

TARTU ÜLIKOOL  
Loodus- ja täppisteaduste valdkond  
Arvutiteaduse instituut  
Informaatika õppekava

Karina Tolli

**GitHub Copilot Chat vestlusroboti rakendamine  
veebiarenduses: Praktiline võrdlus inimarendaja tööga  
Bakalaureusetöö (9 EAP)**

Juhendaja: Lidia Feklistova, PhD

Tartu 2024

# **GitHub Copilot Chat vestlusroboti rakendamine veebiarenduses: Praktiline võrdlus inimarendaja tööga**

## **Lühikokkuvõte:**

Bakalaureusetöö eesmärk on uurida GitHub Copilot Chat kasutamise võimalusi veebiarenduses, võrreldes vestlusroboti abil loodud veebisaidi inimarendaja tööga. Töö keskendub kahe meetodi võrdlusele, arvestades arendusprotsessi efektiivsust ja kasutusmugavust. Lisaks tuginetakse olemasolevatele uuringutele, et hinnata Copiloti vestlusroboti unikaalsust võrreldes teiste sarnaste tööriistadega nagu ChatGPT. Töö sisaldab nelja peatükki, mis käsitlevad vestlusrobotite ülevaadet, arendusmeetodite võrdlust, tulemuste analüüsi ning lisades on esitatud vestlusroboti viipade vastused.

## **Võtmesõnad:**

veebiarendus, tehisintellekt, ChatGPT, GitHub Copilot Chat

**CERCS:** P175 Informaatika

# **Implementing GitHub Copilot Chat in Web Development: A practical comparison with human developer work**

## **Abstract:**

The aim of the bachelor's thesis is to explore the possibilities of using GitHub Copilot Chat in web development, comparing the website created with the help of a chatbot to the work of a human developer. The thesis focuses on comparing the two methods, considering the efficiency of the development process and ease of use. Additionally, it relies on existing studies to assess the uniqueness of the Copilot chatbot compared to other similar tools like ChatGPT. The thesis contains four chapters that discuss an overview of chatbots, a comparison of development methods, analysis of results, and includes responses to chatbot queries.

## **Keywords:**

web development, artificial intelligence, ChatGPT, GitHub Copilot Chat

**CERCS:** P175 Informatics

# SISUKORD

<b>Sissejuhatus</b>	<b>4</b>
<b>1. AI tööriistade rakendamine</b>	<b>5</b>
1.1 Vestlusrobot	5
1.2 AI vestlusrobotite kasutamine erinevates valdkondades	5
1.3 GitHub Copilot Chat kui veebilehe loomise abivahend	7
1.4 ChatGPT kui veebilehe loomise abivahend	8
1.5 Uuringud vestlusrobotite kasutamisest veebiarenduses	9
<b>2. Veebisaidi loomise protsess</b>	<b>11</b>
2.1 Veebisaidi funktsionaalsed ja mittefunktsionaalsed nõuded	11
2.2 Prototüübi loomine	14
2.3 Veebisaidi arendus inimarendaja poolt	14
2.3.1 Tehnoloogiate kirjeldus	14
2.3.2 Arendus	16
2.4 Veebisaidi arendus GitHub Copilot Chat abil	19
2.4.1 Ettevalmistus	19
2.4.2 Arendus	19
<b>3. Valminud veebisaitide analüüs</b>	<b>27</b>
3.1 Arendusprotsessi võrdlus	27
3.2 Valminud lahenduste tehniline võrdlus	28
3.2.1 Laadimiskiirus	28
3.2.2 Vigade arv ja tüübid	30
<b>Kokkuvõte</b>	<b>31</b>
Kasutatud kirjandus	33
<b>Lisad</b>	<b>37</b>
<b>I. Litsents</b>	<b>40</b>

## Sissejuhatus

Veebiarendus on pidevas arengus olev valdkond, mida mõjutavad tehnoloogilised uuendused ja uute tööriistade kasutuselevõtt. Viimastel aastatel on tehisintellekt toonud kaasa olulisi muudatusi, mis on mõjutanud veebiarendust. GitHub Copilot Chat on näide uuendustest, mis on mõjutanud veebiarenduse valdkonda, pakkudes arendajatele täiesti uutset tööriista. Töö kirjutamise ajal puudusid teaduslikud uurimused GitHub Copilot Chati kasutamisest veebilehtede loomisel.

Bakalaureusetöö eesmärk on uurida ja analüüsida GitHub Copilot Chat kasutamise võimalusi veebiarenduses. Töö raames arendatakse veebisaiti, kasutades kahte meetodit: töö autori poolt iseseisvat tööd ning GitHub Copilot vestlusroboti abi. Töö hõlmab nende kahe arendusmeetodi võrdlemist, keskendudes arendusprotsessi efektiivsusele ja kasutusmugavusele. Lisaks tuginetakse olemasolevatele uuringutele vestlusrobotite kasutamise kohta veebiarenduses, et paremini mõista GitHub Copilot Chat potentsiaali ja eristuvaid omadusi võrreldes teiste sarnaste tööriistadega nagu ChatGPT.

Eesmärgi saavutamiseks olid püstitatud järgmised uurimisküsimused:

1. Kas GitHub Copilot Chat abil on võimalik luua veebisait minimaalse inimarendaja sekkumisega toimivat koodi?
2. Kas GitHub Copilot Chat abil veebisaidi arendusprotsess on kiirem võrreldes traditsiooniliste meetoditega?
3. Kas inimarendaja poolt valminud veebisait on parema jõudlusega kui vestlusroboti abil valminud veebisait?

Bakalaureusetöö koosneb kolmest sisulisest peatükist. Esimeses peatükis antakse ülevaade vestlusrobotite laiemast kasutamisest ning tutvustatakse uuringus käsitletavaid vestlusroboteid ning nende kasutusvõimalusi veebiarenduses. Teises peatükis vaadeldakse veebisaidi arendamise protsessi traditsioonilise meetodiga ja GitHub Copilot Chat abil, näidates mõlema protsessi eripärad. Kolmandas peatükis võrreldakse kahel meetodil valminud veebisaitide ja arutatakse saadud andmete üle. Lisades on viide töös kasutatud viipadele ja piltidena toodud mõned Github Copilot vestlusroboti viiba vastused.

## **1. AI tööriistade rakendamine**

Selles peatükis kirjeldatakse, kuidas tehisintellekti (AI) rakendamine vestlusrobotite arenduses on levinud eri valdkondades, tuues esile nende tööpõhimõtted, praktilised rakendused ja mõju kasutajakogemusele ning tööstusharude tõhususele. Lisaks uuritakse ChatGPT ja GitHub Copilot Chat vestlusrobotite kasutamist veebiarenduses, pakkudes ülevaadet nende võimalustest ja eelistest arendajatele ning arendusprotsessidele.

### **1.1 Vestlusrobot**

Vestlusrobot (ingl *chatbot*) on arvutiprogramm, mis suudab simuleerida inimvestlust, vastates koheselt küsimustele ja täites ülesandeid. Vestluse käigus töödeldakse küsimusi ning genereeritakse asjakohast vastust automaatselt. Tehisintellekti (AI) kasutamine vestlusrobotites on üha enam levinud, mis võimaldab paremini mõista kasutaja viipu, sealhulgas ka neid, mis pole grammatiliselt või keeleliselt korrektsed [1]. Tehisintellekti-vestlusrobot integreerib masinõppe ja loomuliku keele töötlemise, et täpsemalt tõlgendada kasutaja soove ning pakkuda viibadele loomulike ja asjakohaseid vastuseid [2].

### **1.2 AI vestlusrobotite kasutamine erinevates valdkondades**

AI vestlusrobotide kasutamine on oluliselt suurenenud ja kiiresti integreerunud igapäevaellu. Antud tööriist on võimeline täita mitmesuguseid ülesandeid, alates sisu loomisest kuni veebis otsimiseni. Vestlusroboteid rakendatakse laialdaselt erinevates sektorites, pakkumaks kiiret vastust või tuge. Edasi on välja toodud mõned näited.

Reisiplaneerija: tehisintellekti tööriistad pakuvad mugavat ja efektiivset viisi reisi korraldamiseks, luues kohandatud reisiplaane, mis arvestavad sihtkohta, eelarvet ja individuaalseid huve. Vestlusrobotid suudavad kiiresti, mõne minutiga, koostada päevapõhised plaanid, hõlmates mitmesuguseid vaatamisväärsusi ning pakkudes erinevaid valikuid majutuse, transpordi ja toitlustuse osas [3]. Humley läbi viidud uuringus selgus, et 87% selles osalenud kasutajatest eelistas oma reise planeerida just tehisintellektil põhineva vestlusroboti abil, kui see väidetavalt mõjutaks nende aja ja raha kokku hoidmist [4]. Vestlusrobotid on muutunud oluliseks abivahendiks aja säästmisel reisi planeerimiseks.

Klienditeenindus: kliendirahulolu uurimuses toodi välja, et vestlusrobotid on võimelised kiiremini suhtlema kliendiga ning reageerima vajadustele ööpäevaringselt. Vestlusrobotitel on võimekus käidelda suurt hulka kliendipäringuid (broneerimine, ostu sooritamine,

konsulteerimine) samal ajal tagades kiiruse, järjepidevuse ja parema kliendirahulolu. Iga kliendi ainulaadsete vajaduste ja probleemidega arvestav personaalne lähenemine suurendab märkimisväärselt rahulolu. Kuigi teenindajatel võivad tekkida probleemid emotsioonide juhtimisega ja empaatia väljendamisega, siis masinal seda probleemi ei ole. Vestlusrobotid ei väljenda negatiivseid emotsioone ning suudavad kohaneda kliendi psühholoogilise seisundiga ja vajadustega, mis viib parema teeninduskvaliteedini ja tõstab kliendi rahulolu [5]. Seega tänu oma tugevatele võimalustele vähendavad vestlusrobotid ettevõtte inimtööjõu koormust ja parandavad tegevuse tõhusust.

Meditatsioon: välja arenenud vestlusrobotid on võimelised analüüsima patsiendi sümptomeid ja andma asjakohast teavet tervisliku seisundi kohta, sealhulgas soovitusi järgmisteks toiminguteks. Lisaks vestlusrobotid aitavad inimestel tulla kaasa psühholoogiliste probleemidega vesteldes ja andes individuaalseid soovitusi. Seda tõestab ka Stanfordi ülikoolis läbi viidud uuring, kus vestlusroboti kasutamine vähendas märkimisväärselt 18–28-aastaste inimeste ärevust ja depressiooni [6]. Seega vestlusrobotide kasutamine suurendab tervishoiuteenuste kättesaadavust ning patsientide jälgimisel parandab haiguste varajast avastamist ja ennetamist, tuvastades riskitegureid.

Finants: automatiseerimisega võimaldavad vestlusrobotid säästa aega ja ressursse ning samal ajal tagada täpsuse. Nad võtavad enda peale rutiinse ja ajakuluka töö nagu arvete maksmine, palgaarvestus, eelarvestamine ja tehingute jälgimine. Samuti vestlusrobotid on abiks pettuste tuvastamisel ja ennetamisel finantssektoris. Nad analüüsivad suuri andmemahutusi reaajas, tuvastavad kahtlaseid tegevusi ja hoiatavad finantsasutusi võimalike ohtude eest. Nende abil tehtav varajane pettuste avastamine ja ennetamine aitab kaitsta nii kliente, kui ka finantsasutusi finantskahjude eest [7]. Vestlusroboteid on võimalik õpetada mõistma finantsdokumente, nagu pangaväljavõtted, kindlustuspoliisiid ja investeerimisaruanded ning seejärel neist automaatselt asjakohast teavet eraldama. See omakorda võib märkimisväärselt täpsustada ja kiirustada finantsandmete analüüsimise protsessi [8]. Lisaks saab vestlusrobot ööpäevaringselt jälgida, analüüsida ja koostada prognoose, mis aitavad ettevõtetel teha teadlikumaid otsuseid [7]. Tehisintellekti vestlusrobotite integreerimine finantssektoris on toonud mitmeid uusi võimalusi finantsteenuste osutamiseks.

Haridus: vestlusrobotid pakuvad õpetajatele ja õpilastele laiaulatuslikku tuge, võimaldades isikupärastatud õppekogemusi ja parandades õpilaste kaasatust. Nad aitavad kaasa õppematerjalide ja hindamisprotsesside toetamisele ning aitavad kaasajastada õppekava ja

kiirendada vastuvõtuprotsesse, aidates sellega kokku hoida aega. Vestlusrobotid pakuvad õpilastele ööpäevaringset tuge informatsiooni otsimisel igas keeles, toetades esseele analüüsimist ja pakkudes isikupärastatud soovitusi. Need tööriistad optimeerivad õppimist, arvestades iga õpilase õpiharjumusi ja -raskusi, pakkudes seeläbi tõhusamaid õpetamismeetodeid [9]. Tehisintellekti edusammud on märkimisväärselt avaldanud mõju hariduse arendusele.

Programmeerimine: vestlusrobotite kasutamine programmeerimises avardab tõhusalt arendajate töövõimalusi, muutes nad võimeliseks kiiremini ja efektiivsemalt lahendama nii igapäevaseid kui ka keerukamaid ülesandeid. AI-põhised vestlusrobotid pakuvad programmeerijatele märkimisväärselt abi, muutes koodi kirjutamise protsessi paindlikumaks, oluliselt säästes arendajate aega. Nad on eriti kasulikud kolmes põhivaldkonnas: ideede genereerimine, suuremate ja keerukamate ülesannete lahendamine ning aeganõudvate, kuid lihtsate ülesannete automatiseerimine. Vestlusrobotid suudavad genereerida koodi, mis on lähedal töötavale lahendusele, kuid vajab siiski arendaja sekkumist ja silumist. Lisaks koodi genereerimisele on vestlusrobot suurepärane vahend vigade tuvastamiseks ja seletamiseks, dokumentatsiooni kirjutamiseks ning erinevate programmeerimiskeelte õppimiseks [10]. Samuti suudavad vestlusrobotid koodi ja andmeid analüüsida, et kirjutada asjakohaseid teste [10]. Vestlusrobotid on muutmas programmeerimismaailma, pakkudes alternatiivseid viise koodi kirjutamiseks ning oskuste omandamiseks ja arendamiseks.

### **1.3 GitHub Copilot Chat kui veebilehe loomise abivahend**

Viimastel aastatel on tehisintellekti integreerimine tehnoloogiasse toonud kaasa olulised muutused, eriti veebisaidi arendamise valdkonnas. GitHub Copilot Chat on tipptasemel vestlusrobot, mis võimaldab suhelda GitHub Copilot vestlusrobotiga otse integreeritud arenduskeskkondades. Praegu toetatakse seda Visual Studio Code-is, Visual Studio-s ja JetBrainsi IDE-de komplektis. Copilot Chat on spetsiaalselt loodud tööriist, et toetada arendajaid koodi kirjutamisega seotud ülesannetes, pakkudes koodi soovitusi, selgitusi koodilõikude funktsionaalsuse kohta, ühiktestide genereerimist, veaparandusi ja palju muud, võimaldades samal ajal arendajatel püsida oma arenduskeskkonnas [11]. GPT-4 toel töötav GitHub Copilot Chat on nutikas AI abiline, mis on tehtud arendajate jaoks ja töötab erinevates keeltes [12]. Vestlusrobot on mõeldud ainult koodi kirjutamisega seotud küsimustele vastamiseks. Kasutaja sisend võib olla koodilõikude või lihtsa keele kujul [11].

Vestlusroboti kasutamine aitab optimeerida arendusprotsessi, pakkudes võimalust lihtsustada ja kiirendada töövooge, mis omakorda aitab arendajatel aega kokku hoida ja keskenduda olulisematele ülesannetele [13]. Uuringute põhjal selgus, et 85% arendajatest, kes pole Copilot funktsioone kasutanud, tundsid ennast koodi kvaliteedis kindlamalt ja 88% arendajatest kinnitas, et tundsid ennast rohkem keskendununa ning koodi ülevaatamine muutus 15% kiiremaks. Uuringul osalenud vanema tarkvaraarendaja sõnul “kood oli nii puhas, et ma sain lihtsalt vaadata koodi, mõista, mis toimus, ja suhteliselt hõlpsalt oma koodibaasi üle tuua. Seega tundsin, et koodi kvaliteet oli väga puhas ja lihtsasti mõistetav. Kui ma tegin Copilot-iga tõmbepäringu ülevaatus, nägin, kui hea see on õige veakäsitluskoodi genereerimisel” [14].

Üheks GitHub Copilot Chat-i võimsamaks omaduseks on selle võime pakkuda reaalajas asjakohaseid soovitusi ja lahendada konkreetseid koodi kirjutamise seotud ülesandeid, analüüsides koodi komponente. Lisaks võib see aidata vähendada turvaülevaatuses leitud haavatavuste arvu, pakkudes juhiseid turva probleemide lahendamiseks. Sellest tulenevalt ei tuvasta Copilot Chat üksnes probleeme, vaid pakub ka soovitusi, põhjalikke selgitusi ja alternatiivseid lähenemisviise, rikastades arendajate tööd ja toetades nende püüdlusi kõrgkvaliteetse tarkvara loomiseks [13].

Github Copilot Chat on praegu beetaversioonis ja võib muutuda [11]. Esmalt avalikustati beetaversioon 2023. aasta juulis ainult ärikasutajatele [15]. Beetaversioon on kõikidele avalikustatud 23. detsembril 2023. aastal ja on saadaval kasutajatele, kellel on GitHub Copilot *Individual* või GitHub Copilot *Business* litsents [11, 14]. *Individual* tellimuse omanikud saavad kasutada vestlusroboti nii Visual Studio-s kui VS Code-is. Samuti *Individual* tellimusel võib liituda ootenimekirja, et saada juurdepääsu JetBrains IDE-le. *Business* litsentsi omanikud saavad valida ühe võimaluse vestlusroboti kasutamiseks, kas Visual Studio-s või VS Code-s. Verifitseeritud tudengitel on võimalus liituda *Individual* tellimusega tasuta ning pääseda kõikide funktsionaalsuste juurde [11].

#### **1.4 ChatGPT kui veebilehe loomise abivahend**

Tehnoloogilised edusammud võivad pakkuda abi veebilehtede loomisel, muutes selle protsessi kiiremaks ja lihtsamaks, üks sellistest abivahenditest on ChatGPT vestlusrobot [16]. See on tehisintellektil põhinev vestlusrobot, mis on tulnud kasutusele OpenAI poolt 2022. aasta novembris [17]. OpenAI GPT mudelid on välja töötatud selleks, et genereerida teksti, mis jäljendab inimkeelt, kasutades selleks ulatuslikku internetist pärinevat



andmekogumit. Vestlusrobot kasutab täiustatud loomuliku keele töötlemise tehnikaid, et mõista kasutaja viipu ja vastata neile [18].

Kuigi ChatGPT pole otseselt loodud veebilehtede loomiseks, pakub see mitmeid võimalusi, kuidas seda tõhusalt kasutada veebilehe loomise abivahendina. Vestlusrobot on võimeline kirjutama vastavalt nõuetele ja reeglitele küsitud koodijuppe, samuti saab see töödelda kasutaja poolt saadetud koodilõike, et siluda koodi, tuvastada vigu ning täiendada koodi. Lisaks võib see kirjutada selgitusi erinevatele koodi ridadele, võimaldades paremini mõista, mida konkreetne kood teeb, andes tagasiside koodi stiili, tõhususe, korrektsuse ja täiustuste kohta. See võib osutuda suureks abiliseks aja säästmiseks uue projektiga tutvumiseks või tõhusaks vahendiks programmeerimiskeele õppimisel. ChatGPT võib pakkuda nõuandeid ja juhiseid API integreerimiseks ja serveripoolsete skriptide arendamiseks, aidates genereerida koodi, mis suhtleb andmebaasidega ja käsitleb kasutaja päringuid serveri tasandil [16]. ChatGPT vestlusroboti tõhusaks kasutamiseks on vaja esitada sisendfraas võimalikult täpselt ja detailselt, kuna see võib oluliselt mõjutada vastust. Kerge ümbersõnastamise järel võib see anda hoopis teise vastuse [18].

14. märtsil 2023. aastal avalikustas OpenAI uut GPT-4 põhist mudelit, mis on töökindlam, loomingulisem ja suudab käsitleda suuremat andmemahutu ja keerulisemaid ülesandeid, kui tema eelkäija [19]. Uus mudel on võimeline analüüsima pilte, lugeda andmeid pildilt, tunda ära häält ja analüüsida dokumente [20, 21], võimaldades luua tekstist inimsarnast heli ja pidada kasutajaga vestlust hääle abil. Samuti uus mudel on võimeline genereerima pilte vastavalt kasutaja soovile, mis võib olla kasulik ideede või andmete visualiseerimiseks [20].

GPT-3.5 põhinev mudel on saadaval kõikidele kasutajatele tasuta ning GPT-4 põhinev mudel on saadaval ainult ChatGPT Plus kasutajatele [21]. ChatGPT Plusi kasutajatel on GPT-4 mudeli kasutamiseks praegu piiranguks 40 päringut iga kolme tunni järel, mille ületamiseks tuleb oodata 1 tund või kasutada GPT-3.5 mudelit [22]. ChatGPT vestlusroboti kasutamine veebilehtede loomisel aitab kiirendada arendusprotsessi ja pakkuda tõhusaid lahendusi keerukatele tehnilistele väljakutsetele veebilehe loomisel.

### **1.5 Uuringud vestlusrobotite kasutamisest veebiarenduses**

ChatGPT vestlusrobot on pidevalt arenenud ning täiendanud oma oskust ja võimet. 11. jaanuaril 2024. aastal avaldatud Elegant Themes artiklis [23] väidetakse, et vestlusrobotite abil on võimalik luua veebileht mõne minutiga. Vestlusrobot aitab kujundada ning luua

veebilehte, kasutades HTML, CSS ja JavaScript tehnoloogiaid. Samuti on vestlusrobot heaks sisu loojaks. Artikli autor rõhutab, et ChatGPT võib veebisaidi arendamisel pakkuda mitmekülgset abi, alates sisuideede genereerimisest kuni koodilõikude loomiseni.

ChatGPT vestlusroboti rakendamine tarkvaraarenduses, eriti veebilehtede loomisel, pakub olulisi võimalusi kogu arendusprotsessi tõhustamiseks. Tampere Ülikooli ja Solita Ltd koostöös läbiviidud uuringus tõestati, et ChatGPT, eriti selle uuem versioon 4, suudab efektiivselt toetada nii uute süsteemikomponentide loomist kui ka olemasolevate koodilõikude täiustamist [24]. Uuringus rõhutati GPT-3.5 ja GPT-4 vahelisi märkimisväärsed erinevusi, kusjuures GPT-4 demonstreeris paremat töömahtu kasutust ja võimet jätkata katkenud vastuseid, mis viis tulemuseks konkreetsete ja vähem korduvate interaktsioonide tekkeni. Seega, GPT-4 integreerimine arendusse võib oluliselt parandada koodi genereerimise kiirust ja kvaliteeti, pakkudes sellega tugeva aluse keerukate veebipõhiste süsteemide väljatöötamiseks.

Neilinn [25] jõudis järelduseni, et ChatGPT on kasulik tööriist veebilehe arendamiseks, kuid vajab täielikuks optimeerimiseks arendaja jälgimist ja korrigeerimist. Samas leidis ta, et vestlusrobot ei suutnud olla järjepidev ning tekitas tihti vigu koodis ning ei suutnud paremal viisil koodi lühemaks teha ning kõiki vigu märgata.

Github Copilot Chat on suhtelist uus tööriist ning töö kirjutamise ajal ei leidnud töö autor teadusuuringuid, mis keskendusid tööriista kasutamisele veebiarenduses. Lisaks ei leitud uuringuid, mis tooksid välja töös käsitletud tööriistade võrdlust. 15. detsembril 2023 avaldatud [26] internetiartiklis toodi esile, et GitHub Copilot Chat näitas üldiselt head tulemust, pakkudes kasulikke kooditäiendusi. Hiljem, 23. jaanuaril 2024 avaldatud internetiartiklis [27] samuti käsitleti GitHub Copiloti vestlusroboti kasutamist veebiarenduses, lihtsama veebimudeli loomisel kasutades Copiloti vestlusrobotit. GitHub Copilot Chat tõstis kodeerimise efektiivsust ja produktiivsust, parandas koodi kvaliteeti ning osutus abiks keerulise loogika puhul. Siiski lisas vestlusrobot mittevajalikke elemente, nagu näiteks lisaparameeter salvestamise töötlejas. Kuigi Copilot suurendab kindlasti tootlikkust, on oluline meeles pidada, et seda ei tohiks täielikult usaldada [27].

Tuginedes eespool toodud tulemustele käesoleva töö autor otsustas uurida Github Copilot vestlusroboti rakendamist veebiarenduses, et analüüsida selle tööriista potentsiaali ja praktilist tõhusust. Otsus keskenduda Github Copilot Chat'i uurimisele veebiarenduses tulenes selle platvormi innovaatsilisusest, tudengitele tasuta ligipääsetavusest ning päringute esitamise piirangu puudumisest.

## 2. Veebisaidi loomise protsess

Selles peatükis uuritakse veebisaidi arendamise protsessi kasutades kahte erinevat lähenemist: traditsioonilisel viisil arendatud meetodit ja GitHub Copilot vestlusrobotit. Alguses pannakse paika funktsionaalsed ja mittefunktsionaalsed nõuded loodava veebisaidi jaoks ning luuakse prototüüp. Seejärel kirjeldatakse veebisaidi arendusprotsesside eripärasid tuues välja kasutatavaid tehnoloogiaid.

### 2.1 Veebisaidi funktsionaalsed ja mittefunktsionaalsed nõuded

Süsteeminõudeid eristatakse kahte tüüpi: funktsionaalsed ja mittefunktsionaalsed nõuded. Funktsionaalsed nõuded selgitavad, mida veebileht peab tegema ehk milliseid konkreetseid ülesandeid ja funktsioone see peab täitma [28]. Mittefunktsionaalsed nõuded määratlevad, kuidas süsteem neid ülesandeid täidab ning veebisaidi üldised omadused ja standardid. Mõlemat tüüpi nõuded on omavahel täiendavad ja neid tuleks arendada koos, et tagada projekti ulatuse ja eesmärkide igakülgne mõistmine.

Käesolevas töös on seadistatud järgmised nõuded loodud veebisaidile. Üldised nõuded on järgmised:

#### Funktsionaalsed nõuded

- Igal alamlehel peab olema keskel sisuplokk.
- Lehe päis ja jalus peavad olema kõikidel lehtedel samad.
- Veebisaidi tööresolutsiooniks on ette nähtud 1920x1080px.

#### Mittefunktsionaalsed nõuded

- Veebisaidi disain peab peegeldama looduslikke värvitoone, et tekitada visuaalne seos loomapargiga.
- Taustavärv ehk veebisaidi keha (ingl *body*) peab olema valgest eristuv ja sobima üldise temaatikaga.
- Kõikide tekstide stiil (ingl *font*) peab olema ühtne kogu veebisaidil, et tagada järjepidevus ja lugemise lihtsus.
- Sisuplokk peaks olema päisest heledam, et luua visuaalset eristuvust ja keskendada kasutaja tähelepanu peamisele sisule.

## **Lehe päis (ingl *header*)**

### **Funktsionaalsed nõuded**

- Lehe päis peab olema umbes 15% ekraani kõrgusest, kuid mitte ületama seda.
- Päise vasakus servas peab asuma logo.
- Logost paremal peab kuvama õhutemperatuuri, mis vastab Tartu hetke temperatuurile.
- Päise paremas servas peab olema navigatsioonimenüü, mis suunab kasutaja valitud vahelehtedele: live, meist, hinnakiri ja kontakt.
- Päise ülemise, vasaku ja parema serva nurgad peavad olema ümarad, seetõttu tuleb nendesse servadesse jätta taane, et rõhutada ümarust ja tagada visuaalne kooskõla.

### **Mittefunktsionaalsed nõuded**

- Navigatsiooniriba tekst peab olema ühtlaselt suurte tähtedega.
- Päise värvitooniks kasutatakse rohelist või heledat rohelist.

## **Lehe jalus (ingl *footer*)**

### **Funktsionaalsed nõuded**

- Jalus ei tohi sisaldada servadel taandeid ega ümaraid nurki.
- Jaluse vasakus osas peavad asuma sotsiaalmeedia ikoonid, mis viivad kasutaja klikkimisel vastavale sotsiaalmeedia lehele.
- Allpool, keskel tuleb kuvada veebisaidi autoriõiguse teave.
- Jaluse paremal poolel peavad olema loomapargi kontaktandmed: telefoninumber, e-posti aadress ja asukoht.

### **Mittefunktsionaalsed**

- Lehe jalus peab olema umbes 15% ekraani kõrgusest, kuid mitte ületama seda.
- Päise värvitoonina kasutatakse rohelist või tumerohelist.

## **Vaheleht Meist**

### **Funktsionaalsed nõuded**

- Lehe vasakul poolel peab olema pildikarussell, kus saab kasutaja naaseda eelmisele ja järgmisele pildile
- Teksti sisu all peab olema interaktiivne lahendus, mis kuvab looma kirjeldust, kui kasutaja läheb hiirega selle peale.

### **Mittefunktsionaalsed nõuded**

- Lehe paremal poolel peab olema pealkiri "Tere tulemast loomaaeda", millele järgneb sisuline tekst.

### **Vaheleht Live**

#### **Funktsionaalsed nõuded**

- Lehe keskel peab kajastuma otseülekanne loomadest.
- Otseülekannde käivitub automaatselt ilma kasutaja sekkumiseta.
- Kasutajal ei ole võimalust otseülekannet ise käivitada ega peatada.

### **Vaheleht Hinnakiri**

#### **Funktsionaalsed nõuded**

- Hinnakirja kuvamine lehe keskel.
- Hinnakirja all peab polema soodustuste hind ja tingimus.

### **Mittefunktsionaalsed nõuded**

- Soodustuse hinnainfo peab olema esitatud väiksemas ja heledamas kirjas,

### **Vaheleht Kontakt**

#### **Funktsionaalsed nõuded**

- Kontaktinfo konteiner vasakul. See peab sisaldama üldtelefoni, e-posti aadressi, teavet lahtiolekuaegade kohta ja loomapargi aadressi.
- "Võta ühendust" vorm paremal.
  - Kasutajatel on võimalus sisestada oma nimi, e-posti aadress ja sõnumi sisu.
  - Nupp, mis aktiveerub, kui kõik vajalikud väljad on täidetud.
  - Nupu vajutamisel saadetakse sisestatud andmed e-postiga ja kuvatakse sõnum: "Aitäh, info on saadetud."
  - Kuvab hoiatussõnumi "Täida kõik väljad", kui nupu vajutamisel kõik väljad pole täidetud.
  - Sisestusväljade automaatne tühjenemine pärast vormi edukat saatmist, valmistades ette vormi uueks kasutuskorras.

## Mittefunktsionaalsed nõuded

- Mõlemad konteinerid, nii kontaktinfo kui ka kontaktvorm, peavad olema lehel selgelt nähtavad ja asetatud strateegiliselt vastavalt lehe vasakule ja paremale poolele.

## 2.2 Prototüübi loomine

Veebisaidi visuaalne prototüüp on oluline vahend veebiarenduse protsessis, mis aitab illustreerida kavandatava veebisaidi funktsionaalsust ja kasutaja interaktsiooni võimalusi. Selle peamine eesmärk on varases projektifaasis idee kujundamine ja probleemsete kohtade välja selgitamine, enne kui asutakse arendama lõplikku versiooni. Veebisaidi prototüüp võib varieeruda lihtsatest pabervisanditest kuni interaktiivsete HTML-prototüüpideni [29]. Prototüübi loomine ja selle varasem testimine võimaldab, koguda tagasisidet kasutajatelt ning katsetada erinevaid disainilahendusi, parandades seeläbi kasutajakogemust, rahulolu ja kasutajasõbralikkust [30].

Prototüübi loomiseks kasutas töö autor Figma platvormi. Figma on veebipõhine platvorm, mis on loodud kasutajaliidese ja kasutajakogemuse disainimiseks. Figma abil saab luua interaktiivseid prototüüpe ilma, et oleks vaja kirjutada koodi. Prototüüpimisfunktsioonid võimaldavad disaineritel simuleerida, kuidas kasutajad lõpptoodet kasutama hakkavad, testides kasutajaliidese interaktsioone [31]. See on mugav tööriist meeskonnatööks, kuna võimaldab samaaegselt muuta ja jälgida projekti. Tänu oma veebipõhisusele ja koostöövõimalustele on Figma ideaalne tööriist nii väikestele kui ka suurtele meeskondadele, pakkudes ligipääsu projektidele igal ajal ja igas kohas [32]. Selle töö raames loodud interaktiivne prototüüp on kättesaadav veebiaadressilt: <https://www.figma.com/design/GNVBUEEpgfTCuW7Hw3VFYP/Untitled?node-id=68%3A100&t=Dms7TuVez8MYIcWd-1>

## 2.3 Veebisaidi arendus inimarendaja poolt

### 2.3.1 Tehnoloogiate kirjeldus

Visual Studio Code on mitmekülgne ja kergesti kohandatav tekstiredaktor, mida väljastab Microsoft. Alates ametlikust väljaandmisest 2016. aastal on see saanud kiiresti populaarseks tööriistaks arendajate seas, tänu intuitiivsele kasutajaliidesele, kiiretele klaviatuurikombinatsioonidele ja paindlikule laienduste süsteemile. VS Code'i edu seisneb

selle võimes kohaneda programmeerijate individuaalsete vajadustega, pakkudes samas tuge erinevatele programmeerimiskeeltele ja tööprotsessidele [33].

HTML (ingl *Hyper Text Markup Language*) on märgistuskeel, mida kasutatakse veebilehe struktuuri loomiseks. HTML võimaldab luua struktureeritud dokumente märkides teksti semantiliste struktuuridega, nagu pealkirjad, lõigud, loendid, lingid, tsitaadid ja muud elemendid [34].

CSS (ingl *Cascading Style Sheets*) kasutatakse veebilehe visuaalse esitluse jaoks, et kujundada veebilehe välimust ja elementide paigutust. CSS juhib mitme veebilehe paigutust korraga ja määrab, kuidas HTML-elemente tuleks kujundada, näiteks värvi, fondi ja paigutuse osas [34].

JavaScript on skriptikeel, mida kasutatakse interaktiivsete ja dünaamiliste lahenduste loomiseks veebilehel. See võib suhelda HTML-i sisuga ja teostada funktsioone nagu animatsioon, arvutamine lennult, multimeedia juhtimine ja paigutusprobleemide lahendamine [34].

Node.js on avatud lähtekoodiga JavaScripti käituskeskkond, mis võimaldab kasutada JavaScripti keelt serveripoolsete rakenduste loomiseks [33, 34]. See on populaarne erineva mahuga projektide jaoks [35]. Käesolevas arendusprotsessis kasutatakse Express.js raamistiku, mis on Node.js põhine. Express.js aitab luua ja hallata serveripoolseid rakendusi kiiremini, pakkudes lisafunktsioone Node.js kasutamiseks nagu failide saatmine ja mitmesuguste HTTP-päringute käsitlemine. Express.js aitab arendajatel ehitada rakendusi kiiremini ja tõhusamalt, pakkudes selleks vajalikke tööriistu nagu marsruutimine, kesktarkvara tugi ja palju muud [36].

SendGrid on pilvepõhine e-posti edastamise teenus, mis annab arendajatel võimalust saata e-kirju läbi oma rakenduse. Tänu API-dele (ingl *Application Programming Interface*) ja SMTP releeteenustele (ingl *Simple Mail Transfer Protocol*) on SendGrid integreeritav paljude erinevate programmeerimiskeelte ja platvormidega, mis teeb selle populaarseks valikuks arendajate seas. SendGridi teenust kasutades saab päevas tasuta saata kuni 100 e-kirja, kasutades selleks SMTP relee ja API-sid [37]. Arendusprotsessis kasutatakse selle teenust vormi e-posti teel edastamiseks.

Vercel on pilveplatvorm, mis pakub hostimise ja juurutamise veebirakendustele. Verceli abil saavad arendajad oma töövoogu sujuvamaks muuta, tagades lihtsa juurutamise. Globaalne sisuedastusvõrk (ingl *Content Delivery Network*) tagab kiire ja töökindla

laadimisaja, mis toetab kasutajakogemust. Platvorm on kujundatud kasutajasõbralikult, vähendades rakenduste implementeerimise keerukust ning võimaldab kasutada oma teenust eri suuruste projektide tarbeks [38]. Järgnevas arendusprotsessis kasutatakse Vercel teenust serveri virtuaalseks käivitamiseks.

### 2.3.2 Arendus

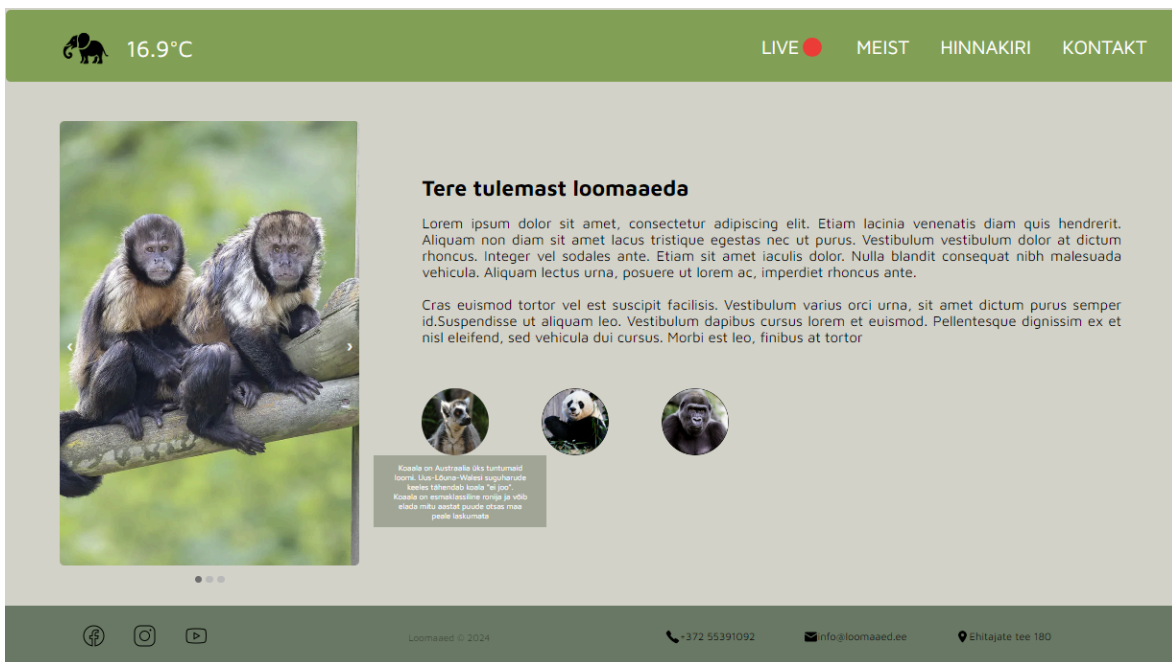
Autori poolt loodud veebisait on kättesaadav veebiaadressilt <https://kodu.ut.ee/~ktolli/Bakat-Iseseisev-veebileht/meist.html> ning kood GitHub repositooriumist <https://github.com/KTolli/Bakat-Iseseisev-veebileht>.

Veebisaidi arendamiseks valis töö autor Visual Studio Code arendusplatvorm. Projekti kausta imporditi vajalikud pildifailid Figma keskkonnast. Enne arendustööde alustamist viidi läbi veebisaidi nõuete täpsustamine, mis põhines juba valmis prototüübil. Nõuetesse lisati spetsiifilised suurusekriteeriumid, et kiirendada arendusprotsessi. Täiendatud dokument on kättesaadav <https://docs.google.com/document/d/1gm2YMxfvqFPEQpbGXPFuWDHpbIHq5k2hvwhFRw4cnSk/edit?usp=sharing>.

Esialgu keskenduti veebisaidi üldiste komponentide arendamisele. Kõikidele lehekülgedele rakendati ühist stiilifaili ja *main.js* skriptifaili temperatuuri näitamise jaoks. Kõigil lehtedel on ühine päis ja jalus ning sama sisuplokk. Temperatuuri kuvamiseks tehakse API päringut, et saada viimased andmed konkreetse asukoha kohta. Pärast andmete saamist ja töötlemist kuvatakse temperatuur ümardatuna täisarvuni veebisaidil.

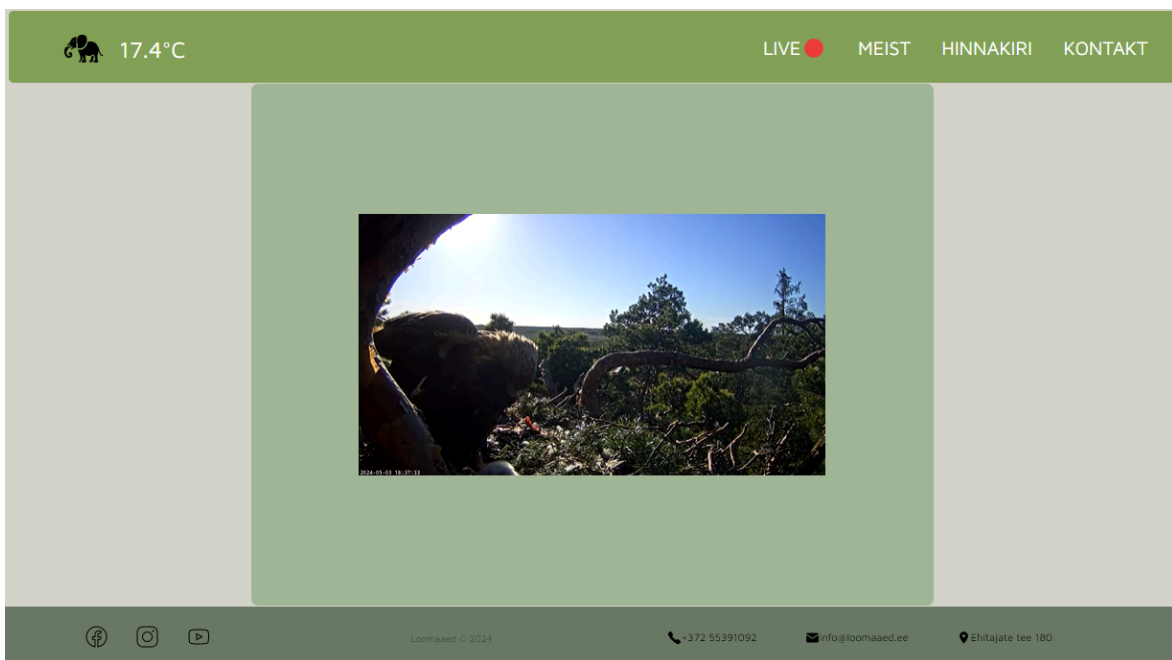
Vahelehe "Meist" sisuplokk on jagatud kaheks osaks (vt joonis 1). Vasakul pool asub pildikarusell, mis võimaldab nooltel klikkides liikuda järgmisele pildile või naasta eelmisele ning kuvab samal ajal all asuvas indikaatoris pildi positsiooni järjestuses. Pildikaruselli funktsionaalsust toetab skriptifail *slide.js*. Paremal pool asub veebisaidi tutvustav tekst, millele järgnevad ringikujulised loomapildid. Piltide all on peidetud vastava looma kirjeldus, mis muutub nähtavaks, kui kasutaja viib hiire pildi peale. Selle interaktiivse lahenduse jaoks on *popup.js* skriptifail.





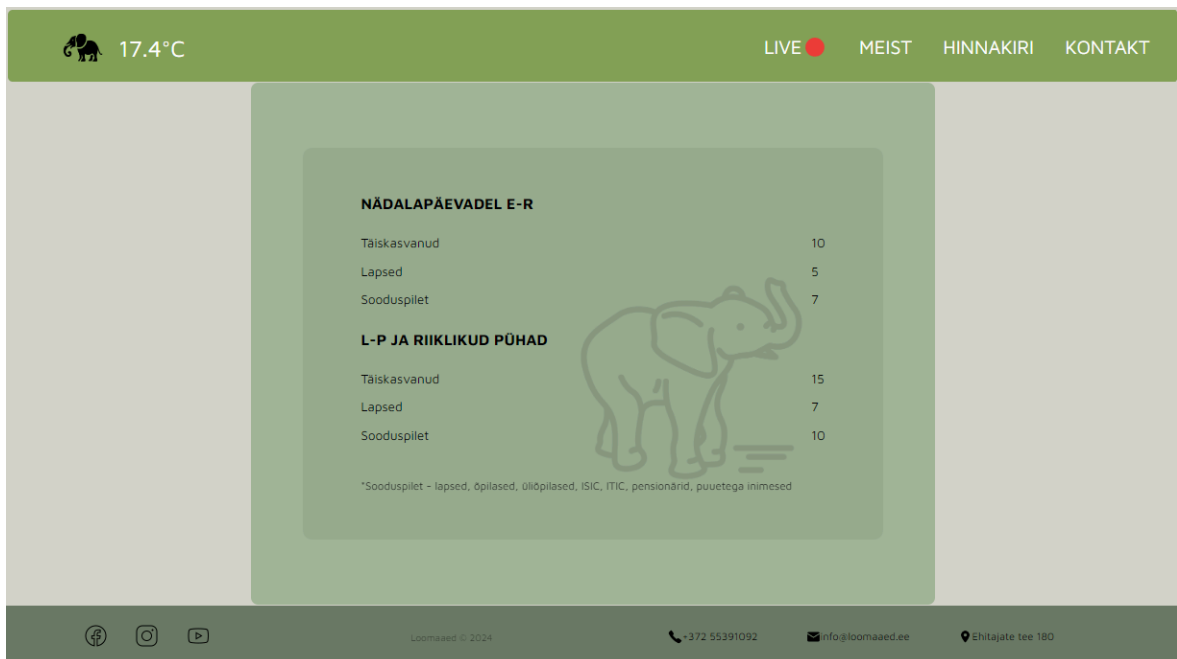
Joonis 1. Inimarendaja loodud vaheleht “Meist” koos looma kirjeldava tekstiga.

Vahelehe "Live" sisuploki jaoks loodi välimine konteiner, mis on mõeldud otseülekande kuvamiseks (vt joonis 2). Otseülekannete edastamiseks kasutatakse *YouTube*'i *iframe*'i, mis on spetsiaalselt ette nähtud *YouTube*'i otseülekannete jaoks. *YouTube*'i *iframe*'i dokumentatsioonist tulenevalt lisati vajalikud parameetrid, et tagada video automaatne käivitamine ja piirata kasutaja interaktsioonivõimalusi videoga.



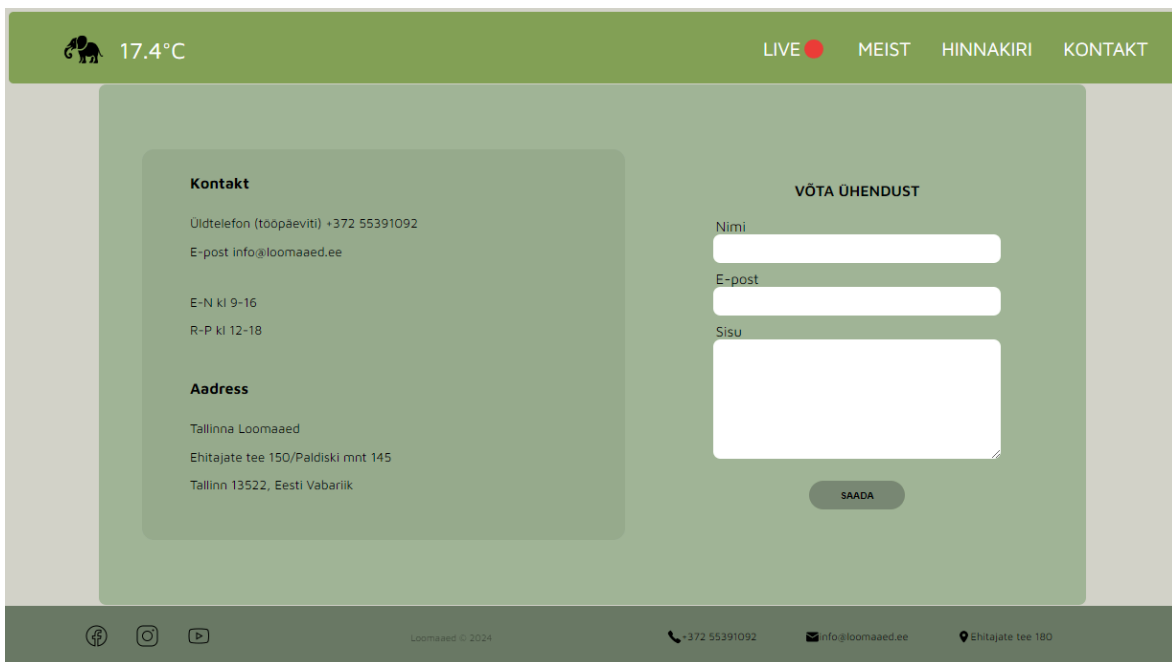
Joonis 2. Inimarendaja loodud vaheleht “Live”.

Vahelehe "Hinnakiri" sisuploki sai lisatud välimine konteiner, mis kasutab samat klassi nagu vahelehe "Live" välimine konteiner. Konteineri sees on veel üks konteiner, mille sees on hinnakirja sisu konteiner (vt joonis 3).



Joonis 3. Inimarendaja loodud vaheleht "Hinnakiri".

Viimaseks vaheleheks on "Kontakt", mille sisuploki sees on laiem välikonteiner (vt joonis 4). See konteiner on jagatud kaheks osaks: vasakul asub kontaktinfo ja paremal kontaktvorm. Kontaktinfo osa koosneb konteinerist, millele on rakendatud taustavärv. See konteiner sisaldab kaks teksti konteinerit, mis sisaldavad pealkirja ja teksti. Kontaktvormi konteineris asub kontavorm, mis kasutab sisseehitatud sisendtüüpe, määrates, et esimene väli on teksti jaoks, teine e-posti jaoks ning kolmas suurem sisestusala.



Joonis 4. Inimarendaja loodud vaheleht “Kontakt”.

Kui saatmisel ei ole kõik korrektselt täidetud, annab vorm vastava veateate. Nupule vajutamisel aktiveerub funktsioon, mis saadab vormi sisu meiliaadressile. Selleks kasutatakse *mail.js* faili, mis võimaldab kasutajatel vormi kaudu andmeid sisestada ning seejärel saata need andmed POST-päringuna serverisse. Pärast andmete saatmist tühjendatakse vormi väljad ning kasutajale kuvatakse teavitus, et kiri on edukalt saadetud.

## 2.4 Veebisaidi arendus GitHub Copilot Chat abil

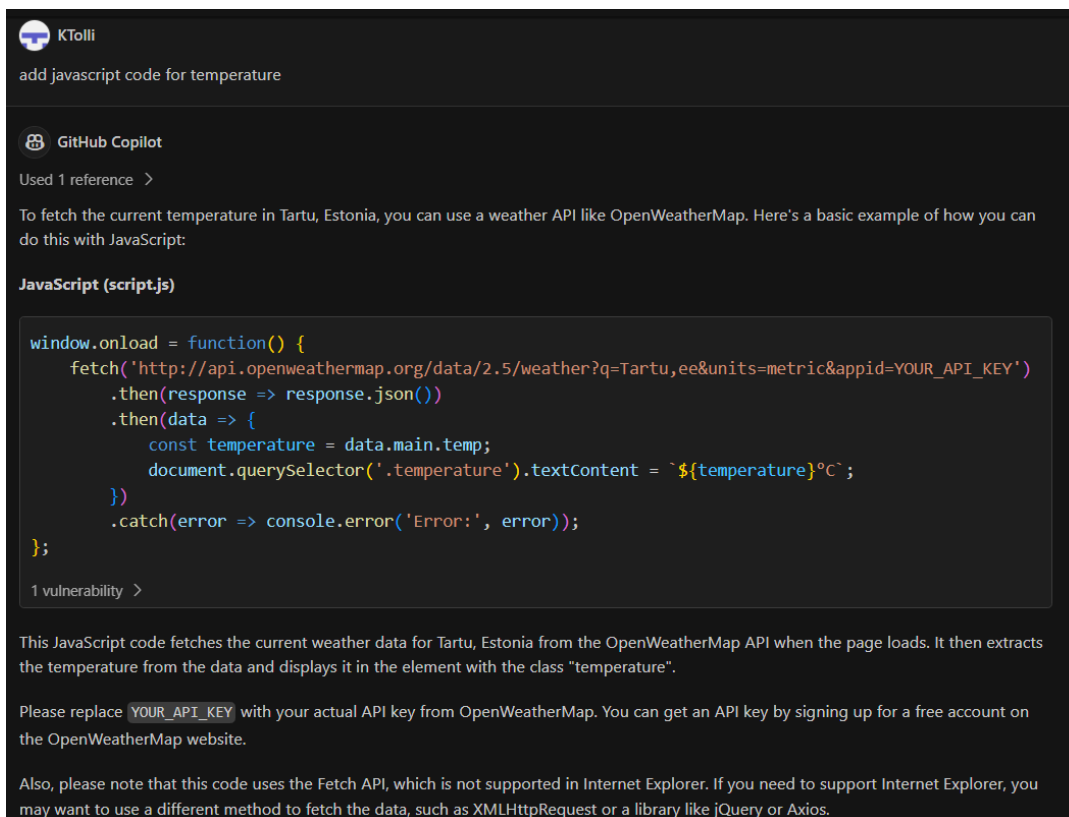
### 2.4.1 Ettevalmistus

Veebisaidi arenduseks tuli esmalt alla laadida Visual Studio Code arenduskeskonda GitHub Copilot Chat laiendus. Protsess nõudis GitHubi autentimise protseduuri läbiviimist. Projekti loomise järel imporditi projekti kausta pildifailid Figma keskkonnast. Visual Studio Code keskkonda vasakule paneelile lisati GitHub Copilot Chat'i dialoogiaken, mis võimaldab kasutajal esitada küsimusi ning kuvab ka vestlusroboti soovitatud küsimused. Arendusprotsessi kiirendamiseks kasutati eelnevalt valmistatud dokumenti veebisaidi kriteeriumite ja nõuetega.

## 2.4.2 Arendus

GitHub Copilot Chat abil loodud veebisait on kättesaadav veebiaadressilt <https://kodu.ut.ee/~ktolli/Baka-copilot/meist.html> ning kood GitHub repositooriumist <https://github.com/KTolli/Baka-copilot>. Veslusrobotiga vestluse ajalugu on kättesaadav Lisas 1 toodud dokumendis.

Veebisaidi arendusprotsess algas lehe struktuuri kirjeldusega, mille põhjal paluti koostada vastavad HTML-failid (vt Lisa 2, Lisa 3). Vastuseks viibale lõi vestlusrobot vajalikud failid, mida sai otse kopeerida või aktiivsesse kursorikohta projekti koodis sisestada. Iga faili kood oli eraldi esitatud, mis lihtsustas kasutamist. Seejärel paluti vestlusrobotil luua lehe päis vastavalt etteantud nõuetele. Vestlusrobot genereeris vajalikud HTML, CSS ja JavaScript koodid. HTML-kood oli varustatud kommentaaridega, sealhulgas juhistega, kus tuleb muuta logo faili nime õigeaks. CSS oli loodud vastavalt spetsiifilistele vajadustele. JavaScripti kood (Joonis 5), sisaldas funktsiooni, mis integreerib OpenWeather API ja oli kommenteeritud osadega, kus tuleb lisada API võti. Samuti juhendas robot, kuidas OpenWeather keskkonnas konto luua ja API võtit hankida. Selle osa töö tulemus vastas täielikult ootustele.



The screenshot shows a GitHub Copilot Chat window. At the top, the user's profile 'KTolli' is visible. The user's prompt is 'add javascript code for temperature'. The Copilot response includes a reference link, an explanation of the task, and a JavaScript code block. The code block is titled 'JavaScript (script.js)' and contains a function that fetches temperature data from the OpenWeatherMap API for Tartu, Estonia. The code uses the Fetch API and includes error handling. Below the code, there is a note about the API key and a warning about Internet Explorer compatibility.

```
window.onload = function() {
  fetch('http://api.openweathermap.org/data/2.5/weather?q=Tartu,ee&units=metric&appid=YOUR_API_KEY')
    .then(response => response.json())
    .then(data => {
      const temperature = data.main.temp;
      document.querySelector('.temperature').textContent = `${temperature}°C`;
    })
    .catch(error => console.error('Error:', error));
};
```

1 vulnerability >

This JavaScript code fetches the current weather data for Tartu, Estonia from the OpenWeatherMap API when the page loads. It then extracts the temperature from the data and displays it in the element with the class "temperature".

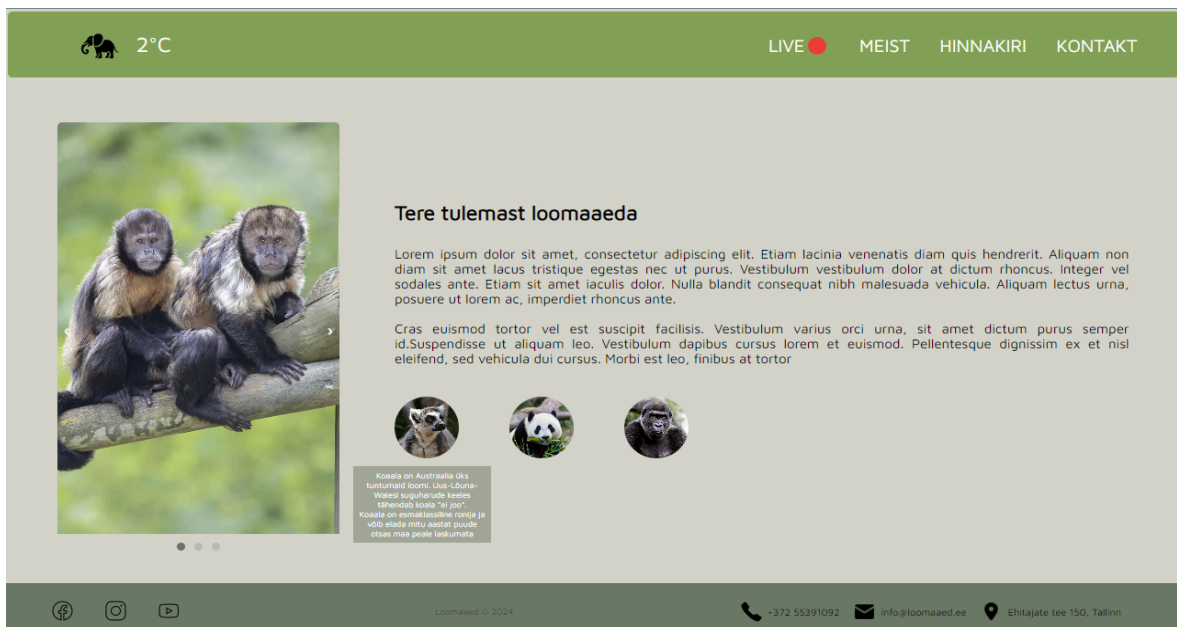
Please replace `YOUR_API_KEY` with your actual API key from OpenWeatherMap. You can get an API key by signing up for a free account on the OpenWeatherMap website.

Also, please note that this code uses the Fetch API, which is not supported in Internet Explorer. If you need to support Internet Explorer, you may want to use a different method to fetch the data, such as XMLHttpRequest or a library like jQuery or Axios.

Joonis 5. Github Copilot Chat vestlus: temperatuuri genereerimine

Järgnevalt paluti teha vastavalt nõuetele jalus. Vestlusrobot genereeris asjakohase koodi, lisades kommentaarid, mis juhendasid, kus tuleb muuta pildi failinime ja teksti sisu. Valminud tulemus vastas täielikult esitatud nõuetele.

Vahelehe "Meist" sisuploki arendamisel tekkisid vestlusrobotiga mõned probleemid, mis vajasisid täiendavaid juhiseid. Esialgne nõue oli tagada, et pildikaruselli kõik pildid oleksid ühesuurused. Pärast täiendavaid juhiseid kohandati pildid vastavalt konteineri suurusele, mis koosnes piltidest. Kuna piltide suuruse muutmine rikkus nende kvaliteeti, paluti robotil katta konteiner piltidega nii, et need kataksid kogu pinna. Järgnev probleem ilmnis indikaatoriga, mis pidi asuma pildikaruselli all. Vestlusroboti poolt genereeritud kood lisas indikaatorid samasse konteinerisse, kus oli pildikarusell, muutes indikaatori nähtamatuks. Mitmete käskude abil õnnestus lõpuks saavutada soovitud tulemus (vt joonis 6).



Joonis 6. Github Copilot Chat abiga valminud vaheleht “Meist” koos looma kirjeldava tekstiga.

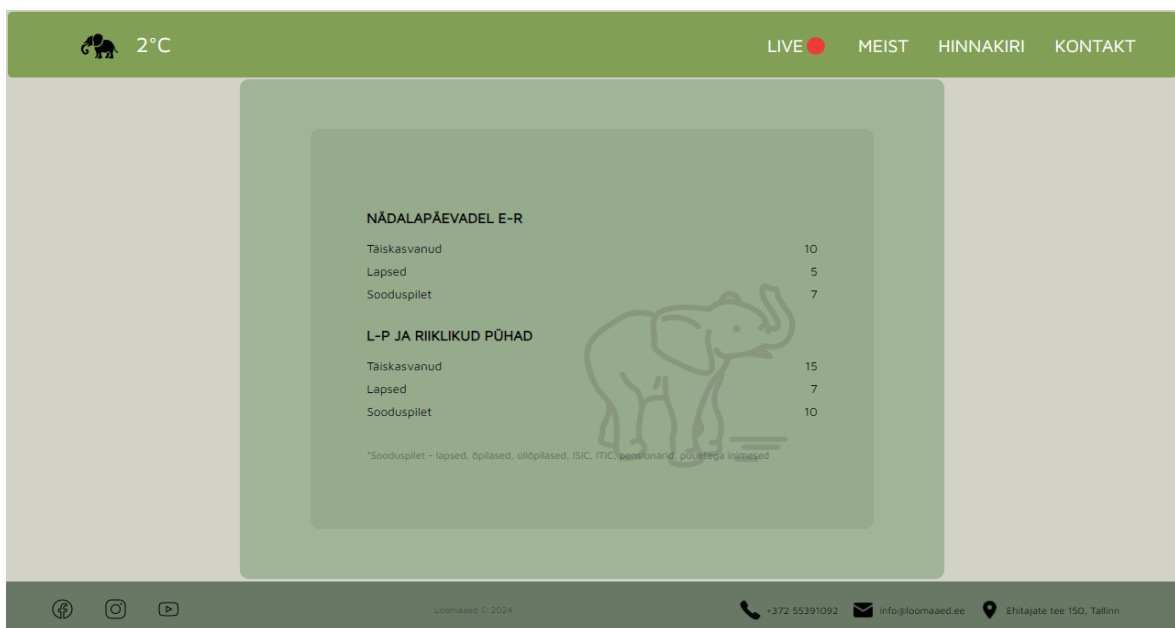
Järgnevas arendusetapis paluti luua paremal asuv tekstikonteiner, mis pidi sisaldama pealkirja ja teksti. Esialgne koodi genereerimine asetas tekstikonteineri pildikaruselli alla, mitte paremale. Sellest tulenevalt ei vastanud genereeritud kood esitatud nõuetele. Hoolimata mitmest vestlusroboti poolt genereeritud koodist ja viibast, ei saavutatud nõutud tulemust, mis nõudis inimarendaja sekkumist koodi.

Manuaalse sekkumisega lisati sisuploki stiili ``display: flex; flex-direction: row;``, mis võimaldas elementide paigutamist horisontaalselt vasakult paremale. Edasi paluti vestlusrobotil konteineritele lisada võrdsed taanded vasakult ja paremalt, mis nõudis

konteinerite põhjalikku inspekteerimist ja täpset viitamist sobivale konteinerile. Peale mitmekordseid kohandusi ja täiustusi saavutati lõpuks nõuetele vastav tulemus.

Järgnevas arendusetapis anti teada, et vaheleht "Hinnakiri" peaks kasutama sama päist ja jalust nagu vaheleht "Meist". Vestlusrobot genereeris päise koodi, rakendades samu stiile, kuid jättis alles vaid navigatsiooni osa paremal. Korduval päringul, et implementeerida täpselt samasugune päis, ei suutnud vestlusrobot siiski vajalikku koodi genereerida. Seetõttu kopeeriti päise ja jaluse koodijupid otse "Meist" leheküljelt.

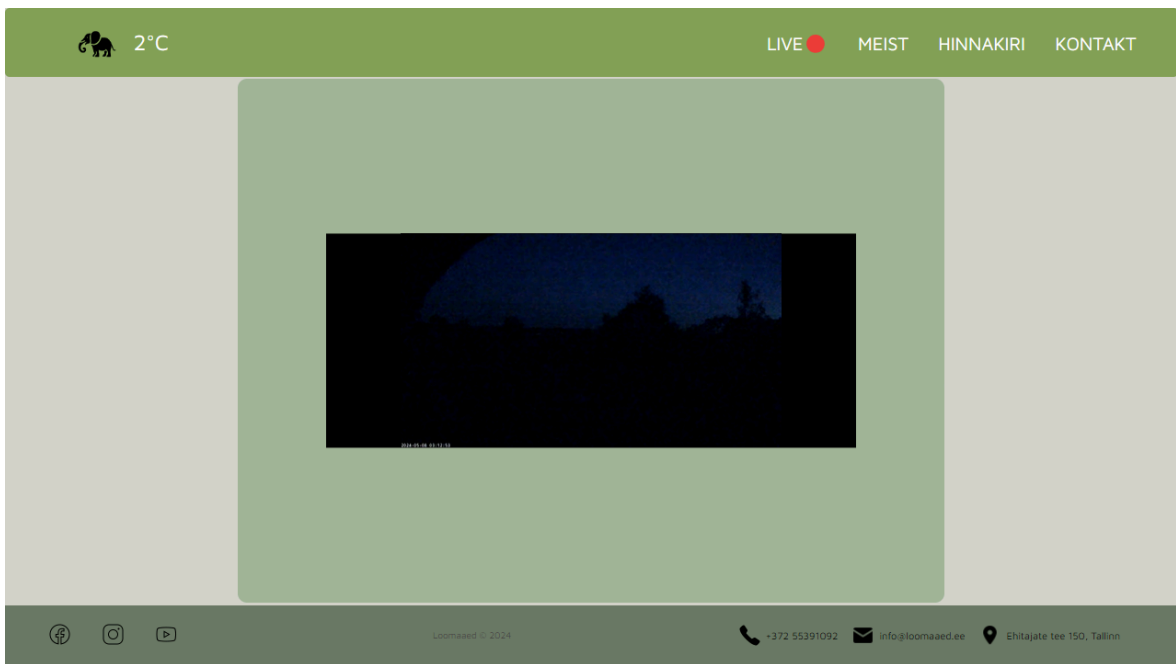
Järgnevalt anti ülesanne luua "Hinnakiri" vahelehe jaoks sisuplokk, mis koosneb välimisest ja sisemisest konteinerist vastavalt nõuetele. Vestlusrobot genereeris algselt eraldi stiilifaili selle lehe jaoks. Siiski pidi töö autor andma täpsustava juhise, et kõik lehed peaksid kasutama ühtset stiilifaili ning viidati konkreetsele stiilifailile. Samuti stiili osas eksis vestlusrobot koodi genereerimisel klassi nimetuse osas. Koodis kasutatava klassi nimetus oli *discount* ning genereeritud koodis klassi nimi *discount-note*. Paluti kohandada koodi vastavalt olemasolevatele klassi nimetustele ja stiilidele. See muudatus vastas ootustele. Vestlusrobot lõi edukalt koodid nõutud plokkide jaoks, kuid tekkis probleem sisuploki kõrgusega, mis oli liialt kõrge, mistõttu ei mahtunud veebisaidi jalus tavapärase suurusega ekraanile. Probleemi lahendamiseks anti korraldus üle vaadata veebisaidi kõrgus. Selle tulemusena genereeriti uus CSS koodijupp, mis arvutas veebisaidi kogukõrguse, lahutades sellest päise ja jaluse kõrguse. Lõpuks esitati nõuded hinnakirja sisu jaoks. Veebileht sai nõuetekohaselt valmis (vt joonis 7).



Joonis 7. Github Copilot Chat abiga valminud vaheleht "Hinnakiri".

Vestlusrobotil paluti kohandada ka "Meist" sisu konteineri kõrgust, rakendades samat nõuet nagu vahelehel "Hinnakiri". Näide viibast on esitatud (vt Lisa 4). Vestlusrobot genereeris CSS-koodi, kus rakendati stiili klassile, millel polnud vastavat nimetust. Järgmises viibas selgitati, et projektis ei ole sellise nimetusega klassi, mille peale vestlusrobot genereeris uue koodi õige klassinimetusega. Uue stiili lisamisega läksid aga paigast ära sisuplokis olevate elementide paigutused. Seetõttu paluti kohandada CSS-faili vastavalt projektinõuetele. Saadud koodijupp sisaldas küll õigeid klassinimetusi, kuid ei hõlmanud kõiki varem kasutusel olnud ja vajalikke stiile. Paluti täiendada ja kohandada juba olemasolevat koodi, et see vastaks kõigile vajadustele. Pärast mitmeid täiendavaid viipaid ja koodi modifitseerimist saavutati lõpptulemus, mis vastas ootustele ning tagas sisuploki visuaalse koostöö ja funktsionaalsuse.

Uuele "Live" vahelehele (vt joonis 8) oli vaja taas lisada päis ja jalus. Siiski kohtas vestlusrobot sama probleemi mis varem, mis sundis päise ja jaluse koodi manuaalselt kopeerima. Seejärel paluti luua "Live" vahelehele sisuplokk, kasutades sama klassi konteinerit, nagu on rakendatud vahelehel "Hinnakiri". Algselt lõi vestlusrobot uue stiilifaili, kuid paluti kasutada juba olemasolevat. Peale seda kohandas vestlusrobot koodi vastavalt olemasolevale stiilifailile. Edasi paluti sisuplokki lisada internetist vabalt leitav otseülekanne loomadest. Kuna vestlusrobotil puudus võimekus internetist otseülekandeid otsida, genereeris see HTML-i jaoks koodijupi, mis võimaldab video lisamist. Hiljem täpsustati, et vajalik on lisada otseülekanne *YouTube*'ist, koos asjakohaste atribuutidega. Vestlusrobot genereeris *iframe*'i otseülekande jaoks, kuid algul oli video suurus liiga väike ja paigutus keset konteinerit probleemne. Mitmete täiendavate viibade järel saavutati video õige suurus ja asetus konteineri keskel. Lisaks paluti keelata kasutajal igasugune interaktiivsus videoga. Selleks lisas vestlusrobot piirangud video sisaldavale konteinerile, tagades, et video oleks passiivne. Lõpptulemusena valmis leht nõuetekohaselt.



Joonis 8. Github Copilot Chat abiga valminud vaheleht “Live”vaheleht.

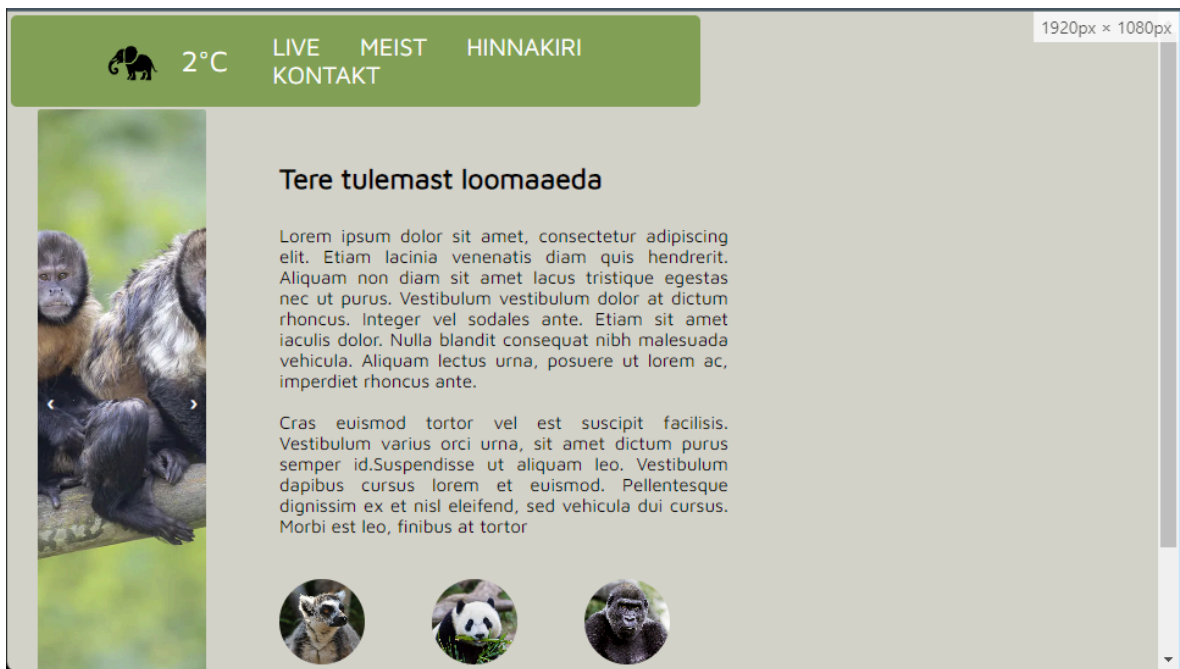
Vahelehele "Kontakt" lisati päis ja jalus ning määrati, et leht kasutab olemasolevat stiilifaili. Paluti luua sisuplokk, mis jaguneb kaheks võrdseks osaks: vasakul kontaktinfo ja paremal kontaktvorm. Vestlusrobotil tekkis probleeme sisu korrektse vasakult paremale paigutamisega, mistõttu oli vajalik käsitsi koodi sekkumine. Pärast täiendavate juhiste saamist vastas kontaktinfo osa ootustele, kuid kontaktvormi elementide paigutamisel – lahtrite nimetused vasakul ning nupud ja pealkirjad keskel – esines raskusi. Vormi elementide grupeerimisel ja stiilide rakendamisel saavutas vestlusrobot nõuetele vastava tulemuse. Kontaktvormi funktsionaalsuse osas soovitas vestlusrobot kasutada serveripoolset programmeerimiskeelt nagu PHP, Node.js või Python (vt Lisa 5). Valik langes Node.js peale. Vestlusrobot genereeris vastava koodi kontaktvormi saatmiseks Gmail kaudu (vt Lisa 6). Esimene katse kontaktvormi saatmiseks ebaõnnestus autentimise probleemide tõttu. Järgnevalt genereeris vestlusrobot uue koodi, kus tuli kasutada OAuth. Arenduse aja kokkuhoidmiseks paluti saata täpne kirjeldus, kuidas seadistada konto Google. Juhendi järgi sai tehtud OAuth 2.0 Client ID ja asendatud koodi, kuid siis tekkis jälle probleem autentimisvõtmega. Sellele vastavalt tegi vestlusrobot skripti autentimisvõtme genereerimiseks, kuid see andis sama vea. Paluti kasutada teist võimalust vormi saatmiseks, mille vastuseks andis vestlusrobot koodi, mida juba on eelnevalt saatnud esimese viiba esitamisel. Paluti kasutada teist võimalust vormi saatmiseks, mitte Google. Selle viiba tulemusena genereeris vestlusrobot SendGridi kasutamise koodi, mis nõudis API võtme vahetamist. Töö autor palus saata ka täpsemad juhiseid SendGridi konto



seadistamiseks ja serveri käivitamiseks. Vastavalt juhiste loodi konto ja täiendati koodi, kuid vormi saatmise funktsionaalsus ei töötanud korrektselt serveri poole pealt. Saadeti vestlusrobotile konsoolis ilmunud veateade. Mitmete koodimuudatuste soovitude tulemusena jõudis vorm küll serverisse, kuid ei edastanud edasi vajalikke andmeid. Järgmises viipas kirjeldati uuesti, milliseid andmeid tuleks meili teel edastada. Sellele vastavalt saatis vestlusrobot uue koodi ja HTML-i muudatused. Siiski ei vastanud vormi muutujad nendele, mis olid juba olemas. Paluti kohandada kood vastavalt olemasolevale, mille järel koodijupp täiendati vastavalt nõuetele. Seejärel tekkis probleem serveri poolse vastusega. Kuigi täidetud vormi sisu jõudis meilisaajani, sai veebisaidil olev kasutaja teate, et meil ei olnud saadetud, kuigi konsooli logisid vaadates näitas, et sõnum oli saadetud. Vestlusroboti mitmed katsed seda probleemi lahendada jäid tulemusteta. Veebileht sai küll välimuselt nõuetekohaselt valmis (vt joonis 9), kuid vormi saatmisel esinesid jätkuvalt tõrked.

Joonis 9. Github Copilot Chat abiga valminud vaheleht “Kontakt”.

Veebisaidi arenduse lõppjärgus soovitas vestlusrobot esitada viiba, kuidas muuta lehe suurust adaptiivseks. Siiski ei vastanud saadud lahendus ootustele, rikkudes veebisaidi esteetilist välimust (vt joonis 10).



Joonis 10. Github Copilot Chat: ebaõnnestunud lehe suuruse muutmine

### **3. Valminud veebisaitide analüüs**

Selles peatükis toimub kahe erineva arendusmeetodiga valminud veebisaitide võrdlev analüüs. Võrdluse eesmärk on esile tuua iga meetodi tugevused ja nõrkused, keskendudes nii arendusprotsessi sujuvusele kui ka tehnilistele aspektidele. Peatükis tuuakse välja, kuidas iga meetodiga loodud veebisait vastas prototüübile ja kuidas erinevad arendusmeetodid mõjutasid lõpptulemuste protsessi, jõudlust ja üldist kasutajakogemust. Lisaks käsitletakse tehnilisi väljakutseid ja probleeme, mis ilmnesisid arendusprotsessi käigus. Võrdlus sisaldab Google PageSpeed Insights analüüsi, mis annab ülevaate iga veebisaidi jõudlusest erinevates kategooriates.

#### **3.1 Arendusprotsessi võrdlus**

Mõlemad valminud veebisaidid vastasid prototüübile ja paika pandud funktsionaalsetele ja mittefunktsionaalsetele nõutele. Kui arendaja rakendas YouTube'i dokumentatsioonis toodud parameetreid, oli kasutajal siiski võimalus panna videopausile ja kasutada teisi YouTube'i platvormi sisseehitatud funktsioone. Seevastu, Copilot vestlusroboti kasutamine andis olulise eelise, kuna see piiras täielikult kasutaja võimalusi videoga interakteerida.

GitHub Copilot Chat'i abil valminud kood oli arusaadav ja hästi struktureeritud. Klassid ja muutujad olid arusaadavalt nimetatud, mis hõlbustas koodi mõistmist. Koodi osad olid loogiliselt grupeeritud, mis lihtsustab tulevikus veebilehe täiendamist. Arenguprotsessi tulemusel selgus, et GitHub Copilot vestlusroboti abil valminud veebileht ei suutnud täita kõiki nõudeid, mis ilmnis veateates kontaktvormi saatmisel, kuid suutis pakkuda lisalahendust kasutaja tegevuse täielikuks piiramiseks, mida inimarendaja meetodil ei olnud.

Arendusprotsessis ilmnis oluline erinevus ajakasutuses. Eksperiment näitas, et inimarendajal läks neli korda rohkem aega veebisaidi loomisele võrreldes ajakuluga, mis läks veebisaidi loomisele vestlusroboti abil. Tuleb märkida, et kuigi vestlusrobot suutis käskluste abil jõuda lahenduseni, nõudsid need täiendavad käsklused inimarendaja sekkumist koodi. Need probleemi kohad näitasid vestlusroboti iseseisva töö tõhusust, rõhutades inimarendaja olulist rolli arendusprotsessis. Üks märkimisväärne probleem ilmnis vestlusrobotil kontaktvormi saatmisel: kuigi vormi sisu saadeti edukalt meilile, sai kasutaja eksitava teate, et kiri ei olnud saadetud ning samuti kulus palju aega selleks, et leida meili saatmise lahendus. Kontaktvormi arendamise protsessile kulus üle kolmandiku

kogu arendusajast, kasutades Copilot vestlusroboti meetodit. Põhjuseks oli, et vestlusrobot genereeris koodijuppe, mis ei suutnud probleemi lahendada.

Inimarendaja pidi uurima lisafunktsioonide ja interaktsioonide võimalusi ning kohandama seda olemasoleva koodiga. Seevastu vestlusroboti abil oli võimalik otse küsida vajalikku teavet, mis tõi kaasa olulise ajasäästu. Samuti võimaldas vestlusrobot anda samm-sammult juhiseid, näiteks SendGrid konto seadistamiseks, mis samuti vähendas arendusressursside kasutust.

Kokkuvõttes näitab eksperiment, et kuigi vestlusrobot võib tekitada tehnilisi probleeme, aitab selle kasutamine oluliselt vähendada arendusprotsessi kestust ja vajadust väliste infoallikate järele. See omakorda kiirendab nõuetele vastavat veebisaidi loomist ning võimaldab lisafunktsioonide efektiivsemat lisamist.

Analüüsi aruande koostamise käigus on kasutatud ChatGPT tekstiroboti abi, et saada tagasisidet aruande põhiteksti keelekasutuse korrektsuse kohta. Saadud tagasisidest lähtuvalt on töö teksti viimistletud ning keelevead parandatud [39].

## **3.2 Valminud lahenduste tehniline võrdlus**

### **3.2.1 Laadimiskiirus**

PageSpeed Insight on Google tööriist, mis analüüsib veebisaidi laadimiskiirust ja jõudlust nii mobiilsetel seadmetel kui ka lauaarvutites. Analüüsi tulemusena esitatakse skoor skaalal 0–100 neljas peamises kategoorias: jõudlus, ligipääsetavus, parimad praktikad ja otsingumootorite optimeerimine (ingl *Search Engine Optimization*, SEO). Jõudluse kategooria keskendub lehe laadimiskiirusele ja kasutajatele esimese sisu näitamise kiirusele. Ligipääsetavuse kategooria hindab, kui ligipääsetav on lehekülg erivajadustega kasutajatele. Parimad praktikad kategoorias hinnatakse üldiste heade praktikate kohta, mis peaksid olema rakendatud veebilehtedel, mis hõlmab turvalisuse, jõudluse ja käideldavuse aspekte. SEO kategooria keskendub lehe optimeerimisele otsingumootorite jaoks, milles vaadatakse üle, kas leht kasutab efektiivselt metaandmeid [40]. Käesolevas töös kasutatakse PageSpeed Insight tööriista ainult lauaarvuti versiooni analüüsimiseks, kuna veebisaidid ei olnud mõeldud mobiiliseadmetes kuvamiseks.

Analüüsi tulemustest (tabel 1) selgub, et inimarendaja loodud “Meise” ja “Live” vahelehtede jõudlus oli üldiselt parem võrreldes vestlusroboti loodud lehtedega. “Live” vahelehe puhul olid SEO ja ligipääsetavuse tulemused võrdsed. Parimate praktikate

kategoorias näitas vestlusrobot paremaid tulemusi “Live” ja “Hinnakiri” vahelehel, mis rõhutab, et automaatseid lahendusi kasutades võib saavutada teatud aspektides paremaid tulemusi. “Hinnakiri” ja “Kontakt” vahelehtede puhul olid arendusmeetodite jõudlus tulemused võrdsed, kuid parimates praktikates ja SEOs esinesid vestlusroboti loodud lehtedel paremad tulemused. See tõstab esile vestlusrobotite potentsiaali parandada veebilehtede üldist kvaliteeti teatud kategooriates, kuigi teatud olukordades on endiselt vajalik inimarendaja osalus, et tagada tervikliku kvaliteedi.

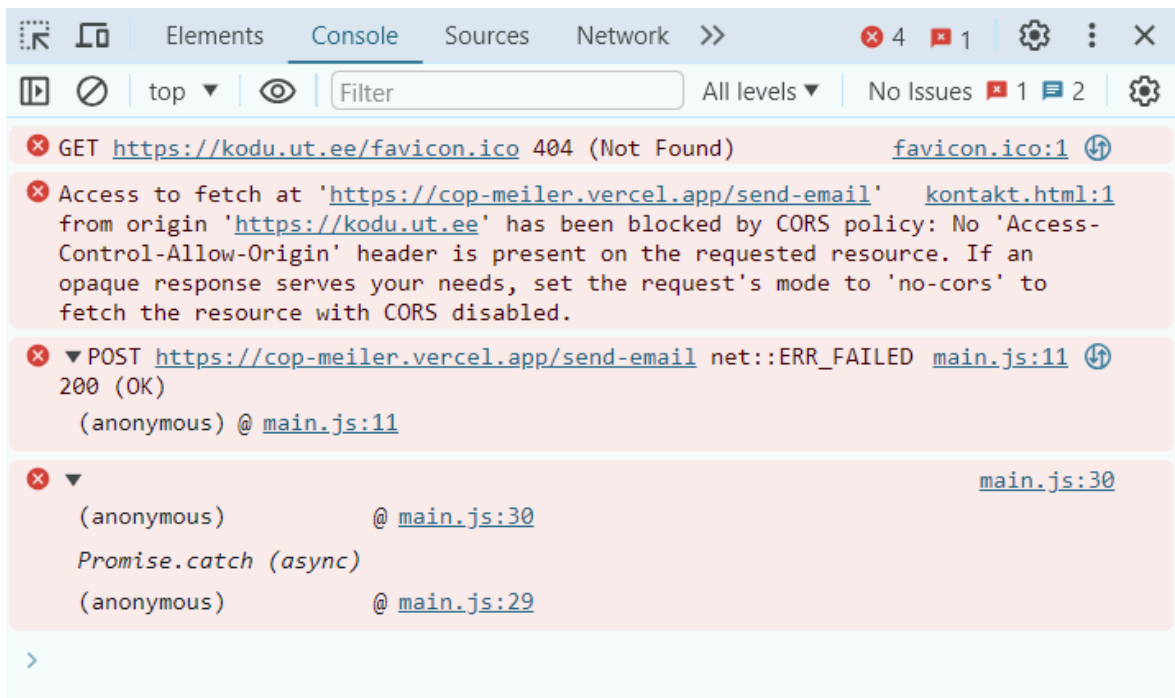
		Inimarendaja poolt valminud lehe hinnang	Github Copilot chat abil valminud lehe hinnang
Meist	Jõudlus	100	72
	Ligipääsetavus	59	76
	Parimad Praktikad	93	96
	SEO	73	82
Live	Jõudlus	82	72
	Ligipääsetavus	67	67
	Parimad Praktikad	81	89
	SEO	91	91
Hinnakiri	Jõudlus	100	100
	Ligipääsetavus	59	76
	Parimad Praktikad	85	93
	SEO	82	91
Kontakt	Jõudlus	99	99
	Ligipääsetavus	82	82
	Parimad Praktikad	89	93
	SEO	91	91

Tabel 1. Google PageSpeed Insight analüüsi tulemuste võrdlus

### 3.2.2 Vigade arv ja tüübid

Kasutades brauseris sisseehitatud arendaja tööriista inspekteerimise võimalusi, analüüsiti mõlema meetodi abil loodud veebisaitide funktsionaalsust. Inimarendaja meetodil loodud veebisaidil kuvatakse veateadet "favicon.ico 404 (Not Found)", mis on seotud kodulehe ikooniga. Kontaktvormi saatmisel ei ilmnenu ühtegi veateadet.

Veebisaidil, mis sai loodud Github Copilot Chat abil, kuvatakse samuti veateadet "favicon.ico 404 (Not Found)" (vt joonis 11). Lisaks esinevad pärast vormi esitamist "Kontakt" vahelehel veel kolm tüüpilist viga. Esimene neist on CORS poliitika rikkumine, mis tekib ressursside laadimisel teisest domeenist ning net::ERR\_FAILED, mis viitab ühendusprobleemidele või päringu tühistamisele, kuigi vastusekood "200 (OK)" justkui kinnitab päringu edukust. Kolmas veateade tekib süsteemi võimetusest tuvastada kirja väljasaatmist, see veateade kuvatakse kasutajal ekraanil.



Joonis 11. Github Copilot Chati abil loodud vahelehe "Kontakt" vead

## Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli analüüsida ja võrrelda GitHub Copilot vestlusroboti rakendamise võimalusi veebiarenduses, tuues välja selle eelised ja erinevused võrreldes inimarendaja tööga. Eksperimendi tulemusel valmisid kahel erineval arendusmeetodil veebisaidid. Uuringu tulemus tõi välja olulised erinevused ajakasutuses. Vaatamata mitmetele tehnilistele tõrgetele, mis nõudsid täiendavaid käsklusi ja täpsustusi, uuringu tulemus tõi välja, et GitHub Copilot Chat suudab oluliselt kiirendada arendusprotsessi, pakkudes samas kvaliteetset koodi ja täiendavaid juhiseid. Töö autor jõudis järelduseni, et vestlusroboti kasutamine nõuab jätkuvalt järelevalvet ja arendajate aktiivse osalemise järele, kuna AI vestlusrobotid ei suuda veel täielikult asendada inimarendajaid, eriti keerukamate arendusülesannete puhul. Vestlusroboti kasutamine ei suuda täielikult asendada inimarendajat ning nõuab koodi mõistmist ja vestlusroboti tegevuse jälgimist, eriti juhtudel, kui vestlusrobot genereerib koodilõike, mis ei vasta olemasolevatele klassinimetustele.

Kuigi tänapäeval saab kasutada erinevaid vestlusroboteid siis üks märkimisväärne eelis GitHub Copiloti vestlusroboti kasutamisel näiteks võrreldes ChatGPT vestlusrobotiga on selle integreeritavus arenduskeskkonnaga. Vestlusroboti kasutamine otse arenduskeskkonnas teeb selle kasutamise oluliselt mugavamaks ning lisab projekti siseseid võimalusi. GitHub Copilot suutis säilitada järjepidevust ning tulla tagasi eelmise viipa juurde, mida ei võimalda ChatGPT vestlusrobot. Suureks eeliseks ChatGPT-4 versiooni mudeliga on see, et see ei oma piiranguid viipade esitamisel ning pakub ka ise viiba soovitusi vastavalt projektile. Lisaks on GitHub Copilot tasuta tudengitele, erinevalt GPT-4 mudelist, pakkudes seeläbi suuremat juurdepääsu ja vähendades kulusid.

Vastates esimesele uurimisküsimusele, töö autor leidis, et GitHubCopilot Chat vestlusroboti abil on võimalik luua veebisait minimaalse sekkumisega, kuigi see nõuab spetsiifilisemate viipade sõnastamist, mis võivad viidata konkreetsele klassile või koodi osale. Samuti ilmnes, et kriitilistes olukordades ei suuda vestlusrobot probleemi lahendada ainult viipade abil, vaid nõuab inimarendaja sekkumist. Vastates teisele uurimisküsimusele eksperimendi tulemused näitasid, et vestlusroboti abil arendusprotsess võttis neli korda vähem aega võrreldes traditsioonilise meetodiga arendades. Vestlusrobot suutis genereerida lihtsamat koodi, mõistes kasutaja küsimusi ja vajadusi. See vestlusrobot oskas järjepidevalt jätkata vestlust ja naasta eelmiste vestlusviipade juurde, ületades selles ChatGPT

vestlusroboti. Vestlusroboti kasutamine arenduskeskkonnas kiirendas oluliselt protsessi ja parandas kasutajakogemust võrreldes varasemate lahendustega.

Vastates viimasele uurimisküsimusele, selgus, et inimarendaja poolt valminud veebisait saavutas üldiselt parema jõudluse, kuid vestlusroboti abil arendatud "Hinnakiri" ja "Kontakt" vahelehed saavutasid samaväärse jõudluse. See tulemus näitab, et teatud kontekstides ja ülesannetes võib GitHub Copilot Chat pakkuda inimarendajaga võrreldavat efektiivsust, eriti standardiseeritud ja korduvate ülesannete puhul.

Töö tulemused näitasid, et kuigi GitHub Copilot Chat võib esitada tehnilisi väljakutseid, aitab selle kasutamine oluliselt vähendada arendusprotsessi kestust ja vajadust väliste infoallikate järele. Lisaks ta pakub eeliseid, mida võivad kasutada nii algajatele kui ka kogenud arendajad veebiarenduse valdkonnas. Algajate jaoks on vestlusrobot eriti väärtuslik, kuna see kirjeldab igat sammu detailsete selgitustega, aidates sellega paremini mõista ja õppida arendusprotsessi. Lisaks annab see otsest juurdepääsu asjakohasele dokumentatsioonile, mis võimaldab kasutajatel kiiresti vajalikku informatsiooni saada. Kogenud arendajatele on vestlusrobot väärtuslik ressurss, mis aitab kiirendada arendusprotsessi, pakkudes kiireid koodijuppide implementatsioone ja asjakohaseid koodisoovitusi. Samuti võib vestlusrobot olla abiliseks veateadete tuvastamiseks ja võimalike lahenduste soovitamiseks. Siiski sõltub tööriista tõhus kasutamine oluliselt inimarendaja pidevast kontrollist ja koodi mõistmisest.

Käesoleva uurimuse piiranguks on selle keskendumine ainult kahele arendusmeetodile, mis ei pruugi kajastada teiste võimalike arendusvahendite või meetodite mõjusid ja efektiivsust veebiarenduses. Samuti võivad uurimuse tulemusi mõjutada autori viibade struktuur ja ülesehitus. Uurimuses keskenduti konkreetsete paika pandud nõuetega veebisaidi loomisele, mistõttu ei pruugi tulemused olla üldistatavad teistele projektidele.

Bakalaureusetöö tulemusi saab kasutada suunistena vestlusrobotite kasutamistel veebiarenduses, eriti algajate korral, kes vajavad täiendavaid abivahendeid korrektse koodi kirjutamisel, samas näidates nende, et inimarendajal peab olema arusaam genereeritud koodist. Käesolev töö pakub ühte esimestest uuringutest GitHub Copilot Chat rakendamisest veebiarenduses. Töö annab aluse edasisteks uurimistöödeks vestlusrobotite ja AI rakendamisest veebiarenduses aidates kaasa uute teadmiste kogumisele ja analüüside läbiviimisele.



## Kasutatud kirjandus

- [1] Coursera Staff. (2024). What Is a Chatbot? Definition, Types, and Examples. <https://www.coursera.org/articles/what-is-a-chatbot> (2.01.2024)
- [2] CM.com. (i.a.) What is an AI chatbot? <https://www.cm.com/glossary/what-is-ai-chatbot/> (2.01.2024)
- [3] Mahmood, S. H. (2023). 7 Free Travel Planning AI and ChatGPT Apps to Get an Instant Itinerary. <https://www.makeuseof.com/free-travel-planning-ai-chatgpt-apps/> (08.01.2024)
- [4] Rokou, T. (2018). Study shows consumers want travel assistance from chatbots. <https://www.traveldailynews.com/regional-news/study-shows-consumers-want-travel-assistance-from-chatbots/> (08.01.2024)
- [5] Zhang, J., Chen, Q., Lu, J., Wang, X., Liu, L., & Feng, Y. (2023). Emotional expression by artificial intelligence chatbots to improve customer satisfaction: Underlying mechanism and boundary conditions. <https://doi.org/10.1016/j.tourman.2023.104835>
- [6] Sergey V. Rybin; Darina S. Ripka. (2023). Overview of Machine Learning Technologies in Medicine. 10.1109/DMEST60476.2023.10339562
- [7] Dante-AI. (2024). Elevating Financial Services: The Role of AI Chatbots. <https://www.dante-ai.com/article/elevating-financial-services-the-role-of-ai-chatbots/> (8.01.2024)
- [8] AIContentfy team. (2023). ChatGPT in action: Real-world use cases. <https://aicontentfy.com/en/blog/chatgpt-in-action-real-world-use-cases> (7.01.2024)
- [9] Selvaraj, S. (2023). Chatbots for education institutions – Use cases & benefits. <https://yellow.ai/blog/chatbots-for-education/> (5.01.2024)
- [10] Merow, C., Serra-Diaz, J. M., Enquist, B. J., & Wilson, A. M. (2023). AI chatbots can boost scientific coding. <https://doi.org/10.1038/s41559-023-02063-3>
- [11] GitHub Docs. (i.a.) About GitHub Copilot Chat. <https://docs.github.com/en/copilot/github-copilot-chat/about-github-copilot-chat> (6.01.2024)
- [12] Zhao, S. (2023). GitHub Copilot Chat now generally available for organizations and individuals.

<https://github.blog/2023-12-29-github-copilot-chat-now-generally-available-for-organizations-and-individuals/#:~:text=December%2029%2C%202023%20At%20GitHub,the%20new%20universal%20programming%20language> (6.01.2024)

[13] Zhao, S. (2023). GitHub Copilot Chat beta now available for all individuals. <https://github.blog/2023-09-20-github-copilot-chat-beta-now-available-for-all-individuals> (6.01.2024)

[14] Rodriguez, M. (2023). Research: Quantifying GitHub Copilot's impact on code quality. <https://github.blog/2023-10-10-research-quantifying-github-copilots-impact-on-code-quality/> (6.01.2024)

[15] Callaham, J. (2023). GitHub Copilot Chat will become generally available in December. <https://www.neowin.net/news/github-copilot-chat-will-become-generally-available-in-december/> (6.01.2024)

[16] Bergström, A. (2023). How to effectively use ChatGPT as a developer. <https://dev.to/andreasbergstrom/mastering-your-ai-coworker-a-developers-guide-to-chatgpt-1gea> (20.02.2024)

[17] Hines, K. (2023). History Of ChatGPT: A Timeline Of The Meteoric Rise Of Generative AI Chatbots. <https://www.searchenginejournal.com/history-of-chatgpt-timeline/488370/> (21.02.2024)

[18] OpenAI. (2022). Introducing ChatGPT. <https://openai.com/index/chatgpt/> (12.03.2024)

[19] OpenAI. (2022). GPT-4. <https://openai.com/research/gpt-4> (12.03.2024)

[20] OpenAI. (2023). ChatGPT can now see, hear, and speak. <https://openai.com/blog/chatgpt-can-now-see-hear-and-speak> (12.03.2024)

[21] OpenAI. (i.a.). ChatGPT. <https://openai.com/chatgpt> (12.03.2024)

[22] OpenAI. (i.a.). What is ChatGPT Plus? <https://help.openai.com/en/articles/6950777-what-is-chatgpt-plus> (10.05.2024)

[23] Aayush. (2024). How to Build a Website With ChatGPT (2024 Guide). <https://www.elegantthemes.com/blog/business/how-to-build-a-website-with-chatgpt> (23.04.2024)

- [24] Abrahamsson, P., Anttila, T., Hakala, J., Ketola, J., Knappe, A., Lahtinen, D., Liukko, V., Poranen, T., & Ritala, T.-M. (2023). ChatGPT as a Fullstack Web Developer - Early Results. [https://doi.org/10.1007/978-3-031-48550-3\\_20](https://doi.org/10.1007/978-3-031-48550-3_20)
- [25] G. Neilinn. (2023). Veebisaidi loomise protsess ChatGPT4 abiga ning ilma. <https://hdl.handle.net/10062/93582>
- [26] Shamoon, M. (2023). EXPERIMENTING WITH GITHUB COPILOT CHAT FOR DEVELOPMENT. <https://mohd-shamoon.medium.com/experimenting-with-github-copilot-chat-for-development-2dffe2b37caf> (9.04.2024)
- [27] Lea, K. (2024). Tips to Use GitHub Copilot for Better Web Development. <https://www.getfishtank.com/blog/github-copilot-essentials-for-web-developers> (09.04.2024)
- [28] Gorbatšenko, P. (i.a.). What are Functional and Non-Functional Requirements and How to Document These. <https://enkonix.com/blog/functional-requirements-vs-non-functional/>
- [29] Experience UX. (i.a.). What is a website prototype? <https://www.experienceux.co.uk/faqs/what-is-a-website-prototype/> (09.04.2024)
- [30] Miniukovitš, A., & Figl, K. (2023). The effect of prototypicality on webpage aesthetics, usability, and trustworthiness. <https://doi.org/10.1016/j.ijhcs.2023.103103>
- [31] Figma Learn. (i.a.). What is Figma? <https://help.figma.com/hc/en-us/articles/14563969806359-What-is-Figma> (9.04.2024)
- [32] Kopf, B. (2018). The Power of Figma as a Design Tool. <https://www.toptal.com/designers/ui/figma-design-tool> (9.04.2024)
- [33] Johnson, B. (2019). Visual Studio Code: End-to-End Editing and Debugging Tools for Web Developers. <https://ebookcentral-proquest-com.ezproxy.utlib.ut.ee/lib/tartu-ebooks/detail.action?docID=5849509> (2.05.2024)
- [34] Challapalli, S. S. N., Kaushik, P., Suman, S., Shivahare, B. D., Bibhu, V., & Gupta, A. D. (2021). Web Development and performance comparison of Web Development Technologies in Node.js and Python. 10.1109/ICTAI53825.2021.9673464.

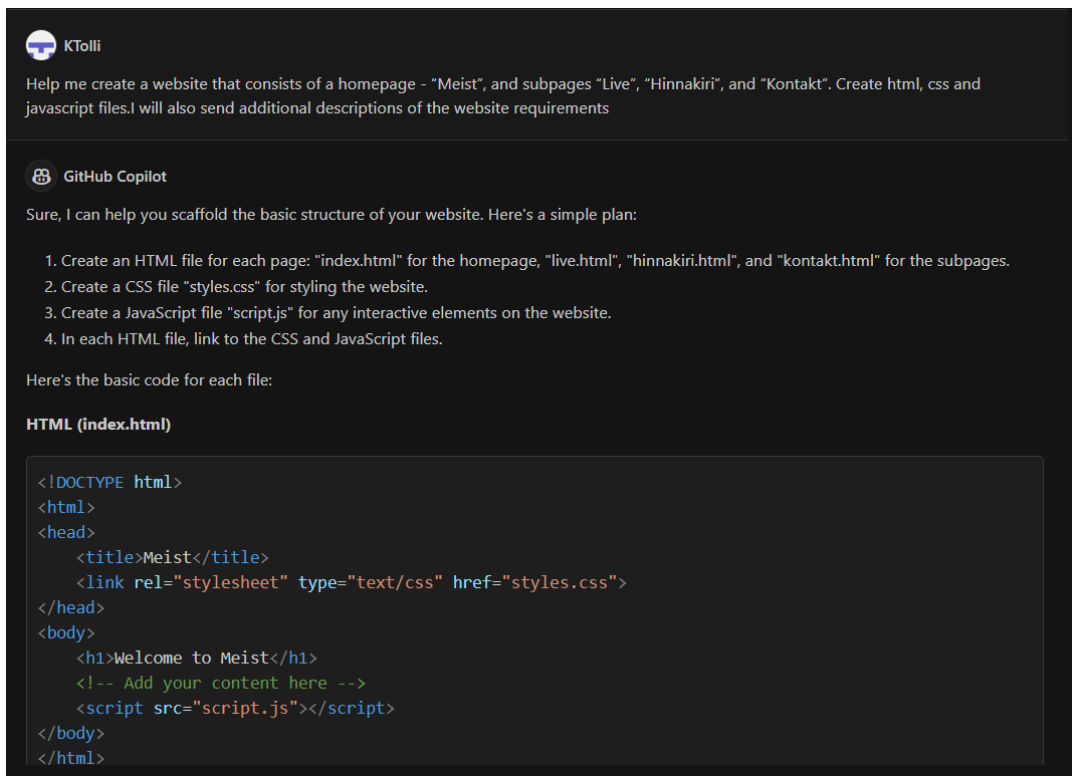
- [35] Node.js contributors. (i.a.). Introduction to Node.js. <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs> (12.05.2024)
- [36] Codecademy Team. (i.a.). What is Express.js? <https://www.codecademy.com/article/what-is-express-js> (15.05.2024)
- [37] SendGrid. (i.a.) Get Started with SendGrid for Free. <https://sendgrid.com/en-us/free> (12.05.2024)
- [38] CodeWalnut. (i.a.) What is Vercel? <https://www.codewalnut.com/learn/what-is-vercel-features-and-benefits> (11.05.2024)
- [39] OpenAI. (2022). ChatGPT (20. detsembri versioon) [suur keelemudel]. <https://chat.openai.com/>
- [40] Wilkinson, M. (2023). Google PageSpeed Insights: What It Is & How to Boost Your Score. <https://www.semrush.com/blog/google-pagespeed-insights/>

## Lisad

**Lisa 1.** GitHub Copilot Chat vestluse ajalugu.

<https://docs.google.com/document/d/1C3SNkAG0ilx-A0lQZuA094G2TD1fT9QwIhdgETAlJmY/edit?usp=sharing>

**Lisa 2.** GitHub Copilot Chat vestlus: Lehtede koodi genereerimine-1.



**KToili**

Help me create a website that consists of a homepage - "Meist", and subpages "Live", "Hinnakiri", and "Kontakt". Create html, css and javascript files. I will also send additional descriptions of the website requirements

**GitHub Copilot**

Sure, I can help you scaffold the basic structure of your website. Here's a simple plan:

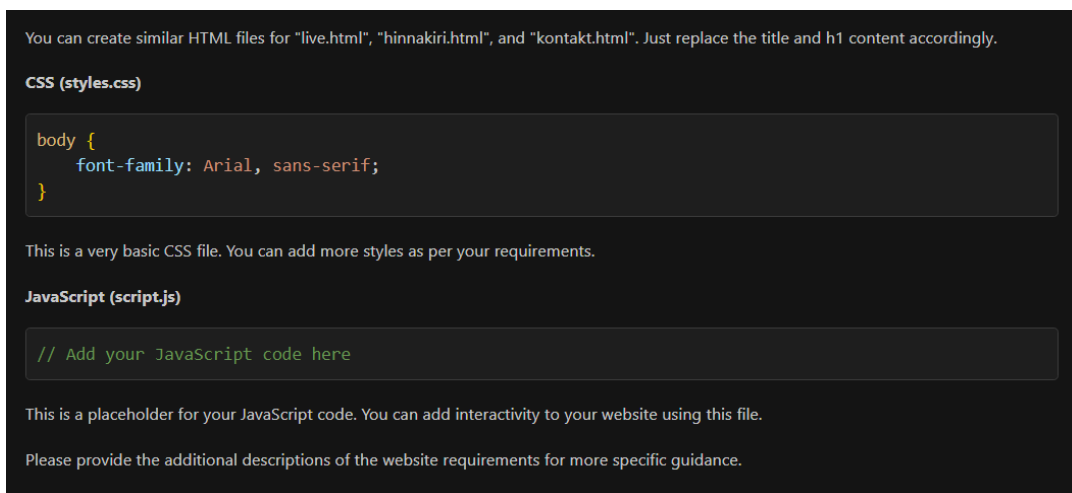
1. Create an HTML file for each page: "index.html" for the homepage, "live.html", "hinnakiri.html", and "kontakt.html" for the subpages.
2. Create a CSS file "styles.css" for styling the website.
3. Create a JavaScript file "script.js" for any interactive elements on the website.
4. In each HTML file, link to the CSS and JavaScript files.

Here's the basic code for each file:

**HTML (index.html)**

```
<!DOCTYPE html>
<html>
<head>
  <title>Meist</title>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
  <h1>Welcome to Meist</h1>
  <!-- Add your content here -->
  <script src="script.js"></script>
</body>
</html>
```

**Lisa 3.** Github Copilot Chat vestlus: Lehtede koodi genereerimine-2.



You can create similar HTML files for "live.html", "hinnakiri.html", and "kontakt.html". Just replace the title and h1 content accordingly.

**CSS (styles.css)**

```
body {
  font-family: Arial, sans-serif;
}
```

This is a very basic CSS file. You can add more styles as per your requirements.


**JavaScript (script.js)**

```
// Add your JavaScript code here
```


This is a placeholder for your JavaScript code. You can add interactivity to your website using this file.

Please provide the additional descriptions of the website requirements for more specific guidance.

#### Lisa 4. Github Copilot Chat vestlus: “Meist” lehel sisuploki kohandamine.

 KTolli

add similar code for meist to fit page

 GitHub Copilot


Used 1 reference >

To make the `meist` page fit the viewport similar to the `.content-block-1` and `.wider-box` classes, you can create a new class (let's call it `.meist-content`) and apply similar styles. Here's how you can do it:


```
.meist-content {
  background-color: #a2b799;
  border-radius: 20px;
  width: 60%;
  height: calc(100vh - 150px - 150px); /* Adjust the header and footer height accordingly */
  margin: 0 auto; /* Center the block horizontally */
  display: flex; /* Enable Flexbox */
  justify-content: center; /* Center content horizontally */
  align-items: center; /* Center content vertically */
  flex-direction: column; /* Stack the content vertically */
}
```

In this setup, the `.meist-content` class has the same styles as the `.content-block-1` and `.wider-box` classes. You can apply this class to the main content of the `meist` page in your HTML file to make it fit the viewport.

#### Lisa 5. Github Copilot Chat vestlus: viip kontaktvormi loomiseks.

 KTolli

form requirement. Upon pressing the button "saada", the entered data is sent to the email `karina.tolli@ut.ee` and displays the message "Aitäh, info on saadetud"

 GitHub Copilot

Used 1 reference >

To achieve this, you can use a form with an action attribute set to a mailto: URL. However, this method has limitations and is not recommended for production websites. It depends on the user's email client settings and might not work in all cases.

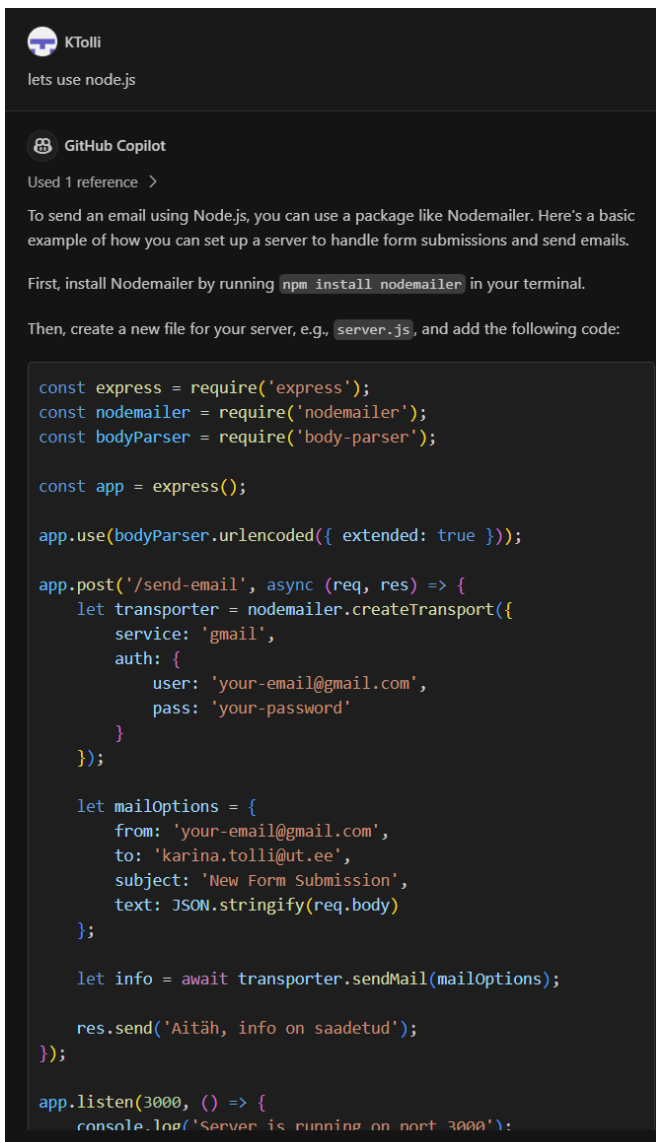
Here's a simple example:

```
<form action="mailto:karina.tolli@ut.ee" method="post" enctype="text/plain">
  Name:<br>
  <input type="text" name="name"><br>
  E-mail:<br>
  <input type="text" name="mail"><br>
  Comment:<br>
  <input type="text" name="comment" size="50"><br><br>
  <input type="submit" value="saada" onclick="alert('Aitäh, info on saadetud')">
</form>
```

In this setup, when the user clicks the "saada" button, their email client will open with a new email drafted to `karina.tolli@ut.ee` and the form data included in the body of the email. The `alert('Aitäh, info on saadetud')` will display the message "Aitäh, info on saadetud".

For a more reliable solution, you would need to use a server-side language like PHP, Node.js, Python, etc., to handle the form submission and send the email. This is beyond the scope of a simple HTML form and would require server-side programming knowledge.

## Lisa 6. Github Copilot Chat vestlus: viip Node.js kasutamiseks.



## Litsents

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

Mina, **Karina Tolli**,

annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose **“GitHub Copilot Chat vestlusroboti rakendamine veebiarenduses: Praktiline võrdlus inimarendaja tööga,”** mille juhendaja on **Lidia Feklistova**, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

1. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i 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.
2. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

*Karina Tolli*

**15.05.2024**