila saadun datan ja lataustilan perusteella.
- Datan Näyttäminen:** Näytä haettu data käyttöliittymässä. Käytä ehdollista renderointiä näyttääksesi lataustilan aikana esim. latausindikaattorin.
- Virheen käsittely:** Lisää virheen käsittely, joka näyttää käyttäjälle ilmoituksen, mikäli datan haku epäonnistuu.

Lisähaasteena voit lisätä ominaisuuksia, kuten sivutuksen tai suodatuksen haettavalle datalle. Tehtävän suorittaminen antaa käytännön kokemusta datan asynkronisesta käsittelystä ja sen integroimisesta React-sovellukseen.

Nyt tehtävänanto on jo kuvan (kuva 14) mukaisesti kattavampi ja yksityiskohtaisempi, mikä parantaa harjoittelun tehokkuutta. Kuten aiemmin mainitti, GPT:tä voidaan päivittää kurssin edetessä ja sitä kannattaa tehdä aina tarpeen vaatiessa. Kuva 14 käy ilmi, että GPT on päivitetty keskittymään yksinomaan funktionaalisen Reactin opetukseen, mikä vähentää sekaannuksen riskiä luokkapohjaisen Reactin kanssa.

Yksittäisten tehtävien lisäksi GPT:tä voi pyytää jo kurssin alussa suunnittelemaan oppikokonaisuus huomioon ottaen jonkin konkreettisen sovelluksen, jota täydennetään ja rakennetaan päivittäin kurssin edetessä. Tässä tapauksessa GPT ehdotti esimerkiksi Kommenttisovelluksen rakentamista, jolla käyttäjät voivat lisäillä, poistaa ja vastata toisten käyttäjien kommentteihin.

### **6.2.3 Ohjelmoinnin haasteet**

Opiskelun edetessä opiskelijat kohtaavat useita ohjelmointiin liittyviä haasteita ja asioita, joiden ymmärtäminen voi olla aluksi vaikeaa. Varsinkin dokumentaatioiden lukeminen ja virheiden korjaus voivat tuottaa varsinkin aloitteleville ohjelmoijille merkittäviä ja aikaa vieviä haasteita, minkä johdosta opiskelumotivaatio voi laskea. ChatGPT:tä voi hyödyntää edellämainituissa ongelmissa esimerkiksi niin, että pyytää sitä selittämään vaikeaksi koetun asian mahdollisimman yksinkertaisesti ja esimerkkien avulla. Mikäli ChatGPT ei tuota heti ymmärrettävää vastausta, kehotetta säädetään ja testataan uudelleen niin kauan, kunnes asia ymmärretään. Kuvan (kuva 15) esimerkissä ChatGPT:tä pyydettiin selittämään, miten filter-funktio toimii Reactissa.

## Kuva 15 Koodin selitys ja opetus



**You**

selitä ja opeta minulle, kuten Reactin aloittelijalle, miten tämä toimii: ""const filteredNumbers = numbers.filter(number => number > 10);""



**Functional React Course Architect**

Tietysti! Käydään läpi tuo esimerkki `.filter()`-metodin käytöstä JavaScriptissa, joka on erityisen hyödyllinen React-sovellusten kanssa. Oletetaan, että olet Reactin aloittelija ja haluat ymmärtää, miten tämä rivi koodia toimii:

```
javascript Copy code
const filteredNumbers = numbers.filter(number => number > 10);
```

### Selitys Osa Kerraltaan:

- Taulukko `numbers`:**
  - Oletetaan, että sinulla on muuttuja nimeltä `numbers`, joka on taulukko (lista) numeroita. Esimerkiksi: `const numbers = [1, 15, 30, 4, 5];`
- Filter-Metodi:**
  - `.filter()` on JavaScriptin taulukkometodi, jota käytetään taulukon alkioden suodattamiseen perustuen johonkin ehtoon.
  - Tämä metodi käy läpi `numbers`-taulukon jokaisen alkion.
- Nuolifunktio:**
  - `number => number > 10` on nuolifunktio. Se on lyhyt tapa kirjoittaa funktio JavaScriptissa.
  - Tässä tapauksessa, `number` on argumentti, joka edustaa taulukon `numbers` kunkin alkion arvoa, kun `.filter()`-metodi käy läpi taulukkoa.
  - `number > 10` on ehto. Se tarkoittaa "valitse tämä `number`, jos se on suurempi kuin 10".
- Suodatusprosessi:**
  - `.filter()`-metodi suorittaa nuolifunktion jokaiselle `numbers`-taulukon alkioille.
  - Jos nuolifunktion ehto (`number > 10`) palauttaa `true` (eli numero on suurempi kuin 10), kyseinen numero sisällytetään uuteen taulukkoon.
  - Jos ehto palauttaa `false` (eli numero on 10 tai pienempi), numeroa ei sisällytetä uuteen taulukkoon.

Kattava selitys jatkuu edelleen ja sen jälkeen saadaan myös tarkentavia esimerkkejä filterin käytöstä. Mikäli aiheeseen haluaa syventyä perusteellisemmin niin omista mieltymyksistä ja oppimistavoista riippuen ChatGPT:tä voi esimerkiksi pyytää tuottamaan lisää esimerkkejä tai harjoitustehtäviä. Tämä mahdollistaa opetettavan asian aktiivisen harjoittelun.

Usein opiskelijoilla kuluu aikaa erilaisten syntaksivirheiden etsimiseen, kuten sulkeiden puuttumiseen. ChatGPT:tä voi hyödyntää niiden ja myös muidenkin virheiden korjaamiseen ja selittämiseen, jolloin tapahtuu samalla niin oppimista kuin myös ajansäästöä.

ChatGPT:tä voidaan käyttää myös koodin katselmoijana, joka antaa huomionarvoisia ajatuksia koodin tarkastuksen lisäksi esimerkiksi saavutettavuusasioista sekä vinkkejä uusien teknologioiden ja kirjastojen käytöstä. Kun React Course GPT:tä pyydettiin tarkastamaan sille syötetty koodi, huomioi se selkeiden virheiden lisäksi paljon muita asioita, kuten saavutettavuusasioita, joita ei välttämättä aloitteleva ohjelmoija tule vielä ajatelleeksi. Lisäksi se huomioi koodin siistimistä ja ehdotti korjauksia, joiden avulla koodista saa hyvien käytäntöjen mukaista, kuten kuva 16 esittää.

Kuva 16 Parannusehdotukset koodiin



### React Course

Kyllä, koodisi näyttää hyvältä perustan osalta kommenttisovellukselle. Olet onnistuneesti luonut keskeiset komponentit ja yhdistänyt ne yhteen sovellukseksi. Tässä on joitakin huomioita ja mahdollisia parannuksia:

- Kommentin Esitys:** `Comment`-komponentissa `return [<div>{id} {text}</div>]` on käytetty hakasulkuja, jotka eivät ole tarpeellisia. Voit yksinkertaisesti palauttaa `<div>{id} {text}</div>`.
- Ainutlaatuinen Avain (`key`) Listassa:** Olet käyttänyt `comment.id`-arvoa avaimena (`key`) listattaessa kommentteja, mikä on hyvä tapa varmistaa, että jokaisella listan elementillä on ainutlaatuinen tunnistus.
- Formin Tila ja Tapahtumankäsittely:** Tällä hetkellä `CommentForm` ei käsittele kommenttien lähetyksiä. Jatkossa, kun opit tilanhallinnasta, voit lisätä tilan hallinnan tähän komponenttiin käyttäen `useState`-koukkaa ja käsittelijäfunctioita kommenttien lisäämiseksi.
- Tyyli ja Ulkoasu:** Vaikka tyyli ei ole tämän vaiheen pääfokus, voit aina lisätä CSS-tyylejä parantaaksesi käyttöliittymän ulkoasua ja käyttäjäkokemusta.
- Saavutettavuus ja HTML-rakenne:** Pidä mielessä saavutettavuus ja HTML:n semanttinen rakenne. Esimerkiksi voit käyttää `form`-elementtiä `CommentForm`-komponentissa ja varmistaa, että käyttöliittymäelementit, kuten painikkeet ja syöttökentät, ovat käyttäjäystävällisiä.
- Koodin Siisteyden Ylläpito:** Kommenttien ja merkintöjen lisääminen koodiin voi auttaa sinua ja muita kehittäjiä ymmärtämään koodin rakennetta paremmin.

Yllä olevat ehdotukset auttavat sinua parantamaan koodin rakennetta paremmin.

- Komponenttien Eristäminen:** Jos sovelluksesi kasvaa, voit harkita komponenttien siirtämistä omiin tiedostoihinsa, mikä tekee koodista helpommin hallittavaa ja luettavaa.

Tässä on päivitetty versio `Comment`-komponentistasi ilman tarpeettomia hakasulkeita:

```
jsx Copy code
const Comment = ({ text, id }) => {
  return <div>{id} {text}</div>;
}
```

## 6.2.4 Muut opiskelumateriaalit

On hyvä tunnistaa, että oppimateriaalien monipuolinen käyttö laajentaa opiskelukokemusta. ChatGPT on yksi osa tätä kokonaisuutta, mutta sen rinnalla käytetään myös aiemmin esiteltyjä Fullstack Openin verkkokurssia sekä Reactin virallista dokumentaatiota. Nämä resurssit täydentävät ChatGPT:n tarjoamaa reaaliaikaista vuorovaikutusta ja yksilöllistä ohjausta, luoden kattavan ja monipuolisen oppimisympäristön. Tämä yhdistelmä edesauttaa kokonaisvaltaista Reactin ymmärrystä ja opiskelua sekä tehostaa ajan käyttöä oppimisprosessissa. Lisäksi rinnalle otettu kurssi edistää kriittistä ajattelua toimien samalla myös laadun tarkastuksena ChatGPT:n antamien vastausten osalta.

Aiheiden monimutkaistuessa tukeudutaan oppimisprosessin tehostamiseksi enemmän strukturoituun verkkokurssiin ja dokumentaatioon, mutta käytetään ChatGPT:tä jatkuvasti rinnalla haastavien aiheiden selkeyttäjänä, koodin tarkastajana, lisäharjoitusten tuottajana ja uudenlaisien näkökulmien luojana.

## 6.2.5 Muut huomiot

Kuten aiemmin mainittiin, on oma aktiivisuus ChatGPT:n kanssa kommunikoidessa tärkeää. Erilaisten kommunikointityylien sekä työkalujen käyttö ja kokeilu edistävät ChatGPT:n kanssa työskentelyä. Kokeilun arvoisia ovat esimerkiksi erilaisten promptaustyylien, valmiiden tai itseluotujen GPT:eiden ja pluginien monipuolinen käyttö. Voidaan esimerkiksi testata muuttuvien tilanteiden vaikutusta ChatGPT:n antamiin vastauksiin niin, että vaihdellaan keskustelun konteksteja teknisistä keskusteluista teoreettisiin selityksiin. Lisäksi voidaan kokeilla erilaisia tyylejä kommunikaatioon ja katsoa miten ne vaikuttavat ChatGPT:n vastauksiin ja selitysten laatuun, kuten esimerkiksi ”Selitä asia X, kuten selittäisit sen aloittelevalle React-opiskelijalle” vs. ”Selitä asia X kuten selittäisit alan Y asiantuntijalle”.

ChatGPT:lle voi myös antaa rooleja, joiden avulla se kykenee omaksumaan erilaisia persoonallisuuksia, ammatillisia rooleja tai jopa historiallisia hahmoja. Tämä mahdollistaa monipuolisemman ja räätälöidymmän vuorovaikutuksen, sillä tekoäly voi vastata kysymyksiin ja osallistua keskusteluihin valitun roolin näkökulmasta. Tällainen roolipohjainen lähestymistapa voi tehdä keskusteluista informatiivisempia, tarjoten samalla uudenlaisia oppimiskokemuksia.

Opiskelua ajatellen voi olla hyödyllistä käyttää kahta tai useampaa eri GPT:tä, erikoistuen sillä hetkellä opeteltaviin asioihin. Toinen voisi keskittyä esimerkiksi vain sanallisen materiaalin tuottamiseen ja asioiden selittämiseen ja toinen voisi olla erikoistunut koodin tuottamiseen ja tarkistamiseen.

## 6.3 Yksisivuisen verkkosivun luominen

Yksisivuisen (SPA) verkkosivun luominen on monivaiheinen prosessi alkaen suunnitteluvaiheesta ja päättyen valmiin sivun julkaisuun. Sivun kehittämisprosessissa hyödynnetään aiemmin luotua tietoperustaa Reactin periaatteista, kuten komponenttien rakennetta ja tilanhallintaa. Käydään läpi esimerkkien avulla ChatGPT:n hyödyntämisestä prosessissa sekä dokumentoidaan teknistä toteutusta myös muutamilla esimerkeillä.

### 6.3.1 Sovelluksen suunnittelu

Verkkosivun kehittämistyö alkaa suunnitelman laadinnalla. Kohderyhmä sekä sivuston aihe on jo tiedossa. Aiheena on ChatGPT:n hyödyntäminen opiskeluprosessissa mahdollisimman tehokkaasti ja koota niistä sekä tehokkaasta kommunikoinnista sivulle konkreettisia esimerkkejä ja vinkkejä niin opiskelijoille, opettajille kuin muillekin aiheesta kiinnostuneille. ChatGPT:tä voidaan hyödyntää suunnitelman laadinnassa esimerkiksi kysymällä, millaisia vaihtoehtoja on layoutin luomiseksi Reactissa tai pyytää apua itse suunnitteluun. Tässä tapauksessa kysyttiin apua suunnitteluun käyttämällä promptia:

Auta laatimaan suunnitelma yksisivuisen verkkosivun luomiseen, kun kohderyhmänä on sekä aloittelevat, että kokeneemmat ChatGPT:n käyttäjät, tarkoituksena on luoda ohjesivusto ChatGPT:n tehokkaaseen hyödyntämiseen sekä kommunikointiin ja sivu sisältää kolme linkkiä: Etusivu, oman GPT:n luominen ja tehokas hyödyntäminen (erityisesti ohjelmoinnissa).

Vastauksena saadaan kattava pohja suunnitelmalle, mikä avaa kokonais käsitystä tulevasta projektista. Esimerkiksi teknisen toteutuksen osalta saadaan vinkkejä muun muassa layoutin ratkaisuun Gridistä ja navigoinnin toteutukseen React Routerista sekä komponenttien suunnittelusta, kuten kuva 17 havainnollistaa.

## Kuva 17 Teknisen toteutuksen suunnitelma



Functional React Course Architect

### 2. Tekninen Toteutus

#### Teknologiat

- **React:** Sovelluksen runko ja komponentit.
- **React Router:** Navigointi yksisivuisessa sovelluksessa, käyttäen `BrowserRouter`` ja `Route`` -komponentteja sivujen väliseen navigointiin.

#### CSS

- Sovelluksen tyylittely, mukaan lukien CSS Grid ja Flexbox responsiivisen layoutin toteutukseen.

#### Komponenttien Suunnittelu

- **Layout Komponentti:** Yhtenäinen kehys, joka sisältää navigointipalkin ja yhteiset osat (header, footer).
- **Sivukohtaiset Komponentit:** Eri komponentit kullekin osiolle, jotka ladataan dynaamisesti käyttäjän navigoidessa.

#### Navigointi ja Sisältö

- **Single Page Application (SPA):** Käytetään React Routerin `Link`` -komponentteja sujuvaan navigointiin sivun eri osioiden välillä ilman sivun uudelleenlatauksia.

Koska GPT:n tarkoitus on opettaa uusia asioita, pyydetään sitä syventämään tietoa esimerkkien avulla selkeästi ja antamalla yksityiskohtaista kontekstia opiskelijan tasosta. Tässä esimerkissä halutaan syventyä React Router DOM:n käyttöön kehoitteella: "Selitä, kuten aloittelevalle ohjelmoijalle, mitä tarkoittaa React Router DOM. Anna lisäksi konkreettinen esimerkki". Vastauksena saadaan kuvan (kuva 18) mukainen selitys siitä, mikä on React Router DOM ja miksi se on hyödyllinen. Lisäksi saatiin asennusohjeet sekä yksinkertainen esimerkki, jota voidaan hyödyntää omassa projektissa.

## Kuva 18 React Router DOM:n selitys



### Functional React Course Architect

React Router DOM on kirjasto, jota käytetään single-page applications (SPA) eli yksisivuisten sovellusten reititykseen Reactissa. Yksisivuisella sovelluksella tarkoitetaan web-sovellusta, joka lataa yhden HTML-sivun ja päivittää sen sisältöä dynaamisesti käyttäjän kanssa vuorovaikutuksessa, sen sijaan että lataisi uusia sivuja palvelimelta. Tämä tekee sovelluksesta nopeamman ja antaa paremman käyttäjäkokemuksen.

### Mikä on React Router DOM?

React Router DOM mahdollistaa URL-osoitteiden käytön navigoinnin hallintaan React-sovelluksessasi. Sen avulla voit määritellä reittejä (eli polkuja tai URL-osoitteita), jotka vastaavat eri komponentteja sovelluksessasi. Kun käyttäjä siirtyy sovelluksessasi tiettyyn osoitteeseen, React Router näyttää sille osoitteelle määritellyn komponentin, ilman että sivua tarvitsee ladata uudelleen selaimessa.

### Miksi React Router DOM on hyödyllinen?

- **SPA:n Tehokkuus:** Se mahdollistaa yksisivuisten sovellusten tehokkaan rakentamisen. Käyttäjät voivat navigoida sovelluksen eri osien välillä ilman sivun uudelleenlatauksia, mikä parantaa suorituskykyä ja käyttäjäkokemusta.
- **Joustavuus:** Voit määritellä reittejä dynaamisesti ja ryhmitellä komponentteja, jolloin sovelluksesi rakenne pysyy selkeänä ja modulaarisena.
- **Käyttöliittymän Säilyttäminen:** Sovelluksen tila (esim. käyttäjän syöttämät tiedot lomakkeissa) säilyy, kun käyttäjä navigoi eri näkymien välillä.

Suunnitelmapohjan avulla kootaan yhteenveto sivun rakenteesta. Sivun layoutissa käytetään Gridiä, sivulle tulee navigaatiopalkki vasemmalle laidalle, joka sisältää linkit etusivulle, oman GPT:n luomiselle sekä ChatGPT:n tehokkaalle hyödyntämiselle. Käytetään React Router DOM:ia reititykseen sekä rakennetaan jokaisesta sivusta oma komponentti, lisäksi luodaan uudelleen käytettäviä komponentteja toiminnallisuuksille, kuten nappuloille. Tyylittely tehdään omaan CSS-tiedostoon. Projektin edetessä lisäillään ja muokataan toiminnallisuutta tarpeen mukaan ja lopulta testataan ja julkaistaan verkkosivu.

### 6.3.2 Tekninen toteutus

Staattisen yksisivuisten verkkosivun kehittämistyössä käytetään Reactin funktionaalisia komponentteja sekä hookeja tilanhallintaan ja käyttöliittymän dynaamiseen päivittämiseen. Projekti aloitetaan luomalla ensin tarvittavat komponentit, jotka edustavat eri sivuja (etusivu, oman GPT:n luominen ja ChatGPT opiskelutyökaluna), jonka jälkeen tuodaan tarvittavat komponentit App.js-tiedostoon, suunnitellaan rakenne sivustolle ja tehdään reititys React Router -kirjaston avulla, kuten ohjelmakoodissa (ohjelmakoodi 1) on tehty.



## Ohjelmakoodi 1 Sivun rakenne ja reititys

```
1 import { BrowserRouter, Routes, Route, Navigate } from "react-router-dom";
2 import Home from "../components/Home";
3 import CreateGPT from "../components/CreateGPT";
4 import Prompting from "../components/Prompting";
5 import NavigationBar from "../components/NavigationBar";
6 import Header from "../components/Header";
7 import Footer from "../components/Footer";
8 import "../styles.css";
9 import config from '../config';
10
11
12
13 function App() {
14   const {home, createGPT, prompting} = config.routes;
15
16   return (
17     <BrowserRouter>
18       <div className="App">
19         <Header />
20         <NavigationBar />
21         <div className="content">
22           <Routes>
23             <Route path={home} element={<Home />} />
24             <Route path={createGPT} element={<CreateGPT />} />
25             <Route path={prompting} element={<Prompting />} />
26             <Route path={"*"} element={<Navigate to={home} />} />
27           </Routes>
28         </div>
29         <Footer />
30       </div>
31     </BrowserRouter>
32   );
33 }
34
35 export default App;
36
```

Seuraavaksi tarkastellaan ohjelmakoodissa (ohjelmakoodi 2) navigaattorakenteen kannalta keskeisin komponentti NavigationBar, jonka tehtävänä on määrittellä sovelluksen päänavigointi.

## Ohjelmakoodi 2 NavigationBar-komponentti

```
1 import React from 'react';
2 import { Link } from 'react-router-dom';
3 import config from '../config';
4
5 const NavigationBar = () => {
6
7   const {home, createGPT, prompting } = config.routes;
8
9   return (
10    <nav className="navigation-bar">
11      <div className="nav-button">
12        <Link to={home} className="nav-link">Etusivu</Link>
13      </div>
14      <div className="nav-button">
15        <Link to={createGPT} className="nav-link">Luo oma GPT</Link>
16      </div>
17      <div className="nav-button">
18        <Link to={prompting} className="nav-link">ChatGPT opiskelutyökaluna</Link>
19      </div>
20    </nav>
21  );
22 };
23
24
25 export default NavigationBar;
26
```

Tuodaan sille tarvittavat komponentit React Router DOM -kirjastosta, mikä mahdollistaa käyttäjän sujuvan liikkumisen sovelluksen eri osien välillä ilman sivun uudelleenlatauksia. NavigationBar käyttää Link-komponenttia, joka on tärkeä osa reitityksen toteutuksessa, ja jotka ohjaavat käyttäjää haluttuihin osioihin sovelluksessa.

Seuraavaksi rakennetaan eri sivut eli komponentit, joiden toiminta perustuu edellä mainittuun reititykseen. Tarkastellaan lähemmin etusivun rakennetta (ohjelmakoodi 3), jossa on myös propsien ja JSX käyttöä sekä tilanhallintaa.

## Ohjelmakoodi 3 Etusivun ohjelmakoodi

```

1  import profileImage from '../assets/profile.jpg'
2  import NavigationButtons from './NavigationButtons';
3  import config from './config';
4  import texts from './texts.json';
5  import Card from './Card';
6  import { renderTextWithLineBreaks } from './utils';
7  import TipSection from './TipSection';
8
9  const Home = () => {
10     const prevPage = null; // Ei edellistä sivua Home-sivulla
11     const nextPage = { path: config.routes.createGPT, name: "Seuraava sivu" };
12     const homeTexts = texts.home;
13     const cards = homeTexts.cards;
14     const tip = texts.tip["text-section"];
15
16     return (
17       <div>
18         <div className="home-section">
19           <div className="profile-info">
20             <img src={profileImage} alt="Profiilikuva" className="profile-image" />
21             <div className="profile-text">
22               <strong>Hanna Moisala</strong><br />
23               Julkaistu: 15.3.2024
24             </div>
25           </div>
26           <div className="intro-section">
27             <h1>{homeTexts.header}</h1>
28             {renderTextWithLineBreaks(homeTexts.text1)}
29           </div>
30           <div className="create-info">
31             {renderTextWithLineBreaks(homeTexts.text2)}
32           </div>
33           <div className="cards-container">
34             {cards.map((card, index) => (
35               <Card key={index} title={card.title} text={card.text} backgroundImage={card.backgroundImage} />
36             ))}
37           </div>
38           <TipSection header={tip.header} text={tip.text} solutionText={tip.solutionText} />
39         </div>
40         <div className="navigation-buttons">
41           <NavigationButtons prevPage={prevPage} nextPage={nextPage} />
42         </div>
43       </div>
44     );
45   };
46
47   export default Home;

```

Koodia tarkastellessa huomion arvoisena asiana nousee esille, miten JSX:ää voidaan hyödyntää Reactissa. Esimerkiksi rivillä 28 ja 31 kutsutaan JavaScript-funktiota ”renderWithLineBreaks” suoraan HTML-rakenteen sisältä aaltosulkeiden avulla. Propsien käytöstä löytyy esimerkki riviltä 38, jossa Home-komponentti välittää dynaamista tietoa alikomponentilleen TipSectionille. Tässä tapauksessa, Home toimii vanhempana, joka lähettää TipSection-lapsikomponenteilleen tarvittavat propsit: header, text ja solutionText. Seuraavaksi siirrytään tarkastelemaan tarkemmin tilanhallintaa TipSection-komponentissa (ohjelmakoodi 4).

## Ohjelmakoodi 4 Esimerkki tilanhallinnasta

```
1 import React, { useState } from 'react';
2 import { renderTextWithLineBreaks } from '../utils';
3
4 const TipSection = ({ header, text, solutionText }) => {
5   const IdeaIcon = () => (
6     
7   );
8
9   const [isClicked, setIsClicked] = useState(false);
10  const handleClick = () => {
11    setIsClicked(!isClicked);
12  }
13
14  return (
15    <div className="tip-section">
16      <h2>{header} <IdeaIcon /></h2>
17      {renderTextWithLineBreaks(text)}
18      <button onClick={handleClick}>Ratkaisu</button>
19      {isClicked && renderTextWithLineBreaks(solutionText)}
20    </div>
21  );
22 };
23
24
25 export default TipSection;
```

Tilanhallintaa tarvitaan esimerkin tapauksessa näyttämään ja piilottamaan kuvan (kuva 19) pähkinäarvoituksen ratkaisu. Tämä tapahtuu käyttämällä useState-hookia, joka on Reactin tarjoama tapa hallita komponentin tilaa. Se mahdollistaa tilan muutosten seurannan ja komponentin päivittämisen vastaamaan näitä muutoksia. Yksinkertaisesti sanottuna, kun tila muuttuu (klikkauksen seurauksena), komponentti renderöidään uudelleen näyttämään muuttuneen tilan mukainen sisältö eli tässä tapauksessa asettamalla useStaten alkuarvo falseksi, jolloin ratkaisu on piilossa. Käyttäjän klikatessa päivitetään isClicked sen hetkisen arvon vasta-arvoksi, eli falsesta trueksi ja truesta falseksi (rivillä 11). Näin ilmestyy vastaus napin alle ja klikattaessa uudestaan, vastaus katoaa näkyvistä.

## Kuva 19 Esimerkki tilanhallinnasta UI:ssa

### Pähkinä 💡

Mieti mitä eroa on kehoitteilla:

1. Kerro minulle Reactista
2. Kuinka voin siirtää tietoja yhdestä komponentista toiseen Reactissa käyttäen propseja?

Ratkaisu



### Pähkinä 💡

Mieti mitä eroa on kehoitteilla:

1. Kerro minulle Reactista
2. Kuinka voin siirtää tietoja yhdestä komponentista toiseen Reactissa käyttäen propseja?

Ratkaisu

Ensimmäinen kehote on epämääräinen, laaja ja siitä puuttuu selkeä tavoite tai ongelma, mikä johtaa myös epämääräisiin ja laajoihin vastauksiin.

Etusivusta ohjelmakoodi 5 esittelee myös lähemmin Card-komponenttia. Kyseisen komponentin käyttötarkoitus on esittää osa sivun sisällöstä korttimuodoissa, jonka propseina kulkee korttien tarvittavat tiedot, kuten tekstit ja taustakuvat (Card-komponentista esimerkki UI:ssa kuva 21). Kortit mapataan id:n avulla sivulle vierekkäisiksi korttikomponenteiksi, joilla jokaisella kortilla on oma sisältönsä.

### Ohjelmakoodi 5 Card-komponentti etusivulla

```

34 {cards.map((card, id) => (
35   <Card key={id} title={card.title} text={card.text} backgroundImage={card.backgroundImage} />
36 ))}

```

Toiminnallisuuksia lisätään muillekin sivuille käyttämällä mahdollisuuksien mukaan uudelleen aiemmin luotuja komponentteja, kuten ohjelmakoodi 6 havainnollistaa, jossa on "ChatGPT opiskelutyökaluna" -sivulle lisätty Card-komponentti ohjeistusten esittelyille.

## Ohjelmakoodi 6 Komponenttien uudelleen käyttö

```

147   const openModal = (rule) => {
148     setSelectedRule(rule);
149     setIsOpen(true);
150   };
151
152   const closeModal = () => {
153     setIsOpen(false);
154   };
155
156   return (
157     <div>
158       <h2>{prompting.header}</h2>
159       <p>{prompting.text}</p>
160       <div className="rules-container">
161         {rules.map((rule) => (
162           <Card
163             key={rule.id}
164             title={rule.title}
165             text={rule.description}
166             onClick={() => openModal(rule)}
167           />
168         ))}
169       </div>
170       <Modal
171         isOpen={modalIsOpen}
172         onRequestClose={closeModal}
173         style={customStyles}
174         contentLabel="Rule Details"
175       >
176         <h2>{selectedRule?.title}</h2>
177         <p>{selectedRule?.description}</p>
178         {selectedRule?.expandedContent}
179         <button onClick={closeModal}>Sulje</button>
180       </Modal>
181       <NavigationButtons prevPage={prevPage} />
182     </div>
183   );
184 };
185
186 export default Prompting;


```

Toiminnallisuutta lisätään myös Modal-kirjaston ja tilanhallinnan avulla avaamaan jokaisesta ohjeesta oma suurempi ikkuna riveillä 170–180 (esimerkki UI:ssa kuva 23).

Käyttöliittymän muuttuvat tekstit, kuten otsikot ja sisältö, tuodaan texts.json-tiedostosta, mikä mahdollistaa sisällön helpon päivittämisen ja mahdollistaisi myös tekstien määrittelyn eri kielillä. Ainoastaan muuttumattomina pidettyjä tekstejä voidaan kirjoittaa suoraan koodiin. Sovelluksen visuaalinen ilme, kuten värit, fontit ja elementtien asetelut, määritellään keskitetysti CSS-tiedostossa. Tämä eriytetty lähestymistapa varmistaa, että ulkoasu on johdonmukainen koko sovelluksessa ja helpottaa tyylitietojen hallintaa ja muokkausta.

Seuraavaksi huolehditaan versionhallinnasta ja samalla saadaan varmuuskopiot, jotta koodi ei ole pelkästään paikallisesti olemassa ja ratkaisuna tähän käytetään GitHub-repositoriota. ChatGPT:tä voidaan hyödyntää pyytämällä niin kutsuttua step-by-step-ohjeistusta kyseenomaisen vaiheittaisen asian ratkaisemiseksi, kuten kuva 20 esittää. ChatGPT:lle annetaan kehote: "Anna step-by-step ohjeet siihen, miten voidaan Visual Studio Coden terminaalin kautta liittää koodit GitHubiin". Vastauksena saadaan käyttäjäystävälliset ohjeet, joita noudattamalla onnistutaan saamaan koodit GitHubiin.

Kuva 20 Step-by-step-ohjeet

 **ChatGPT**

Jos haluat liittää koodisi GitHubiin käyttäen Visual Studio Code (VS Code) -terminaalia, noudata näitä askel askeleelta -ohjeita:

- Luo GitHub-repository:**
  - Mene GitHubiin ja luo uusi repository klikkaamalla "+"-ikonia oikeassa yläkulmassa ja valitsemalla "New repository". Anna repositorylle nimi ja halutessasi kuvaus. Valitse julkisuusasetus ja luo repository.
- Avaa projektisi VS Codessa:**
  - Käynnistä VS Code ja avaa projektikansiosi valitsemalla "File" > "Open Folder".
- Alusta Git projektissasi (jos ei vielä tehty):**
  - Avaa terminaali VS Codessa valitsemalla "Terminal" > "New Terminal".
  - Tarkista, onko Git jo alustettu kirjoittamalla `git status`. Jos ei, alusta Git kirjoittamalla `git init`.
- Lisää tiedostot versionhallintaan:**
  - Lisää kaikki projektitiedostot Gitin seurantaan kirjoittamalla `git add .`.
- Tee ensimmäinen commit:**
  - Tallenna muutokset versionhallintaan kirjoittamalla `git commit -m "First commit"`.
- Linkitä GitHub-repository:**
  - Siirry GitHubiin ja kopioi luomasi repositorion URL.
  - Linkitä paikallinen repository GitHubiin kirjoittamalla `git remote add origin REPOSITORY_URL`, korvaa `REPOSITORY_URL` kopioimallasi osoitteella.
- Työnnä koodi GitHubiin:**
  - Työnnä koodisi GitHub-repositoryyn kirjoittamalla `git push -u origin master` tai `git push -u origin main`, riippuen siitä, käytätkö `master`- vai `main`-haaraa oletuksena.

Tämän prosessin jälkeen koodisi on liitetty GitHubiin, ja voit nähdä sen GitHub-repositoriossasi. Jos kohtaat ongelmia autentikoinnissa tai muissa vaiheissa, tarkista GitHubin dokumentaatio tai Gitin ohjeet lisätietoja varten.

### 6.3.3 Testaus ja julkaisu

Varmuuskopioinnin ja versionhallinnan jälkeen koodi viimeistellään, minkä jälkeen siirrytään testaus- ja julkaisuvaiheeseen. Staattisen ja yksinkertaisen verkkosivun osalta usein riittää

manuaalinen testaus varmistamaan, että kaikki toiminnallisuudet toimivat odotetusti. React-sovellusten testaamisessa voidaan yleisesti hyödyntää esimerkiksi JEST-testikehystä sekä React Testing Library -kirjastoa, jotka mahdollistavat komponenttien yksikkö- ja integraatiotestauksen, mutta tässä tapauksessa testaus hoidettiin manuaalisesti käyttöliittymän kautta. Testauksen jälkeen sovellus on valmis julkaisuun, jolloin se siirretään tuotantoympäristöön ja tehdään saataville loppukäyttäjille. Julkaisuprosessi voi sisältää myös viime hetken optimointeja suorituskyvyn parantamiseksi.

Julkaisu toteutetaan käyttäen GitHub Pages -palvelua, joka mahdollistaa verkkosivun julkaisemisen suoraan GitHub-repositoriosta. Tämä vaatii lisäyksiä package.json-tiedostoon, kuten kotisivun domainin määrittelyn ([www.opitehokkaasti.com](https://www.opitehokkaasti.com)), sekä julkaisuskriptien lisäämisen, kuten ohjelmakoodi 7 havainnollistaa.

Ohjelmakoodi 7 Julkaisua varten tarvittavat lisäykset package.json-tiedostoon

```

1  {
2    "homepage": "https://www.opitehokkaasti.com",
3    "name": "opitehokkaasti",
4    "version": "0.1.0",
5    "private": true,
6    "dependencies": {
7      "@testing-library/jest-dom": "^5.17.0",
8      "@testing-library/react": "^13.4.0",
9      "@testing-library/user-event": "^13.5.0",
10     "gh-pages": "^6.1.1",
11     "react": "^18.2.0",
12     "react-dom": "^18.2.0",
13     "react-modal": "^3.16.1",
14     "react-router-dom": "^6.21.3",
15     "react-scripts": "5.0.1",
16     "web-vitals": "^2.1.4"
17   },
18   "scripts": {
19     "predeploy": "npm run build",
20     "deploy": "gh-pages -d build",
21     "start": "react-scripts start",
22     "build": "react-scripts build",
23     "test": "react-scripts test",
24     "eject": "react-scripts eject"
25   },

```

Lisäksi tehdään Visual Studio Coden terminaalin komennot (komento 1), joiden avulla onnistuu Create React Appilla alustetun projektin julkaisu GitHubin Pagesilla.

Komento 1 Julkaisukomennot VS Coden terminaalissa

```

npm install --save gh-pages
npm run deploy

```



## 6.4 Päiväkirjamerkintöjen analysointi ja pohdinta

Päiväkirjamerkintöjä kirjoitettiin 30 päivän ajan. Ensin opiskeluprosessista ja sen jälkeen verkkosivun tekemisestä. Teemoittelun avulla saatiin nostettua koko opiskeluprosessista esille viisi keskeistä teemaa, jotka toistuivat merkinnöissä: omien GPT-mallien rooli, haasteet, saavutukset, muiden materiaalien käyttö ja kommunikoinnin merkitys. Tarkastellaan ja pohditaan näitä teemoja tarkemmin.

### 6.4.1 Opiskeluprosessi yleisesti

Oppimisen edistyminen Reactin opettelussa näytti kasvavan jatkuvasti ajan kuluessa. Alkuvaiheessa keskityttiin perusteiden ymmärtämiseen, minkä jälkeen siirryttiin yhä monimutkaisempiin aiheisiin, kuten tilanhallintaan ja komponenttien yhdistelyyn. harjoitustehtävistä selviytyttiin onnistuneesti ja useissa merkinnöissä nousi esille innostuneisuus ja motivaatio, kun oppimista tapahtui. ChatGPT oli päivittäisessä käytössä esimerkiksi selittäen erilaisia funktioita, kertoen esimerkkejä päivän aiheista ja antaen harjoitustehtäviä, mutta sen tehokkuus oppimisen edistämässä vaihteli päivittäin, riippuen teknisistä haasteista ja kommunikaation tehokkuudesta.

Alkuvaiheessa käytettiin pelkästään ChatGPT:tä ja erityisesti omaa, mukautettua GPT:tä opettajana ja kurssipohjana, luoden sen avulla niin teoriaosiot kuin harjoitustehtävät, mutta aiheiden monimutkaistuessa turvaututtiin enemmän muihin resursseihin, kuten Fullstack-kurssiin. Tämän jälkeen ChatGPT:n rooli oli pikemminkin olla haastavien ja vaikeasti ymmärrettävien asioiden selittäjä kuin kokonainen verkkokurssi.

### 6.4.2 Omien GPT-mallien rooli

Omien GPT:eiden luominen eri käyttötarkoituksiin toi osaltaan helpotusta oppimisprosessiin. Jo alussa huomattiin, että omat mukautettavat GPT:t mahdollistivat huomattavasti tehokkaamman käytön kuin perinteiset keskusteluikkunat, jotka saattavat unohtaa pitkien keskustelujen yksityiskohdat ja vaativat toistuvaa uudelleenalustamista. GPT:n harkittu alustus, antaen tarpeeksi kontekstia johti siihen, ettei promptaustaidoilla ollut enää niin suurta merkitystä kuin aiemmin.

Päiväkirjamerkinnät osoittavat, että tehokkaan yhteistyön kehittyessä GPT:n kanssa, opiskelumateriaalien hyödyntäminen mahdollisti syvemmän tason saavuttamisen oppimisprosessissa. Prosessi edellytti useiden GPT-mallien kehittämistä ja niiden jatkuvaa

tarkentamista ja päivittämistä, jotta ne vastaisivat kulloinkin vallitseviin tarpeisiin mahdollisimman hyvin. Esimerkiksi neljäntenä päivänä toteutettiin GPT-mallien räätälöinti seuraavasti: ”Loin uuden GPT:n (React Course), jolle syötin aikataulun ja pyysin tekemään kattavat opetusmateriaalit aina kahdelle tunnille, sisältäen teorian ja harjoitukset. Tämä toimii paremmin kuin yksi GPT, joka koittaa hoitaa niin aikataulun, opiskelumateriaalien luonnin ja koodien selitykset”.

### 6.4.3 Haasteet

Aineisto osoittaa, että opiskelija kohtasi haasteita opintojensa aikana, jotka liittyvät teknisiin ongelmiin ja ChatGPT:n kykyyn ymmärtää sekä tuottaa asianmukaisia materiaaleja. Esimerkiksi jo kolmannen ja neljännen päivän aikana ilmeni merkkejä turhautumisesta, kun käytössä ollut oma mukailtu GPT ei pystynyt tuottamaan oppimateriaaleja aikaisemmin käsiteltyjen aiheiden pohjalta. Sen sijaan se loi harjoitustehtäviä myös sellaisista aiheista, joita ei ollut vielä opiskelussa käsitelty.

Teknisten haasteiden toistuminen, kuten ChatGPT:n rajoittunut vastausten generointikyky ja keskustelujen paisuminen liian laajoiksi, aiheutti ongelmia. Vastausten generointimäärän rajoitus, 40 vastausta kolmessa tunnissa, pakotti välillä turvautumaan GPT-3-mallin generoimiin vastauksiin, mikä ei aina tuottanut toivottua tulosta. Erityisesti ongelma nousi esille ohjelmointiin liittyvissä kysymyksissä.

Myöskään GPT-4 ei aina pystynyt tunnistamaan koodivirheitä tehokkaasti. Tämä haaste ilmaantui verkkosivuston kehitysvaiheessa, kun projektin laajuus kasvoi useiden sivujen koodimäärään ja niiden väliset vuorovaikutukset aiheuttivat ongelmia. ChatGPT yritti korjata virheet tuloksetta, mikä johti opiskeluajan käyttöön tehtävissä, jotka eivät suoranaisesti edistäneet oppimistavoitteita. Nämä ongelmat ratkesivat lopulta aina opiskelijan omien korjausten kautta.

Kuitenkin opiskeluprosessin edetessä ja ymmärryksen kasvaessa, monet alkuvaiheen haasteet saatiin ratkaistua. Ratkaisuna teknisiin ongelmiin toimi uusien keskusteluikkunoiden avaaminen ongelmien ilmaantuessa, mikä auttoi välttämään token-rajoitusten aiheuttamat generointiongelmat. Lisäksi omalle GPT:lle annettiin lisää yksityiskohtaista tietoa tiedostomuodoissa, mikä mahdollisti yksityiskohtaisemman ja syvällisemmän materiaalin tuottamisen. Tämä paransi merkittävästi opiskelutehokkuutta ja auttoi keskittymään paremmin päivän oppimistavoitteisiin.

Nämä kokemukset korostavat, että vaikka tekniset haasteet ja ChatGPT:n rajoitukset asettivat esteitä, luovilla ratkaisuilla ja jatkuvalla sopeutumisella näihin haasteisiin pystyttiin vastaamaan. Tällainen opettaa hyödyllisiä taitoja opiskeluprosessin hallinnassa ja tekoälyn hyödyntämisessä oppimisessa.

#### 6.4.4 Saavutukset

Päiväkirjamerkinnöistä nousi esille myös myönteinen vaikutus opiskeluun ja verkkosivun rakentamiseen. ChatGPT ja varsinkin oma, mukailtu kurssipohjainen GPT toimi keskeisenä resurssina koko opiskeluprosessin ajan, tarjoten sekä materiaalia että tukea haasteiden ratkaisemisessa. Sen avulla luotiin kurssille myös alustava aikataulurakenne, jota käytettiin pohjana koko kurssin ajan. Aikataulusta ei juurikaan poikettu perusteiden oppimisen aikana, mutta verkkosivun rakentamisen aikataulu tarkentui vasta sitä tehdessä. Tämä ilmenee päiväkirjamerkinnöistä, joissa viitataan aina päivän opiskeluaiheisiin ja GPT:n tuottamiin materiaaleihin.

Kun teknisiä ongelmia ei ilmaantunut, nousi merkinnöistä esille ChatGPT:n kyky tuottaa laadukkaita ratkaisuja ja vastauksia aloittelevan Reactin ohjelmoijan kysymyksiin. ChatGPT:n avulla kysymyksiin saatiin reaaliaikaisia vastauksia ja näin ollen säästyä aikaa ja opiskelu oli tehokasta. GPT osasi selittää opiskeltavat asiat selkeästi ja ymmärrettävästi. Esimerkiksi päivän 11 aiheena oli listojen käsittely, missä filter-funktion toiminta aiheutti hämmennystä, hyödynnettiin GPT:tä sen ymmärtämiseen:

Yritin ymmärtää, miten listalta poistaminen toimii, mutta Reactin dokumentaation lukeminen ei antanut täyttä selkeyttä sen toiminnasta. Tämän jälkeen pyysin ChatGPT:tä selittämään minulle yksityiskohtaisesti, miten se toimii, ja se avasi periaatteen todella hyvin. Yllätyin positiivisesti, miten helposti ymmärsin asian sen avulla. Myös map-funktion ymmärtämisessä ChatGPT osoitti kykynsä selittää selkeästi ja tarjota hyödyllisiä esimerkkejä.

Harjoitusmateriaalien osalta nousi esille, että esimerkkitehtävien syöttämisen jälkeen omalle GPT:lle, alettiin saada laadukkaita harjoitustehtäviä. Esimerkiksi päivän 12 ehdollisen renderöinnin aiheesta lukee näin: ”Ehdollinen renderöinti tuntui aluksi yksinkertaiselta, mutta kun GPT esitti haastavampia tehtäviä, harjoittelun myötä aiheen monimutkaisuus avautui. GPT:n ohjeistus oli kuitenkin niin selkeää, että aloin pääsin nopeasti aiheesta kartalle ja onnistuin suorittamaan kaikki harjoitustehtävät”. ChatGPT osasi hyödyntää

esimerkkitehtävien tyyliä ja muuttaa omia tehtävänantojaan tyyllillisesti samaan yksityiskohtaisempaan suuntaan.

Koodin selittäjän ja materiaalien luonnin lisäksi ChatGPT toimi tehokkaana virheiden korjaajana tiettyyn pisteeseen saakka. Merkinnöistä ilmenee, että Reactin perusteiden käsittelyssä ChatGPT toimi loistavasti. Esimerkiksi päivän 8 tilanhallinnan harjoituksesta lukee näin: "Sain jatkuvasti virheen, josta en meinannut päästä millään yli. Annoin koodin ChatGPT:lle ja se korjasi koodin toimivaksi sekunneissa ja antoi samalla kattavan selityksen, mistä virheeni johtui. Opin taas paljon". Monimutkaisemmissa ja pidemmissä koodeissa ChatGPT ei enää suoriutunut niin hyvin, kuten haasteosuudessa mainittiin.

Eryityisesti ChatGPT:n kyky tuoda esiin uusia näkökulmia osoittautui hyödylliseksi. Tällainen yhteistyö tekoälyn kanssa ei ainoastaan syvennä teknistä osaamista, vaan myös opettaa uusia tapoja ajatella ja oppia. Se kertoi saavutettavuusasioita, muistutti tietoturvallisuudesta ja yleensäkin laajensi näkökulmia eri asioihin. Esimerkiksi, kun opiskelija mietti verkkosivua tehdessään, millaisia vaihtoehtoja on sivun asettelulle eli layoutille niin ChatGPT antoi useita eri vaihtoehtoja ja kattavasti tietoja jokaisesta ja mitä yksityiskohtaisempia kysymyksiä ChatGPT:lle esitti, sitä tarkempia perusteluja siltä sai vastaukseksi. Merkinnöistä nousi esille myös, ettei ChatGPT tuottanut juurikaan virheellistä tietoa, todennäköisesti johtuen kysymysten oikeanlaisella asettelulla sekä joissain tapauksissa myös lähteiden käytöllä.

#### **6.4.5 Muiden resurssien merkitys**

Kun opiskeltavat aiheet muuttuivat monimutkaisemmiksi, muiden oppimateriaalien, kuten Fullstack Open -kurssin, käyttö yleistyi. Tämä ilmeni päiväkirjamerkinnöistä, jotka osoittivat, että valmiiksi suunnitellut ja jäsenellyt opintokokonaisuudet tarjosivat selkeämmän ja helpommin seurattavan oppimispolun. Esimerkiksi 14. päivänä tehdyssä avoimessa pohdintaosiossa todettiin: "Käytin Fullstack-kurssia saadakseni kokonaiskuvan navigoinnista ja reitityksestä. Vaikka oma GPT-mallini oli antanut hyvän johdatuksen aiheeseen, tunsin tarvetta saada lisävahvistusta ymmärtääkseni kaikki siihen liittyvät seikat perusteellisesti." Myös harjoitustehtävien laatiminen osana opiskeluprosessia muuttui prosessin edetessä työlääksi. Aluksi tehtävien generointi koettiin virkistävänä ja jopa innostavana, mutta opintojen edetessä se alkoi tuntua työläältä ja vei aikaa varsinaiselta opiskelulta. Tällöin käännyttiin mieluummin valmiiksi mietittyjen ja koottujen materiaalien puoleen.

Lisäksi päiväkirjamerkinnät paljastivat, että ChatGPT:n avulla oli mahdollista nopeasti syventää ymmärrystä, saada selkeitä selityksiä ja käytännön esimerkkejä, joita perinteiset

kurssimateriaalit eivät aina tarjonneet. Toisaalta Fullstack Open tarjosi laajan ja rakenteellisen opintopolun ohjelmoinnin perusteisiin, mitä ChatGPT ei yksinään voinut tarjota. ChatGPT:n nopea reagointikyky ja yksilöllinen ohjaus yhdistettynä Fullstack Openin kattavaan ja järjestelmälliseen tapaan opettaa Reactia, auttoi löytämään tasapainon itseohjautuvan oppimisen ja strukturoidun opetuksen välille. Tämän yhteistyön merkityksen ymmärtäminen on ollut keskeistä, sillä se on mahdollistanut molempien metodien vahvuuksien täysimittaisen hyödyntämisen opiskelussa.

#### **6.4.6 Kommunikoinnin merkitys**

Kommunikoinnin merkitys nousi esille päiväkirjamerkintöjen analyysissä viimeisenä keskeisenä teemana. Analyysissä havaittiin, että mukautettujen GPT-mallien käyttö mahdollisti tarkoituksenmukaisempien vastausten saamisen vähemmällä vaivalla verrattuna perinteisiin keskusteluikkunoihin. Tämä johtui siitä, että mukautetuissa malleissa keskustelurakennetta ei tarvinnut suunnitella yhtä huolellisesti token-rajoitusten puitteissa.

Päiväkirjamerkintöjen perusteella ilmeni, että kun kommunikointia ei optimoitu, se saattoi aiheuttaa aikaa vieviä ja tarpeettomia keskusteluja ChatGPT:n kanssa. Tämä korostui tilanteissa, joissa pyrittiin saamaan ymmärrettäviä vastauksia ilman huolellista kysymysten asettelua. Usein kehotusten hienosäätö johti toivottuihin vastauksiin, mutta se ei aina ollut riittävää ratkaisemaan kaikkia ongelmia, erityisesti monimutkaisemmissa tehtävissä kuten verkkosivun kehittämisessä, missä erityisesti tyylittelyssä koettiin haasteita.

Kommunikoinnin kannalta keskeisinä havaintoina oli, että ChatGPT:lle tulee tarjota riittävästi kontekstia sekä kehotuksissa että tarvittaessa lähteissä. Lisäksi osittamisen merkitys korostui: yleisluontoisiin kysymyksiin saatiin yleisluontoisia vastauksia, kun taas tarkemmin määriteltyihin kysymyksiin saatiin käyttökelpoisempia ja täsmällisempiä vastauksia. Hyväksi käytännöksi suurten kokonaisuuksien hallinnassa muodostui aluksi luoda yleiskatsaus aiheeseen ja sen jälkeen edetä vaiheittain yksityiskohtaisempiin osiin. Tämä lähestymistapa mahdollisti jäsenyntyneemmän ja tehokkaamman etenemisen työskentelyssä. Alkuvaiheessa hyödynnetty Prompt Perfect -työkalu jäi vähemmälle käytölle, kun ymmärrys ja taidot kommunikoida ChatGPT:n kanssa kehittyivät, mikä viittaa siihen, että kommunikointitaitojen kasvaessa tarve ulkopuolisille työkaluille väheni.

## 7 Tuloksena yksisivuinen verkkosivu

Konkreettisenä tuloksena ja näyttönä sille, että Reactin oppimista tapahtui ChatGPT:n avustuksella syntyi verkkosivu, jolle koottiin sekä tehokkaan kommunikoinnin että ChatGPT:n hyödyntämisen ohjeita ja esimerkkejä sekä oman GPT:n luomisprosessin ohjeet. Tässä luvussa esitellään lopputulos kuvin.

Valmiin verkkosivun etusivu näyttää kuvan (kuva 21) mukaiselta. Verkkosivu on yksisivuinen Reactilla toteutettu staattinen kokonaisuus, jossa hyödynnettiin ChatGPT:tä niin Reactin opiskelussa kuin verkkosivun suunnittelussa ja toteutuksessa. ChatGPT:tä ei pyydetty missään vaiheessa kirjoittamaan koko koodia vaan nimenomaan sen tarkoitus oli antaa ideoita, opettaa asioita selittämällä termejä sekä koodia ja antamalla esimerkkejä ja harjoitustehtäviä. Lopputuloksen ulkoasun pohjana eli CSS-tyylien rakentamisessa pyydettiin ChatGPT:tä antamaan pohja selkeälle ulkoasulle, mikä helpotti niin sanotun tyhjän sivun syndroomassa.

Verkkosivulle rakennettiin navigaatiopalkki vasemmalle reunalle, josta pääsee kätevästi toisille sivuille. Lisäksi tehtiin myös harjoituksen vuoksi NavigationButtons-komponentti sivun alareunaan edelliselle ja seuraavalle sivulle siirtymiselle. Tilanhallinnan konkreettiseksi esimerkiksi luotiin Pähkinä-osio, jossa tilaa muuttamalla joko näytetään tai piilotetaan kysymyksen ratkaisu. Tekstit taas tulevat omasta JSON-tiedostosta, jossa niiden muokkailu, lisääminen ja poistaminen on helppoa. Etusivulle luotiin lisäksi Card-komponentti, joka sisältää korttien rakenteen ja tilanhallinnan niiden näyttämiseksi. Aluksi kortit olivat pelkästään etusivulla, mutta myöhemmin niitä hyödynnettiin myös tehokkaan kommunikoinnin sivulla, jolloin niihin lisättiin tilanhallintaa.

## Kuva 21 Etusivu

ChatGPT opiskelun tukena

Etusivu

Luo oma GPT

ChatGPT opiskelutyökaluna

**Hanna Moisala**  
Julkaistu: 15.3.2024

### ChatGPT – Uuden aikakauden opiskelutyökalu

Tekoäly, erityisesti ChatGPT on jo nyt mullistanut niin työ- kuin vapaa-ajan opiskelumaailmasta puhumattakaan. ChatGPT:tä voi käyttää opiskelun työkaluna usealla eri tavalla, mutta mitä nämä tavat oikein ovat ja miten ChatGPT:tä voi käyttää mahdollisimman tehokkaasti?

Tällä sivustolla keskitymme ChatGPT:n tehokkaaseen käyttöön ohjelmoinnin opiskelussa sekä oman GPT:n rakentamiseen.

ChatGPT on OpenAI:n luoma generatiivinen tekoälysovellus, joka mahdollistaa vuorovaikutuksen ihmisen kanssa luonnollisella kielellä. Käyttäjä voi siis kirjoittaa kysymyksiä ja ChatGPT vastaa näihin kysymyksiin uskottavalla tavalla. Se perustuu GPT-3-kielimaliin, joka ilmestyi vuoden 2022 lopulla yleisön saataville. Periaatteessa ChatGPT:n voisi kuvaila sen olevan erittäin taitava chatbot, joka osaa vastata kysymyksiin monilla eri kielillä. Ilmaisversio on tällä hetkellä GPT-3.5 ja noin 20 euron kuukausihintaan saa Plus-version käyttöön, mikä perustuu GPT-4 malliin. ChatGPT tuottaa vastauksena tekstiä käyttäjän antamien syötteiden, eli ohjeiden tai toisin sanoen kehoitteiden (prompt), perusteella. Promptien avulla sovellusta ohjataan suorittamaan tehtäviä halutulla tavalla ja sen ansiosta sitä voidaan hyödyntää monipuolisesti esimerkiksi sisällön luonnissa, koodin kirjoittamisessa, tekstin tiivistämisessä, keskustelutehtävissä ja luovassa kirjoittamisessa. ChatGPT:n toiminta perustuu syvä- ja koneoppimiseen, neuroverkkoihin, GPT-malleihin ja luonnollisen kielen käsittelyyn. Se on koulutettu laajalla aineistolla, joka sisältää erilaisia tekstejä kuten kirjoituksia ja artikkeleita. Näiden ansiosta ChatGPT pystyy tunnistamaan kielen rakenteita ja ennustamaan tekstiä.

#### Tehokas hyödyntäminen

ChatGPT:n tehokas hyödyntäminen ja kommunikointi on avain tehokkaaseen oppimiseen. Joskus uuden työkalun käyttöönotto voi tuntua turhautavalta ja raskaalta. Näiden ohjeiden avustuksella pääset suoraan asiaan tehostamalla kommunikointia ja kehittämällä promptaustaitojasi.

#### Oman GPT:n rakentaminen

Tutustu prosessiin, kuinka luot oman GPT:n. Tämä osio tarjoaa perustiedot GPT-mallien toimintaperiaatteista, ohjeet mukautettavien mallien luomiseen ja vinkkejä niiden soveltamiseen omissa projekteissasi, olipa ajatuksenasikin opiskella tai tehdä sovellusta.

**Pähkinä** 💡

Mieti mitä eroa on kehotteilla:

1. Kerro minulle Reactista
2. Kuinka voin siirtää tietoja yhdestä komponentista toiseen Reactissa käyttäen propseja?

[Ratkaisu](#)

[Seuraava sivu](#)

Seuraava sivu eli oman GPT:n luomissivu rakentui kuvan (kuva 22) mukaisesti vieritettävästä pääosiosta, joka sisältää yksityiskohtaiset ohjeet ja esimerkin oman GPT:n luomisesta. Kuvien kokoa pystyy kasvattamaan, kun hiiren vie niiden päälle.

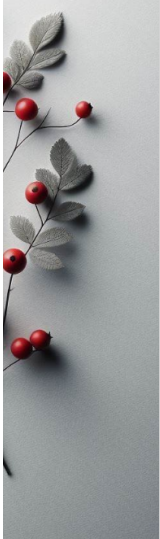
## Kuva 22 Oman GPT:n luomissivu

ChatGPT opiskelun tukena

Etusivu

Luo oma GPT

ChatGPT opiskelutyökaluna




### Oma GPT

OpenAI on kehittänyt GPT:t, jotka ovat mukautettavia ChatGPT-versioita erityisiin käyttötapoihin. GPT:t mahdollistavat käyttäjille luoda personoituja ChatGPT-versioita ilman koodauslaitoja. Luomisprosessi alkaa vuorovaikutuksella luomistyökalun kanssa, jossa käyttäjät antavat ohjeita ja määrittelevät GPT:n halutut ominaisuudet, kuten verkkoseläitön, kuvien luomisen tai datan analysoinnin. Tämä prosessi on suunniteltu helppokäyttöiseksi, jolloin käyttäjät voivat suunnitella GPT:n toiminnot ja ominaisuudet vuoropuhelun kautta.

Personoidut GPT:t tarjoavat etuja perinteiseen ChatGPT:hen verrattuna, erityisesti niiden kyvyssä käsitellä ajantasaista ja käyttäjäkohtaista tietoa. Toisin kuin normaali keskusteluikkuna ChatGPT:n kanssa, personoidut GPT:t voivat hyödyntää käyttäjien lataamia tiedostoja. Tämä mahdollistaa niiden tietopohjan laajentamisen ja päivittämisen uusimmilla tiedoilla. Lisäksi personoidut GPT:t mahdollistavat tarkempien ja yksityiskohtaisten vastausten tuottamisen, sillä ne voidaan ohjelmoida vastaamaan erityistarpeisiin ja erilaisiin konteksteihin. Niiden käytölliittymässä on myös asetuksia, jotka tukevat laajempaa soveltuvuutta, kuten kykyä yhdistää kolmannen osapuolen sovelluksiin ja palveluihin.

### Luo oma GPT

Tulemme luomaan tässä esimerkissä GPT:stä Reactin opiskelua varten kurssipohjan. Oman GPT:n luominen lähtee siitä, että kirjaututaan ensin osoitteeseen <https://chat.openai.com/>. Mikäli tunnuksia ei vielä ole niin tässä kohtaa rekisteröidytään ja hankitaan plus -jäsenyys, muuten ei omaa GPT:tä saa luotua. Kirjautumisen jälkeen avautuu uusi chat-ikkuna ja vasemmalla yläreunassa on Explore-valikko, jota klikkaamalla avautuu omat GPT:t. Klikkaamalla "Create a GPT" päästään luomaan oma, personoitu GPT.



[Edellinen sivu](#)   [Seuraava sivu](#)

@Hanna Moisala 2024

Sivulle "ChatGPT opiskelutyökaluna" on koottu keskeiset ohjeet ChatGPT:n tehokkaaseen hyödyntämiseen ja sen kanssa kommunikointiin. Ohjeet on esitetty korttimuodossa käyttäen uudelleenkäytettävää Card-komponenttia. Jokainen osio voidaan avata tilanhallinnan avulla klikkaamalla, jolloin saadaan konkreettisia esimerkkejä kustakin aiheesta, kuten kuva 23 havainnollistaa.



Kuva 23 Ohjesivun kortti

Tehokas kommunikointi ja ChatGPT:n hyödyntäminen

Tehokas kommunikointi koostuu useista eri osa-alueista ja ChatGPT:n avulla voit hyödyntää niitä tehokkaasti.

### Selkeät ohjeet

Tarkkojen ja yksityiskohtaisten ohjeiden käyttö vähentää mallin arvailua ja parantaa vastausten tarkkuutta.

### Aika

Ohjeista mallia käyttämään enemmän aikaa tehtävän ratkaisemiseksi.

### Opiskelun ja ohjelmoinnin tuki

Hyödynnä ChatGPT:tä opiskelun ja ohjelmoinnin tukena henkilökohtaisena opinto-ohjaajana, reaaliaikaisena palautteenantajana, tiedon lähteenä ja

## Selkeät ohjeet

Tarkkojen ja yksityiskohtaisten ohjeiden käyttö vähentää mallin arvailua ja parantaa vastausten tarkkuutta.

Se ei kuitenkaan tarkoita, että kehotteiden pitäisi olla mahdollisimman lyhyitä. Enemmän ohjeistuksella tarkoitetaan sitä, että mallille tulisi antaa enemmän kontekstia selkeästi ilmaistuna, kuten alla olevissa esimerkeissä on tehty.

- Esimerkki 1: Kysy ChatGPT:tä, miten voit parantaa koodisi laatua näin "Katso antamani koodi laadun näkökulmasta kriittisesti ja anna parannusehdotukset"
- Esimerkki 2: Pyydä ChatGPT:tä selittämään, mitä tarkoittaa "state" Reactissa "Selitä kuin olisin aloitteleva ohjelmoija, mitä "state" tarkoittaa Reactissa"

**You**  
Selitä kun olisin aloitteleva ohjelmoija, mitä "state" tarkoittaa Reactissa

**ChatGPT**  
Käytetään, että React-sovellus on vain se, jossa on muutama huoneisto-komponentti. Sovellus huoneisto on käyttämä ja näyttö (esimerkiksi valit, televisio, josta muuttavat tilaan (pöytä tai pöytä) perustuen siihen, miten niitä käytetään. Tässä vertailussa "state" on kuin muiden laitteiden tila se kertoo, ovatko valot päällä, onko televisio päällä, ja niin edelleen. Reactissa "state" on tapa pitää kukaan komponenttien tiedot tai tila, joka voi muuttua ajan myötä käyttäjän toimien tai sovelluksen sisäisen logiikan seurauksena.

**Mikä on State?**  
"State" on JavaScript-objekti React-komponentissa, joka tallentaa tietoja, jotka voivat muuttua ajan kuluessa. Aika kerta kun komponentin tila muuttuu, React päivittää ja renderoi uudestaan kopsen komponentin näyttämään uudet tiedot.

**Miksi State on Tärkeä?**  
Käytetään, että rakennat laskinta Reactilla. Laskimen numeronäyttö on hyvä esimerkki "state":sta. Kun painat numeronäyttöä ja laskutoimituksia, näyttö muuttuu vastaamaan syötteitä ja laskutoimitusten tuloksia. Ilman "state" sovelluksessa ei muuttaisi, mitä olet tehnyt, eikä se voisi näyttää tai päivittää tuloksia dynaamisesti.

**Kuinka Statea Käytetään?**  
Voi käyttää "state" ja luokkakomponentissa ja funktionaalisissa komponentissa React Hooks -koulujen, kuten "useState", avulla. Tässä on esimerkki käyttäen "useState"-hookkia funktionaalisissa komponentissa.

```

import React, { useState } from 'react';

function Laskin() {
  // useState() luodessasi uuden "state" muuttujan. Tässä "laskin" on alustettu
  // "0:n" arvoksi, mikä alustaa pöytä, ja "useState" on funktio,
  // joka voi muuttaa tämän tilan.
  const [lasku, setLasku] = useState(0); // Alku on pöytä

  return (
    <div>
      <p>Laskin on lasku? <input type="text" /> </p>
      <p>Muuta <input type="button" value="useState(lasku)" />
      <p>Kirja <input type="button" value="pöytä" />
    </div>
  );
}

```

Sulje

ja tarkentamalla ohjeita yksittäisistä yrittämällä yhä mallin mahdollisuuksista.

Hyödynnä verkkotietojen täydentämistä arvioimalla ja haastamalla ChatGPT:n antamia vastauksia.

Tausta peitetään vaalealla verholla tyylitellyn avulla kortin aukaisemisen yhteydessä, jotta avattu kortti erottuu muusta taustasta. Kortin voi sulkea Sulje-painikkeesta tai klikkaamalla kortin ulkopuolelle. Sivulle kootut ohjeet koostuvat kuvan (kuva 24) aiheista.

Kuva 24 Tehokas kommunikointi ja ChatGPT:n hyödyntäminen opiskelussa

Selkeät ohjeet	Kontekstin lisääminen lähteiden avulla	Osittaminen	Aika	Ulkoiset työkalut
<ul style="list-style-type: none"> <li>Ei tarkoita lyhyitä kehoitteita</li> <li>Anna yksityiskohtaisia ohjeita selkeästi</li> </ul>	<ul style="list-style-type: none"> <li>Tiedostojen lisäys</li> <li>Lisää lähdeteksti osaksi kehoitetta</li> <li>Erottimien käyttö</li> <li>Syvenny aiheeseen vuorovaikutuksen avulla</li> </ul>	<ul style="list-style-type: none"> <li>Laajoissa ja monimutkaisissa aiheissa</li> <li>Jos kysyt laajasti, saat laajan vastauksen</li> </ul>	<ul style="list-style-type: none"> <li>Anna mallille aikaa ajatella</li> <li>Pyydä pohtimaan vastaukseen johtava prosessi ennen vastauksen antamista</li> </ul>	<ul style="list-style-type: none"> <li>Pluginit, kuten Prompt Perfect</li> <li>(Korvataan pian GPT:ellä)</li> </ul>
Kommunikointityylien testaus	Opiskelun tuki	Ohjelmoinnin tuki	Iteratiivinen kehitys	Kriittinen ajattelu
<ul style="list-style-type: none"> <li>Erilaiset kommunikointityylit (asiantunteva, ystävällinen)</li> <li>Roolien antaminen mallille ja käyttäjälle</li> </ul>	<ul style="list-style-type: none"> <li>Tekstien tiivistäminen ja selittäminen</li> <li>Pääpointtien löytäminen</li> <li>Henkilökohtainen tuki ja opettaja</li> <li>Uusien näkökulmien esille nostaminen</li> <li>Reaaliaikainen palaute</li> <li>Opiskelumateriaalit</li> <li>Suunnittelutyökalu</li> </ul>	<ul style="list-style-type: none"> <li>Koodin selittäjä ja selventäjä</li> <li>Harjoitus- ja esimerkkikoodit</li> <li>Koodin luonti</li> <li>Koodin korjaus</li> </ul>	<ul style="list-style-type: none"> <li>Promptin testaaminen, kehittäminen ja testaaminen uudelleen</li> </ul>	<ul style="list-style-type: none"> <li>Vastausten oikeellisuuden tarkistus</li> <li>ChatGPT voi antaa uskottavan kuuloisia, virheellisiä vastauksia</li> </ul>

Lopullinen sivu julkaistiin 15.3.2024 osoitteessa [www.opitehokkaasti.com](http://www.opitehokkaasti.com) ja on vuoden verran näkyvillä, kunnes domainin sopimus tulee päätökseen.

## 8 Johtopäätökset ja pohdinta

Tämän opinnäytetyön tavoitteena oli selvittää, miten ChatGPT voi tukea Reactin oppimisprosessia, miten kommunikoida tehokkaasti ChatGPT:n kanssa, sekä mitä haasteita ChatGPT:n käytössä opiskelussa voi ilmetä. Tutkimuksen myötä havaittiin, että ChatGPT tarjoaa merkittävää tukea Reactin oppimisessa, muun muassa tarjoamalla reaaliaikaisia vastauksia kysymyksiin, selkeitä esimerkkejä ja mahdollisuuden syventää ymmärrystä iteratiivisella keskustelulla ja harjoitustehtävien avulla. Alussa ChatGPT:n kyky tuottaa oppimateriaalia osoittautui varsin kehittyneeksi. Aiheiden muuttuessa vaativammiksi, huomattiin nopeasti, että sen suorituskyky olikin rajallisempi ja arvokasta oppimisaikaa valui yrityksiin generoida tarvittavia opiskelumateriaaleja. Tehokas kommunikointi ChatGPT:n kanssa vaatii selkeiden ja täsmällisten kysymysten esittämistä, jotta saadaan mahdollisimman relevantteja ja hyödyllisiä vastauksia. Tämä edellyttää käyttäjältä kykyä muotoilla kehoitteensa tavalla, joka ohjaa tekoälyä oikeaan suuntaan.

Tutkimus toi esiin haasteiden esiintyvyydestä, jotka liittyvät erityisesti teknisiin rajoitteisiin, kuten generoitavien vastausten määrän rajoitukseen sekä tekoälyn kykyyn ymmärtää monimutkaisia ohjelmointiongelmia. Nämä haasteet vaativat käyttäjiltä lisäperehtymistä aiheisiin muista lähteistä ja kriittistä suhtautumista ChatGPT:n tarjoamaan informaatioon.

Päiväkirjamerkintöjen avulla kerätty aineisto syvensi ymmärrystä ChatGPT:n käytöstä Reactin oppimisprosessissa. Tutkimuksen validiteettia ja reliabiliteettia vahvistivat menetelmien huolellinen valinta ja niiden soveltaminen, mahdollistaen tutkimuskysymysten perusteellisen tarkastelun. Vaikka ChatGPT:n päivitykset ja muutokset voivat ajan mittaan vaikuttaa tuloksiin, käytetyt menetelmät ja analyysiprosessit tukivat tulosten johdonmukaisuutta.

Opinnäytetyöprosessi osoitti, että ChatGPT:n käyttö opiskelun tukena on arvokas työkalu, mutta sen tehokkuus riippuu suuresti käyttäjän taidoista ja kyvystä työskennellä tekoälyn rajoitteiden puitteissa. Tulevaisuudessa on tärkeää kehittää strategioita, jotka mahdollistavat vielä tehokkaamman yhteistyön ihmisen ja tekoälyn välillä, erityisesti koulutuksen osa-alueella.

Jatkosuunnitelman ideana olisikin tehdä laajempi tutkimus, jossa otettaisiin huomioon useampien opiskelijoiden kokemukset. Tutkimuksen tavoitteena olisi syventää ymmärrystä siitä, miten ChatGPT ja muut tekoälytyökalut voivat tukea oppimista erilaisissa

oppimisympäristöissä. Erityisesti mielenkiintoista olisi tietää, miten nämä työkalut voisivat suuremmassa mittakaavassa auttaa oppijoita omaksumaan uusia taitoja, tarjota yksilöllisiä oppimiskokemuksia ja vastata oppimisen haasteisiin. Henkilökohtaisten kokemusten perusteella on selvää, että tekoälyn hyödyntämisellä on suuri potentiaali oppimisen tehostamisessa, mutta tämän potentiaalin täysimääräinen hyödyntäminen vaatii lisää tutkimusta ja ymmärrystä.

Tämä opinnäytetyö on antanut arvokkaita ajatuksia siitä, miten tekoäly voi tukea ja tehostaa oppimista, mutta myös muistuttanut strukturoitujen ja ammattilaisten suunnitteleminen opintokokonaisuuksien ja opettajien tärkeydestä opetuksessa. Tekoäly tarjoaa työkalun oppimisprosessiin, mutta sen tehokas hyödyntäminen edellyttää työkalun tehokasta käyttöä sekä ymmärrystä omasta oppimistyylistä.

## 9 Yhteenveto

Tässä opinnäytetyössä keskityttiin tutkimaan, kuinka suuren kielimallin tekoälyohjelmaa, kuten ChatGPT:tä voidaan hyödyntää tehokkaasti Reactin oppimisprosessissa sekä haasteita, joita ChatGPT:n käyttöön liittyy.

Tutkimuskysymyksiin vastaaminen onnistui kattavasti. Analyysin ja konkreettisten harjoitusten kautta osoitettiin, että ChatGPT tarjoaa merkittävän tuen ohjelmoinnin oppimiselle, antaen käyttäjälleen sekä teoreettista ymmärrystä että käytännön taitoja. Toisaalta haasteilta ei välttytty ja ChatGPT:n käyttöön liittyy edelleen teknisiä haasteita sekä riskejä sen väärinkäyttöön. Laadullisessa analyysissä korostui, että ChatGPT:n tehokas hyödyntäminen vaatii harkittua kommunikointia, mutta taitojen karttuessa tekoälyn avulla voidaan nopeuttaa oppimisprosessia ja syventää ymmärrystä ohjelmoinnin eri osa-alueista.

Henkilökohtaisella tasolla opinnäytetyön myötä oma ymmärrykseni Reactista ja ohjelmoinnin periaatteista on syventynyt huomattavasti. Olen oppinut, että tekoälyä voi käyttää luovasti oppimisen tukena, ja se voi tarjota oivalluksia ja ratkaisuja, joita ei välttämättä tulisi muuten ajatelleeksi. Erityisesti ChatGPT:n kyky käsitellä monenlaisia ongelmia ja tarjota yksilöllistä ohjausta on osoittautunut arvokkaaksi. Näiden ansiosta koen, että kynnykseni kehittää osana oppimisprosessia Reactilla toteutettu staattinen verkkosivu alun opiskelusta julkaisuun asti oli pienempi kuin ilman ChatGPT:tä. ChatGPT toimi koko matkan oppaana ja jo pelkästään tieto siitä, että apu on tarvittaessa lähellä, madalsi kynnystäni tarttua tähän haasteeseen.

Tulevaisuuden näkymät tekoälyn hyödyntämisestä oppimisessa ovat kiehtovia ja odotan innolla, mihin kaikkeen tulevaisuuden GPT:t ja muut tekoälysovellukset pystyvät. Vaikka tänä päivänä ChatGPT ei korvaa suunniteltuja kursseja tai opettajia niin on kuitenkin selvää tämän opinnäytetyön perusteella, että tekoälyn rooli opetuksessa ja oppimisessa tulee kasvamaan. Tulevaisuudessa voimme nähdä vieläkin syvällisempiä ja personoidumpia oppimiskokemuksia, kun tekoälyn kehitys jatkuu ja ymmärryksemme sen mahdollisuuksista syvenee.

Yhteenvetona voidaan todeta, että tämän opinnäytetyön myötä ei ainoastaan saavutettu syvempää ymmärrystä Reactista ja ohjelmoinnista, vaan myös avattiin uusia näkökulmia tekoälyn hyödyntämiseen oppimisen tukena. Se tarjoaa vankan perustan niin minulle itselleni kuin muillekin aiheesta kiinnostuneille, jotka haluavat hyödyntää ChatGPT:tä tai muita tekoälytyökaluja osana omaa oppimispolkuaan.

## Lähteet

Aaltonen, M. (2019). *Tekoäly. Ihminen ja kone*. Alma Talent.

Ausat, A. M. A., Massang, B., Efendi, M., Nofirman, & Riady, Y. (2023). Can ChatGPT Replace the Role of the Teacher in the Classroom: A Fundamental Analysis. *Journal on Education*, 5(4), 16100-16106. <https://jonedu.org/index.php/joe/article/view/2745>

Becker, B. A., Denny, P., Finnie-Ansley, J., Luxton-Reilly, A., Prather, J. & Santos, E. A. (2023). Programming Is Hard – Or at Least It Used to Be: Educational Opportunities and Challenges of AI Code Generation. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education*, 1(500-506). <https://doi.org/10.1145/3545945.3569759>

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). *Language models are few-shot learners*. arXiv:2005.14165v4 [cs.CL].  
<https://doi.org/10.48550/arXiv.2005.14165>

Chincha, D. (2023). *Number of ChatGPT Plugins (2023)*. Whatplugin.ai. Haettu 6.12.23 osoitteesta <https://www.whatplugin.ai/blog/chatgpt-plugins>

Coeckelberg, M. (2021). *Tekoälyn etiikka* (K. Pietiläinen, käänt.). Terra Cognita.  
(Alkuperäisteos julkaistu 2020)

Davis, V. (2023). *How to Make Your Own GPT: A Step-by-Step Guide*. Cool Cat Teacher.  
<http://tinyurl.com/2dhn8487>

DeepLearning.AI. (2023). *A COMPLETE GUIDE TO Natural Language Processing*. Haettu 2.12.2023 osoitteesta <http://tinyurl.com/3nfd6ry6>

Dell'Acqua, F., McFowland III, E., Mollick, E., Lifshitz-Assaf, H., Kellogg, K. C., Rajendran, S., Krayner, L., Candelon, F. & Lakhani, K. R. (2023). *Navigating the Jagged Technological*

*Frontier: Field Experimental Evidence of the Effects of AI on Knowledge Worker Productivity and Quality*. Harvard Business School. <http://tinyurl.com/2dcyjcs5>

Du Plessis, N. (vetäjä). (16.2.2024). Microsoft 365 Copilot: edistyneet toiminnallisuudet [webinaari]. Eduhouse. <https://tinyurl.com/4w4v84hw>

Elastic. (n.d.). *What is large language model (LLM)?* <https://www.elastic.co/what-is/large-language-models>

Elements of AI. (n.d.-a). *Tekoälyn filosofiaa*. <https://course.elementsofai.com/fi/1/3>

Elements of AI. (n.d.-b). *Miten neuroverkkoja rakennetaan?* <https://course.elementsofai.com/fi/5/2>

Euroopan komissio. (2021). *Euroopan parlamentin ja neuvoston asetukset. Tekoälyä koskevista yhdenmukaistetuista säännöistä (tekoälyäädös) ja tiettyjen unionin säädösten muuttamisesta*. Ehdotus asetuksesta. <http://tinyurl.com/yus7nuhu>

Euroopan parlamentti. (2023a). *EU:n tekoälyäädös on ensimmäinen laatuaan*. Haettu 23.10.2023 osoitteesta <http://tinyurl.com/4k9s4hbp>

Euroopan parlamentti. (2023b). *Mitä tekoäly on ja mihin sitä käytetään?* Haettu 22.10.2023 osoitteesta <http://tinyurl.com/4ubi7pxf>

Fulford, I. & Ng, A. (vetäjät). (n.d.). *ChatGPT Prompt Engineering for Developers*. [video]. DeepLearning.AI. <https://learn.deeplearning.ai/chatgpt-prompt-eng/lesson/1/introduction>

Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Machine Learning Basics*. Teoksessa I. Goodfellow, Y. Bengio & A. Courville (toim.), *Deep Learning* (ss. 96-161). MIT Press. <https://www.deeplearningbook.org>

Günther, K., Hasanen, K. & Juhila, K. (n.d.) Johdanto: Analyysi ja tulkinta. Teoksessa J. Vuori (toim.), *Laadullisen tutkimuksen verkkokäsikirja*. Tampere: Yhteiskuntatieteellinen tietoarkisto. <http://tinyurl.com/57y6efk2>

Herbert, D. (2023). *What is React.js? Uses, Examples, & More*. HubSpot. <https://blog.hubspot.com/website/react-js>

IBM. (n.d.). *What is Deep Learning?* <https://www.ibm.com/topics/deep-learning>

Indla, B. V. S. & Puranik, Y. (2021). Review on React JS. *International Journal of Trend in Scientific Research and Development (IJTSRD)*, 5(4), 1137-1139.

[www.ijtsrd.com/papers/ijtsrd42490.pdf](http://www.ijtsrd.com/papers/ijtsrd42490.pdf)

Iqbal, U., Kohno, T. & Roesner, F. (2023). *LLM platform security: Applying a systematic evaluation framework to OpenAI's ChatGPT plugins*. arXiv:2309.10254 [cs.CR].

<https://doi.org/10.48550/arXiv.2309.10254>

Johnson, J. (2020). *What's a Deed Neural Network? Deep Nets Explained* [kuva]. Bmc.

<https://www.bmc.com/blogs/deep-neural-network/>

Juhila, K.. (n.d.-a). Laadullisen tutkimuksen ominaispiirteet. Teoksessa J. Vuori (toim.), *Laadullisen tutkimuksen verkkokäsikirja*. Tampere: Yhteiskuntatieteellinen tietoarkisto.

<http://tinyurl.com/mwbrre4c>

Juhila, K. (n.d.-b). Teemoittelu. Teoksessa J. Vuori (toim.), *Laadullisen tutkimuksen verkkokäsikirja*. Tampere: Yhteiskuntatieteellinen tietoarkisto.

<http://tinyurl.com/2wtn249b>

Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., Krusche, S., Kutyniok, G., Michaeli, T., Nerdel, C., Pfeffer, J., Poquet, O., Sailer, M., Schmidt, A., Seidel, T., ... & Kasneci, G. (2023). *ChatGPT for good? On opportunities and challenges of large language models for education*.

<https://doi.org/10.1016/j.lindif.2023.102274>

Kelleher, J. (2020). *Syväoppiminen* (K. Pietiläinen, käant.) [Kuva]. Terra Cognita.

(Alkuperäisteos julkaistu 2019)

Kelleher, J. (2020). *Syväoppiminen* (K. Pietiläinen, käant.). Terra Cognita. (Alkuperäisteos

julkaistu 2019)

Kolari, J. & Kallio, A. (2023). *Tekoäly 123*. Docento.

Kukkurainen, M L. (2019). *Autoetnografia – päiväkirjaan perustuva tutkimus*. LAMK Pro.

<https://www.lamkpub.fi/2019/01/04/autoetnografia-paivakirjaan-perustuva-tutkimus/>



Liu, G. (2023). *Perspectives on the social impacts of reinforcement learning with human feedback*. <https://doi.org/10.48550/arXiv.2303.02891>

Lutkevich, B. (2023). *Custom GPT:s: Examples and how to build*. TechTarget. <http://tinyurl.com/22zczd62>

Malekos, N. (2023). *A.I. Course Creation: How to Use ChatGPT to Create eLearning Content*. LearnWorlds. <http://tinyurl.com/yw282fdy>

Matthias, A. (2021). *What is Artificial Intelligence?* Daily Philosophy. <https://daily-philosophy.com/what-is-artificial-intelligence/>

Meyer, J. G., Urbanowicz, R. J., Martin, P. C. N., O'Connor, K., Li, R., Peng, P.-C., Bright, T. J., Tatonetti, N., Won, K. J., Gonzalez-Hernandez, G. & Moore, J. H. (2023). ChatGPT and large language models in academia: opportunities and challenges. *BioData Mining*, 16(20). <https://doi.org/10.1186/s13040-023-00339-9>

Mhlanga., D. (2023). *Open AI in Education, the Responsible and Ethical Use of ChatGPT Towards Lifelong Learning*. <http://dx.doi.org/10.2139/ssrn.4354422>

Mykkänen, J. (n.d). *Tutkimusmenetelmät ja tutkimusaineistot*. <https://www.mv.helsinki.fi/home/jmykkane/tutkielma/Tutkimusmenetelmat.html>

Myllymäki, P. (2021). Tekoälyn älykkyydestä. Teoksessa Gaudeamuksen työryhmä (toim.), *Älykäs huominen* (ss. 9–31). Gaudeamus Oy.

Niemi, H. (2021). Miten tekoäly muuttaa oppimista ja koulutusta? Teoksessa Gaudeamuksen työryhmä (toim.), *Älykäs huominen* (ss. 130–147). Gaudeamus Oy.

Niemi, H., Pea, R. D., & Lu, Y. (2023). Introduction to AI in Learning: Designing the Future. Teoksessa Y.Lu (toim.), *AI in Learning: Designing the Future* (ss. 1–15). Springer. [https://doi.org/10.1007/978-3-031-09687-7\\_1](https://doi.org/10.1007/978-3-031-09687-7_1)

Niemi, V. (2021). Yksityisyys tekoälyn aikakaudella. Teoksessa Gaudeamuksen työryhmä (toim.), *Älykäs huominen* (ss. 114–129). Gaudeamus Oy.

Ojanperä, T. (2023). *Tekoälyn vallankumous – Käsikirja*. Alma Talent.

Opara, E. C., Adalikwu, M.-E. T. & Tolorunleke, C. A. (2023). ChatGPT for teaching, learning and research: Prospects and challenges. *Global Academic Journal of Humanities and Social Sciences*, 5(2), 33-40. <https://doi.org/10.36348/gajhss.2023.v05i02.001>

OpenAI. (n.d.). *Prompt engineering*. <https://platform.openai.com/docs/guides/prompt-engineering>

OpenAI. (2019). *Better language models and their implications*. <https://openai.com/research/better-language-models>

OpenAI. (2022). *Introducing ChatGPT*. <https://openai.com/blog/chatgpt>

OpenAI. (2023a). *ChatGPT plugins*. <https://openai.com/blog/chatgpt-plugins>

OpenAI. (2023b). *Introducing GPTs*. <https://openai.com/blog/introducing-gpts>

OpenAI. (2024). *ChatGPT — Release Notes*. Haettu 7.1.2024 osoitteesta <http://tinyurl.com/3nnykpzi>

Palojoki, P. (1997). Laadullisen päiväkirjan käyttö terveystutkimuksessa. *Hoitotiede*, 9(3), 127-135. <http://tinyurl.com/9wcz7sct>

Du Plessis, N. (vetäjä). (16.2.2024). *Microsoft 365 Copilot: edistyneet toiminnallisuudet* [webinaari]. Eduhouse. <https://tinyurl.com/4w4v84hw>

PromptPerfect. (n.d.). *Prompt Engineering at its finest: Design. Optimize. Deploy. Instantly*. Haettu 13.12.2013 osoitteesta <https://promptperfect.jina.ai/>

Ravikiran, A.S. (2022). *A Comprehensive Look at React Hooks*. Haettu 20.2.2024 osoitteesta <https://www.simplilearn.com/tutorials/reactjs-tutorial/reactjs-hooks>

React. (n.d.-a). *JavaScript in JSX with Curly Braces*. Haettu 14.2.2024 osoitteesta <https://react.dev/learn/javascript-in-jsx-with-curly-braces>

React. (n.d.-b). *Passing Props to a Component*. Haettu 21.2.2024 osoitteesta <https://react.dev/learn/passing-props-to-a-component>

React. (n.d.-c). *Sharing State Between Components*. Haettu 20.2.2024 osoitteesta <https://react.dev/learn/sharing-state-between-components>

React. (n.d.-d). *State: A Component's Memory*. Haettu 15.2.2024 osoitteesta <https://react.dev/learn/state-a-components-memory>

React. (n.d.-e). *Using Hooks*. Haettu 20.2.2024 osoitteesta <https://react.dev/learn#using-hooks>

React. (n.d.-f). *Writing Markup with JSX*. Haettu 14.2.2024 osoitteesta <https://react.dev/learn/writing-markup-with-jsx>

Salo, I. (2023). *Luova tekoäly mullistaa kaiken*. Kauppakamari.

Siukonen, T. & Neittaanmäki, P. (2019). *Mitä tulisi tietää tekoälystä?* Docendo Oy.

Tarkoma, S. (2021). Miten tekoäly vaikuttaa kokonaisturvallisuuteen. Teoksessa Gaudeamuksen työryhmä (toim.), *Älykäs huominen* (ss. 93–112). Gaudeamus Oy.

Thorp, H. H. (2023). ChatGPT is fun, but not an author. *Science*, 379(6630), 313. <https://doi.org/10.1126/science.adg7879>

Tiedekulma. (4.10.2021). *Tarvitseeko tekoäly sääntöjä?* [video]. Youtube. <https://www.youtube.com/watch?v=mfkNrPTZspw>

Tietosuojavaltuutetun toimisto. *Mikä on henkilötieto?* Tietosuojavaltuutetun Toimisto. <https://tietosuoja.fi/mika-on-henkilotieto>

Turing. (n.d.). *What Is the Necessity of Bias in Neural Networks?* <https://www.turing.com/kb/necessity-of-bias-in-neural-networks>

Työ- ja elinkeinoministeriö. (2017). *Suomen tekoälyaika. Suomi tekoälyn soveltamisen kärkimaaksi: Tavoite ja toimenpidesuosituksset*. <http://tinyurl.com/bdfd37v6>

UX 24/7. (2019). *The Advantages and Limitations of Diary Studies*. <http://tinyurl.com/56jrnkun>

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need*. arXiv:1706.03762v7 [cs.CL].

<https://doi.org/10.48550/arXiv.1706.03762>

Whatplugin.Ai. (2024). *Best GPTs for Academic Research With ChatGPT (2024)*. Haettu 8.2.2024 osoitteesta <https://www.whatplugin.ai/category/academic-research>

Wu, X., Duan, R. & Ni, J. (2023). *Unveiling security, privacy, and ethical concerns of ChatGPT*. Journal of Information and Intelligence. <https://doi.org/10.1016/j.jiixd.2023.10.007>

**Liite 1: Aineistonhallintasuunnitelma****Kehitysprojekti:**

Kehitysprojektin aikana pidetään päiväkirjaa (aineisto), johon kerätään kokemuksia ja tietoa opinnäytetyöprojektista. Tämä tieto analysoidaan opinnäytetyötä varten. Päiväkirjaa säilytetään tekijän tietokoneen C-asemalla, ja siitä tehdään säännöllisesti varmuuskopioita muistitikulle. Päiväkirjaa säilytetään C-asemalla ainakin vuoden verran opinnäytetyön valmistumisesta, minkä jälkeen se poistetaan ja tyhjennetään myös muistitikku.

**Opinnäytetyöaineiston jatkokäyttö työn valmistumisen jälkeen**

Tutkimusaineistoa ei jatkokäytetä. Opinnäytetyön tekijä säilyttää aineiston tietoturvallisesti vuoden ajan opinnäytetyön hyväksymispäivästä, jotta opinnäytetyön tulokset voidaan tarvittaessa varmistaa ja hävittää tämän jälkeen aineiston tietoturvallisesti.