

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Informaatika õppekava

**Grete Neilinn**

# **Veebisaidi loomise protsess ChatGPT4 abiga ning ilma**

Bakalaureusetöö

Juhendaja: Lidia Feklistova, PhD

Tartu 2023

## **Veebisaidi loomise protsess ChatGPT4 abiga ning ilma**

### **Lühikokkuvõte:**

Käesoleva bakalaureusetööga sooviti teada saada, kas üks tänapäeva võimsamatest vestlusrobotitest ChatGPT4 on võimeline looma veebisaiti ainult päringute põhjal ning kas vestlusrobot on piisavalt arenenud, et asendada veebiarendajaid. ChatGPT4 jõudluse parema arusaamise nimel loodi paralleelselt kaks veebisaiti prototüübi järgi, üks ChatGPT4 abita ja teine ChatGPT4-ga, ning mõlemad veebisaidid pidid vastama prototüübi lehekülgede nõuetele. Tulemusena leiti, et ChatGPT4 on suuteline looma prototüübi nõuetele vastava veebisaidi, kuid vestlusrobotil oli mitmeid vigu. Nimelt ei olnud vestlusroboti vastused järjepidevad. Lisaks oli ChatGPT4-l raskusi vigade leidmisega ja koodi lühemaks muutmisega. Vaatamata sellele, on tegemist väga võimsa vestlusrobotiga. ChatGPT4, aga ei ole piisavalt võimas, et asendada täielikult veebiarendajaid. Kõige efektiivsem viis veebisaitide loomiseks oleks kasutada nii veebiarendajaid kui ka ChatGPT4 koos.

### **Võtmesõnad:**

Veebiarendus, veebisaidi disain, vestlusrobotid, ChatGPT4

**CERCS:** P175 Informaatika

## **Website Development Process With ChatGPT4 and Without**

### **Abstract:**

The aim of this bachelor's thesis was to find out, whether ChatGPT4 is capable of creating a website by only answering queries, and whether the chatbot is advanced enough to replace web developers. In order to better understand the performance of ChatGPT4, two websites were created in parallel, one without the help of ChatGPT and the other with ChatGPT4, and both websites had to meet the requirements of the prototype. As a result, ChatGPT4 was found to be able to create a website that met the requirements of the prototype, but the chatbot had several faults. Namely, the answers of the chatbot were not consistent. Additionally, ChatGPT4 struggled to find bugs and ways to shorten the code. Despite this, it is a very powerful chatbot. ChatGPT4, however, is not powerful enough to completely replace web developers. The most efficient way to create websites would be to use both web developers and ChatGPT4 together.

### **Keywords:**

Web development, website design, chatbots, ChatGPT4

**CERCS:** P175 Informatics

# Sisukord

<b>Sissejuhatus</b>	<b>4</b>
<b>1. AI vestlusrobotid</b>	<b>5</b>
1.1. Vestlusrobotid	5
1.2. ChatGPT	6
1.2.1. OpenAI	6
1.2.2. ChatGPT4	6
1.2.3. Varasemad uuringud	7
<b>2. Veebisaitide loomine</b>	<b>8</b>
2.1. Disain	8
2.2. Prototüüp ja nõuded	10
2.3. Arendus vestlusroboti abita	16
2.3.1. Tehnoloogia valik	16
2.3.2. Arendus	17
2.4. Arendus ChatGPT4 abil	20
2.4.1. Protsess	20
2.4.2. Arendus	20
<b>3. Võrdlus</b>	<b>26</b>
3.1. Koodi võrdlus	26
3.2. Arendusprotsessi võrdlus	28
<b>4. Tulemus</b>	<b>30</b>
<b>5. Võimalikud arengusuunad</b>	<b>32</b>
<b>Kokkuvõte</b>	<b>33</b>
<b>Kasutatud kirjandus</b>	<b>34</b>
<b>Lisad</b>	<b>38</b>
Lisa 1. Prototüübi täispikkuses pildid	38
Lisa 2. Päring navigatsioonirea loomiseks kasutades HTML ja CSS'i	40
Lisa 3. Päring Vue.js navigatsioonirea loomiseks	40
Lisa 4. Päring Vue.js avalehe loomiseks	41
Lisa 5. Päring Vue.js tutvustuse sektsiooni loomiseks	41
Lisa 6. ChatGPT4 vastus töötava animatsiooni koodiga tutvustuse sektsiooni jaoks	42
Lisa 7. Päring Vue.js kaartide sektsiooni loomiseks	44
Lisa 8. Päring Vue.js jaluse loomiseks	45
Lisa 9. ASCII kood Vue.js galerii loomiseks	46
Lisa 10. Esimene päring Vue.js galerii loomiseks	47
Lisa 11. ASCII kood Vue.js kontakti loomiseks	47
<b>Litsents</b>	<b>48</b>

# Sissejuhatus

Internet on lahutamatu osa inimeste elust. Seda kasutab üle poole inimkonnast ning see on peamine ühenduse, hariduse ja meelelahutuse saamise meetod maailmas. Interneti kasutajate arv kasvab pidevalt ning selle tagajärjena on ka veebiarendus kiiresti arenev tööstusharu (Decker, 2022). Kuid tehisintellekti kiire areng tekitab palju küsimusi inimeste seas - kas see on võimeline asendama töökohti, kas tehisintellekt on ohtlik, mida suudab tehisintellekt teha jne.

Hetkeseisuga puuduvad uuringud ChatGPT4 mudeli rakendamisest veebiarenduses. Käesolev töö üritab kõrvaldada seda lünka. Töös uuritakse, kas ChatGPT4 on suuteline arendama veebisaiti täielikult päringute põhjal ning seejärel võimeline asendama veebilehtede arendajaid. ChatGPT4 jõudluse parema arusaamise nimel luuakse paralleelselt kaks veebilehte prototüübi järgi, üks ChatGPT4 abita ja teine ChatGPT4-ga, ning hiljem võrreldakse tulemusi.

Bakalaureusetöö koosneb 5 peatükist: esimeses peatükis antakse ülevaade vestlusrobotitest ning ChatGPT neljandast versioonist; teises peatükis tuuakse esile veebilehe disaini head tavad, kirjeldatakse prototüüpi ja arenduse nõudeid kui ka arendusprotsesse; kolmandas peatükis võrreldakse ChatGPT abita ning ChatGPT4 poolt loodud veebisaite; neljandas peatükis esitatakse tulemused; viiendas peatükis tuuakse välja antud töö võimalikud arengusuunad.

# 1. AI vestlusrobotid

## 1.1. Vestlusrobotid

Vestlusrobot (ingl *chatbot*) on arvutiprogrammid, mis töötavad tehisintellekti ja loomuliku keeletöötuse baasil (IBM, n.d.). Vestlusroboteid kasutatakse tihti IT-teenuste kogemuse täiustamiseks (Oracle, n.d.). Näiteks on populaarne kasutada vestlusroboteid iseteeninduse võimaldamiseks ning kliendikontaktikeskustes sissetuleva suhtluse suunamiseks õigetele ressurssidele (IBM, n.d.), sest vestlusrobotid on suutelised pakkuma tuge 24/7 (Frankenfield, 2022). Vestlusrobotitel on ka palju teisi kasutamismõimalusi ning neid on implementeeritud paljudesse valdkondadesse. Näiteks kasutatakse vestlusroboteid sotsiaalmeedia vestlusrakendustes, erinevates äriprotsessides ning isegi nutikates köögiseadmetes (IBM, n.d.). Üks tuntud vestlusrobot on Siri, Apple poolt loodud virtuaalne assistent, mis kasutab lisaks tehisintellekti elementidele ka hääletuvastuse tarkvara (Frankenfield, 2022).

Digitaliseeritud ühiskonnas, kus sõnumirakenduste populaarsus kasvab, mängivad vestlusrobotid üha suuremat rolli (Oracle, n.d.), sest vestlusrobotid hõlbustavad teabe leidmist vastates kasutajate küsimustele ja taotlustele ilma inimese sekkumiseta (IBM, n.d.). Seetõttu on ka paljud firmad implementeerinud vestlusrobotid klientidega suhtlemiseks. Kuigi paljude klientide küsimused on korduvad ja lihtsad, mida saaks vastata ettevõtte KKK sektioonis, pakuvad vestlusbotid personaalset kliendikogemust (Oracle, n.d.), tänu mille paraneb klientide brändilojaalsus (IBM, n.d.). Samuti on vestlusbotide kasutamine pikemas perspektiivis odavam võrreldes inimeste palkamisega, aga see-eest vestlusbotid ei pruugi alati kliendi küsimustest aru saada ning võivad tagajärjena kliendile valesti vastata (Frankenfield, 2022).

Ajalooliselt olid vestlusrobotid tekstipõhised, kus nad olid programmeeritud vastama limiteeritud päringutele (IBM, n.d.). Kuid tänu loomulikule keeletöötusele NPL (ingl *natural language processing*) on tänapäeval vestlusrobotid võimelised tõlgendama inimkeelt väljaspool eelprogrammeeritud käske ning seejärel pidama inimsarnaseid vestlusi (Drift, n.d.). Täpsemalt käsitleb antud keeleteadus inimese ja arvutite suhtluse arendamisega ehk inimkeelte nagu eesti, inglise ja läti keele muutmisega masinakeelde, mis koosneb tuhandetest erinevatest numbrite 0 ja 1 kombinatsioonidest (Günceleme, 2022). Hea näide loomuliku keeletöötuse kasutusest on Google Translate. Samuti kasutavad vestlusrobotid tehisintellekti elemente, nagu masinõpet ning sügavõpet (ingl *deep learning*), et laiendada

programmi küsimuste ja vastuste teadmisaasi, mis võimaldab chatpotidel efektiivsemalt ning täpsemalt kasutajate vajadusi ennustada ning vastavalt reageerida (IBM).

## 1.2. ChatGPT

### 1.2.1. OpenAI

ChatGPT arendaja OpenAI on tehisintellektiga tegelev ettevõte. OpenAI sai alguse 2015. aastal kui mittetulunduslik tehisintellekti uurimisettevõte, mille arendus oli avatud lähtekoodiga - sealt tuligi nimi OpenAI (Harrison, 2023). Nüüdseks on firma tulu teeniv eraettevõte suletud lähtekoodiga (Harrison, 2023), kuid inimestel on võimalik ligipääseda ChatGPT-3 API-le tasuta, mis on keelemooduli pakitud ja modereeritud versioon, ning ChatGPT-4 API-le läbi ootenimekirja (Chowdhury, 2023).

OpenAI eesmärk on algusest peale olnud luua AGI ehk generaalne tehisintellekt (ingl *artificial general intelligence*) (Altman, 2023). Täpsemalt on nende missioon luua AGI, mis toob kasu kogu inimkonnale. Generaalne tehisintellekt on hüpoteetiline arvutiprogramm, mis on suuteline täitma ülesandeid samal tasemel või paremini kui inimesed (Hodson, 2019). Generaalne tehisintellekt AGI ja tehisintellekt AI erinevad intelligentsuse taseme poolest. AGI on mõeldud lahendama üldise intelligentsusega seotud ülesandeid ning kohanema uute olukordadega. See-eest AI on loodud töötama kindlate parameetrite piires ning täitma konkreetseid ülesandeid (Atuhaire, 2023). Siiani ei ole keegi suutnud luua generaalset tehisintellekti (Avcontentteam, 2023).

### 1.2.2. ChatGPT4

ChatGPT4 on OpenAI poolt loodud vestlusrobot, mis töötab närvivõrgu abil ja on treenitud tohutul hulgal andmemahitudel. Tänu suurele andmemahule, millega on vestlusroboti mudel eelkoolitatud, suudab ChatGPT4 aru saada ning toota loomulikku keelt. Neljanda versiooni vestlusrobot saab paremini aru kasutaja kavatsustest võrreldes varasemate ChatGPT mudelitega (Shen, 2023). Samuti pakub ChatGPT4 täpsemat väljundit, omab suuremat mälu ning saab aru rohkematest keeltest (Shen, 2023; Shillsalot, 2023). ChatGPT4 kasutamiseks on vaja ChatGPT Plus kontot, mis maksab ligikaudu 22€ kuus. Samuti on vestlusrobotil päringute saatmisel piirang - saab saata 25 päringut 3 tunni jooksul.

### 1.2.3. Varasemad uuringud

ChatGPT4 avaldati 13. märtsil 2023 ning töö kirjutamise hetkel puuduvad uuringud ChatGPT4 võimest luua veebilehti. See-eest leidub uuringuid ChatGPT varasemate mudelite kohta. Tian et al. (2023) läbi viidud uuringus uuriti ChatGPT koodi genereerimise ja parandamise oskust. Koodi genereerimise testimiseks kasutati Leetcode'i andmestikku ning tulemusena leiti, et ChatGPT saab hakkama kõikide Leetcode'i kergete ülesannetega ning 75% raskete ülesannetega. Eeldades, et enamus Leetcode'i koodilahendused on lisatud inimeste poolt, leiti, et ajalise keerukuse poolest on ChatGPT kood potentsiaalselt parem kui enamike inimeste kirjutatud kood. Samuti näitasid tulemused, et päringu pikkus mängib rolli koodi korrektsuses. Nimelt, mida pikem on päring, seda suurem on tõenäosus, et koodis leidub viga.

Qureshi (2023) poolt läbiviidud uuringus näitasid tulemused, et ChatGPT on tõhus juhendaja õppeprotsessis tänu kohandatud juhiste. Samas leiti, et ChatGPT-l on piiranguid. Nimelt uuringu käigus oli koodi väljundis ebatäpsusi, mille sagedus tõusis ülesannete raskusastmega, ning vestluroboti poolt genereeritud kood ei olnud järjekindel.

## 2. Veebisaitide loomine

### 2.1. Disain

Iga hea veebiarendaja järgib arendamisstandardeid nagu asjakohaste nimedega muutujate kasutamine, koodi segmenteerimine, taanete lisamine, jne., et kirjutada loetavat ja selget koodi (Bose, 2021). Ka veebisaidi disainimisel tuleb järgida teatud tavaid. Üks nendest tavadest on nutitelefonidele kohandatud vaade, kuna 60.67% veebisaitide liiklusest tuleb mobiiliseadetest ning 92.3% internetikasutajatest pääseb internetti mobiiltelefoni kasutades (Howarth, 2023). Veebisaidi disainimisel on tähtis planeerida veebisaidi visuaalset hierarhiat. Elementide paigutus aitab tõsta tähtsa informatsiooni esile ning suunata saidi külastajat intuitiivselt soovitud toiminguni (Juviler, 2021). Lisaks paigutusele on võimalik elemente esile tõsta nende värvi või suurust kohandades, kuid siinkohal peaks meeles pidama veebisaidi üldist välimust. Nimelt on hea tava kasutada ühtseid värvilahendusi ja fonte, sest veebisaidi järjepidevus mõjutab positiivselt kasutatavust (ingl *usability*) ning kasutajakogemust (ingl *user experience*) (Juviler, 2021).

Kuigi veebisaidi disainimisel on olemas head tavad, puuduvad sellel tehnilised standardid, mida iga veebisait peaks järgima. Crestiodina (2021) viis läbi uuringu, et leida, mida täpsemalt võib pidada veebidisaini standardiks ning mis on lihtsalt hea tava. Crestiodina võrdles 500 erineva veebisaidi disaini tunnusjooni ning kui tunnusjoont leidis rohkem kui 80% veebisaitides, siis pidas ta seda standardiks. Uuringu tulemus näitas, et väheseid tunnusjooni, nagu logo positsioon üleval vasakul ja põhi navigatsiooni positsioon päises, võib standardiks pidada ja isikupärastatud disain on domineerivam.

Esmamuljed on tähtsad, sest nad mängivad suurt rolli veebilehe külastaja lahkumises või jäämises. Uuringud on näidanud, et mõned esimesed arvamused kujunevad välja 17ms jooksul (Tuch et al., 2012), kuid enamasti veebisaidi esmamulje tekib 50 millisekundiga (Lindgaard et al., 2006; Tuch et al. 2012). Kuna 97% veebisaidi esmamuljetest on seotud disainiga (Sillence et al., 2004), on oluline tähelepanu pöörata veebisaidi kujundusele ja millise mulje see külastajatele jätab. On tähtis teada, kuidas kujunevad kasutajate esmamuljed veebisaidist. Varem mainitud Tuch et al. (2012) uuringus on välja toodud ka, kuidas visuaalne keerukus ja tüüpilisus (ingl *prototypicality*) mõjutavad esmamuljet. Nimelt madala visuaalse keerukusega ja kõrge tüüpilisusega veebisaiti peeti väga atraktiivseks. Silmade jälgimistehnoloogiat kasutades uuriti disaini elementide rolli esmamulje



kujundamisel (Sheng et al., 2013). Selle käigus analüüsiti osalejate tagasisidet disaini kohta ja silmade fikseerimise mustreid ning tulemused kategoriseeriti. Järeldused olid järgmised:

- eelistatakse heledate ning lõõgastavate värvide kasutamist;
- ebaoluliste ja mõttetute piltide kasutamine jätab negatiivse mulje;
- põhi navigatsioon peaks asuma veebilehe üleval või vasakul poolel. Palju navigeerimislinke veebisaidi allosas või paremal põhjustab segadust ja negatiivseid muljeid;
- kõrge kontrastsusega tekst on oluline paremaks lugemiseks. Teksti font ja suurus peavad olema loetavad;
- suure ja tähendusrikka pildi kasutamine avalehel/silmapaistvas kohas jätab positiivse mulje. Liiga palju pilte, linke ja teksti ei tõmba tähelepanu veebilehele;
- liiga palju vaba ala (ingl *white space*) veebilehel on tõrjuv, kuna pakuvad vähe sisu.

Veebisaidi disainimisel tuleb tähelepanu pöörata ka veebisaidi eesmärgile ja võimalike külastajate eelistustele (Baierova et. al., 2003). Nimelt inimesed, kes kasutavad veebisaiti meelelahutuseks, hindavad kõige kõrgemalt nauditavust (ingl *enjoyability*), juurdepääsetavust ja sisu kvaliteeti. See-eest inimesed, kes kasutavad veebisaiti informatsiooni saamiseks, peavad tähtsaks sisu kvaliteeti, navigeerimist, struktuuri ja juurdepääsetavust. Lisaks tuleks värvi valimisel veebisaidi eesmärki meeles pidada. Varem mainiti, et veebisaidi külastajad eelistavad heledaid ja lõõgastavaid värve (Sheng et al., 2013). Seda võib põhjendada värvi psühholoogiat kasutades. Nimelt heledad värvid nagu valge jätab positiivse ja avatud tunde (Turner, 2017); heledate värvustega veebisaite peeti usaldusväärsemaks kui tumeda režiimiga veebisaite (Nissen & Riedl, 2021). Lisaks heledatele värvidele on võimalik veebisaidi atraktiivsust tõsta ning eesmärki paremini edastada kasutades värvi psühholoogiat, sest värvid on seotud erinevate tähendustega (Tabel 1). Sellega arvestamine aitab disaineril valida õige värvipaletti, mis sobib kokku veebisaidi meeleoluga ning eesmärgiga. Näiteks on roheline värv assotsieeritud loodusega (UserTesting 2019), seega veebisaidid, mille eesmärk või produkt on seotud loodusega võiks kasutada rohelist värvi. Siin oleks hea näide RMK veebisait, mis on Eesti riigimetsa majandamise keskus, kus kasutatakse rohelist värvust ühe põhi värvina oma veebisaidis.

**Tabel 1.** Värvide tähendus ja sümboliseerimine

Värv	Tähendus (UserTesting, 2019)	Sümboliseerimine (Trajectory, n.d.)
punane	elevus, armastus, tugevus	kutsub tegutsema ja võib olla liiga stimuleeriv
oranž	vaprus, enesekindlus, edu	sümboliseerib liikuvust ja tekitab kiirsutamise tunde
kollane	loovus, õnn/õnnelikkus, soojus	soodustab entusiasmi ja tõstab emotsioone, kuid võib tekitada ärevust
roheline	loodus, värskus, kvaliteet, tervistav	sümboliseerib tasakaalu ja stabiilsust
sinine	usaldus, rahu, lojaalsus, pädevus	soodustab lõõgastumist ja väga mitmekülgne värv
roosa	kaastundlikkus, siirus, armas	stimuleerib magusaisu ja on ekslikult peetud naiste lemmikuks
lilla	luksuslikkus, kuninglikkus, ambitsioon	populaarne värv naiste seas ning sümboliseerib luksust
pruun	lihtne, usaldusväärne	sümboliseerib õueala/keskkonda (ingl <i>outdoors</i> ) ning mullasust
must	formaalne, dramaatiline, turvalisus, tarkus	ajatu ja klassikaline värv, aga võib olla liiga domineeriv
valge	puhas, aus, lihtsus	soodustab rahutunnet ning parim värvide rõhutamiseks

Siinkohal aga peaks meele pidama, et värvid ei tähenda ning tekita samu emotsioone igas inimeses. See, kuidas värv inimesi mõjutab, võib oleneda inimese iseloomust, kultuurist ja vanusest (Jue & Ha, 2022).

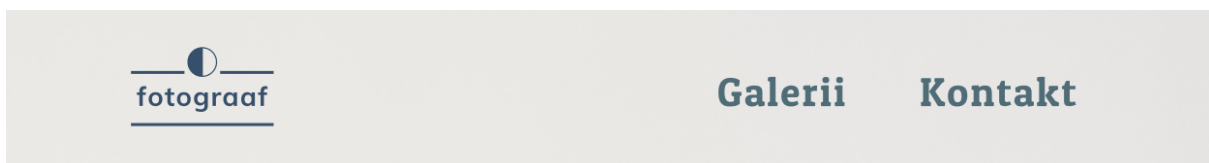
## 2.2. Prototüüp ja nõuded

Veebisait, mida bakalaureusetöö käigus arendatakse kahel erineval viisil, koosneb kolmest leheküljest: avaleht, galerii ja kontakt. Veebisaidi teemaks valiti fotograafia ning prototüübina loodi portfoolio oletatavale fotograafile. Figma valminud prototüübi (Lisa 1) loomisel võeti arvesse Sheng et al. (2013) uuringu tulemused, mille tõttu koosnes prototüübi värvipalett

neljast heledast värvist, valge (#FFFFFF), helesinine (#A7C4D4), helehall (#EAEDEF) ja tuhmvalge (#4F6C77), ning ühest tumedast kontrastsuse jaoks, tumesinine (#4F6C77). Kasutati siniseid ja valgeid toone, et jätta veebisaidist puhas, rahulik ja lihtne mulje. Veebisaidi põhifondina kasutati fonti Patua One (LatinoType) ning sekundaarse fondina kasutati Playfair Display (Sørensen). Figma kasutatakse teksti mõõtühikuks dp-d ehk tihedusest sõltumatuid piksleid (ingl *density-independent pixels*). Kahjuks antud mõõtühikut CSS keeles ei eksisteeri, seega järgnevates nõutes puuduvad teksti suurusega seotud nõuded. Arendamisel kasutatakse vabalt valitud teksti mõõtühikut (rem, px, protsendid vms) ning jälgitakse, et vastav tekst näeks prototüübiga sarnane.

Iga veebilehekülg koosneb päisest (joonis 1) ja jalusest (joonis 2). Päise loomisel jälgiti Crestiodina (2021) poolt välja toodud disaini standardeid, milleks olid: logo positsioon üleval vasakul ja põhi navigatsiooni positsioon päises. Päise nõuded on järgmised:

- ‘Galerii’ ja ‘Kontakt’ hex värvikoodid on #4F6C77;
- teksti font on Patua One (LatinoType);
- logo, ‘Galerii’ ja ‘Kontakt’ on klikitavad;
- logole klõpsates avaneb avaleht;
- tekstile ‘Galerii’ klõpsates avaneb galerii lehekülg;
- tekstile ‘Kontakt’ klõpsates avaneb kontakti lehekülg;
- kursori hõljumisel ‘Galerii’ või ‘Kontakt’ kohal muutub tekst heledamaks;
- olles galerii või kontakti leheküljel on vastava lehekülje tekst allajoonitud;
- avalehel olles ei ole midagi allajoonitud;
- mobiili versioonis asub logo vasakul ning parempoolne menüü asendatakse hamburgeri ikooni menüüga.



**Joonis 1:** Prototüübi päis logoga (Hatchful)

Jaluse nõuded on järgmised:

- tausta hex värvikood on #4F6C77;
- teksti font on Playfair Display (Sørensen);
- teksti pealse joone ja teksti hex värvikood on #FFFFFF;

- tekst ei ole klikitav;
- teksti positsioon on reageeriv;
- mobiili versioonis on tekst väiksem, aga asub kõrvuti.



**Joonis 2:** Prototüübi jalus

Avaleht koosneb maandumise vaateavast (joonis 3), tutvustuse sektsioonist (joonis 4) ja kaartide sektsioonist (joonis 5). Sheng et al. (2013) uuring näitas, et suure tähendusrikka pildi kasutamine jätab positiivse mulje. Seetõttu kasutati avalehe taustaks antud veebisaidi temaatikaga seotud pilti, mille autoriks on Karolina Grabowska (2020). Avalehel alla kerides on elemendid, nagu tekstid ja kaardid, animeeritud liikuma alt üles läbipaistvast olekust nähtavasse olekusse. Samuti kasutatakse avalehel SVG-failivormingut. See on XML-keelel põhinev vektor failivorming 2D piltide jaoks ning SVG-sid on võimalik renderdada mis tahes suuruses kvaliteeti kaotamata (Mozilla, 2023). Tänu sellistele omadustele on SVG-faile mugav kasutada reageeriva disaini puhul. Maandumise vaateava nõuded:

- teksti font on Patua One (LatinoType);
- avalehe pealkirja ja alusparagraafi hex värvikood on #4F6C77;
- pealkirja all olev joon on pealkirjaga sama pikk ning hex värvikood on #A7C4D4;
- avalehe avanedes liigub pealkiri ja paragraaf üles alt üles läbipaistvast olekust nähtavasse olekusse;
- mobiilses versioonis asuvad pealkiri ja alusparagraaf ekraani keskel.



**Joonis 3:** Maandumise vaateava.

Tutvustuse sektsiooni nõuded:

- tausta hex värvikood #A7C4D4;
- teksti värvi kood on #FFFFFF;
- paragraafi tekst on joondatud äärest-ääreni;
- pealkirja font on Patua One (LatinoType);
- paragraafi font on Playfair Display (Sørensen);
- sektsioon on horisontaalselt pooleks lõigatud, kus mõlema poole elemendid, pealkiri ja paragraaf, on vastavas lõigus nii horisontaalselt kui ka vertikaalselt tsentreeritud;
- mobiili versioonis on lõigud üksteise all ehk pealkiri asub paragraafi üleval ning elemendid on tsentreeritud;
- animeeritud alt üles liikuma ning nähtavust muutma alla kerides.



**Joonis 4:** Tutvustuse sektsioon

Kaartide sektsiooni nõuded:

- tausta hex värvikood on #FFFFFF;
- kaartide hex värvikood on #4F6C77;
- kaartide pealkirjade hex värvikood on #FFFFFF;
- pealkirjade font on Patua One (LatinoType);
- kaardil on ümardatud nurgad;
- kaardid on klikitavad;
- klikkimisel pööravad kaardid teise külje;
- kaardi tagumine pool koosneb samast kaardi pealkirjast, mis asub kaardi üleval keskel, ja paragraafist, mis on samuti tsentreeritud kaardi keskele;
- paragraafi font on Playfair Display (Sørensen);
- kaardi esikülje piltideks on SVG-d. Kasutatakse šampanja (SVG Repo - a), statiivi (SVG Repo - b) ja raami (SVG Repo - c) SVG-sid;
- animeeritud alt üles liikuma ning nähtavust muutma alla kerides;
- mobiili versioonis asuvad kaardid üksteise all.



**Joonis 5:** Kaardi esi- ja tagapool

Galerii veebileheküljel (joonis 6) on kasutatud internetist leitud pilte. Kokku on kaheksa erineva suurusega pilti: pilt sinise maja nurgast (Pixabay, 2016), puu vees (Volk, 2020), siniste seintega vahetänav (Esquivel, 2018), vahetänav siniste/pruunide majade vahel (Neal, 2019), kohvik (Torres, 2019), staadium (Dach, 2017), mustikad (Veeterzy, 2016), potilill

(Larissa Farber, 2021). Piltide järjestus ja paigutus ei pea olema üksühene prototüübiga. Põhiülesanne arendamisel oleks paigutada pildid nii, et pildid ei oleks deformeerunud ning dimensioon jääks samaks peale pildi suuruse muutmist. Galerii nõuded on järgmised:

- tausta hex värvikood on #FFFFFF;
- pildid on ümardatud nurkadega;
- piltide positsioon on reageeriv ehk piltide suurused muutuvad vaateavale vastavalt;
- mobiiliversioonis liiguvad pildid üksteise alla.



**Joonis 6:** Galerii lehekülg

Kontakti lehekülg (joonis 7) on mõeldud meili saatmiseks. Selleks peab veebisaidi külastaja sisestama oma täisnime, meiliaadressi ja meili sisu. Samuti, et vältida spämmimist ei saa saata meili ilma, et kõik lüngad oleks täidetud ning nupule 'Saada' vajutamine peab kustutama vormi lünkade sisu. Nõuded galerii lehele on järgmised:

- tausta hex värvikood on #FFFFFF;
- kontakti vormi hex värvikood on #A7C4D4;
- pealkirja 'Võta Ühendust' ja nupu teksti 'Saada' hex värvikood on #FFFFFF;
- vormi lünkade pealkirjade hex värvikood on #4F6C77;
- nupu hex värvikood on #4F6C77;
- vormi lünkade hex värvikood on #FFFFFF;

- teksti font on Patua One (LatinoType);
- nupule 'Saada' vajutamine kustutab lünkade sisu ning saadab meili sisestatud infoga;
- nupule 'Saada' vajutamine ei saada meili kui kõik lüngad pole täidetud;
- kõikide elementide positsioonid on reageerivad ehk elementide suurused muutuvad vaateavale vastavaks, tagades, et elemendid mahuksid ekraanile;
- mobiilversioonis on täisnime, meiliaadressi ja sisu lüngad sama pikad ning kõik elemendid on tsentreeritud.

The image shows a contact form with a light blue background. At the top, the title 'VÕTA ÜHENDUST' is written in white. Below it are three white input fields with labels 'Täisnimi', 'Meiliaadress', and 'Sisu'. At the bottom right of the form is a dark blue button with the text 'SAADA' in white.

**Joonis 7:** Kontakti lehekülg

Veebilehe funktsionaalsuse nullist ülesehitamise asemel saab kasutada Javascript'i raamistikke, mis annavad arendajatele erinevaid mallide komplekte teatud funktsionaalsuste loomiseks (Ekainu, 2023). Tänapäeval kasutavad paljud veebiarendajad Javascript'i raamistikke, sest tänu raamistikele saavad arendajad veelgi keerukamaid rakendusi kiiremini ja tõhusamalt ehitada (Ekainu, 2023). Kuna raamistike kasutamine on hetkel aktuaalne, siis veebisaidi arendamise lisa nõudeks on Javascript'i raamistiku kasutamine, täpsemalt Vue.js kasutamine.

## 2.3. Arendus vestlusroboti abita

### 2.3.1. Tehnoloogia valik

Veebisait loodi HTML märgistukeele, CSS kaskaadlaadistiku ning Javascript'i programmeerimiskeelega. Veebisaidi arendamiseks kasutati Vue.js raamistikku. Lisaks kasutati päise arendamisel Bootstrap'i, avatud lähtekoodiga raamistikku, mida kasutatakse eelkõige reageeriva arenduse jaoks (Zola, 2022), ning meili saatmiseks SMTP.js, veebisaidi



välist teeki, mis lihtsustab meilide saatmist. Loodud Vue projekt sisaldas funktsioone babel, Vuex ja CSS-preprocessors. Arendamiseks kasutati Vue.js raamistiku kolmandat versiooni ning komponentide ehitamiseks kasutati Options API-d, mis tähendab seda, et kasutati valikuid, nagu näiteks mounted, methods, data, et defineerida komponendi olekut ja käitumist. Veebisaidi koodi on võimalik vaadata Githubi hoidlast: [https://github.com/GreteNeilinn/loputoo\\_projekt](https://github.com/GreteNeilinn/loputoo_projekt).

### 2.3.2. Arendus

Arendamisel jaotati veebisait erinevateks komponentideks, et struktureerida projekt loogiliselt ning taaskasutada koodi. Nendeks komponentideks olid päis, jalus, avaleht, galerii, kontakt, kaart ja juurkomponent app. Veebisaidi üheks nõudeks oli päise ja jaluse olemasolu igal lehel, seega juurkomponent koosnebki päise ja jaluse komponendist, mille vahel on sõltuvalt aktiivsest lehest, kas avalehe, galerii või kontakti komponent (joonis 8). Aktiivse lehe näitamiseks kasutati Vue direktiivi *v-if*, mis renderdab vastava HTML elemendi või Vue komponendi kui direktiivi sisene tingimus vastab tõe.

```
<template>
  <Header
    :pages="pages"
    :active-page="activePage"
    :nav-link-click="(index) => (activePage = index)"
  />
  <GaleriiPage v-if="activePage == 0" />
  <KontaktPage v-if="activePage == 1" />
  <MainPage v-if="activePage == 2" />
  <Footer />
</template>
```

**Joonis 8:** Algkomponent

Päise arendamisel kasutati Bootstrap'i. Täpsemalt kasutati Bootstrap'i navigatsiooni loomisel arendamisprotsessi lihtsustamiseks. Bootstrap'i abil oli võimalik luua reageeriv navigatsioon, mis asendas mobiiliversioonis menüü nõutud hamburgeri ikooniga (joonis 9).



**Joonis 9:** Navigatsioonimenüü mobiiliversioon

Avalehe üheks nõudeks oli elementide, nagu tekstide ja kaartide, animeerimine alt üles läbipaistvast olekust nähtavasse olekusse. Selleks kasutati Vue Composable funktsiooni `vueuse/motion`, mis animeeris vastavaid elemente alla kerides. Tegemist oli väga lihtsa, aga võimsa API-ga, mille abil sai paari koodireaga elemendid animeeritud. Selle lisamiseks pidi `vueuse/motion` teegi projekti alla laadima ning deklareerima põhi Javascript'i failis `main.js`. Motion funktsiooni oli võimalik lisada elemendile kasutades valmis seadistust, mille jaoks pidi HTML-märgendi sisse lisama tahetud animatsiooni direktiivi, nagu näiteks `v-motion-roll-visible-once-right`. Samuti oli võimalik lisada kohandatud animatsiooni. Antud veebisaidil kasutati elementide animeerimiseks valmisseedistuste asemel kohandatud versiooni. Lisaks sellele oli avalehe nõudeks pööratavate kaartide lisamine. Koodi korduse vältimiseks, loodi eraldi kaardi komponent, mida lisati veebilehte 3 korda kasutades Vue `v-for` direktiivi (joonis 10). Direktiiv `v-for` töötab massiivi baasil, mis tähendab seda, et `v-for` direktiiv renderdab elementi vastavalt massiivi suurusest. Antud veebilehe korral koosnes massiivi hulk kolmest elemendist, seega kaardi komponenti korrati kolm korda.

```
<Card
  v-for="(card, index) in cards"
  :key="index"
  :title="card.title"
  :svg="card.svg"
/>
```

**Joonis 10:** Kaardi komponendi silmus

Meili saatmiseks on võimalik implementeerida mitmeid meetodeid. Kõige kergem viis oleks lisada HTML-märgend <a>, mis sisaldaks *href* atribuuti mailto süntaksiga. Kuid sel juhul peaks arendaja olema kindel, et igal veebisaidi külastajal oleks arvutisse alla laetud e-posti rakendus, nagu näiteks Outlook, sest märgendi <a> peale vajutades peaks avanema e-posti rakendus. Teine variant oleks kasutada PHP programmeerimiskeelt. PHP on hea lahendus meilide saatmiseks kiiruse ja paindlikkuse tõttu (Tijūnaitis, 2022), kuid selle implementeerimine on keerukas ning vajab serverit. Kolmas variant, mida kasutati ka valminud veebilehe korral, on kasutada veebisaidi välist teeki. Antud veebilehe korral kasutati SMTP.js teeki, mis on tasuta ning ei vaja serveripoolset (ingl *server-side*) programmeerimiskeelt nagu PHP, ainult Javascript'i (Hebda, 2022). SMTP.js kasutamiseks tuli lisada script-märgend teegi allika lingiga ning meetod (joonis 11), mis sisaldas turvatud tokenit (viide krüpteeritud SMTP.js kasutaja isikutunnistusele), saaja ja saatja meiliaadressi, teemat ja meili sisu.

```
sendEmail(e) {
  Email.send({
    SecureToken: "dfd4a3e6-48d1-4193-bab9-c74d81906563",
    To: "loputoo.test@gmail.com",
    From: "loputoo.test@gmail.com",
    Subject: "Email Sending Test",
    Body:
      this.values[0].text +
      "<" +
      this.values[1].text +
      "> saatis meili: \n" +
      this.values[2].text,
  }).then((message) => alert(message));
  this.$refs.contactForm.reset();
}
```

**Joonis 11:** Meili saatmine

Kokkuvõtvalt oli arendusprotsess bakalaureusetöö autori jaoks sujuv ning väheste probleemidega. Raskust tekitasid eelkõige piltide allikate hoidmine Vue *data* funktsioonis ning meili saatmise funktsiooni integreerimine.

## 2.4. Arendus ChatGPT4 abil

### 2.4.1. Protsess

Enne arenduse algust ChatGPT4-ga, loodi Vue projekt, mis sisaldas funktsioone babel, Vuex ja CSS-preprocessors ning arendamiseks kasutati Vue.js raamistiku kolmandat versiooni. Veebisaidi arendus toimus puhtalt ChatGPT4 vastuste põhjal ning küsimusi küsiti inglise keeles (joonis 12). Põhjus, miks inglise keelt kasutati, tulenes andmetest, millega ChatGPT4 on treenitud. Vestlusrobotit on koolitatud erinevate keelte andmetega, kuid ingliskeelse andmehulk on kordades suurem võrreldes teiste keeltega (Kemper, 2023). Arendamisprotsessi käigus järgiti ChatGPT4 poolt soovitatuid samme ning kopeeriti vestlusroboti poolt saadetud koodijupid vastavasse projekti faili. Probleemide või vigase koodi korral esitati ChatGPT4-le päring, mis sisaldas vea kirjeldust ning käsku vea parandamiseks, ning uuriti, kas vestlusrobot sai hakkama vea parandamisega. Samuti uuriti ChatGPT4 oskust refaktoreerida koodi. Valminud veebisaidi koodi on võimalik vaadata Githubi hoidlast: [https://github.com/GreteNeilinn/loputoo\\_chatgpt/tree/master/src](https://github.com/GreteNeilinn/loputoo_chatgpt/tree/master/src).

### 2.4.2. Arendus

Veebisaidi arendus ChatGPT4 abil algas navigatsioonirea loomisega. Alustati lihtsa päringuga (lisa 2), kus anti ette päise nõuded ning paluti genereerida vastav HTML ja CSS kood. ChatGPT4 saatis vastuseks kolm erinevat faili index.html, style.css ning script.js vaatamata sellele, et algselt paluti ainult HTML ja CSS genereerida. Tulemusena loodi õigesti joondatud navigatsiooniriba, kus elementide suurused olid proportsioonis. Samuti sai ChatGPT hakkama Google fonti lisamisega kui talle oli ette antud ainult fonti nimi. Mobiiliversioonis oli loodud hamburgeri ikoon div-märgendeid kasutades, kuid antud ikooni peale vajutades ei avanenud menüü. Koodi uurides leidis töö autor, et Javascript'i failis lisatakse hamburgeri ikoonile vajutades navigatsioonile klass "nav-active", kuid CSS failis puudus antud klassi CSS kood, mille tulemusena ei ilmunud ekraanile menüü lingid ikoonile klikkides. ChatGPT4 käest küsides, kas ta teab, miks menüü ei avane, suutis vestlusrobot vastata korrektselt ning lisas koodi parandused. Peale muudatusi vastas navigatsioon nõuetel, kuid nägi välja natukene erinev. Selle parandamiseks paluti ChatGPT4-l muuta teksti ja elementide vahe suurust.



could you change the code so that the font is a Google font called Playfair Display



Certainly! You can add the Playfair Display Google font by updating the ``<style>`` section in your Footer.vue component. First, import the font at the beginning of the ``<style>`` section:

```
html
Copy code

<style scoped>
@import url('https://fonts.googleapis.com/css2?family=Playfair+Display:wght@

/* ...rest of the CSS... */
</style>
```

Then, change the font-family for the ``.footer-title`` and ``.footer-list`` to use the Playfair Display font:

```
html
Copy code

.footer-title {
  font-family: "Playfair Display", serif;
  font-size: 1.5rem;
  color: white;
  margin: 1rem 0;
}

.footer-list {
  list-style-type: none;
  padding: 0;
  margin: 0;
  font-family: "Playfair Display", serif;
  font-size: 1.25rem;
  color: white;
  line-height: 1.5;
}
```

Now your footer should use the Playfair Display font for the title and the list items.

## Joonis 12: Näide küsimusest ja ChatGPT4 vastusest

Peale päise muudatusi saadeti ChatGPT4-le päring muuta navigatsiooni kood Vue.js raamistikule vastavaks. ChatGPT4 vastas detailselt ning lisas vastavad sammud koodi implementeerimiseks. Vastus sisaldas komponenti Navbar.vue, kus oli eelnevalt saadetud HTML, CSS ja Javascript kokku pandud. Samuti mainiti, kuhu peaks lisama Google fonti link projektis ning kuidas peaks välja nägema App.vue või mõni teine komponent, et sinna lisada Navbari komponent. Detailsest vastusest vaatamata oli kood vigane. Nimelt ei ilmunud ekraanile logo ning lisatud font ei kasutatud. Kui ChatGPT4-le esitati antud vead ning küsiti,

kas ta teab, kuidas neid vigu parandada, vastas vestlusrobot, mis samme peab järgima, et antud vead parandada ning lisas ka koodi parandused. Samuti küsis töö autor, kas antud Vue.js koodi on võimalik teha efektiivsemaks või lühemaks kasutades Vue direktiive. ChatGPT poolt soovitatud muudatused oli vahetada `:class="{ 'display-mobile-nav': showMobileNav }"` direktiiviga `v-show="showMobileNav"` ning `v-on:click` asemel kasutada `@click`. Vue raamistikus `@click` on `v-on:click` lühendatud süntaks, seega loogika poolest ei toimunud muudatust. `V-show` on Vue direktiiv, mis renderdab vastava HTML elemendi, kuid teeb vastava HTML elemendi nähtavaks juhul kui direktiivi sisene tingimus või muutuja vastab tõele (Vue, 2021). Kui lisada `v-show` direktiiv `nav-märgendile` nagu ChatGPT4 soovitas, siis oleksid menüü lingid 'Galerii' ja 'Kontakt' nähtavad, ainult juhul kui klikkida hamburgeri ikooni, sest klikkimise funktsioon muudab `v-show` sisese muutuja tõeseks. Seega antud soovitus vestlusroboti poolt muutis navigatsiooni vigaseks ning küsides ChatGPT4 käest, kas ta oskab seda parandada, vastas vestlusrobot taaskord vigase koodiga, kus muudatus ei parandanud `v-show` direktiiviga kaasnevaid probleeme.

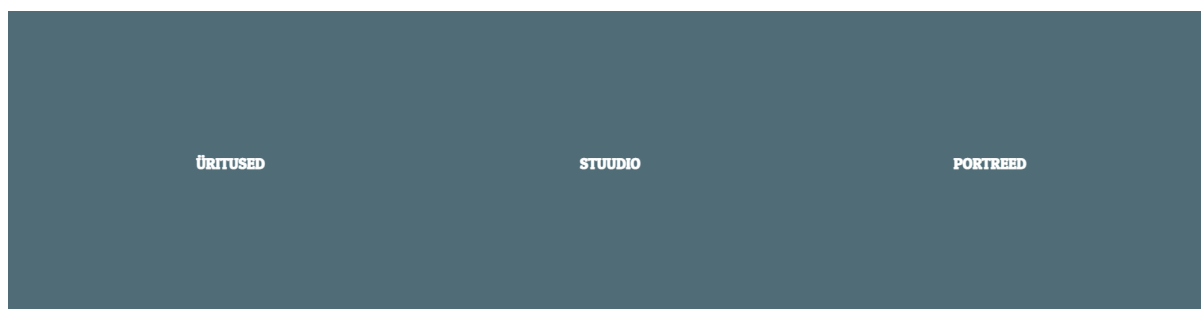
Päise arendamine algas HTML ja CSS koodi genereerimisega, kus hiljem konverteeriti vastav kood Vue.js raamistikule sobivaks. Samuti paluti ChatGPT4-l luua Vue.js raamistikule sobiv kood otse nõuetest (lisa 3) - HTML ja CSS koodi genereerimine jäeti vahelt ära. Tulemuseks saadi loogika poolest sarnane kood, kuid seekord kasutati teeki `vue-router`, seega HTML-märgend `<a>` asendati Vue `<router-link>` märgendiga. Vue ruuter renderdab ainult veebilehe muudetud osasid, mis võimaldab veebisaidi külastajal navigeerida sujuvalt lehekülgede vahel ilma lehti uuesti laadimata (Benjamin, 2023). Lisaks esines erinevus navigatsioonilinkide sihtkohtades. Nimelt HTML failis puudus navigatsiooni linkidel sihtkoht, seetõttu puudus ka konverteeritud Vue navigatsioonis linkide sihtkoht. See-eest soovitas ChatGPT4 lisada vaated (ingl *views*) 'Galerii' ja 'Kontakt' Vue projekti kui jäeti HTML genereerimine vahelt välja. Siinkohal oli näha, et otse Vue.js raamistikule sobiva koodi küsimine annab sarnase, kuid natukene asjakohasema tulemuse. Päise loomise küsimuste ja vastuste faili on võimalik vaadata lingil: <https://drive.google.com/file/d/18aqBCKNKeTpW2bGPCqJlwICePj0Ud2Fx/view?usp=sharing>.

Avalehe arendus algas ChatGPT4-le nõuete esitamisega ning küsiti sellele vastavat Vue.js koodi (lisa 4). Taaskord sai ChatGPT4 üllatavalt hästi hakkama, aga osad veebilehe aspektid ei vastanud nõuetele. ChatGPT4-l paluti lisada joon avalehe tiitli alla ning vestlusrobot sai sellega mõnesmõttes hakkama. Nimelt oli tiitel allajoonitud, kuid joon oli terve lehekülje laiune. ChatGPT4 sai parandamisega hakkama, aga selle saavutamine vajas kolme päringu

saatmist, kus öeldi, et antud kood ei tööta õigesti ning joon on lehekülje laiune. Samuti täpsustati nõuete all, et avalehe jaoks on loodud vaade Home.vue ja navigatsioonirea komponent Navbar.vue ning paluti, et loodavas veebilehes asuks avaleht navigatsioonirea all. Antud vea sai kergelt parandatud, mainides vestlusrobotile, et veebisaidi projektis asub navigatsiooniriba App.vue failis.

Tutvustus sektsiooni päring (lisa 5) tagastas detailse vastuse, kuidas seda implementeerida ning milline peab kood olema. Antud ChatGPT4 vastus vastas kõikidele tutvustuse sektsiooni nõuetele välja arvatud animatsiooni nõudele. ChatGPT4 ei olnud võimeline luua esimese korraga animatsiooni, kus sektsiooni tiitel ja paragraaf muutuvad nähtavaks kui kerida alla. See-eest saadeti 6 erinevat päringut, kus kirjeldati täpsemalt, et animatsioon ei tööta ning anti detaile olukorra kohta, ning viimase päringu (lisa 6) tulemusena saadi töötav kood. Lisaks saadeti seitsmes päring, et konverteerida antud App.vue kood Vue.js raamistiku kolmandale versioonile vastavaks. ChatGPT4 andis erinevaid võimalusi, kuidas animatsiooni implementeerida. Näiteks pakkus ta kasutada CSS animatsioone, Vue.js *transition* süsteemi ning *vue-scrollama* teeki. Kahjuks ükski nendest ei töötanud ning peale kuue erineva päringu saatmist, suutis ChatGPT4 anda töötava lahenduse (lisa 6), kus loodi kohandatud Vue direktiiv. Selle jaoks lisati uus Javascript'i faili, mis sisaldas Javascript'i *IntersectionObserver* rakendusliidet.

Avalehe kaartide sektsiooni loomise protsess nõudis mitmeid päringuid, et tagada veebilehe vastavust nõuetele. Nimelt esimene päring (lisa 7) ChatGPT4-le sisaldas 15 erinevat nõuet, mis oli 8 nõuet rohkem tutvustuse sektsiooni päringust. Peale esimest päringut ei olnud paljud nõuded täidetud vestlusroboti poolt saadetud koodis ning vajasisid lisapäringuid vigade parandamiseks. Esimese päringu (lisa 7) tulemuseks saadi vigane sektsioon (joonis 13), kus animatsioon ja kaartide ümber pööramine ei töötanud, samuti puudusid kaartide pildid ning sektsiooni taustavärviks oli #4F6C77, kuigi tegemist oli kaardi värvusega mitte sektsiooni. ChatGPT4 vastusest oli aru saada, et vestlusrobot ei tea, mida mõeldakse kaardi all. Selletõttu saadeti uus päring, kus täpsustati, et kaart on ristkülikukujuline ning lisaks anti kaardi pikkus, milleks oli 200px. Antud päring vastuseks saadud kood andis kaartidele õige kuju ja pikkuse. Samuti sai ChatGPT4 kiirelt hakkama animatsiooni parandusega. Nimelt saadeti päring, kus mainiti, et animatsioon ei tööta, ning ChatGPT4 mäletas varasemast vestlusest, et projekti on lisatud kohandatud Vue direktiiv *v-fade-in*, ning soovitas antud direktiivi lisada kaardi elemendile. Samuti kasutati ChatGPT4 saadetud koodis *v-for* direktiivi, mis tänu silmusele eemaldas koodi korduse.



**Joonis 13:** Kaartide sektsiooni algversioon

Jaluse loomine möödus sujuvalt. Esimese päringuga (lisa 8) saadi vestlusroboti käest töötav kood, kus vajasid muutust ainult teksti suurus ning elementide vahelised kaugused. Samuti suutis ChatGPT4 muuta jaluse koodi efektiivsemaks ja lühemaks kui seda küsiti. Nimelt suutis ta eemaldada koodi korduse kasutades varem mainitud Vue *v-for* direktiivi.

Galerii loomisel kasutati ASCII (Ameerika Informatsioonivahetuse Standardkood) koodi (lisa 9), et luua piltide paigutuste plaan. Ascii kood jagati neljaks ning saadeti eraldi päringutena ChatGPT4-le (lisa 10), sest ASCII täispikkuse koodi saatmine ületas tähemärkide limiidi. Vestlusrobot sai aru, et tegu oli piltide kastidega ning lisas igale pildile allika ning samuti suutis ta lisada sama piltide vahelise kauguse igale pildile. ChatGPT4 ei järginud saadetud ASCII koodis piltide kuvasuhet, vaid seda pidi eraldi päringu sees mainima. Galerii loomisel ei saanud ChatGPT4 hakkama koodi lühendamisega. Nimelt soovitas vestlusrobot kasutada *v-for* direktiivi, kuid antud kood oli vigane, sest jäeti välja klassi atribuudid, mis andsid piltidele õige paigutuse ja suuruse, ning tulemusena saadi küll kordusteta koodi, kuid antud kood ei vastanud nõuetele. Kui ChatGPT4-lt küsiti, miks pildid on valesti paigutatud ning paluti antud viga ära parandada, ei suutnud ChatGPT4 sellega toime tulla. Iga uue päringuga implementeeriti eelnevalt antud vigase koodi osad ning korrektset koodi ei saavutatud. Kuhjunud vigase koodi parandamiseks saadeti ChatGPT4-le töötav galerii komponendi kood, mis ei sisaldanud *v-for* direktiivi, ning paluti kasutada *v-for* direktiivi, nii et CSS koodi ei peaks muutma. Tulemuseks saadi algselt vigane kood süntaks vea tõttu, kuid teise päringu vastus, kus seda viga mainiti ja paluti ära parandada, oli korrektne (joonis 14).





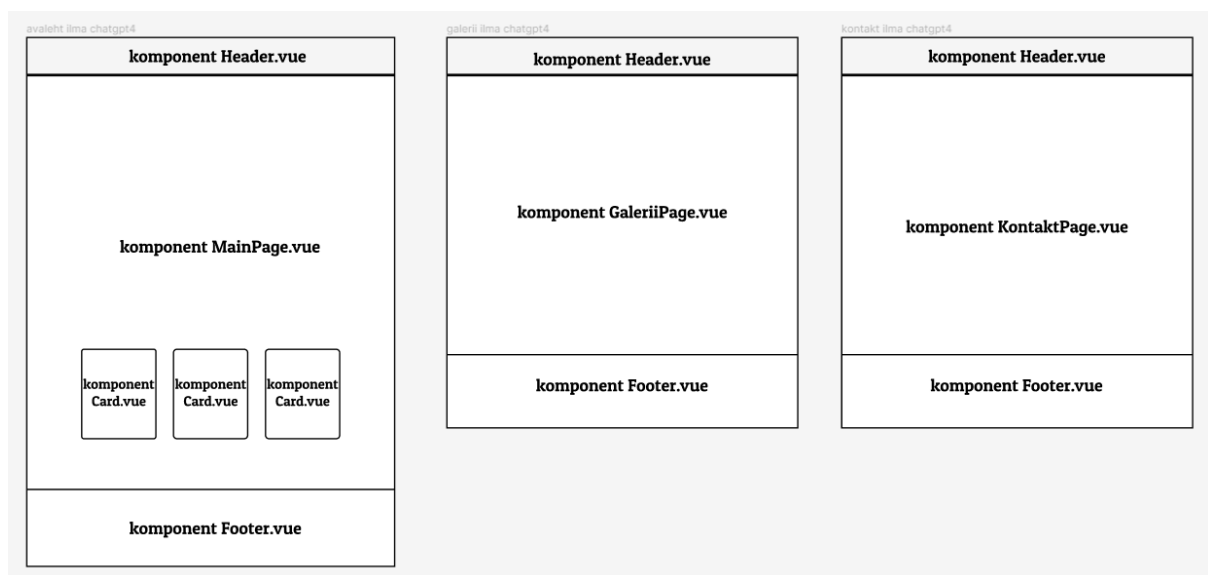
**Joonis 14:** ChatGPT4 poolt loodud galerii

Kuna ASCII koodi kasutamine tegi galerii loomise väga kergeks, kasutati seda ka kontakti lehekülje loomisel. ChatGPT4 suutis saadetud ASCII koodi (lisa 11) järgi luua identse kontakti vormi, ainuke erinevus oli tiitli paigutus. Meili saatmiseks soovitas ChatGPT4 kasutada EmailJS teenust. Tegemist on Javascript'i teegiga, mis aitab meile saata ning ei nõua serverit. ChatGPT4 saatis sammudega juhendi, kuidas lisada EmailJS teek projekti ning kuidas luua EmailJS leheküljel emaili mall. Samas vastuses mainis vestlusrobot ka seda, et EmailJS ja teised kliendipoolsed (ingl *client-side*) meili saatmise teenused ei pruugi olla kõige turvalisemad, kuid on sobilikud väikeste projektide või kontseptsiooni tõestamise (ingl *proof-of-concept*) rakenduste jaoks. Kõige raskem oli ChatGPT4 jaoks taaskord mobiilversiooni loomine, jättes suure ekraani versiooni samaks.

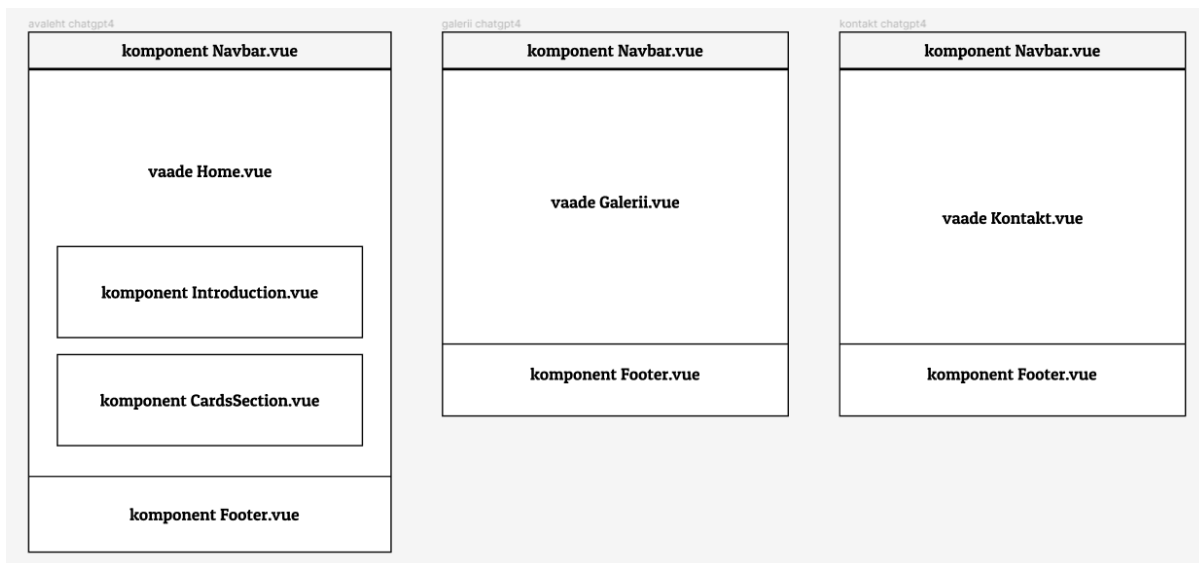
## 3. Võrdlus

### 3.1. Koodi võrdlus

Veebisaitide projekti ülesehitus suurim erinevus seisnes Vue ruuteri kasutuses ja veebisaidi sektsioonide jaotavuses. Nimelt veebisait, kus ChatGPT4 abi ei kasutatud, oli jaotatud ainult komponentideks (joonis 15) ning veebisait, mis loodi ChatGPT4-ga, oli jaotatud nii vaadeteks kui ka komponentideks (joonis 16). Tehniliselt on ka vaated Vue komponendid, kuid neid lisatakse eraldi vaadete kausta kui projektis kasutatakse ruuterit, et määrata, mis komponentide vahel ruuteriga liigutakse. Seega põhjus, miks ChatGPT4 loodud veebisaidis kasutati nii vaateid kui ka komponente, tuleneb ruuteri kasutamisest. Vestlusroboti abita loodud veebisaidis kasutati ruuteri asemel lehekülgede vahel liikumiseks Vue *@click.prevent* sündmust ja klikitud lingi indeksit. Nimelt lisati igale navigatsiooni lingile *@click.prevent* sündmus, mis aktiveerub lingile klikkimisel ning *.prevent* osa sellest hoiab ära vaiketoimingu pärast lingile klõpsamist ehk tagab, et lehekülge ei laadita uuesti. Kui lingile klikitakse, edastab *@click.prevent* juurkomponendile App.vue lingi indeks ning vastav komponent esitatakse ekraanile.



**Joonis 15:** Komponentide jaotus ChatGPT abita veebisaidil



**Joonis 16:** Komponentide ja vaadete jaotus ChatGPT4 abiga veebisaidil

Avalehe kood on erinev veebisaitidel. Näiteks on avalehe tiitli alusjoon erinevalt implementeeritud. ChatGPT4 abita tehtud veebisaidil kasutati div-märgendit, mis asus tiitliga samas konteineris ning seega sai sama laiuse, mis tiitlilgi. See-eest kasutas ChatGPT4 poolt loodud veebisaidil kolme erinevad span-märgendit, kus esimene span-märgend sisaldas sõna 'Fotograafi', teine tühikut ning kolmas sõna 'stuudio', ning igale märgendi alla lisati joon. Samuti erines animatsiooni implementatsioon. Vestlusroboti abita veebisaidis kasutati Vue teeki `@vueuse/motion` ning ChatGPT4 poolt loodud veebisaidil oli lisatud kohandatud Javascript'i fail `fadeIn.js`, mis sisaldas Javascript'i `IntersectionObserver` rakendusliidet, mille abil tuntakse, kas element on vaateavas. Veebisaitide avalehtedel oli ka sarnasusi. Nimelt mõlema veebilehe puhul kasutati Vue `v-for` direktiivi, et esitada kaarte ekraanil ning eemaldada koodi kordus, kuid ChatGPT4 abita loodud veebisaidis kasutati `v-for` direktiivi kaardi komponendil ning ChatGPT4 poolt loodud veebisaidis kasutati `v-for` direktiivi kaartide sektsiooni komponendi sees.

Galerii erinevus esineb piltide positsioneerimises. Nimelt kasutatakse vestlusroboti abita tehtud veebilehes CSS `grid` paigutust ning ChatGPT4 poolt loodud veebilehes CSS `flexbox` paigutust. Põhi erinevused `grid`'i ja `flexbox`'i vahel on paigutuste dimensioonid. CSS `grid` on mõeldud kahemõõtmeliste paigutuste ning CSS `flexbox` on mõeldud ühemõõtmeliste paigutuste jaoks. See tähendab, et grid paigutusega saab paigutada elemente nii ridade kui veergude järgi, kuid `flexbox`'i kasutades saab ainult ühe järgi ehk kas rea või veeru järgi paigutada (Gupta, 2023). Ka galerii puhul kasutasid mõlemad veebilehed Vue `v-for` direktiive, et lisada pilte ekraanile ja lühendada koodi pikkust. Mõlema veebilehe piltide

listid, mida kasutati *v-for* direktiivi jaoks, sisaldasid piltide allikaid ning tekste, mida esitatakse, kui mingil juhul ei esine pilt ekraanil. ChatGPT4 abita loodud veebilehes sisaldas piltide list ka klassi nime, mille järgi lisati CSS koodis vastavale pildile õige paigutus.

Kontakti komponendis kasutasid mõlemad veebilehed kliendipoolseid meili saatmise teenuseid. Vestlusroboti abita loodud veebisaidis oli selleks teenuseks SMTP.js ja ChatGPT4 poolt loodud veebisaidis EmailJS. Samuti kasutasid mõlemad veebilehed Vue *v-model* direktiivi, et saada kontaktvormi sisestatud info kätte. Samuti leidis ka erinevusi veebilehtede vahel. Nimelt kontrollis ChatGPT4 poolt loodud veebileht enne meili saatmist, kas kõik vormi lüngad on täidetud, if-lauset kasutades. See-eest kasutas ChatGPT4 abita loodud veebileht HTML required-atribuuti, mis takistas meili saatmist tühjade lünkade korral.

## 3.2. Arendusprotsessi võrdlus

Veebisaidi loomine ChatGPT4 abita oli otsekohene, kuid aeganõudev protsess. Kõigepealt jagati veebisait erinevateks komponentideks ning alustati arendust juurkomponendist App.vue. Komponentide arendus algas alati HTML raamistiku paika panemisega ning peale seda lisati CSS ja Javascript'i kood vastavalt nõuetele. Arenduse käigus kasutati Google otsingumootorit võimalike koodi lahenduste leidmiseks ning Stack Overflow veebisaiti vigade leidmiseks ning parandamiseks.

See-eest veebisaidi loomine ChatGPT neljanda versiooniga oli korduv protsess. Vestlusrobot vajab mitmeid sarnaseid päringuid ja küsimusi, et luua töötav ning nõuetele vastav kood. Kuigi ChatGPT4 suutis päringutele kiiresti vastata, ei olnud vestlusroboti vastused järjepidevad. Vastuste järjepidevuses mängis rolli vestluse ajalugu, sest ChatGPT4 kasutab kontekstilist mälu ehk vestlusrobot peab meeles varem saadetuid päringuid ning suudab viidata eelnevatele sisenditele, et tagada asjakohased vastused (Luansing, 2023). Samuti sõltus järjepidevus päringu kui ka varem saadetud koodi kompleksuses. Siin saab näiteks tuua kaartide sektsiooni loomise. Kui kaartide sektsioon oli peaaegu valmis ning vastas enamus nõuetele, täpsemalt oli lisatud animatsioon, kaartide ümberpööramisfunktsioon ning kaartide paigutus ja disain suurel ekraanil, paluti ChatGPT4-l tsentreerida kaardid mobiilversioonis. Algselt oli mobiilversiooni CSS kood selline:

```

@media (max-width: 768px) {
  .card-section {
    flex-direction: column;
  }
  .card-container {
    margin-bottom: 2rem;
  }
  .card {
    height: 300px; /* Set card height for mobile as well */
  }
}

```

Peale tsentreerimise päringut tagastati koodijupp, kus atribuut *flex-direction* oli viidud teise klassi, kuigi tegemist ei olnud muudatuse soovitusel, sest antud atribuudi juurde ei pandud, et seda peaks komponendi CSS faili juures muutma, ning antud muudatus ei ole seotud kaartide tsentreerimisega. Tagastatud koodilõik:

```

@media (max-width: 768px) {
  .cards-section {
    padding-top: 2rem; /* Add padding to the top */
  }

  .card-container {
    flex-direction: column;
    align-items: center; /* Add this line */
    justify-content: center; /* Add this line */
  }

  .card {
    margin-bottom: 1rem;
  }
}

```

Selliste vigade tegemine vestlusroboti poolt tegi arendusprotsessi pikemaks ning korduvaks, sest järjepidevuse vigade leidmine oli ChatGPT4 jaoks raske ning vigade parandamine nõudis mitmeid päringuid.

## 4. Tulemus

ChatGPT4 on suuteline looma lihtsa struktuuriga veebisaiti ning tuleb ka toime animatsiooni lisamisega, erinevate suurustega piltide paigutusega, kaartide ümberpööramise efektiga ning meilide saatmisega. Kuigi tegemist ei ole kõige efektiivsema meetodiga. Nimelt vastab ChatGPT4 päringutele kiirelt, mis muudab näiliselt arendusprotsessi lühemaks, kuid vestlusroboti puudulik järjepidevus muudab arendusprotsessi mõttetult pikaks. Nagu varem mainitud, arendusprotsessi käigus muutis ChatGPT4 koodilõikudes teavitamata erinevaid osasid, mis muutis koodi vigaseks. Näiteks vahetas ChatGPT4 kontaktvormi loomise käigul HTML input-märgendi nime *'name'* *'fullname'* vastu, mille tõttu ei sisaldanud meili sisu vormi sisestatud kliendi nime. Selliste väikeste muudatuste parandamine on arendaja jaoks kerge, kuid ChatGPT neljandal versioonil on sellega suured raskused. Vestlusroboti järjekindluse puudust on varem täheldatud ka Qureshi (2023) poolt läbiviidud uuringus. Vestlusroboti järjepidevus sõltub vestluse ajaloost ning päringu kui ka varem saadetud koodi kompleksusest. Samuti mõjutab ChatGPT4 järjepidevust päringute sõnastus, sest vestlusrobot vastab erinevalt erineva sõnastuse korral. Kuid arendusprotsessi käigus oli ka juhtumeid, kus ChatGPT4 vastas erinevalt täpselt samasugustele küsimustele, sest ChatGPT võib anda stohhastilise proovivõtuprotsessi (ingl *sampling process*) tõttu iga kord erineva vastuse (Jalil et. al, 2023). Näiteks saadeti vestlusrobotile päise loomise päring (lisa 3) mitmes tühjas vestluses, et välistada ChatGPT4 kontekstilise mälu rolli vastuses, ning iga kord oli vastus erinev vaatamata, et päringus muudatusi ei tehtud. Vastused ei erinenud ainult sõnade ja tooni poolest, vaid ka loogika poolest. Näiteks ühes vestluses andis ChatGPT4 päise loomise päringu (lisa 3) vastuseks koodilõigu, kus kasutati päise loomiseks Bootstrap'i raamistikku ja lehekülgede vahel liikumiseks HTML a-märgendit. Teises vestluses soovitas vestlusrobot kasutada navigeerimiseks *vue-router* teeki ning kohandatud CSS koodi.

Kuigi ChatGPT4 oskab vigast koodi parandada, võib selle paranduse lahenduseni jõudmine kaua aega võtta. Tihti peale tuleb lahenduse leidmiseks saata mitmeid päringuid, kus mainitakse, mis veaga on tegu ning palutakse seda parandada. Siin võib näiteks tuua kaartide sektsiooni loomise. Nimelt sai vestlusrobot hakkama kaartide positsioneerimisega üksteise alla mobiiliversioon, kuid selle tagajärjena ei vastanud kaartide positsioon suurel ekraanil nõuetele. Autor leidis, et vea põhjus oli kaardi konteineri *display* atribuudis. Kui ChatGPT4-lt küsida, kas antud probleem võib olla kaardi konteineris, soovitas vestlusrobot proovida lisada *flex-direction* atribuuti kaardi konteinerile, kuigi mobiiliversiooni arenduse käigus sai fikseeritud, et see atribuut käib kaardi sektsiooni juurde. Kuna ChatGPT4 kasutab

kontekstilist mälu ehk vestlusrobot peab meeles varem saadetuid päringuid (Luansing, 2023), alustati kaks uut vestlus. Ühes vestluses anti ChatGPT4-le ette kaartide sektsiooni komponenti kood ning paluti positsioneerimine parandada ning teises anti ette ainult HTML osa ning paluti lisada kaartide positsioneerimiseks vajalik CSS kood. Ainult teine variant töötas. Seega on mõndadel juhtudel parem proovida uues vestluses saata soovitud funktsionaalsusega päring, koodi parandamise päringu saatmise asemele. Siinkohal näeme samuti, kuidas mängib koodi kompleksus ChatGPT4 vastustes rolli. Päringu, kus koodilõigust oli suur osa loogikast eemaldatud, vastus oli esimese korraga korrektne, võrreldes päringuga, mis sisaldas peale HTML koodi ka CSS'i ja Javascript'i, kus vastus oli vigane. Kokkuvõtvalt korraga andes palju nõudeid, ei taga et kõik on kaetud, ükshaaval nõuete esitamine on mõndadel juhtudel kasulik. See tulemus on kooskõlas Tian et al. (2023) läbi viidud uuringuga, kus leiti, et lühema päringu korral on suurem võimalus saada korrektne koodilahendus.

Samuti oli juhtumeid, kus töö autor küsis vestlusroboti käest, kas koodi on kuidagi võimalik efektiivsemaks või lühemaks teha ning vestlusrobot vastas soovitusetega, mille järgimisel muutus midagi veebilehel nõuetele mittevastavaks või kood muutus vigaseks. Näiteks avalehe loomise lõpus küsiti, mida võiks muuta, et kood oleks lühem või efektiivsem, vastuseks pakkus vestlusrobot välja, et tiitli klassi CSS koodi hulgast võiks eemaldada atribuuti *align-items*, sest tiitli konteineril oli see atribuut juba olemas. Antud atribuuti eemaldamine tiitli klassist joondas tiitli vasakule, seega ChatGPT4 poolt soovitus ei sobinud kokku päringuga, sest rea eemaldamine muutis loogikat ja disaini. Vaatamata sellele oli ka mitmeid kordi, kus ChatGPT4 oskas koodi muuta lühemaks vigadeta, nagu näiteks jaluse loomisel.

ChatGPT4 sai väga hästi hakkama klasside, komponentide, vaadete, meetodite, parameetrite ja paljude muude nimetamisega. Näiteks saadeti vestlusrobotile päring (lisa 7), kus paluti luua uus sektsioon, mis sisaldab kaarte. ChatGPT4 oskas selle päringu järgi anda komponendile relevantse nime CardsSection.

Kuigi ChatGPT4 on puudujääke, on see siiski väga võimas ja kiire vestlusrobot. Antud vestlusrobot, aga ei ole piisavalt võimas, et asendada täielikult veebiarendajaid, sest korduva koodi ja vigade parandamine nõuab veebiarendajal palju vähem aega ja ressursse. Kõige efektiivsem viis veebisaitide loomiseks oleks kasutada nii veebiarendajaid kui ka ChatGPT4 koos. Nimelt on ChatGPT4 väga efektiivne algse struktuuri ja paigutuse loomises ning ChatGPT4-ja implementeerimine veebisaitide arendusse, muudaks protsessi kordades kiiremaks.

## 5. Võimalikud arengusuunad

Võimalikud arengusuunad oleks testida ChatGPT neljanda versiooni toimetulekut andmebaasi implementeerimises. Võimalik viis selle uurimiseks oleks lasta ChatGPT neljandal versioonil lisada veebisaiti konto lisamise funktsioon, kus veebisaidi külastajad saavad luua konto ning sellega sisse logida. Sealt saaks omakorda edasi uurida, kas ChatGPT4 saab hakkama andmete krüpteerimisega ja valideerimisega.

Samuti kui ChatGPT4-ja pildianalüüsamise funktsioon avalikustatakse publikule, oleks võimalik uurida vestlusrobot toimetulekut veebilehe arendamisega etteantud pildi põhjal. Samuti saaks proovida, kas ChatGPT4 saab hakkama veebisaidi prototüübi genereerimisega.



## Kokkuvõte

Käesoleva bakalaureusetööga sooviti teada saada, kas üks tänapäeva võimsamatest vestlusrobotitest ChatGPT4 on võimeline looma veebisaiti ainult päringute põhjal ning kas vestlusrobot on piisavalt arenenud, et asendada veebiarendajaid. Selle uurimiseks lõi autor prototüübi Figma disainiplatvormil järgides veebilehe disaini häid tavasid ning värvi psühholoogiat. Loodud prototüüp oli portfoolio oletatavale fotograafile ning koosnes kolmest erinevast lehest: avaleht, galerii ja kontakt. ChatGPT4 jõudluse parema arusaamise nimel loodi paralleelselt kaks veebisaiti prototüübi järgi, üks ChatGPT abita ja teine ChatGPT4-ga, ning mõlemad veebisaidid pidid vastama prototüübi lehekülgede nõuetele.

ChatGPT4 oli suuteline looma prototüübi nõuetele vastava veebisaidi, kuid vestlusrobotil oli mitmeid vigu. Nimelt ei olnud vestlusroboti vastused järjepidevad. Arendusprotsessi käigus muutis ChatGPT4 koodilõikudes teavitamata erinevaid osi, mis muutis koodi vigaseks. Selliste väikeste muudatuste parandamine on kerge, kuid ChatGPT neljandal versioonil oli sellega suured raskused ning tagajärjena muutus arendusprotsess pikaks ja korduvaks. Lisaks oli ChatGPT4-l raskusi vigade leidmisega ja koodi lühemaks muutmisega. Vaatamata sellele, on tegemist väga võimsa vestlusrobotiga, tänu millel on võimalik kiirelt ja kergelt luua veebilehe algstruktuuri ja paigutust. ChatGPT4, aga ei ole piisavalt võimas, et asendada täielikult veebiarendajaid. Kõige efektiivsem viis veebisaitide loomiseks oleks kasutada nii veebiarendajaid kui ka ChatGPT4 koos.

# Kasutatud kirjandus

Altman, S. (2023). Planning for AGI and beyond.

<https://openai.com/blog/planning-for-agi-and-beyond> (30.04.2023)

Atuhaire, R. (2023). Artificial Intelligence (AI) Vs Artificial General Intelligence (AGI).

<https://www.dignited.com/108411/artificial-intelligence-ai-vs-artificial-general-intelligence-agi/> (30.04.2023)

Avcontentteam (2023). The Future is Here: Rise of Artificial General Intelligence (AGI).

<https://www.analyticsvidhya.com/blog/2023/04/artificial-general-intelligence/> (09.05.2023)

Baierova, P., Tate, M., Hope, B. (2003). The Impact of Purpose for Web Use on User Preferences for Web Design Features. *ResearchGate*.

[https://www.researchgate.net/publication/228710526\\_The\\_impact\\_of\\_purpose\\_for\\_web\\_use\\_on\\_user\\_preferences\\_for\\_web\\_design\\_features](https://www.researchgate.net/publication/228710526_The_impact_of_purpose_for_web_use_on_user_preferences_for_web_design_features) (07.04.2023)

Benjamin (2023). Vue Router (Why Use It, Plus 3 Things You Need To Know).

<https://beginnersoftwaredeveloper.com/vue-router-why-use-it/> (02.05.2023)

Bose, S. (2021). Coding Standards and Best Practices To Follow.

<https://www.browserstack.com/guide/coding-standards-best-practices> (28.03.2023)

Chowdhury, A. (2023). Is ChatGPT open source?

<https://www.videogamer.com/tech/is-chatgpt-open-source/> (29.04.2023)

Crestiodina, A. (2021). Web Design Standards vs. Website Best Practices.

<https://www.orbitmedia.com/blog/web-design-standards> (28.03.2023)

Dach, M. (2017). Pexels. <https://www.pexels.com/photo/blue-athletic-field-332835/> (12.04.2023)

Decker, A. (2022). The Beginner's Guide to Website Development.

<https://blog.hubspot.com/website/website-development> (08.05.2023)

Drift. An Introduction To AI Chatbots. <https://www.drift.com/learn/chatbot/ai-chatbots/> (29.04.2023)

Ekainu, G.A. (2023). 9 Best JavaScript Frameworks to Use in 2023.

<https://ninetailed.io/blog/best-javascript-frameworks/> (13.04.2023)

Zola, A. (2022). Bootstrap. <https://www.techtarget.com/whatis/definition/bootstrap> (26.04.2023)

Esquivel, R. (2018). Pexels. <https://www.pexels.com/photo/blue-wall-alley-1580173/> (11.04.2023)

Farber, L. (2021). Pexels.

<https://www.pexels.com/photo/vase-with-fresh-chamomile-flowers-on-table-in-sunlight-7573942/> (12.04.2023)

Frankenfield, J. (2022) Chatbot Definition, Types, Pros & Cons, Examples.

<https://www.investopedia.com/terms/c/chatbot.asp> (29.04.2023)

Grabowska, K. (2020). Pexels.

<https://www.pexels.com/photo/cameras-on-white-background-4397899/> (09.04.2023)

Gupta, A. (2023). CSS Grid Vs. Flexbox: A Tutorial to Understand the Key Differences.

<https://www.simplilearn.com/tutorials/css-tutorial/css-grid-vs-flexbox> (07.05.2023)

Günceleme (2022). Mis on loomuliku keele töötlemine (NLP)? Mis on loomuliku keele töötlemise kasutusala?

<https://et.rayhaber.com/2022/02/dogal-dil-isleme-nlp-nedir-dogal-dil-isleme-kullanim-alanlar-i-nelerdir/> (29.04.2023)

Harrison, M. (2023). OpenAI Seems Like a Very Sleazy Company to Be Creating

World-Changing AGI. <https://futurism.com/openai-sleazy-company-creating-agi>

(29.04.2023)

Hatchful. Shopify. <https://www.shopify.com/tools/logo-maker> (09.04.2023)

Hebda, A. (2022). Sending Emails with JavaScript.

<https://mailtrap.io/blog/javascript-send-email/> (26.04.2023)

Hodson, H. (2019). DeepMind and Google: the battle to control artificial intelligence.

<https://www.economist.com/1843/2019/03/01/deepmind-and-google-the-battle-to-control-artificial-intelligence> (30.04.2023)

Howarth, J. (2023). Internet Traffic from Mobile Devices (Mar 2023).

<https://explodingtopics.com/blog/mobile-internet-traffic> (28.03.2023)

IBM. What is a chatbot? <https://www.ibm.com/topics/chatbots> (29.04.2023)

Jalil, S., Rafi, S., LaToza, T. D., Moran, K., Lam, W. (2023). ChatGPT and Software Testing

Education: Promises & Perils. *Arxiv*, doi: 10.48550/arXiv.2302.03287

Jue, J., Ha, J.H. (2022). Exploring the relationships between personality and color

preferences. *Frontiers*, doi: 10.3389/fpsyg.2022.1065372

Juviler, J. (2021). 9 Guidelines & Best Practices for Exceptional Web Design and Usability.

<https://blog.hubspot.com/blog/tabid/6307/bid/30557/6-guidelines-for-exceptional-website-design-and-usability.aspx> (29.03.2023)

Kemper, J. (2023). ChatGPT Guide: 10 prompt strategies for better output.

<https://the-decoder.com/chatgpt-guide-prompt-strategies/> (09.05.2023)

LatinoType. Google Fonts. <https://fonts.google.com/specimen/Patua+One> (09.04.2023)

Lindgaard, G., Fernandes, G., Dudek, C., & Brown, J. (2006). Attention web designers: You have 50 milliseconds to make a good first impression. *Taylor & Francis*, doi: 10.1080/01449290500330448

Luansing J. (2023). Does ChatGPT Learn From User Conversations? <https://www.makeuseof.com/does-chatgpt-learn-from-user-conversations/> (06.05.2023)

Mozilla (2023). SVG: Scalable Vector Graphics. <https://developer.mozilla.org/en-US/docs/Web/SVG> (09.04.2023)

Neal, M. (2019). Pexels. <https://www.pexels.com/photo/perspective-of-a-three-storey-brick-walled-building-3049880/> (11.04.2023)

Nissen, A., Riedl, R. (2021). Design Mode, Color, and Button Shape: A Pilot Study on the Neural Effects of Website Perception. *Springer*. [https://link.springer.com/chapter/10.1007/978-3-030-88900-5\\_22](https://link.springer.com/chapter/10.1007/978-3-030-88900-5_22) (07.04.2023)

Oracle. What is a chatbot? <https://www.oracle.com/chatbots/what-is-a-chatbot/> (29.04.2023)

Pixabay (2016). Pexels. <https://www.pexels.com/photo/blue-concrete-2-storey-house-208560/> (11.04.2023)

Qureshi, B. (2023). Exploring the Use of ChatGPT as a Tool for Learning and Assessment in Undergraduate Computer Science Curriculum: Opportunities and Challenges. *ResearchGate*. [https://www.researchgate.net/publication/370227070\\_Exploring\\_the\\_Use\\_of\\_ChatGPT\\_as\\_a\\_Tool\\_for\\_Learning\\_and\\_Assessment\\_in\\_Undergraduate\\_Computer\\_Science\\_Curriculum\\_Opportunities\\_and\\_Challenges](https://www.researchgate.net/publication/370227070_Exploring_the_Use_of_ChatGPT_as_a_Tool_for_Learning_and_Assessment_in_Undergraduate_Computer_Science_Curriculum_Opportunities_and_Challenges) (09.05.2023)

Shen, M. (2023). An Ultimate Introduction to GPT-4. <https://www.awesomescreenshot.com/blog/knowledge/gpt-4> (08.05.2023)

Sheng, H., Lockwood, N.S., Dahal, S. (2013). Eyes Don't Lie: Understanding Users' First Impressions on website design using eye tracking, doi: 10.1007/978-3-642-39209-2\_71

Shillsalot, S. (2023). Top 5 ways in which ChatGPT 4 is better than ChatGPT 3.5. <https://ambcrypto.com/top-5-ways-in-which-chatgpt-4-is-better-than-chatgpt-3-5/> (08.05.2023)

Sillence, E., Briggs, P., Fishwick, L., Harris, P. (2004). Trust and Mistrust of Online Health Sites. *ResearchGate*, doi:10.1145/985692.985776

Sørensen., C.E. Google Fonts. <https://fonts.google.com/specimen/Playfair+Display> (09.04.2023)

SVG Repo - a. Champagne SVG Vector. <https://www.svgrepo.com/svg/201476/champagne> (09.04.2023)

SVG Repo - b. Tripod SVG Vector. <https://www.svgrepo.com/svg/147402/tripod> (09.04.2023)

SVG Repo - c. Frame SVG Vector. <https://www.svgrepo.com/svg/413253/frame> (09.04.2023)

Tian, H., Lu, W., Li, T. O., Tang, X., Cheung, S. C., Klein, J., Bissyandé, T. F. (2023). Is ChatGPT the Ultimate Programming Assistant -- How far is it? *University of Luxembourg*, doi: 10.48550/arXiv.2304.11938

Tijūnaitis, T. (2022). Guide: How to send emails in PHP (with examples). <https://www.mailersend.com/blog/php-send-email> (26.04.2023)

Torres, J.J. (2019). Pexels. <https://www.pexels.com/photo/blue-metal-bistro-sets-near-potted-flowers-and-road-2277653/> (12.04.2023)

Trajectory. How to Benefit from Color Psychology in Your Website Design. <https://www.trajectorywebdesign.com/blog/web-design-color-psychology/> (07.04.2023)

Tuch, A. N., Presslauer, E., Stöcklin, M., Opwis, K., & Bargas-Avila, J. (2012). The role of visual complexity and prototypicality regarding first impression of websites: Working towards understanding aesthetic judgments. *International Journal of Human-Computer Studies*. <https://research.google/pubs/pub38315/> (01.04.2023)

Turner, A.L. (2017). The psychology of Web design: How colors, typefaces and spacing affect your mood. <https://thenextweb.com/news/psychology-web-design> (07.04.2023)

UserTesting (2019). Color Psychology: How color impacts conversion rates and UX <https://www.usertesting.com/blog/color-ux-conversion-rates> (08.04.2023)

Veeterzy (2016). Pexels. <https://www.pexels.com/photo/blueberry-fruit-on-gray-container-70862/> (12.04.2023)

Volk, J. (2020). Pexels. <https://www.pexels.com/photo/branches-of-leafless-tree-in-river-5273088/> (11.04.2023)

Vue (2021). v-if vs. v-show - Vue Conditional Rendering. <https://learnvue.co/articles/v-if-vs-v-show> (01.05.2023)

# Lisad

## Lisa 1. Prototüübi täispikkuses pildid

Avaleht:



Galerii:



Kontakt:

fotograaf

Galerii Kontakt

VÕTA ÜHENDUST

Täisnimi

Melliadress

Sisu

SAADA

ASUKOHT  
Linn  
Aadress 14  
52087

MEEDIA  
Facebook  
Instagram  
Pinterest

## Lisa 2. Päring navigatsioonirea loomiseks kasutades HTML ja CSS'i

I would like to make a navigation bar for my website. Could you generate html and css for this website. These are the requirements:

- A navigation bar on the top of the page
- On the left side of the navigation bar is a logo and the source of the logo file is logo.png
- On the right side of the navigation bar are two links 'Galerii' and 'Kontakt'
- The text has a color of #4F6C77
- The text font is a Google font called Patua One
- the background color is white
- The navigation bar is responsive and in the mobile version the logo is on the left side of the navigation bar and on the right side is a hamburger icon
- when clicking on the hamburger icon, it shows the links 'Galerii' and 'Kontakt' under each other
- Logo is clickable
- The links 'Galerii' or 'Kontakt' change color to #83989f on hover

## Lisa 3. Päring Vue.js navigatsioonirea loomiseks

I would like to make a navigation bar for my Vue.js website. Could you generate the necessary code for this website? These are the requirements:

- A navigation bar on the top of the page
- On the left side of the navigation bar is a logo and the source of the logo file is logo.png
- On the right side of the navigation bar are two links 'Galerii' and 'Kontakt'
- The text has a color of #4F6C77
- The text font is a Google font called Patua One
- the background color is white
- The navigation bar is responsive and in the mobile version the logo is on the left side of the navigation bar and on the right side is a hamburger icon
- when clicking on the hamburger icon, it shows the links 'Galerii' and 'Kontakt' under each other



- Logo is clickable
- The links 'Galerii' or 'Kontakt' change color to #83989f on hover

## Lisa 4. Päring Vue.js avalehe loomiseks

I would like to make a homepage for my website using Vue.js. I already made a view Home.vue for the homepage. I also already have made a navigation bar for the website and it is inside a component called Navbar.vue. Could you generate the necessary code for the homepage? These are the requirements for the homepage:

- homepage is under the navigation bar; the homepage with the navigation bar fills the whole viewport
- the homepage has a background image and the image file is 'avaleht.jpg'
- it has a title 'FOTOGRAAFIA STUUDIO' which is aligned to the left and vertically centered
- the title is underlined; the underline color is #A7C4D4
- under the title is a paragraph 'Lorem ipsum dolor sit amet, consectetur adipiscing elit.'
- text color is #4F6C77
- the text font is a Google font called Patua One
- the homepage is responsive and in the mobile version the title and paragraph are centered vertically and horizontally;

## Lisa 5. Päring Vue.js tutvustuse sektsiooni loomiseks

I would like to add an introduction section right under the homepage. Could you generate the necessary code that I could add to my Vue.js project. Here are the requirements for the introduction section:

- the height of the section should be half of the viewport;
- the section is split in to half horizontally, where in the first half is a title 'Tutvustus' and in the second half is a paragraph 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore.'

- both title and paragraph are centered vertically and horizontally in their half of the section; section background color is #A7C4D4;
- text color is white; title font is a Google font called Patua One; paragraph font is a Google font called Playfair Display;
- the paragraph text is aligned side to side;
- the section is responsive and in the mobile version the section is split in half vertically, so the title and paragraph are under each other;
- when this section comes in to view when scrolling down in the homepage, the title and paragraph should be animated to ease in and should go from transparent to visible

## Lisa 6. ChatGPT4 vastus töötava animatsiooni koodiga tutvustuse sektsiooni jaoks

Apologies for the confusion. Let's try a different approach using a custom Vue directive for the fade-in effect.

Create a new file named `fadeIn.js` in your `src` directory, and add the following code:

```
function fadeIn(el, binding) {
  const observer = new IntersectionObserver(
    (entries) => {
      entries.forEach((entry) => {
        if (entry.isIntersecting) {
          el.style.opacity = 1;
          el.style.transition = 'opacity 1s';
        } else {
          el.style.opacity = 0;
        }
      });
    },
    {
      threshold: 0.1,
    }
  );

  observer.observe(el);
}

export default {
```

```

    bind: fadeIn,
  };

```

This custom directive will handle the fade-in effect when the element becomes visible in the viewport.

Register the custom directive in your main.js file:

```

import Vue from 'vue';
import App from './App.vue';
import router from './router';
import store from './store';
import fadeIn from './fadeIn';

Vue.directive('fade-in', fadeIn);
Vue.config.productionTip = false;

new Vue({
  router,
  store,
  render: h => h(App)
}).$mount('#app');

```

Update your Introduction.vue file to use the custom directive:

```

<template>
  <section class="introduction">
    <div class="intro-left" v-fade-in>
      <h2 class="intro-title">Tutvustus</h2>
    </div>
    <div class="intro-right" v-fade-in>
      <p class="intro-text">
        Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
        eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad
        minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip
        ex ea commodo consequat. Duis aute irure dolor in reprehenderit in
        voluptate velit esse cillum dolore.
      </p>
    </div>
  </section>
</template>

```

Remove the previous script and style changes related to the visibility and fade-in effect. The custom directive should handle the effect.

With these changes, the title and paragraph in the introduction section should be hidden initially and appear with a fade-in effect when you scroll down. This approach should provide a more accurate way to detect when the elements become visible in the viewport. If you still encounter issues, please double-check your code and ensure the `Introduction.vue` component is correctly imported and registered in your `Home.vue` file.

## Lisa 7. Päring Vue.js kaartide sektsiooni loomiseks

I would like to add a new section right under the introduction section. This new section will contain three cards. Could you generate the necessary code that I could add to my Vue.js project? Here are the requirements for this new section that contains cards:

- the height of the section should be half of the viewport;
- the section contains three cards that are all aligned in the same row;
- the cards have rounded edges; the cards are flippable;
- a card is flipped on click; the front side of every card contains an image in svg format and under that image is the card title;
- the image on the front side of the cards takes up 80% of the card space;
- the images are centered horizontally on the front side of the cards;
- the image file for the first card is 'champagne.svg', the image file for the second card is 'tripod.svg' and the image file for the third card is 'frame.svg';
- the title of the first card is 'ÜRITUSED', the title of the second card is 'STUUDIO' and the title of the third card is 'PORTREED';
- the back side of the cards contain the same title that the card has on the front side and also this paragraph 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam';
- the title is above the paragraph on the back side of the cards and the paragraph takes up 75% of the card space;
- the titles are centered horizontally on both sides of the card and the text font is a Google font called Patua One;

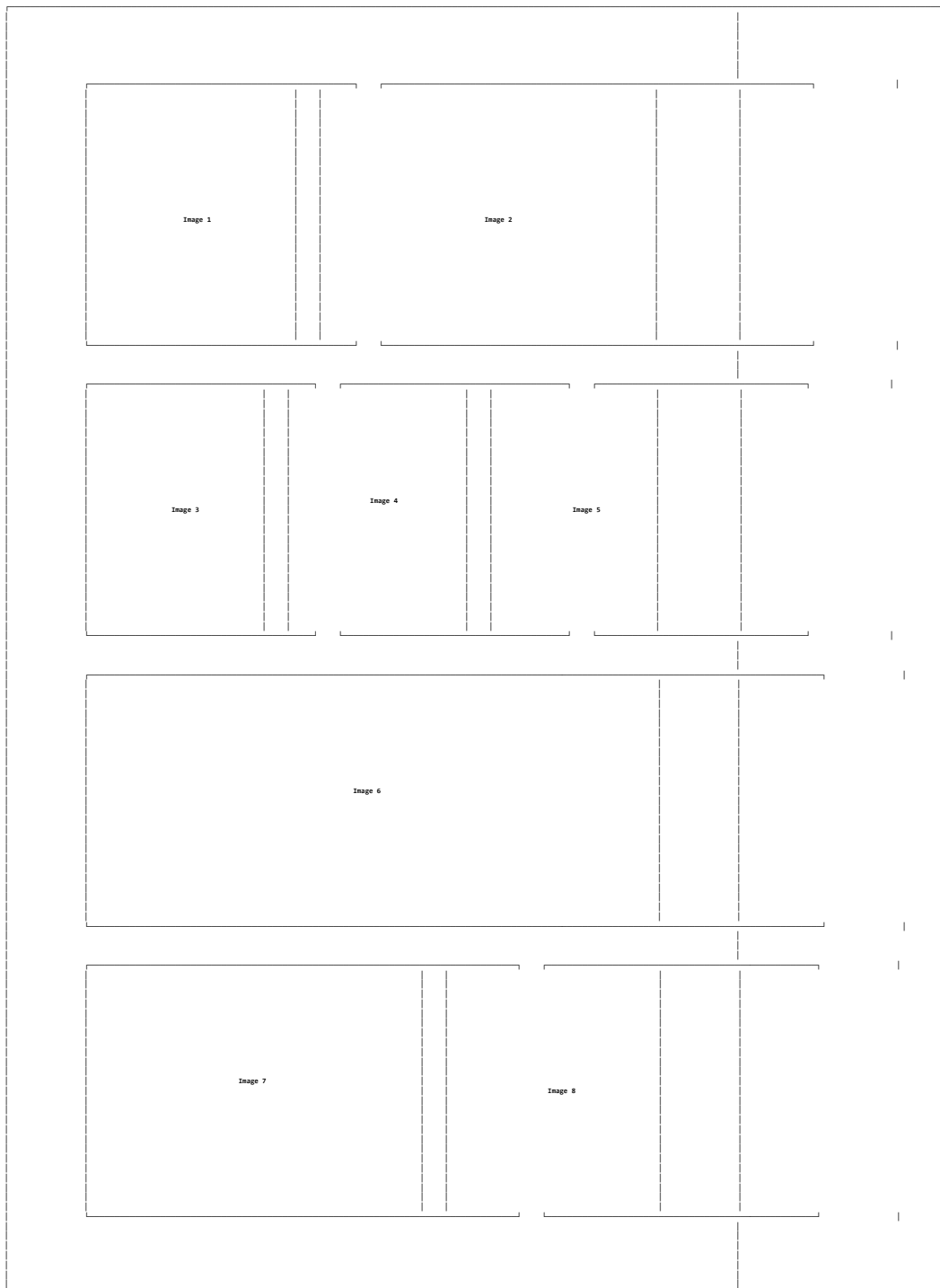
- the paragraph is centered horizontally on the backside of the cards the text font is a Google font called Playfair Display;
- the text color is white; the background color of the cards is #4F6C77;
- the section is responsive and in the mobile version the cards are aligned in a column, so the cards are positioned under each other;
- when this section comes in to view when scrolling down in the homepage, the cards should be animated to ease in and should go from transparent to visible;

## Lisa 8. Päring Vue.js jaluse loomiseks

I would like to add a footer under the cards section. Could you generate the necessary code that I could add to my Vue.js project? Here are the requirements for the footer:

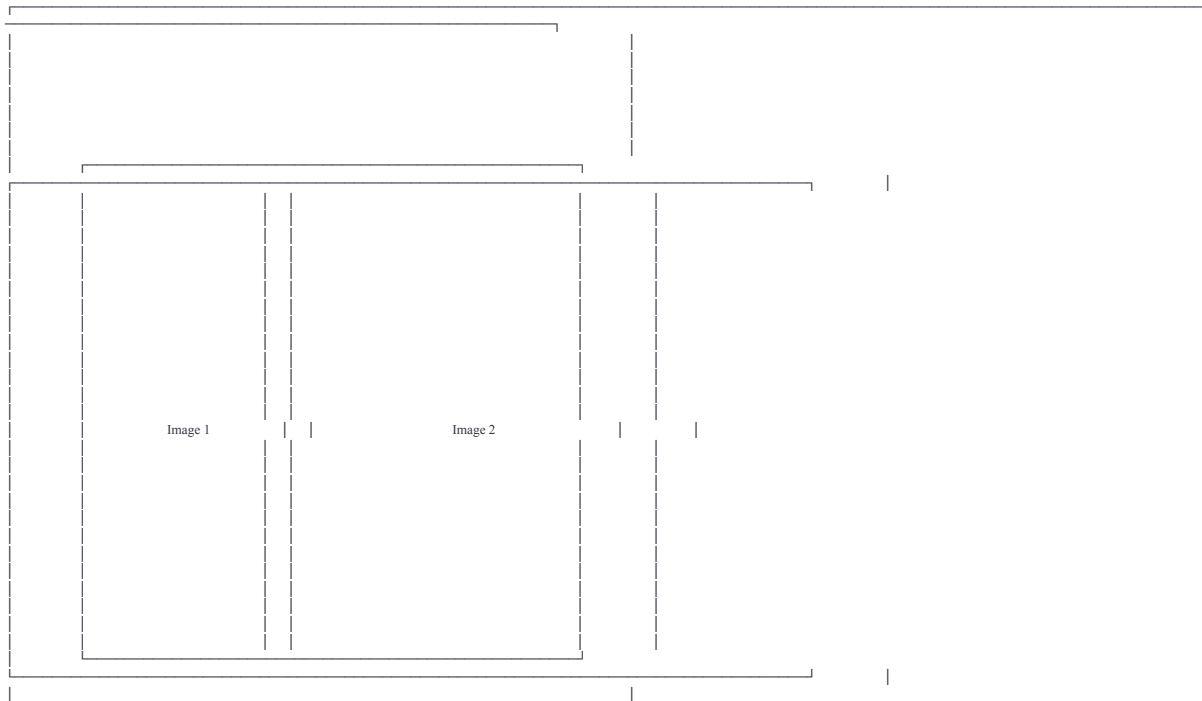
- the height of the section should be a third of the viewport;
- the footer has a white line on the top part that is centered horizontally;
- the line is 5px high and it's width is 60% of the footers width;
- the footer under the line is split into half horizontally;
- both halves contain a title and under the title are three words that are positioned under each other;
- the title of the first half is 'ASUKOHT' and the words beneath it are 'Linn', 'Aadress 14' and '52087';
- the title on the second half is 'MEEDIA' and the words beneath it are 'Facebook', 'Instagram' and 'Pinterest';
- the background color of the footer is #4F6C77;
- the text font is a Google font called Playfair Display;
- when the footer comes in to view when scrolling down in the homepage, the text should be animated to ease in and should go from transparent to visible;

## Lisa 9. ASCII kood Vue.js galerii loomiseks

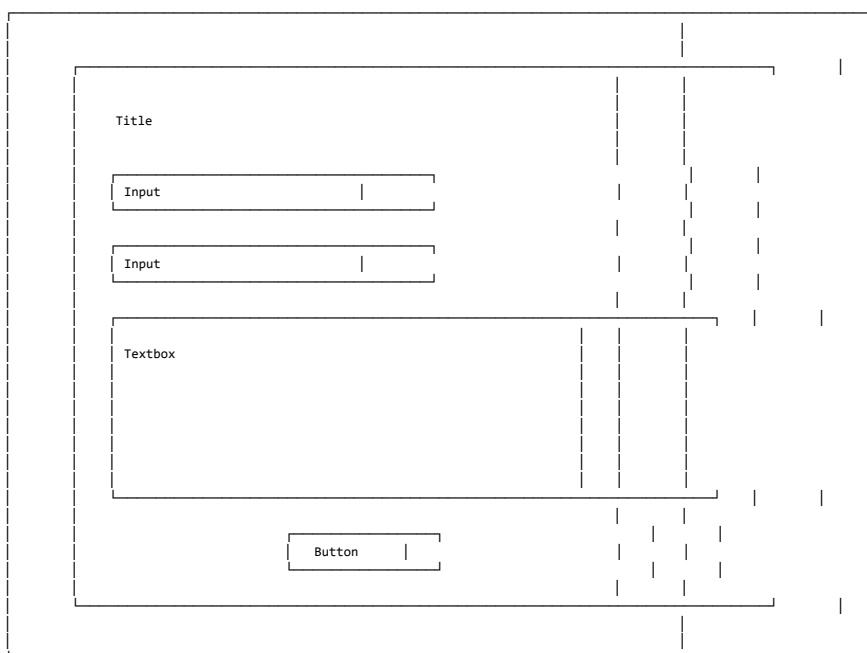


## Lisa 10. Esimene päring Vue.js galerii loomiseks

I would like to make a gallery for my website using Vue.js. I already made a view Galerii.vue for the Gallery. This is the ascii representation of fourth of the Gallery view and could you generate the necessary code for the this:



## Lisa 11. ASCII kood Vue.js kontakti loomiseks



# Litsents

## **Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks**

Mina, Grete Neilinn,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose “Veebisaidi loomise protsess ChatGPT4 abiga ning ilma”, mille juhendaja on Lidia Feklistova, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

*Grete Neilinn*

**08.05.2023**