

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Janar Kallson

Veebisaidi loomine Dr. Diana Kirss OÜ-le

Bakalaureusetöö (9 EAP)

Juhendaja: Lidia Feklistova, MSc

Tartu 2021

Veebisaidi loomine Dr. Diana Kirss OÜ-le

Lühikokkuvõte:

Käesolevas bakalaureusetöös luuakse veebisait Dr. Diana Kirss OÜ-le. Sisu operatiivsemaks ja mugavamaks haldamiseks on lisatud administraatori liides. Töö sisaldab eelneva veebisaidi puudujääkide analüüsi, uue veebisaidi funktsionaalsete ja mittefunktsionaalsete nõuete nimekirja, ülevaadet kasutatud tehnoloogiatest ning kokkuvõtet valminud veebisaidist. Samuti on antud ülevaade sooritatud testidest ja võimalikest arengusuundadest.

Võtmesõnad:

Veebisait, Vue.js, Vuetify, Node.js, Express, GraphQL, MongoDB

CERCS: P175 Informaatika

Creating a website for Dr. Diana Kirss OÜ

Abstract:

The aim of the bachelor thesis is to create a website for Dr. Diana Kirss OÜ with administrator interface. This thesis also contains previous website analysis, list of functional and non-functional requirements, overview of used technologies, created website and information about tests that were made on the website. The thesis ends with ideas on how created website could be improved.

Keywords:

Website, Vue.js, Vuetify, Node.js, Express, GraphQL, MongoDB

CERCS: P175 Informatics

Sisukord

1.	Sissejuhatus	4
2.	Eelnev veebisait	5
3.	Nõuete analüüs	7
3.1	Kliendi poolt esitatud nõuded	7
3.2	Paberil maketti loomine	9
3.3	Disaini loomine	12
3.4	Prototüübi loomine.....	13
4.	Kasutatud tehnoloogiad.....	14
4.1	Vue.js	14
4.2	Vuetify.....	15
4.3	Vuex	17
5.	Valminud veebisaidi ülevaade	20
5.1	Päis ja jalus.....	20
5.2	Avaleht	20
5.3	Kodukorrast.....	23
5.4	Tedaanded	23
5.5	Administraatori vaade	24
5.6	API loomine	26
5.6.1	GraphQL	26
5.6.2	Autentimine.....	30
5.7	Otsingumootori optimeerimine	32
5.8	Loodud veebisaidil teostatud testid.....	34
5.8.1	Operatsioonisüsteemide ning brauserite testid.....	35
5.8.2	Mobiilisõbralikkus	35
5.8.3	Lighthouse raportid	35
6.	Võimalikud arengusuunad	37
7.	Kokkuvõte	38
8.	Kasutatud kirjandus.....	39
Lisad.....		41
I.	Vana veebisaidi pildid.....	41
II.	Disain	44
III.	Uue veebisaidi pildid.....	47
IV.	Litsents	50

1. Sissejuhatus

Dr. Diana Kirss OÜ asutas Ahja perearstikabineti 2008. aastal. Tegemist on meditsiiniteenuseid osutava asutusega. Oma patsientide paremaks informeerimiseks ja kättesaadavamaks muutmiseks pööras firma tähelepanu veebikanalitele ja võimalustele. Esimene versioon veebisaidist oli loodud 2020. aastal. Vana veebisait on nüüdseks jäänud poolikuks nii sisult, kui ka ülesehituselt. Infoajastu ja pandeemia mõjutusel otsustas perearstikabinet oma veebisaidile suuremat rõhku pöörata.

Bakalaureusetöö on rakenduslik töö ning selle eesmärk on kaasajastada Ahja perearstikabineti veebisaiti, et olemasolevatel või tulevastel külastajatel oleks võimalik mugavalt saada asjakohast informatsiooni interneti teel - varasemalt oli põhjaliku informatsiooni saamine võimalik ainult helistades või kohapeale minnes. Tänu uuendatud veebisaidile, saab klient juba enne külastamist ennast viia kurssi perearstikeskuse kodukorraga ning hinnakirjaga. Samuti võimaldab loodav veebisait jätta Ahja perearstikabineti töötajatel erinevaid teadaandeid oma klientidele või veebisaidi külastajatele ning külastajatel on võimalus saata küsimuste korral kiri töötajatele. Veebisaidi kliendipoolle (ingl. k. *frontend*) loomiseks kasutatakse Vue tehnoloogiat, tagateenuse (ingl. k. *backend*) loomiseks kasutatakse Node.js-i, Expressi ning GraphQL-i ning andmebaasina MongoDB-d.

Töö koosneb mitmest peatükist:

- esimeses peatükis analüüsitakse vana veebisaiti,
- teises peatükis on toodud kliendi nõuded uuele veebisaidile,
- kolmas peatükk räägib kasutatud tehnoloogiatest,
- neljas peatükis antakse ülevaade valminud veebisaidist,
- töö lõpus on välja toodud võimalikud arengusuunad.

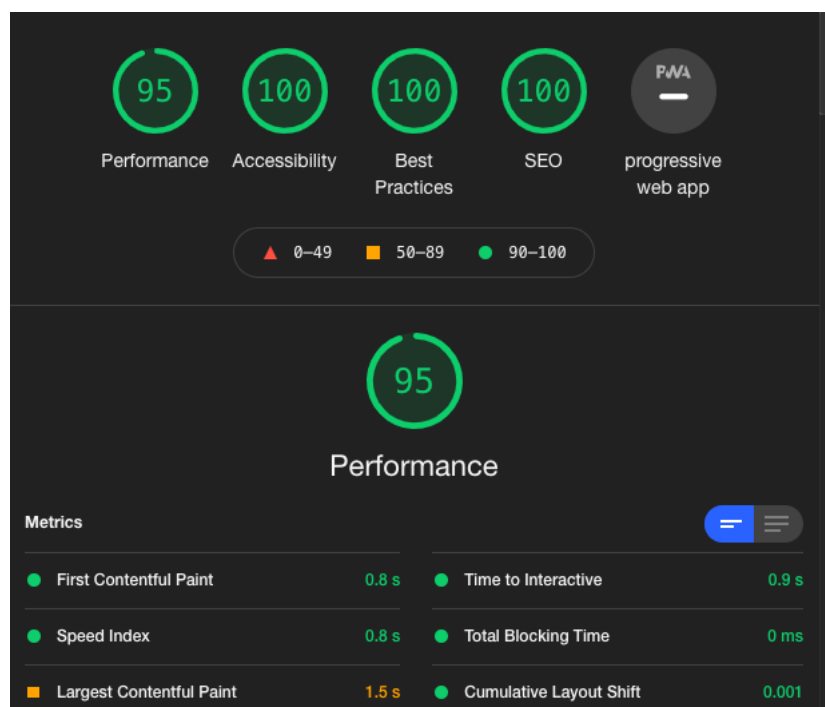
2. Eelnev veebisait

Enne uue veebisaidi loomist analüüsis töö autor varasemalt loodud veebilehte, et leida ülesse kohad, mis vajavad arendamist ning mida tuleb arvesse võtta uue veebisaidi loomisel. Järgnevalt on välja toodud autori poolt avastatud murekohad ning on pakutud võimalikud lahendused, mida realiseeritakse veebisaidi uuendamisel (pikem nõuete analüüs uue veebisaidi vastu on toodud järgmises peatükis):

- veebisaidi esimesel laadimisel võttis info kuvamine umbes neli sekundit aega, uue veebisaidi loomisel parandatakse esialgse laadimise kiirust või vajadusel kuvatakse veebilehe külastajale, et veebileht on hetkel veel laadimas;
- varasema saidi avalehel oli kohesest hästi palju infot ning selle tõttu oli keeruline koheselt aru saada sellest, et mis informatsiooni sealt leiab (vt. Lisa 1 joonis 34). Uuel veebisaidil jaotatakse informatsioon paremini sektsioonidesse, et kasutajal oleks võimalik kiirelt leida otsitavat informatsiooni;
- Avalehel puudusid pildid, uuele veebisaidile lisatakse pilte, et veebisait kaasahaaravam tunduks;
- Avalehe vahelehel olevad pealkirjad olid joondatud vasakule (vt. Lisa 1 joonis 34), uuel veebisaidil joondatakse pealkirjad keskele;
- Töötajate vahelehel oli väga palju valget tausta ning töötajaid ei olnud hästi välja toodud (vt. Lisa 1 joonis 35). Uue lehe loomisel töötajatele eraldi vahelehte ei tehta, vaid töötajad lisatakse koheselt avalehele, et anda veebisaidile terviklikum tunne;
- Töötajate vahelehel oli vertikaalne kerimisriba, mida tegelikult seal vaja ei läinud;
- Hinnakirja vahelehel oli liiga palju tühja ruumi ning uuel veebisaidil paigutatakse hinnakiri avalehele (vt. Lisa 1 joonis 37);
- Kasulikud lingid vahelehel oli tabelis igale lingile lisatud juurde seletus sektsioon, kuhu polnud midagi kirjutatud (vt. Lisa 1 joonis 38). Uuel veebisaidil elimineeritakse seletuse osa ning jäetakse alles ainult klikitavad lingid;
- Hinnakiri ning kasulikud lingid vahelehel olev info ei eristunud taustast ning oli raske aru saada kus algab informatiivne osa ning kus on taust. Uuel veebisaidil peab olema tausta ning informatiivse osa vahel suurem värvivahe;
- kõikidel vahelehtedel puudus jalus. Uuele lehele peab lisama jaluse, kust on võimalik näha Ahja perearstikabineti kontaktandmeid;

- koodibaasi analüüsidest oli näha, et arendamisel oli kasutatud läbisegi nii eesti kui ka inglise keelt. Uue veebilehe arendamise juures hoitakse ühtset stiili ning töö autor otsustas, et uue lehe arenduse käigus kasutatakse ainult inglise keelt;
- veebisaidil puudus võimalus saata Ahja perearstikabinetile kirja. Nii klient kui ka töö autor leidis, et uuele veebisaidile tuleb emaili saatmise funktsionaalsus kindlasti lisada.

Töö autor kasutas ka Lighthouse-t Google Chrome DevToolsist, et näha, mis on selle tööriista arvates lehe hetke kvaliteet. Lighthouse on tööriist, mille abil on võimalik analüüsida milline on veebisaidi jõudlus (ingl. k. *performance*), ligipääsetavus, otsingumootori optimeerimine, kas on kasutatud häid arendustavasid jne [1]. Lighthouse tulemus (vt. Joonis 1) oli väga hea, millest võib järeldada, et vana veebisait toimib hästi. Vaatamata sellele olid kliendil tekkinud uued soovid nii funktsionaalsuse kui ka veebisaidi välimuse osas. Koodiga tutvudes leidis autor, et nende realiseerimine vana koodi baasil on raskendatud ning mõistlik on luua täiesti uus lahendus.



Joonis 1. Vana veebisaidi Lighthouse raport

3. Nõuete analüüs

Antud peatükis on toodud ülevaade kliendipoolsetest nõuetest ja nende analüüsist. Kliendiga sai läbi viidud koosolek, kus vaadeldi koos varasemalt loodud veebisaiti ning analüüsiti, mida saaks paremini teha ning mida soovitakse muuta ja lisada. Koosoleku käigus pani töö autor kirja kliendi soovid ning nõuded. Peale koosolekut jagati nõudmised funktsionaalseteks ning mittefunktsionaalseteks nõueteks.

3.1 Kliendi poolt esitatud nõuded

Kliendi funktsionaalsed nõuded uuendatud veebisaidi kohta on esitatud tabelites 1-10.

Nõude kood	Nõue
REQ 1	Administraatorina peab olema võimalik lisada andmeid tööle olevatest töötajatest
REQ 2	Administraatorina peab olema võimalik muuta andmeid tööle olevatest töötajatest
REQ 3	Administraatorina peab olema võimalik kustutada andmeid tööle olevatest töötajatest
REQ 4	Iga töötaja nime all peab olema nupp, mille abil on võimalik näha töötaja vastuvõtuaegu
REQ 5	Iga töötaja nime all peab olema nupp, mille abil on võimalik näha töötaja telefoniaegu

Tabel 1. Funktsionaalsed nõuded töötajate kohta

Nõude kood	Nõue
REQ 6	Administraatorina peab olema võimalik muuta hinnakirjas olevaid hindu
REQ 7	Administraatorina peab olema võimalik lisada hinnakirja hindu
REQ 8	Administraatorina peab olema võimalik kustutada hinnakirjas olevaid hindu
REQ 9	„Hinnakiri“ osas toodud tõendite, visiitdidasude ning vaktsineerimiste hinnad peavad olema esitatud tabelites
REQ 10	Veebisaidil peab olema nupp, mille vajutamisel suunatakse hinnakiri lehele

Tabel 2. Funktsionaalsed nõuded hinnakirja kohta

Nõude kood	Nõue
REQ 11	Administraatorina peab olema võimalik muuta kasulikke linke
REQ 12	Administraatorina peab olema võimalik eemaldada kasulikke linke
REQ 13	Administraatorina peab olema võimalik juurde lisada kasulikke linke
REQ 14	Administraatorina peab olema võimalik ümber järjestada kasulikke linke

Tabel 3. Funktsionaalsed nõuded kasulikke linkide kohta

Nõude kood	Nõue
REQ 15	Administraatorina peab „Meist“ osasse olema võimalik lisada infot
REQ 16	Administraatorina peab „Meist“ osa infot olema võimalik muuta
REQ 17	Administraatorina peab „Meist“ osa infot olema võimalik kustutada
REQ 18	Administraatorina peab olema võimalik „Meist“ lehel olevat teksti vormindada

Tabel 4. Funktsionaalsed nõuded "Meist" osa kohta

Nõude kood	Nõue
REQ 19	Veebisaidil peab olema sektsioon teadaannete jaoks
REQ 20	Veebisaidil peavad teadaanded olema järjestatud avaldamise kuupäeva aluses – uuemad teadaanded on eespool, vanemad tagapool
REQ 21	Iga teadaande juures peab olema nähtav avalikustamise kuupäev
REQ 22	Administraatorina peab olema võimalik lisada uusi teadaandeid
REQ 23	Administraatorina peab olema võimalik muuta eelnevalt lisatud teadaandeid
REQ 24	Administraatorina peab olema võimalik teadaandeid ära kustutada
REQ 25	Administraatorina peab olema võimalik vormindada teadaande teksti

Tabel 5. Funktsionaalsed nõuded teadaannete kohta

Nõude kood	Nõue
REQ 26	Ahja perearstikabineti asukoha näitamiseks peab olema kasutatud Google Mapsi kaarti
REQ 27	Administraatorina peab olema võimalik muuta Ahja perearstikabineti kontakttelefoninumbrit
REQ 28	Administraatorina peab olema võimalik muuta Ahja perearstikabineti lahtiolekuaegu
REQ 29	Administraatorina peab olema võimalik muuta Ahja perearstikabineti aadressi
REQ 30	Administraatorina peab olema võimalik muuta Ahja perearstikabineti kontaktemaili
REQ 31	Veebisaidil peab olema kontaktvorm, mille abil on võimalik külastajatel saata Ahja perearstikabinetile kiri
REQ 32	Veebisaidil olevas kontaktvormis peavad olema väljad saatja nime, emaili, teema ning sisu jaoks
REQ 33	Kirja saatja ei pea saama koopiat saadetud kirjast
REQ 34	Kontaktvormis olevat kirja ei pea saama vormindada

Tabel 6. Funktsionaalsed nõuded kontakti kohta

Nõude kood	Nõue
REQ 35	Veebisaidil peab olema nupp, mille vajutamisel suunatakse kodukorra lehele
REQ 36	Kodukorra lehelt peab olema võimalik alla tõmmata nimistusse registreerimise blankett
REQ 37	Administraatorina peab olema võimalik lisada kodukorda teksti
REQ 38	Administraatorina peab olema võimalik muuta kodukorras olevat teksti
REQ 39	Administraatorina peab olema võimalik kustutada kodukorras olevat teksti

Tabel 7. Funktsionaalsed nõuded kodukorra kohta

Nõude kood	Nõue
REQ 40	Veebisaidi jaluses peab olema koht Ahja perearstikabineti lisamiseks
REQ 41	Veebisaidi jaluses peab olema koht Ahja perearstikabineti telefoni lisamiseks
REQ 42	Veebisaidi jaluses peab olema koht Ahja perearstikabineti emaili lisamiseks
REQ 43	Veebisaidi pealdises peab olema „Meist“ nupp
REQ 44	Veebisaidi pealdises peab olema „Töötajad“ nupp
REQ 45	Veebisaidi pealdises peab olema „Hinnakiri“ nupp
REQ 46	Veebisaidi pealdises peab olema „Kontakt“ nupp

Tabel 8. Funktsionaalsed nõuded pealdise/jaluse kohta

Nõude kood	Nõue
REQ 47	Veebisait peab olema heledates toonides, värvi lisamiseks peab kasutama oranži
REQ 48	Veebisait olev tekst peab olema vähemalt 12px, kuna väiksemat teksti võib vanematel inimestel olla raske lugeda

Tabel 9. Üldised funktsionaalsed nõuded

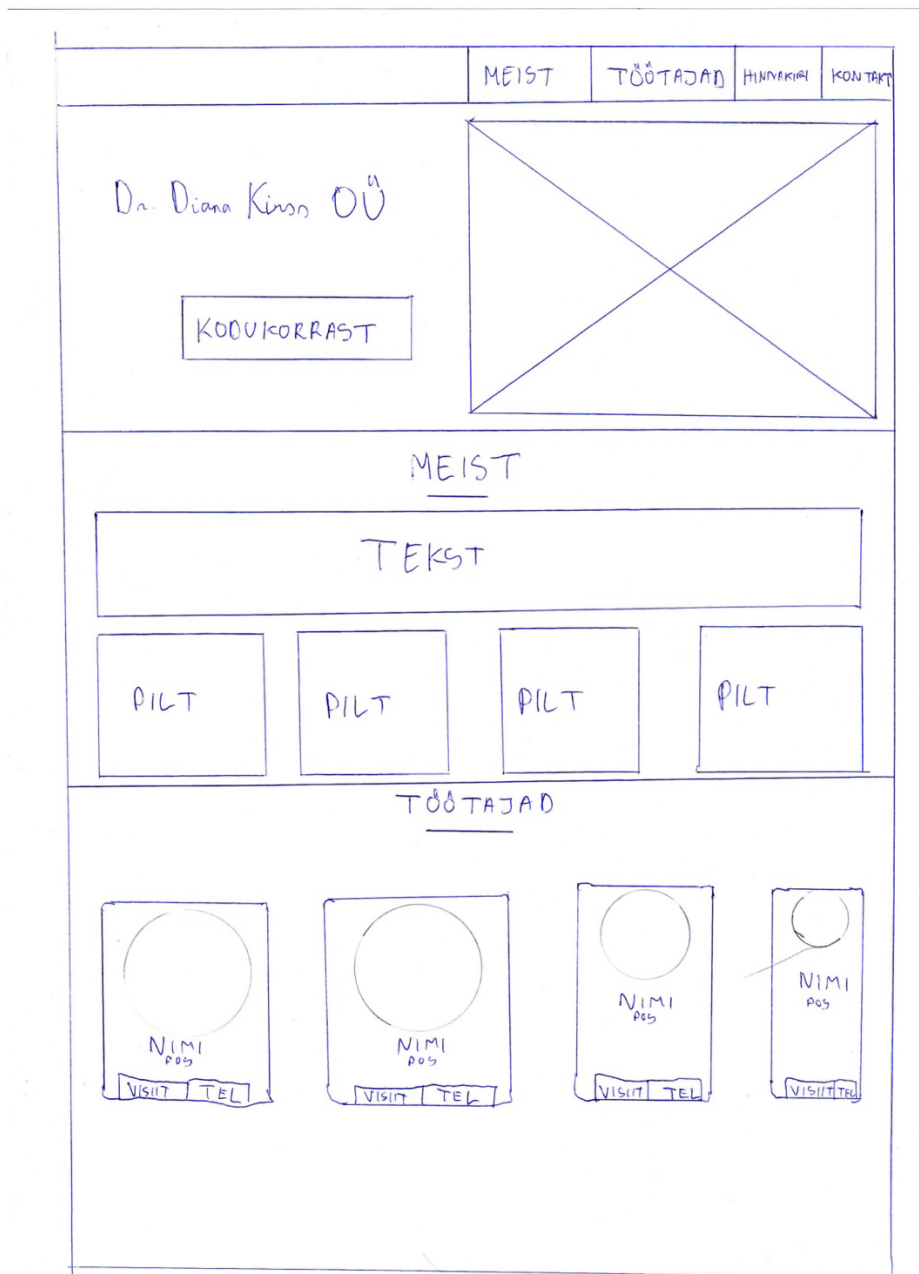
Nõude kood	Nõue
REQ 49	Veebisait peab olema mobiilisõbralik
REQ 50	Veebisait peab toetama Google Chrome't
REQ 51	Veebisait peab toetama Mozilla Firefox'i
REQ 52	Veebisait ei pea toetama Internet Explorer'it

Tabel 10. Mittefunktsionaalsed nõuded

3.2 Paberil maketti loomine

Peale nõuete kogumist alustas töö autor esialgse maketti (ingl. k. *mockup*) loomisega. Maketti loomise ajal lähtus töö autor kliendi poolt esitatud nõuetest. Maketi loomise eesmärgiks oli paika saada esialgne visioon sellest, milline võiks loodav veebisait tulevikus

välja näha. Makett loodi ainult avalehele, kuna avalehel on peaaegu kogu veebisaidi info kirjas. Töö autor ei pidanud vajalikuks teiste vahelehtede makette valmistada, kuna selleks hetkeks oli loodava veebisaidi visioon saavutatud ning hiljem prototüüpi luues saab seda visiooni kasutada vahelehtede loomiseks. Loodud maketti esitleti ka kliendile ning klient oli rahul loodud disainiga. Esialgne makett nägi välja järgnev (joonised 2a ja 2b moodustavad ühtse terviku):



Joonis 2a. Avalehe makett (lehe algus)

HINNAKIRI

TÕENDID

TEENUS	HIND

VISIIDITASUD

TEENUS	HIND

VAKTSINEERIME

TEENUS	HIND

VÕTA ÜHENDUST

TEKST

NIMI

EMAIL

SAADA

ADDRESS
TELEFON
EMAIL

Joonis 2b. Avalehe makett (lehe lõpp)

3.3 Disaini loomine

Peale paberil maketti loomist alustas töö autor veebisaidi disaini loomist. Loodava veebisaidi disain sai loodud kasutades Figma keskkonda (<https://www.figma.com/>). Disaini aluseks sai võetud varasemalt paberil loodud makett ning kliendi poolne nõue REQ47, mille alusel peab veebisait olema heledates toonides ning värvi lisamiseks võib kasutada näiteks oranži. Kõige esimesena sai loodud veebisaidi avalehe disain. Avalehele tulles näevad kasutajad koheselt nuppe, mis viivad kasutajad kodukorra juurde või teadaannete juurde. Kliendiga arutledes olid need vahelehed nimetatud kõige olulisemateks. Sellele järgneb osa “Meist”, kus on informatsiooni Ahja perearstikabineti kohta ning mõned pildid sellest kabinetist. Edasi on “Töötajate” osa, kus on nähtaval kõik Ahja perearstikabineti töötajad ning samuti on võimalik vaadata iga töötaja vastuvõtuaegu ning telefoni aegu. Järgmisena on toodud hinnakiri, kus on välja toodud kõikide pakutavate teenuste hinnad. Avalehe lõpus on Google Maps Ahja perearstikabineti asukohaga, kontaktandmed ning vorm Ahja perearstikabinetile kirja saatmiseks.

“Kodukorra” vahelehel on info kuvamiseks kasutatud lipikute (ingl. k. *tab*) süsteemi. Lipikute kasutamine on selles olukorras mõistlik, sest kuna infot on palju, siis kogu info üksteise alla panemine võib tekitada külastajal raskusi nende poolt otsitava info kättesaamisega. Luues võtmesõnadega lipikud leiab külastaja kiirelt üles vajamineva. Samuti võimaldab lipikute kasutamine kokku hoida ruumi.

Viimaseks loodud vaheleheks oli “Teadaanded” vaheleht. “Teadaanded” vaheleht näeb välja sarnane “Kodukorrast” vahelehe, kuid siin oli kasutatud laienduspaneeli (ingl. k. *expansion panel*), mille abil on võimalik hoida kokku vertikaalset ruumi kui on vaja kuvada palju informatsiooni [2].

Loodud disaini sai esitletud kliendile ning klient jäi rahule loodud disainiga. Töö autor ei loonud eraldi disaini mobiilide jaoks, kuna Vuetify’st kasutatud komponendid on koheselt mobiilisõbralikud. Loodud disaini pildid on nähtaval Lisad II osas.

3.4 Prototüübi loomine

Peale disaini loomist lõi töö autor prototüübi, kasutades selleks Marveli keskkonda (<https://marvelapp.com/>). Prototüübi loomiseks tuli sisestada ekraanipildid varasemalt loodud disainist ning peale seda sai ekraanipiltide peal olevad nupud teha dünaamiliseks. Peale nuppude dünaamiliseks tegemist andis prototüüp kliendile edasi selle, milline näeb uus veebisait välja ning kuidas erinevate vahelehtede vahel liikumine käib. Töö autor tõi kliendile välja selle, et näiteks “Kodukorrast” vaatest liikudes päise abil “Töötajad” osasse, siis päriselt loodavas veebisaidis viib see nupp tagasi avalehele ning kerib töötajate juurde, mitte ei jää lihtsalt avalehe algusesse. Kliendilt tagasiside oli positiivne ning klient jäi väga rahule loodud prototüübiga. Prototüüpi on võimalik vaadata aadressilt <https://bit.ly/3rWmEeF>.

4. Kasutatud tehnoloogiad

Käesolevas peatükis annab töö autor ülevaate kasutatud tehnoloogiate kohta, nende lähiajaloo kohta ning põhjendab tehnoloogiate valikut.

4.1 Vue.js

Veebisaidi loomiseks kasutab töö autor kasutajaliideste loomiseks Vue tehnoloogiat. Vue on progressiivne raamistik kasutajaliideste loomiseks [3]. Esimene versioon jõudis avalikkuseni aastal 2013 ning selle raamistiku loojaks oli Evan You, kes oli eelnevalt töötanud Google's tarkvara arendajana [4]. Katsetades uut reaktiivsussüsteemi ning avastades, et see töötab kohati paremini kui väga sarnase raamistiku Angular.js-i reaktiivsussüsteem, otsustas Evan, et võiks loodud süsteemi internetti üles panna ning lasta ka teistel inimestel seda kasutada. Nüüdseks ajaks on üles pandud rakendus (ehk Vue.js) üle maailma väga laialt kasutuses ning väga armastatud tööriist [4]. Kuigi esialgu tegeles Vue.js-i arendamisega Evan küll ainult üksinda, siis nüüdseks ajaks on Vue.js tiimis üle 30 erineva arendaja [5]. Vue on väga võimas programmeerimiskeel veebisaitide loomiseks. Paljud arendajad eelistavad seda sellepärast, et Vue lubab luua erinevaid komponente ning neid veebisaidi erinevates kohtades ära kasutada – selle tõttu ei pea veebisaidi erinevates sektsioonides kirjutama duplikaat koodi, vaid saab kasutada varasemalt kirjutatud komponente. Tänu sellele väheneb kirjutatud koodi hulk, mis parandab koodi loetavust ning hallatavust.

Vue.js on väga sarnane tuntud raamistikele nagu React.js ning Angular.js. Põhilised sarnasused on [6]:

- aktiivne kasutajatugi;
- komponentidele orienteeritud kood;
- komponentide mitmekesisus.

On oluline märkida, et Vue.js-i, React.js-i ning Angular.js-i raamistikutel on erinevad omanikud. React.js loodi 2011. aastal, kui Facebookis töötav tarkvara arendaja nimega Jordan Walke alustas selle arendamisega ning selle raamistiku omanikuks ongi Facebook [7]. Angular.js-i haldajaks on aga Google [8]. Vue.js raamistik ei ole omatud mingi suure ettevõtte poolt ning selle tõttu on neil palju suurem vabadus oma ideede realiseerimisel. Sellel on ka omad miinused. Kuna React.js ning Angular.js on toetatud suurte firmade poolt, siis on ka neil lihtsam leida rahastust. Vue.js-l polnud selliseid omanikke, seega oli neil

raskem leida töötajaid ning finantsi erinevate asjade jaoks (näiteks reisid konverentsidele, oma raamistiku reklaamimine) [9].

Töö autor valis Vue sellepärast, et Vue on väga populaarne raamistik, sellel on põhjalik dokumentatsioon ning Vue template'd on kirjutatud HTML-s, mis on varasemalt töö autorile juba tuttav. Teiseks on Vue-s võimalik luua taaskasutatavaid komponente, mille abil väheneb duplikaatkoodi kirjutamise hulk, sest saab kasutada mitmes kohas varasemalt loodud komponente. Samuti pakkus Vue töö autorile huvi ning ta tahtis selles keeles paremaks arendajaks saada.

4.2 Vuetify

Veebisaidi loomisel kasutab autor kasutajaliidese raamistikuna Vuetify-d, mis põhineb Vue.js-l [10]. Vuetify esialgne väljalase oli aastal 2014 ning nüüdseks on Vuetify kasvanud üheks kõige populaarsemaks JavaScripti raamistikuks terves maailmas [10]. Vuetify pakub arendajale erinevaid komponente, mida on lihtne oma koodi implementeerida ning sellega arendusprotsessi oluliselt kiirendada. Näiteks on vaja veebisaidile disainida mingisugune nupp koos teatud funktsionaalsusega. Kasutades lihtsalt Vue.js-i on vaja luua nupp ning see disainida (on vaja täpselt paika panna suurused, on vaja mõelda selle peale, et mis juhtub selle nupuga siis, kui kasutatakse näiteks nutitelefoni). Kuid võttes kasutusele Vuetify kasutajaliidese raamistiku, on nupu loomine oluliselt lihtsam – Vuetify veebisaidil toodud dokumentatsioonist tuleb otsida peatükki „Button“ ning tutvuda olemasolevate näidetega sellest, kuidas nuppu luua ning kuidas saab selle stiili muuta. Tänu sellisele dokumentatsioonile saavad arendajad tegeleda rohkem arendusprotsessiga ning ei pea kulutama liigselt aega väga madala taseme disainimise peale. Vuetify-d kasutades tuleb kopeerida nupu kood ning asetada see enda veebisaidi koodi ning ülejäänud on juba Vuetify poolt kaetud. Samuti on Vuetify poolt antud komponente väga lihtne muuta enda soovide järgi, sest igal komponendil on propsid (ingl. k. *properties*). Propside abil on võimalik muuta komponentide erinevaid aspekte. Propsid võivad olla järgnevat tüüpi [11]:

- *string*,
- *number*,
- *boolean*,
- *array*,
- *object*,
- *date*,
- *function*,
- *symbol*.

Kui arendaja tahaks luua punast värvi nuppu, siis selleks peab ta leidma Vuetify-st nupu koodi ning panema leitud koodi enda koodi sisse. Samuti tuleb nupu puhul ära defineerida *color prop* ning anda talle väärtuseks „red“ (vt. Joonis 3). Selle tulemusel muutub nupp punaseks.

```
<v-btn  
color="red"  
>  
Punane nupp  
</v-btn>
```

Joonis 3. Punase nupu näide

Samas, kui arendaja tahaks nüüd sinist nuppu, siis ei peaks arendaja tegema muud, kui ainult muutma *color prop*-i väärtust ning tulemuseks olekski sinine nupp (vt. Joonis 4).

```
<v-btn  
color="blue"  
>  
Sinine nupp  
</v-btn>
```

Joonis 4. Sinise nupu näide

Vuetify puhul on võetud lähenemine, et mobiilirakendused on prioriteet [10]. See tähendab seda, et kõik Vuetify-st võetud komponendid töötavad ka mobiilivaates ning skaleeruvad korrektselt [10].

Tänu Vuetify kasutamisele võtmisele väheneb koodi pikkus. Tihtilugu arendajad kohtuvad sellise probleemiga, et mingit nuppu on vaja kasutada mitmes kohas, kuid ei soovita kirjutada duplikaat koodi. Antud raamistik lubab nupu luua valmis ühes kohas ning hiljem saab seda nuppu kasutada igal pool.

Töö autor valis Vuetify sellepärast, et see on spetsiaalselt Vue jaoks loodud ning seda oli lihtne integreerida Vue projektiga. Samuti on Vuetify-l põhjalik dokumentatsioon ning erinevad koodinäited loodud komponentide kasutamisest.

4.3 Vuex

Lisaks Vue-le ning Vuetify-le kasutab töö autor ka Vuex'i. Vuex on raamistik Vue.js rakendustele, mille abil on võimalik hoida rakenduses kesksel seisundit (ingl. k *state*) erinevate komponentide vahel [12]. Vuex-i abil on võimalik pääseda ligi erinevatest komponentidest samadele andmetele, mida on töö autoril loodava veebisaidi juures vaja. Näiteks töö autor kasutab sellist printsiipi jaluse lisamisel igale vahelehele, kus on välja toodud Ahja perearstikabineti kontaktandmed. Ilma Vuex-i kasutamata tuleks need andmed erinevatel vahelehtedel olevatele jalustele kohale toimetada propside abil, kuid see muudab kogu süsteemi ning andmete jagamise erinevate komponentide vahel väga keeruliseks. Palju parem lahendus on luua keskne seisund, kuhu veebilehe esialgse laadimise korral laetakse sisse kontaktandmed, hoiustatakse need ning seejärel saab jaluse komponendi siseselt küsida seisundist kontaktandmeid. Sellega piiratakse andmete jagamine ainult komponendi siseselt ning vahet pole kus jalust kasutatakse – sellega tulevad alati kontaktandmed kaasa.

Joonisel 5 on välja toodud Vuex-i keskse seisundi loomine. Kõigepealt antakse Vue-le teada, et kasutatakse Vuex-i. Seejärel luuakse konstant, milleks ongi keskne seisund. Sellele on võimalik nüüd igast teisest komponendist ligi pääseda. Seejärel defineeritakse, milliseid andmeid selles keskses seisundis hoitakse. Kuna soovitakse hoida jaluse andmeid, siis ongi seal defineeritud aadress, telefon ning email. Samuti kasutab töö autor VuexPersistence pistikprogrammi (ingl. k. *plugin*), mis tagab selle, et brauserit värskendades/mõnele muule vahelehele liikudes hoitakse seisundit alles, kuna ainult Vuex-i kasutades kaoks eelnevalt välja toodud tegevuste korral keskne seisund ära. VuexPersistence pistikprogrammi abil hoitakse seisundit Local Storages JSON objektina. Seejärel on loodud meetodid *getterid* ning mutatsioonid. *Getter*-ite abil on võimalik küsida kesksest seisundist andmeid ning mutatsioonide abil on võimalik keskse seisundi andmeid muuta.

```

import Vue from "vue";
import Vuex from "vuex";
import VuexPersistence from "vuex-persist";

Vue.use(Vuex);

export const store = new Vuex.Store( options: {
  state: {
    address: "",
    phone: "",
    email: ""
  },
  plugins: [
    new VuexPersistence( options: {
      modules: ["address", "phone", "email"]
    }).plugin
  ],
  getters: {
    getContactInformation(state) {
      return {
        address: state.address,
        phone: state.phone,
        email: state.email
      };
    }
  },
  mutations: {
    setContactInformation(state, payload) {
      state.address = payload.address;
      state.phone = payload.phone;
      state.email = payload.email;
    }
  }
});

```

Joonis 5. Vuex-i keskse seisundi loomine

Sellega on Vuex-i keskne seisund üles seatud ning nüüd veebilehe laadimisel saab sinna andmeid hoiustada. Selleks tuleb välja kutsuda Vuex-i *commit* meetodit, mis võtab kaks parameetrit – mutatsiooni nimi ning andmed, mida soovitakse kesksesse seisundisse lisada/muuta. Joonise 6 pealt on näha, et mutatsiooni nimeks on eelnevalt Vuex-is defineeritud mutatsioon *setContactInformation* ning andmeteks on JSON objekt, mis koosneb aadressist, telefonist ning emailist.

```

function setFooterData(contactInformation) {
  store.commit( type: "setContactInformation", payload: {
    address: contactInformation.address,
    phone: contactInformation.phone,
    email: contactInformation.email
  });
}

```

Joonis 6. Jaluse andmete Vuex-i kesksesse seisundisse salvestamine

Sellega on nüüd kontaktandmed keskses seisundis olemas ning neid saab kätte igast komponendist. Joonisel 7 on näha kuidas töö autor sai kätte keskse seisundi andmed jaluses.

```
<script>
export default {
  name: "Footer",
  data() {
    return {
      address: "",
      email: "",
      phone: ""
    };
  },
  created() {
    const contactInformation = JSON.parse(localStorage.getItem( key: "vuex"));
    this.address = contactInformation.address;
    this.phone = contactInformation.phone;
    this.email = contactInformation.email;
  }
};
```

Joonis 7. Vuex-i keskse seisundi andmete kättesaamine jaluses

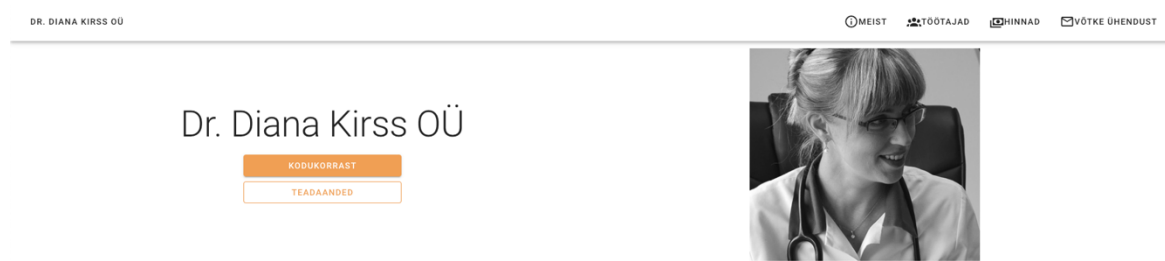
Kui komponent luuakse, siis otsitakse kohealt Local Storaigest objekti nimega *vuex*. Saades see objekt kätte, asendatakse varasemalt defineeritud väärtused kesksest seisundist saadud andmetega. Nüüd on jaluses keskse seisundi andmed ning need on nähtaval igal pool, kus on kasutatud jalust. Nagu eelnevalt mainiti, siis see on palju parem lahendus kui see, et propside abil erinevatele vahelehtedele saata kontaktandmeid, et neid jaluses kuvada.

5. Valminud veebisaidi ülevaade

Käesolevas peatükis antakse ülevaade loodud veebisaidist (<https://ahjaperearst.ee/>) ning kirjeldatakse täpsemalt, kuidas loodi veebisaidi API.

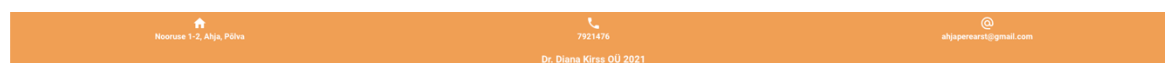
5.1 Päis ja jalus

Iga vahelehe alguses on koheselt nähtav päis¹. Seal on välja toodud nupud „Meist“, „Töötajad“, „Hinnad“ ning „Võtke ühendust“, mis suunavad avalehel oleva vastava info juurde. Päise all on Ahja perearstikabineti asutuse nimi, pilt perearstist ning nupud, mille abil on võimalik liikuda kodukorra või teadaannete juurde (vt. Joonis 8).



Joonis 8. Loodud päis

Iga vahelehe lõpus on jalus, kus on toodud välja Ahja perearstikabineti aadress, telefoninumber ning kontaktemail. Kuigi esialgses disainis sai jalus disainitud halli värvi, siis lõpplahenduses muudeti jaluse värv oranžiks, kuna see värv sobis töö autori arvates jaluseks paremini (vt. Joonis 9).

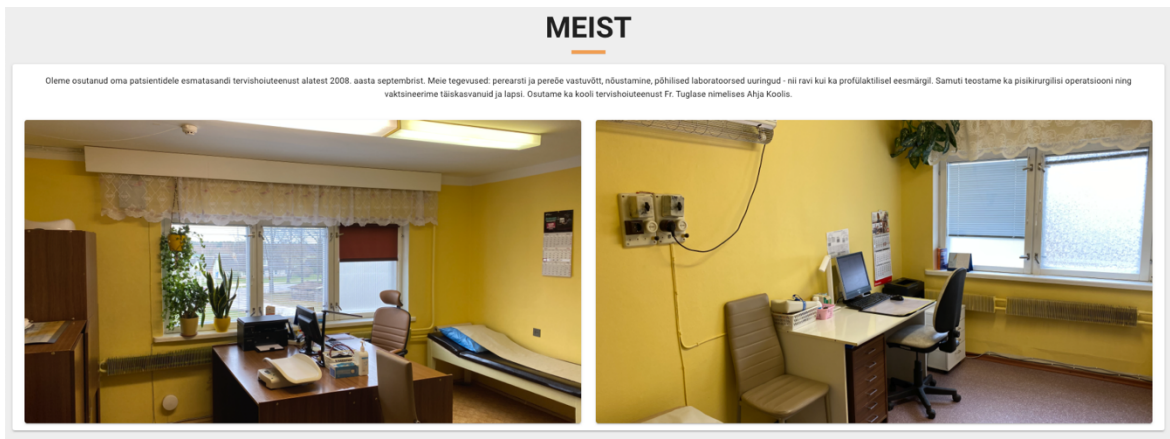


Joonis 9. Loodud jalus

5.2 Avaleht

Nagu eelnevalt mainiti, siis avalehel on kõigepealt nähtav päis, millele järgneb „Meist“ osa. „Meist“ osas on välja toodud Ahja perearstikabineti lühikirjeldus ning mõned pildid perearstikabinetist (vt. Joonis 10).

¹ Projekti kliendipool (ingl. k. *frontend*) on kättesaadav GitHub repositooriumist https://github.com/jkallson/new_perearst_frontend ning projekti API on kättesaadav GitHub repositooriumist <https://github.com/jkallson/perearst-graphql>




Joonis 10. Meist seksioon

„Meist“ osale järgneb „Töötajad“ osa, kus on välja toodud kõik Ahja perearstikabineti töötajad. Iga töötaja seksioonis on selle töötaja pilt ning kui töötajast veel pilti ei eksisteeri, siis on lisatud selle asemele kohatäide (ingl. k. *placeholder*). Iga töötaja juurde on märgitud tema ametikoht (perearst, pereõde) ning vajutades vastuvõtu või telefoni aegade nupu peale on võimalik tutvuda vaadeldava töötaja vastuvõtuaegadega ning telefoni aegadega (vt. Joonis 11).

DR. DIANA KIRSS OÜ

MEIST TÖÖTAJAD HINNAD VÕTKE ÜHENDUST

TÖÖTAJAD




Diana Kirss
Perearst - hetkel lapsehoolduspuhkusel

VASTUVÕTU AJAD

TELEFONI AJAD

Vastuvõtu ajad

Päev	Kellaajad
Esmaspäev	09.00-13.00
Teisipäev	09.00-13.00
Kolmapäev	09.00-13.00
Neljapäev	09.00-17.00
Reede	09.00-13.00



Katrin Mölder
Asendusarst

VASTUVÕTU AJAD

TELEFONI AJAD

TÖENDID

Teenus	Hind
Tervisetõend (sisaldab kopsuröntgenit)	18.00 €
Tervisetõend (kopsuröntgen ja bakteriaalsed analüüsid)	60.00 €
Tervisetõend (sisaldab bakteriaalseid analüüse)	47.00 €
Tervisetõend juhilubade taotlemiseks või vahetamiseks	20.00 €
Tervisetõend relvaloa taotlemiseks	25.00 €

Joonis 11. Tabel töötajate vastuvõtu või telefoni aegadega

Järgmisena tuleb „Hinnad“ osa. Teenused on jaotatud kategooriatesse (Tõendid, Visiiditasud, Vaktsineerimine) ning iga kategooria all on sellele vastavad teenused (vt. Joonis 12).

HINNAD	
TÕENDID	
Teenus	Hind
Tervisetõend (sisaldab kopsaröntgenit)	18.00 €
Tervisetõend (kopsaröntgen ja bakteriaalsed analüüsid)	60.00 €
Tervisetõend (sisaldab bakteriaalseid analüüse)	47.00 €
Tervisetõend juhulbade taotlemiseks või vahetamiseks	20.00 €
Tervisetõend relvaloa taotlemiseks	25.00 €
Tervisetõend täiskasvanule (kindlustusele, kohtu ja uurimisorganitele, migratsiooniametile, turismifirmale)	10.00 €
Tervisetõend ilma lisauringuteta	5.00 €
Tervisetõend abipolitseinkule	15.00 €
VISIIDITASUD	
Teenus	Hind
Ambulatoorne vastuvõtt	Tasuta
Kodukutsed	Tasuta
Kindlustamata inimese vastuvõtt	15.00 €
VAKTSINEERIMINE	
Teenus	Hind
Difteetia (teetanuse vastane vaktsineerimine - peab revaktsineerima iga 10 aasta järel)	Tasuta
Puukentseliit	Tasuta (lisandub vaktsiini hind)
Gripp	Tasuta (lisandub vaktsiini hind)

Joonis 12. Hindade sektsioon

Järgneva osaks on „Võtke ühendust“ sektsioon (vt. Joonis 13) järgnevate elementidega:

- *Google Maps* kaart, mille abil näidatakse Ahja perearstikabineti asukohta;
- kontaktinformatsioon, kus on välja toodud telefon, email ning aadress;
- kontaktvorm, mille kaudu on võimalik Ahja perearstikabinetile saata emaili.

VÕTKE ÜHENDUST

KONTAKT

Küsimuste või probleemide korral helistage või kirjutage meile. Vastame Teile esimesel võimalusel!

Kontakt informatsioon

Telefon
7921 476

Email
ahjaperearst@gmail.com

Aadress
Nooruse 1-2, Ahja, Põlva

Nimi

Email

Teema

Küsimus

[SAADA](#)

Joonis 13. Võtke ühendust sektsioon

Avalehe lõpetab eelnevalt kirjeldatud jalus, kus on välja toodud Ahja perearstikabineti aadress, telefon ning email. Täispikk avaleht on nähtav lisan 3, joonisel 43.

5.3 Kodukorrast

Kodukorrast vaheleht algab samuti eelnevalt kirjeldatud päisega. Peale päist tuleb sektsioon, kus on välja toodud Ahja perearsti kodukord ning kasulikud lingid. Esialgses disainis ei olnud kasulikud lingid Kodukorrast vahelehele paigutatud, kuid arenduse käigus selgus, et kasulike linkide sinna lisamine sobis kõige paremini ning jättis veebisaidist parema mulje (vt. Joonis 14). Kodukorra info on välja toodud lipikute abil, kus lipikule vajutades kuvatakse kasutajale vastava lipiku info. Lingid on välja toodud tabelis, kus on kirja pandud iga lingi kirjeldus. Lingile vajutades suunatakse kasutaja lingil oleva informatsiooni juurde. Kodukorrast vaheleht lõppeb jalusega. Täispikk kodukorra vaheleht on nähtav lisan 3, joonisel 44.



Joonis 14. Kodukorrast ning kasulikud lingid sektsioon

5.4 Teadaanded

Teadaannete vaheleht algab päisega. Edasi on välja toodud Ahja perearstikabineti poolt postitatud teadaanded. Teadaannete kuvamiseks kasutati laienduspaneeli, laienduspaneeli päise pealkirjaks on teadaande pealkiri. Laienduspaneelile peale vajutades kuvatakse vastava teadaande sisu (vt. Joonis 15). Teadaanded on sorteeritud uuemast vanemaks.

Dr. Diana Kirss OÜ

KODUKORRAST

TEADAANDED



TEADAANDED

Siit leiate viimased teadaanded Ahja perearstikabineti poolt!

D-vitamiini vajalikkus 21/04/2021 ^

Soovitame kõikidel inimestel tarvitada igapäevaselt D-vitamiini toidulisandeid.

D-vitamiin on hädavajalik kaltsiumi imendumiseks, luustiku tervise tagamiseks ning immuunsüsteemi talituseks. Samuti on uuringutel jõutud järelduseni, et D-vitamiinil on kaitsev toime pahaloomuliste kasvujate tekke eest.

D-vitamiin tekitab nahas päikese toimet, kuid meie kliimas ei ole kunagi päikest piisavalt, et tagada normaalset D-vitamiini taset organismis. Seega me vajame seda juurde igapäevaselt. Soovitame võtta õlikapsleid või tilku, tablettidest imendub D-vitamiin halvemini.

Profülaktilised D- vitamiini annused:

- Vastsündinutele soovitatakse kolmandast-neljandast elupäevast anda (400IU) 10 µg päevas,
- 1-12. aastased võiksid võtta (400-800 IU) 10-20 µg päevas,
- 12-18. aastased (600-1000 IU) 15-25 µg päevas,
- Terved täiskasvanud (1000 IU.) 25 µg päevas,
- Vanurid ja tubased täiskasvanud (1200-2000 IU) 30-50 µg päevas.

Kui Te ei ole mitu aastat D-vitamiini võtnud, siis on mõistlik alustada suurena annusega, nt. 4000 IU päevas kuu jooksul ning edasi jätkata profülaktilises annuses.

Raviannused võivad olla suuremad, neid määrab arst vastavalt vajadusele.

Olge terved!

Uus veebileht valmis 20/04/2021 v

Joonis 15. Uue veebisaidi teadaannete vaheleht

5.5 Administraatori vaade

Administraatori vaatesse saamine eeldab autoriseerimist. Selleks käesoleva töö autor lõi vahelehe sisselogimise võimalusega (vt. Joonis 16).

ADMINISTREERIMISLIIDES

Joonis 16. Administreerimisliidese sisselogimine

Peale edukat sisselogimist suunatakse kasutaja edasi administraatorivaatesse. Administraatorivaates on võimalik teha järgnevat tegevusi:

- muuta „Meist“ teksti;
- lisada/muuta/kustutada töötajaid;
- lisada/muuta/kustutada hindu;
- lisada/muuta/kustutada/ümber järjestada linke;
- muuta kontaktandmeid;
- lisada/muuta/kustutada teadaandeid;
- lisada/muuta/kustutada kodukorra infot.

Alljärgneval joonisel 17 on toodud välja kodukorra osa administraatori vaates. Kogu informatsioon on kuvatud tabelis, tabeli päises on nupp „Lisa uus“. Selle abil on võimalik lisada uut kodukorra informatsiooni.

KODUKORD		LISA UUS
VASTUVÕTULE REGISTREERIMISEST		✎
RETSEPTIDE TELLIMISEST		✎
TÖÖVÕIMETUS- JA HOOLDUSLEHTEDEST		✎
TELEFONIAJAD		✎
ANALÜÜSIDE VÕTMINE		✎
NIMISTUSSE REGISTREERIMINE		✎
JUHILUBADE TAOTLEMISEST		✎
VÄLJASPOOL PEREARSTI TÖÖAEGA ERAKORRALISTE PROBLEEMDEGA SAATE PÕRÕUDUA		✎
MUUD KONTAKTNUMBRID		✎

Joonis 17. Administraatorivaate kodukorra informatsioon

Iga tabeli elemendi lõpus on muutmise ning kustutamise nupp. Vajutades muutmise nupu peale tuleb ette dialoog, kus on võimalik muuta kodukorra infot (vt. Joonis 18). Samasugune dialoog tuleb ka ette uue kodukorra informatsiooni lisamisel. Täispikk administraatori vaheleht on nähtav lisas 3, joonisel 45.

Joonis 18. Kodukorra muutmise dialoog

5.6 API loomine

Veebilehe API (ingl. k. *Application Programming Interface*) loodi kasutades Node.js-i, Express-i ning GraphQL-i. Node.js-i ning Express-i kasutatakse peamiselt tagateenuste (ingl. k. *back-end*) loomiseks. Kõigepeal sai valmis loodud JavaScript-i fail, mis programmi käivitamisel serveri tööle paneb ning loob andmebaasiga ühenduse. Samuti lisas töö autor sinna kaks funktsionaalsust: esiteks, enne igat andmete päringut kontrollib API, kas päringu sooritaja on autenditud või mitte. Päringutel, millega saab ainult andmeid kätte, ei ole autentimist vaja, kuid päringud, mille abil on võimalik andmebaasi andmeid muuta, nõuavad autenditud kasutajat. Samuti defineeriti loodud failis ära, et kasutatakse GraphQL-i ning et GraphQL-ga saab suhelda siis, kui teha päringud /graphql lõpp-punkti pihta (ingl. k. *endpoint*). Sellega sai server üles pandud ning edasi oli vaja luua GraphQL-s vajaminevad funktsioonid.

5.6.1 GraphQL

Kui töö autor alustas API loomist, siis esialgu oli plaanis luua täiesti tavaline REST API, milles on tavalised API funktsionaalsused – GET, POST, PUT, DELETE. API-t luues töö autor aga märkas, et avalehele andmete kätte saamine on sellist meetodit kasutades väga ebaefektiivne. Andmebaasis paiknes avalehe info mitmetes tabelites ning selleks, et sealt kõik kätte saada on vaja teha mitu päringut. Selleks oli kaks võimalust – kas teha kliendi poole pealt mitu päringut API-s defineeritud lõpp-punktide pihta või luua ainult üks lõpp-punkt, kus tehakse kõik baasipäringud. Töö autor leidis, et mitmete päringute tegemine koormab serverit liigselt ning kogu info võiks realsuses kätte saada ühe päringuga. Seega töö autor lõi sellise lõpp-punkti (vt. Joonis 19).

```
exports.test = (req, res) => {
  workerInformation.find().exec().then(workers => {
    contactInformation.find().exec().then(contact => {
      aboutUsText.find().exec().then(aboutUs => {
        Links.find().exec().then(links => {
          prices.find().exec().then(prices => {
            return res.status(200).json({workers: workers, contact: contact, aboutUsText: aboutUs, prices: prices});
          })
        })
      })
    })
  })
}

.catch(err => {
  console.log(err)
})
}
```

Joonis 19. Lõpp-punkt kogu avalehe info saamiseks

Loodud lahendus nägi välja väga kohmakas ning oli keeruline aru saada, mis seal täpsemalt toimub. Samuti oli ka selline lahendus väga aeglane ning keskmiselt võttis sellise päringu lahendamine aega umbes 1.5 sekundit. Arvestades neid aspekte otsustas töö autor võtta kasutusele GraphQL-i.

GraphQL on päringukeel (ingl. k. *Query Language*), mille abil on võimalik jooksutada päringuid/mutatsioone kasutades eelnevalt defineeritud tüübisüsteemi [13]. Uurides, kuidas GraphQL töötab, leidis töö autor, et GraphQL-i kasutamine lahendab kõik varasemalt eksisteerinud probleemid ning GraphQL ei nõua mitmete lõpp-punktide loomist, vaid piisab ühest lõpp-punktist, milleks on /graphql (vt. Joonis 20).

```
app.use('/graphql', graphqlHTTP({ options: {  
  schema: graphqlSchema,  
  rootValue: graphqlResolvers,  
  graphiql: true,  
} }));
```

Joonis 20. GraphQL lõpp-punkti loomine

GraphQL lõpp-punkti luues tuleb ära defineerida skeem (ingl. k. *schema*) ning juurväärtus (ingl. k. *root value*). Skeemis on ära defineeritud järgnevad elemendid:

- tüübid (ingl. k. *type*);
- sisendid (ingl. k. *input*);
- päringud (ingl. k. *query*);
- mutatsioonid (ingl. k. *mutation*).

Tüüpidenäiteks defineeritakse ära mis kujul on GraphQL-i päringute ning mutatsioonide vasted. Näiteks on defineeritud kaks tüüpi – AboutUsText ning Prices (vt. Joonis 21). Hiljem saab neid tüüpe kasutada, et defineerida, mis tüüpi vastuseid päringud/mutatsioonid oodata võivad. Kui luua päring ning oodata vastuse tüübina AboutUsText-i, siis GraphQL teab, et vastuses peab olema leitav nii *_id* kui ka *text*. Kusjuures kui lisada hüüumärk tüübi lõppu, siis see tähendab seda, et see väli ei tohi kunagi olla tühi.

```
type AboutUsText {  
  _id: ID!  
  text: String!  
}  
  
type Prices {  
  _id: ID!  
  name: String!  
  price: String!  
  class: String!  
}
```

Joonis 21. Tüüpide defineerimine

Sisendeid kasutatakse mutatsioonides ning nende abil defineeritakse ära, millist sisendit mutatsioon ootab. Kui kasutaja üritab mingil muul kujul andmeid saata defineeritud mutatsiooni, siis ei lähe päring läbi ning tagastatakse veateade.

Päringud on loodud selleks, et saaks andmebaasist pärida infot. GraphQL päring on sama asi, mis tavalises REST API-s on GET päring. Päringute juures defineeritakse ära päringu nimi ning tüüp, mida see päring vastusena ootab. Alljärgneval joonisel 22 on näha, et on defineeritud aboutUs päring, mis ootab vastusena AboutUsText tüüpi, kusjuures see vastus ei saa olla null.

```
type RootQuery {
  aboutUs: AboutUsText!
  prices: [PricesQuery!]!
  contactInformation: ContactInformation!
  links: [Links!]!
  workers: [Workers!]!
  login(username: String!, password: String!): AuthData!
  news: [News!]!
  regulations: [Regulation!]!
}
```

Joonis 22. GraphQL päringute defineerimine

Mutatsioonid on loodud selleks, et võimaldada kasutajal andmebaasis olevaid andmeid muuta või midagi lisada [14]. Näiteks allpool oleval joonisel 23 on näha, et on defineeritud mutatsioon nimega updateAboutUsText, mis ootab sisendina AboutUsTextInput tüüpi sisendit ning tagastab AboutUsText tüüpi vastuse.

```
type RootMutation {
  createPrice(priceInput: PriceInput): Prices
  removePrice(priceID: ID!): Prices
  updatePrice(priceInput: PriceInput): Prices
  updateContactInformation(contactInformationInput: ContactInformationInput): ContactInformation
  updateAboutUsText(aboutUsTextInput: AboutUsTextInput) : AboutUsText
  createLink(linkInput: LinkInput!): Links
  deleteLink(linkID: ID!): Links
  updateLink(linkInput: LinkInput!): Links
  updateLinkOrder(linkInput: [LinkInput!]!) : [Links!]!
  createWorker(workerInput: WorkerInput!): Workers
  deleteWorker(workerID: ID!): Workers
  updateWorker(workerInput: WorkerInput!): Workers
  createNews(newsInput: NewsInput!): News
  updateNews(newsInput: NewsInput!): News
  deleteNews(newsID: ID!): News
  createRegulation(regulationsInput: RegulationsInput!): Regulation
  updateRegulation(regulationsInput: RegulationsInput!): Regulation
  deleteRegulation(regulationID: ID!): Regulation
}
```

Joonis 23. GraphQL mutatsioonide defineerimine

Peale skeemi loomist oli vaja valmis luua juurväärtuse (ingl. k. *root value*) jaoks vajalikud lahendajad (ingl. k. *resolvers*). Lahendajate abil sooritatakse GraphQL-i saadetud päring või mutatsioon. Eelnevalt defineeriti ära `aboutUs` päring. Kui see päring saadetakse, siis GraphQL otsib lahendajat, mis oskaks selle päringu sooritada. Lahendaja nimi peab olema sama nimega, mis oli mutatsiooni/päringu nimi. Näiteks, kui kasutaja tahab leida meist teksti, siis teeb ta päringu `aboutUs` kohta. Peale päringu sooritamist otsib GraphQL ülesse lahendaja (vt. Joonis 24).

```
const AboutUsModel = require("../models/aboutUsModel")

module.exports = {
  aboutUs: async () => {
    try {
      return await AboutUsModel.findOne()
    } catch (err) {
      throw new err
    }
  },

  updateAboutUsText: async (args, req) => {
    if (!req.isAuthenticated()) {
      throw new Error("Unauthenticated!")
    }
    try {
      await AboutUsModel.updateOne({ filter: { _id : args.aboutUsTextInput._id }, update: { $set: { text: args.aboutUsTextInput.text, } } });
      return await AboutUsModel.findById(args.aboutUsTextInput._id)
    } catch (err) {
      throw err
    }
  }
}
```

Joonis 24. Meist päringute/mutatsioonide lahendaja

Selles lahendajas sooritatakse andmebaasi päring ning saadakse tagasi andmed meist teksti kohta. Kui päring ei lähe läbi, siis tagastatakse veateade.

Samuti kui saadetakse näiteks mutatsioon `updateAboutUsText`, siis GraphQL otsib järjekordselt lahendajat, mis oleks sama nimega. Üleval pool oleval joonisel 24 on näha seda lahendajat. Antud lahendaja otsib andmebaasist üles teksti ning uuendab seda. Peale uuendamist tagastab ta teksti.

GraphQL pakub väga palju paindlikkust ning seda kasutades pole vaja defineerida mitmeid lõpp-punkte. Selle asemel tuleb defineerida erinevad päringud ning mutatsioonid ning hiljem saab vastavalt vajadusele GraphQL-st neid küsida. Samuti on väga suur boonус GraphQL-i puhul see, et kasutaja saab ise valida, milliseid andmeid ta vastusest tahab. Näiteks olime eelnevalt defineerinud päringu, mille abil saab kätte kõik lingid. Joonisel 25 on vasakul pool päring ning paremal pool päringu tulemus. Päringu sooritamiseks tuleb defineerida päringu nimi (hetkel on päringu nimeks *links*) ning andmed, mida soovime vastusena kätte saada (hetkel soovime kätte saada `_id`, `link`, `name` ning `orderIndex`-t).

```

query {
  links {
    _id
    link
    name
    orderIndex
  }
}

```

```

{
  "data": {
    "links": [
      {
        "_id": "60644ed17e9b2104617f63cb",
        "link": "sadqweqweqwe janar",
        "name": "qweqwe",
        "orderIndex": 0
      },
      {
        "_id": "606af21bf6847a0a07de8860",
        "link": "asdasd",
        "name": "test",
        "orderIndex": 1
      },
      {
        "_id": "606af220f6847a0a07de8861",
        "link": "gsagas",
        "name": "asfasgagas",
        "orderIndex": 2
      },
      {
        "_id": "606af2cff9e1c90a19f08659",
        "link": "asdasd",
        "name": "yeyeye",
        "orderIndex": 3
      }
    ]
  }
}

```

Joonis 25. Päringu sooritamine

Samuti on ka võimalik korraga sooritada mitu pärinut (vt. Joonis 26).

```

query {
  links {
    _id
    link
    name
    orderIndex
  }
}

```

```

{
  "data": {
    "links": [
      {
        "_id": "60644ed17e9b2104617f63cb",
        "link": "sadqweqweqwe janar",
        "name": "qweqwe",
        "orderIndex": 0
      },
      {
        "_id": "606af21bf6847a0a07de8860",
        "link": "asdasd",
        "name": "test",
        "orderIndex": 1
      },
      {
        "_id": "606af220f6847a0a07de8861",
        "link": "gsagas",
        "name": "asfasgagas",
        "orderIndex": 2
      },
      {
        "_id": "606af2cff9e1c90a19f08659",
        "link": "asdasd",
        "name": "yeyeye",
        "orderIndex": 3
      }
    ]
  }
}

```

Joonis 26. Mitme päringu sooritamine

5.6.2 Autentimine

Üheks väga tähtsaks osaks veebisaidi arendamisel oli autentimise lisamine. On oluline vältida olukorda, kus igaüks, kes teab API URL-i, saaks muuta või kustutada andmeid andmebaasist. Selleks lisas töö autor validatsiooni mutatsioonidele, et kontrollida, kas

mutatsiooni saatjal on selleks õigused olemas või ei ole. Selleks lisati vahevara (ingl. k. *middleware*), mis kontrollib kas kasutaja on autenditud või mitte (vt. Joonis 27).

```
const jwt = require("jsonwebtoken")

module.exports = (req, res, next) => {
  const authHeader = req.get('Authorization')
  if (!authHeader) {
    req.isAuth = false
    return next();
  }
  const token = authHeader.split(' ')[1]
  if (!token || '') {
    req.isAuth = false
    return next();
  }
  let decodedToken;
  try {
    decodedToken = jwt.verify(token, process.env.SECRET_KEY)
  } catch (err) {
    req.isAuth = false
    return next();
  }
  if (!decodedToken) {
    req.isAuth = false
    return next();
  }
  req.isAuth = true
  next()
}
```

Joonis 27. Autentimist kontrolliv vahevara

Kõigepealt otsitakse päringu header-ist *Authorization* väärtusele vastavaid vasteid. Kui see pole kaasa pandud, siis saab päring mitte autenditud staatuse ning vahevara lõpetab oma töö. Peale seda on teada, et mingisugune väärtus on leitud ning nüüd üritatakse sellest väärtusest välja eraldada pääsmik (ingl. k. *token*), mis on *jsonwebtoken*. JSON Web Token on standard, mille abil on võimalik valideerida ning verifitseerida, kas kasutajal, kes mutatsiooni saadab, on selleks õigused olemas või mitte [15]. Kui pääsmikku ei ole või pääsmik on tühi, siis saab päring järjekordselt mitte autenditud staatuse ning vahevare lõpetab oma töö.

Nüüdseks on teada, et mingisugune pääsmik on päringuga koos kaasa pandud. Pääsmikule sooritatakse kontroll ning kui see on edukas, siis saab päring autenditud staatuse, kui aga ei ole edukas, siis saab päring mitte autenditud staatuse.

Sellega on tehtud kindlaks kas kasutaja saab sooritada mutatsiooni või mitte. Enne mutatsiooni sooritamist, millel on vaja autenditud kasutajat, vaadatakse üle, kas päring on autenditud või mitte (vt. Joonis 28).

```

updateAboutUsText: async (args, req) => {
  if (!req.isAuthenticated()) {
    throw new Error("Unauthenticated!")
  }
  try {
    await AboutUsModel.updateOne( filter: { _id : args.aboutUsTextInput._id }, update: {$set: {text: args.aboutUsTextInput.text,}});
    return await AboutUsModel.findById(args.aboutUsTextInput._id)
  } catch (err) {
    throw err
  }
}
}

```

Joonis 28. Päringu autentimise kontroll

JSON Web Token lisatakse kasutajale külge siis, kui kasutaja logib admin-vaatesse sisse. Sisselogimise korral otsitakse andmebaasist kasutajanime, mille sisselogimise korral kasutaja sisestab. Kui see kasutaja leitakse, siis võrreldakse tema parooli sisestatud parooliga. Kuna andmebaasis olev parool on krüpteeritud, siis kasutatakse töö autor võrdlemiseks teeki bcrypti, mille sees olev meetod `compare()` võrdleb kasutaja poolt sisestatud parooli krüpteeritud parooliga. Kui parool on õige, siis luuakse JSON Web Token ning määratakse selle aegumise ajaks üks tund (vt. Joonis 29).

```

const UserModel = require("../models/usersModel")
const bcrypt = require("bcrypt")
const jwt = require("jsonwebtoken")

module.exports = {
  login: async ({username, password}) => {
    const user = await UserModel.findOne( filter: {username: username})
    if (!user) {
      throw new Error("Invalid credentials!")
    }
    const isEqual = await bcrypt.compare(password, user.password)
    if (!isEqual) {
      throw new Error("Invalid credentials!")
    }
    const token = jwt.sign( payload: {userId: user.id, username: user.username}, process.env.SECRET_KEY, options: {
      expiresIn: '1h'
    })
    return {
      token: token,
      tokenExpiration: 1
    }
  }
}

```

Joonis 29. Kasutaja valideerimine ning JWT lisamine

5.7 Otsingumootori optimeerimine

Peale veebisaidi loomist tegeles töö autor otsingumootori optimeerimisega ehk SEO-ga (ingl. k. *search engine optimization*). Tänu SEO'le on Google või mõni muu otsingumootor võimeline aru saama, mis informatsiooni leiab veebisaidilt ning tänu sellele suudavad otsingumootorid lehti järjestada relevantsuse alusel [16]. Kui loodud veebisaidi SEO on hea,

siis on loodud veebisait ka paremini leitav otsingumootoritest. Selleks peab silmas pidama järgmiseid aspekte [16]:

- veebisaidil olev sisu peab olema kvaliteetne ning relevantne;
- veebisaidil peab olema hea jõudlus (ingl. k. *performance*) – veebisait peab laadima võimalikult kiiresti;
- veebisait peab olema mobiilisõbralik ning ligipääsetav iga seadmega;
- veebisait peab olema turvaline (SSL sertifikaat);
- veebisaidil peavad olema rikkalikud metaandmed (ingl.k. *rich metadata*), mille abil tuleb kirjeldada võimalikult hästi, mis sisu veebisaidilt leida võib.

Eelnevalt välja toodud informatsiooni alusel alustas bakalaureusetöö autor loodud veebisaidi SEO parandamist. Alguses sai loodud fail *robots.txt*. Selle faili abil saavad otsingumootorites olevad otsirobotid aru, milliseid osasid veebisaidist võivad nad külastada ja indekseerida ning milliseid osasid mitte [17]. Töö autor lõi *robots.txt* faili, millega lubas kõikidel otsingurobotitel veebisaiti indekseerida. Samuti lubati otsirobotitel indekseerida ainult URI'sid, mis omavad reaalselt sisu (avaleht, teadaanded, kodukord ning administraatori avaleht). Muid URI'sid pole vaja indekseerida, sest need ei oma sisu, millest võiksid otsirobotid kasu saada ning veebisaidi relevantsust tõsta. Sellega väheneb otsirobotite töö loodud veebisaidil, mille tõttu on suurem võimalus, et otsirobotid leiavad ülesse informatsiooni, mida lehel päriselt kuvatakse. Samuti defineeriti *robots.txt* failis ära, et kust on otsirobotitel võimalik leida loodud veebisaidi saidikaarti (ingl. k. *sitemap*) (vt. Joonis 30).

```
# Group 1
User-agent: *
Allow: /admin/avaleht
Disallow: /src/
Disallow: /admin/

Sitemap: https://ahjaperearst.ee/sitemap.xml
```

Joonis 30. Loodud *robots.txt* fail

Seejärel loodi saidikaardi fail *sitemap.xml* (vt. Joonis 31), mille abil on võimalik välja tuua kogu informatsioon veebisaidil leitavate lehtede, piltide, videote ning muude failide kohta [18]. Töö autor otsustas saidikaardis välja tuua informatsiooni ainult veebisaidil leitavate lehtede kohta. Saidikaarti abil suudavad otsirobotid intelligentsemalt otsida infot loodud veebisaidilt [18]. Töö autor kasutas saidikaarti loomiseks saidikaarti XML generaatorit

(<https://www.xml-sitemaps.com/>), mis lõi valmis esialgse saidikaarti malli. Peale seda lisas töö autor loodud saidikaarti malli loodud lehtede URL'id.

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset
  xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9
    http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd">
  <!-- created with Free OnLine Sitemap Generator www.xml-sitemaps.com -->
  <url>
    <loc>https://ahjaperearst.ee/</loc>
    <lastmod>2021-04-28T12:09:55+00:00</lastmod>
  </url>
  <url>
    <loc>https://ahjaperearst.ee/kodukord</loc>
    <lastmod>2021-04-28T12:09:55+00:00</lastmod>
  </url>
  <url>
    <loc>https://ahjaperearst.ee/teadaanded</loc>
    <lastmod>2021-04-28T12:09:55+00:00</lastmod>
  </url>
</urlset>
```

Joonis 31. Loodud saidikaart

Peale eelnevalt mainitud failide loomist oli vaja veel muuta veebisaidi metaandmeid. Töö autor lisas metaandmetesse veebisaidi autori nime, veebisaidi kirjelduse ning samuti ka erinevaid märksõnu, mis loodud veebisaidi kohta käivad.

Selleks, et oleks võimalik analüüsida, kuidas veebisait otsingumootorites esineb ning leida võimalikke murekohtasid, kasutas töö autor Google poolt pakutavat otsingu konsooli (ingl.k. *search console*) tööriista. Sinna tuli lisada domeeni nimi ning peale selle lisamist oli võimalik näha erinevaid andmeid veebisaidi kohta – milliste võtmesõnadega veebisaiti üles leitakse, mis riikidest veebisaiti külastatakse, mis vahelehti veebisaidil enamasti külastatakse, kuna ning mis seadmetega veebisaiti külastatud on.

5.8 Loodud veebisaidil teostatud testid

Testimise käigus kontrolliti, kuidas veebisait töötab erinevate operatsioonisüsteemide ja brauseritega, kui hea on veebisaidi SEO ja jõudlus ning kui kiire on veebisait.

5.8.1 Operatsioonisüsteemide ning brauserite testid

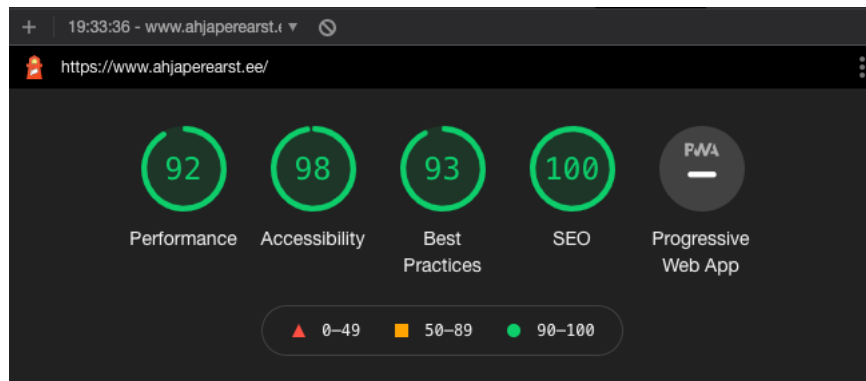
Töö autor testis esimesena loodud veebisaidi töötamist erinevate operatsioonisüsteemide ja brauseritega, kuna veebisaidi arendamise käigus kasutati ainult ühte operatsioonisüsteemi ning brauserit. Nende testide sooritamiseks kasutati Comparium'it (<https://comparium.app/>), mille abil on võimalik kontrollida veebisaidi töötamist erinevate operatsioonisüsteemidega, brauseritega ning samuti ka erinevate seadmetega. Töö autor testis veebisaiti Chromiumis järgnevate ekraanilaiustega: 500px ning 1920px. Nende laiustega simuleeriti veebisaidi tööd nii telefonis kui ka arvutis. Operatsioonisüsteemidena kasutati Windows 10-t, macOS'i ning Linux'it ning brauseritena kasutati Google Chrome't (versioonid 78.0-91.0), Firefox'i (versioonid 68.0-85.0) ning Safari't (versioonid 12.0-13.0). Eelnevalt välja toodud konfiguratsioonidega läksid testid läbi ning probleeme ei esinenud.

5.8.2 Mobiilisõbralikkus

Järgmisena testiti mobiilisõbralikkust. Selleks kasutati Google otsingu konsooli (ingl. k. *search console*) poolt pakutud mobiilisõbralikkuse testi tööriista (<https://search.google.com/test/mobile-friendly>). Mobiilisõbralikkuse test sooritati kõikide vahelehtedega – Avaleht, Kodukord ning Teadaanded. Kõikide testide tulemused näitasid, et veebisait on mobiilisõbralik.

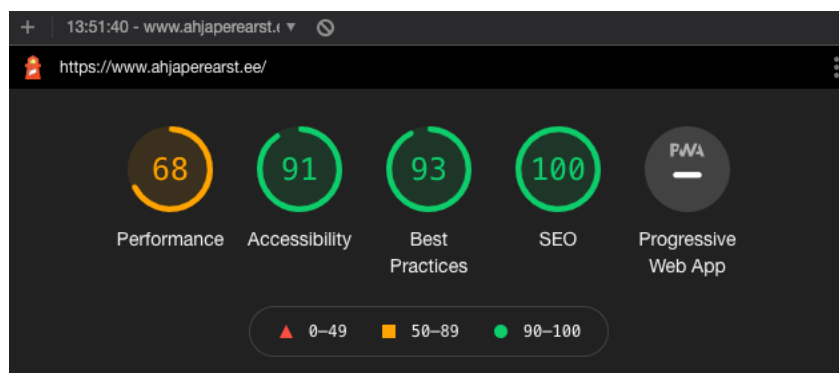
5.8.3 Lighthouse raportid

Viimasena kasutati Google Devtoolsi poolt pakutud tööriista Lighthouse. Nagu eelneva veebisaidi analüüsis sai välja toodud, siis Lighthouse abil on võimalik kindlaks teha, milline on veebisaidi kvaliteet – milline on lehe jõudlus, ligipääsetavus, kas on kasutatud häid arendustavasid ning kui hea on veebisaidi otsingumootorite optimeeritus. Töö autor genereeris kaks Lighthouse raportit – arvutile (ingl. k. *desktop version*) suunatud raport ning telefonile (ingl. k. *mobile version*) suunatud raport. Esimene raport näitas, et veebisaidi kvaliteet on väga hea (vt. Joonis 32). Võrreldes varasema veebisaidiga on tulemus küll natukene langenud, kuid see on põhjustatud sellest, et uuel veebisaidil kasutati ka pilte. Eelneval veebisaidil polnud ühtegi pilti, mistõttu ei pidanud veebisaidi laadimisel pilte laadima.



Joonis 32. Arvutile suunatud Lighthouse raport

Teine raport näitas, et kõik tulemused peale jõudluse olid enamjaolt samasugused (vt. Joonis 33). Jõudluse tulemus on mobiili raporti puhul aeglasem sellepärast, et Lighthouse kasutab mobiili testide sooritamiseks Moto G4 telefoni, mis ei oma head protsessorit [19]. Selle tõttu võtab sellisel telefonil veebisaidi loomine rohkem aega. Samuti on mobiilsel testil kasutusel aeglasem internetiühendus, mille tõttu võtab veebisaidi laadimine kauem aega [20]. Seda arvestades leiab töö autor, et enamus telefonidel probleeme ei esine.



Joonis 33. Mobiilile suunatud Lighthouse raport

6. Võimalikud arengusuunad

Kuigi on valmis saadud töötav veebisait, siis on alati võimalik seda veel edasi arendada arvestades kliendi uusi soovet. Hetkel puudub veebisaidil keele valik ning ainsaks keeleks on eesti keel. Kuna kõik patsiendid või veebisaidi külastajad ei pruugi osata eesti keelt, siis oleks väga hea, kui veebisait oleks ka näiteks inglise või vene keeles. Töö alustamisel seda prioriteediks ei võetud, kuid nüüd oleks see väga hea edasiarendus.

Teiseks saaks valminud veebisaidile luua reaajas vestluse võimaluse (ingl.k. *chatbot*), mille kaudu oleks patsientidel/klientidel võimalik pöörduda perearsti või pereõe poole. Vestluse lisamine lihtsustaks taaskord perearstikabineti kättesaamist.

Kolmandaks arengusuunaks on otsingumootorite optimeerimise parandamine. Hea otsingumootori optimeerimine on oluline, kuna tänu sellele on loodud veebisait otsingumootoritest paremini leitav ning on võimalus, et tänu sellele võib perearstikabinetile tekkida rohkem kliente.

7. Kokkuvõte

Käesoleva bakalaureusetöö raames loodi uus veebisait Dr. Diana Kirss OÜ-le, mis on nähtav aadressil <https://ahjaperearst.ee/>.

Bakalaureusetöös sooritati eelneva veebisaidi analüüs, pandi kliendiga paika uue veebisaidi funktsionaalsed ning mittefunktsionaalsed nõuded, kirjeldati kasutatuid tehnoloogiaid, anti ülevaade valminud veebisaidist, teostati teste uuel veebisaidil ning toodi välja võimalikud arengusuunad.

Arendamise käigus suudeti lahendada varasema veebisaidi juures esinenud probleemid ning uus veebisait sai terviklikuma mulje. Kõik kliendi poolt välja toodud funktsionaalsed ning mittefunktsionaalsed nõuded said lahendatud ning ellu viidud. Veebisaidi esimest laadimiskiirust sai kõvasti parandatud – kui varasemalt oli see umbes neli sekundit, siis loodud veebisaidil on esialgse laadimise kiirus alla ühe sekundi. Demonstreerides loodud veebisaiti kliendile oli klient rahul ning mainis, et täpselt sellist veebisaiti nad ette kujutasidki. Peale demonstratsiooni hakkas klient koheselt uut veebisaiti kasutama.

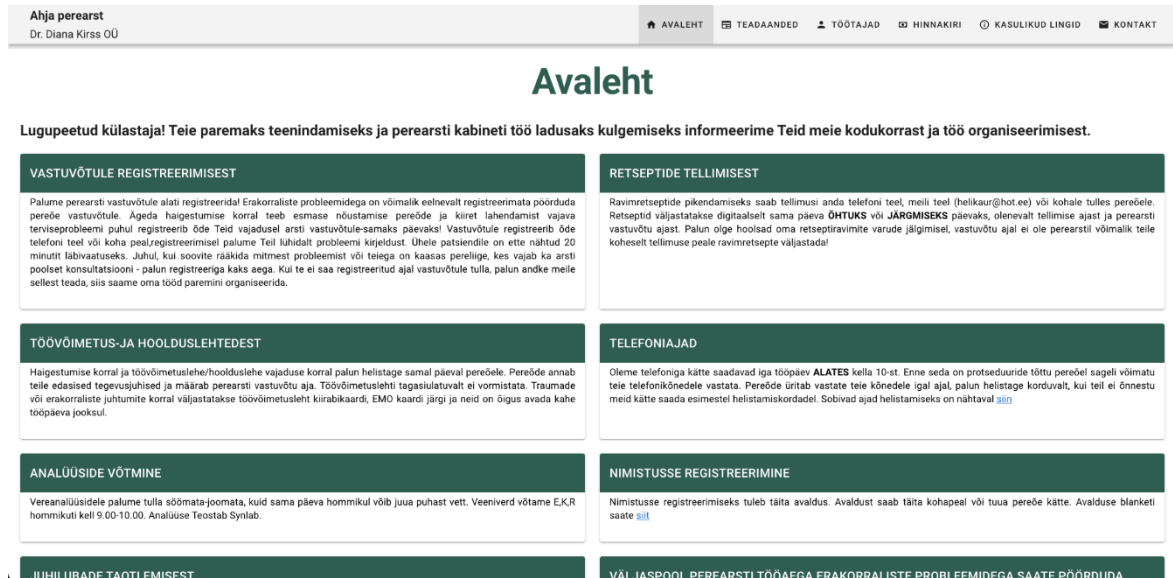
8. Kasutatud kirjandus

- [1] Lighthouse. <https://developers.google.com/web/tools/lighthouse> (06.04.2021)
- [2] Vuetify. Expansion panels <https://vuetifyjs.com/en/components/expansion-panels/> (09.03.2021)
- [3] Vue.js. What is Vue.js <https://vuejs.org/v2/guide/#What-is-Vue-js> (07.12.2020)
- [4] Vue.js in depth. What is Vue.js. <https://worldline.github.io/vuejs-training/presentation/#history> (07.12.2020)
- [5] Vue.js in depth. Vue.js Development team. <https://worldline.github.io/vuejs-training/presentation/#development-team> (07.12.2020)
- [6] Vue.js in depth. Comparison between the frameworks <https://worldline.github.io/vuejs-training/presentation/#comparison-between-the-frameworks> (07.12.2020)
- [7] Education Ecosystem. React.js history. <https://www.education-ecosystem.com/guides/programming/react-js/history> (07.12.2020)
- [8] AngularJs. <https://angularjs.org/> (07.12.2020)
- [9] Vue.js in depth. Vue.js business model and funding. <https://worldline.github.io/vuejs-training/presentation/#business-model-and-funding> (07.12.2020)
- [10] Vuetify. Why Vuetify? <https://vuetifyjs.com/en/introduction/why-vuetify/#why-vuetify> (07.12.2020)
- [11] Vue.js. Type checks <https://vuejs.org/v2/guide/components-props.html#Type-Checks> (07.12.2020)
- [12] Vuex. <https://vuex.vuejs.org/#what-is-vuex> (27.04.2021)
- [13] GraphQL. <https://graphql.org/learn/> (06.04.2021)
- [14] GraphQL. Mutations and Input Types <https://graphql.org/graphql-js/mutations-and-input-types/> (06.04.2021)
- [15] JWT. Introduction to JSON Web Tokens <https://jwt.io/introduction> (06.04.2021)
- [16] MadeWithVueJS. How to make Vue.js Single Page Applications SEO-friendly: A Beginners Guide <https://madewithvuejs.com/blog/how-to-make-vue-js-single-page-applications-seo-friendly-a-beginner-s-guide> (28.04.2021)

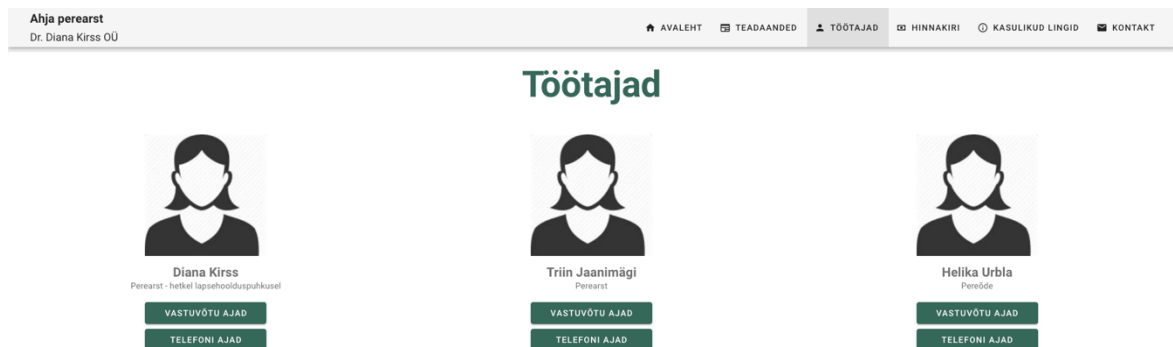
- [17] Veebimajutus. Milleks on robots.txt fail ja mida Sa sellega tegema pead?
<https://www.veebimajutus.ee/blogi/milleks-on-vaja-robots-txt-faili> (28.04.2021)
- [18] Google Search Central. Learn about sitemaps
<https://developers.google.com/search/docs/advanced/sitemaps/overview> (28.04.2021)
- [19] PageSpeed Insights. About PageSpeed Insights
<https://developers.google.com/speed/docs/insights/v5/about#faq> (29.04.2021)
- [20] Github. Lighthouse. Network Throttling
<https://github.com/GoogleChrome/lighthouse/blob/master/docs/throttling.md> (29.04.2021)

Lisad

I. Vana veebisaidi pildid



Joonis 34. Vana veebisaidi avaleht



Joonis 35. Vana veebisaidi töötajate vaheleht

Ahja perearst
Dr. Diana Kirss OÜ

[AVALEHT](#)
[TEADAANDED](#)
[TÖÖTAJAD](#)
[HINNAKIRI](#)
[KASULIKUD LINGID](#)
[KONTAKT](#)

Teadaanded

D- vitamiini vajalikkus 06/09/2020 ^

Soovitame kõikidel inimestel tarvitada igapäevaselt D- vitamiini toidulisandeid.

D vitamiin on hädevajalik kaltsiumi imendumiseks, luustiku tervise tagamiseks ning immuunsüsteemi talituseks. Samuti on uuringutel jõutud järelduseni, et D- vitamiinil on kaitsv toime pahaloomuliste kasvajate tekke eest. D-vitamiin tekib nahas päikese toime, kuid meie kliimas ei ole kunagi piisavalt, et tagada normaalset D vitamiini taset organismis. Toiduga saab D- vitamiini väga piiratud. Seega, me vajame seda juurde igapäevaselt. Soovitame võtta õlikapsleid või tilku, tablettidest imendub D vitamiin halvemini.

Profülaktilised D vitamiini annused:

Vastsündinutele soovitatakse kolmandast-neljandast elupäevast anda (400 IU)10 µg päevas.

1-12. aastased võiksid võtta(400–800 IU) 10–20 µg päevas.

12-18.aastased(600–1000IU) 15-25 µg päevas.

Terved täiskasvanud(1000 IU)25 µg/päevas.

Vanurid ja tubased täiskasvanud 1200–2000 IU/p 30–50 µg/päevas.

Kui te ei ole mitu aastat D vitamiini võtnud, siis on mõistlik alustada suurema annusega, nt. 4000 IU päevas kuu jooksul ja edasi jätkata profülaktilises annuses.

Raviannused võivad olla suuremad, neid määrab arst vastavalt vajadusele.

Joonis 36. Vana veebisaidi teadaannete vaheleht

Ahja perearst
Dr. Diana Kirss OÜ

[AVALEHT](#)
[TEADAANDED](#)
[TÖÖTAJAD](#)
[HINNAKIRI](#)
[KASULIKUD LINGID](#)
[KONTAKT](#)

Hinnakiri

Teenuste eest saab maksta ainult sularahas!

TÖENDID	VISIIDITASUD	VAKTSINEERIMISE JÄRGNEVATE HAIGUSTE VASTU
Ambulatoorne vastuvõtt	Tasuta	
Kodukutsed	Tasuta	
Kindlustamata inimese vastuvõtt	15.00 €	

Joonis 37. Vana veebisaidi hinnakirja vaheleht

Kasulikud lingid

Link	Seletus
Eesti haigekassa	-
Terviseamet	-
Patsiendiloortaal	-
Raviühendid	-
Informatsioon loomumise kohta	-
Informatsioon diabeedi kohta	-
Informatsioon valmiste tervise kohta	-
Informatsioon vaktsineerimise kohta	-
Informatsioon süttetamisest loomumise kohta	-
Seksuaaltervis	-
Naistele rinnavähist	-
Urinöödamatus	-
Meesete	-
Meesete eesnäilmerst	-
Peavalu	-

Joonis 38. Vana veebisaidi kasulikud lingid vaheleht

Ahja perearst Dr. Diana Kirss OÜ

[A VALEHT](#)
[TEADAANDED](#)
[TÖÖTAJAD](#)
[HINNAKIRI](#)
[KASULIKUD LINGID](#)
[KONTAKT](#)

Address
 Nooruse 2-1, Ahja, Põlva vald

Vastuvõetusjad
 E,T,K,R 09.00-13.00 N 13.00-17.00

Telefon
 7921476

E-mail
heljuur@hot.ee

Nooruse 2
 Nooruse 2, Ahja, 63710 Põlva maakond
 Vaadake suuremat kaarti

Kaardidraht 02021 Kasutusõigused Teatage veast kaardi

Joonis 39. Vana veebisaidi kontakti vaheleht

II. Disain



Joonis 40. Avalehe disain

Dr. Diana Kirss OÜ

KODUKORRAST

TEADAANDED



TEADAANDED

Siit leiate viimased teadaanded Ahja perearstikabineti poolt!

TEADAANNE 1	06/09/2020	✓
TEADAANNE 2	06/09/2020	✓
TEADAANNE 3	06/09/2020	✓
TEADAANNE 4	06/09/2020	✓
TEADAANNE 5	06/09/2020	✓
TEADAANNE 6	06/09/2020	✓

Address
Nooruse 1-2, Ahja, Põlva vald

Telefon
7921476

Email
email@email.com

Dr. Diana Kirss OÜ, 2021

Joonis 41. Teadaannete disain

Dr. Diana Kirss OÜ

KODUKORRAST

TEADAANDED



KODUKORRAST

Palun tutvuge meie kodukorraga! Küsimuste tekkimisel helistage või saatke meile kiri!

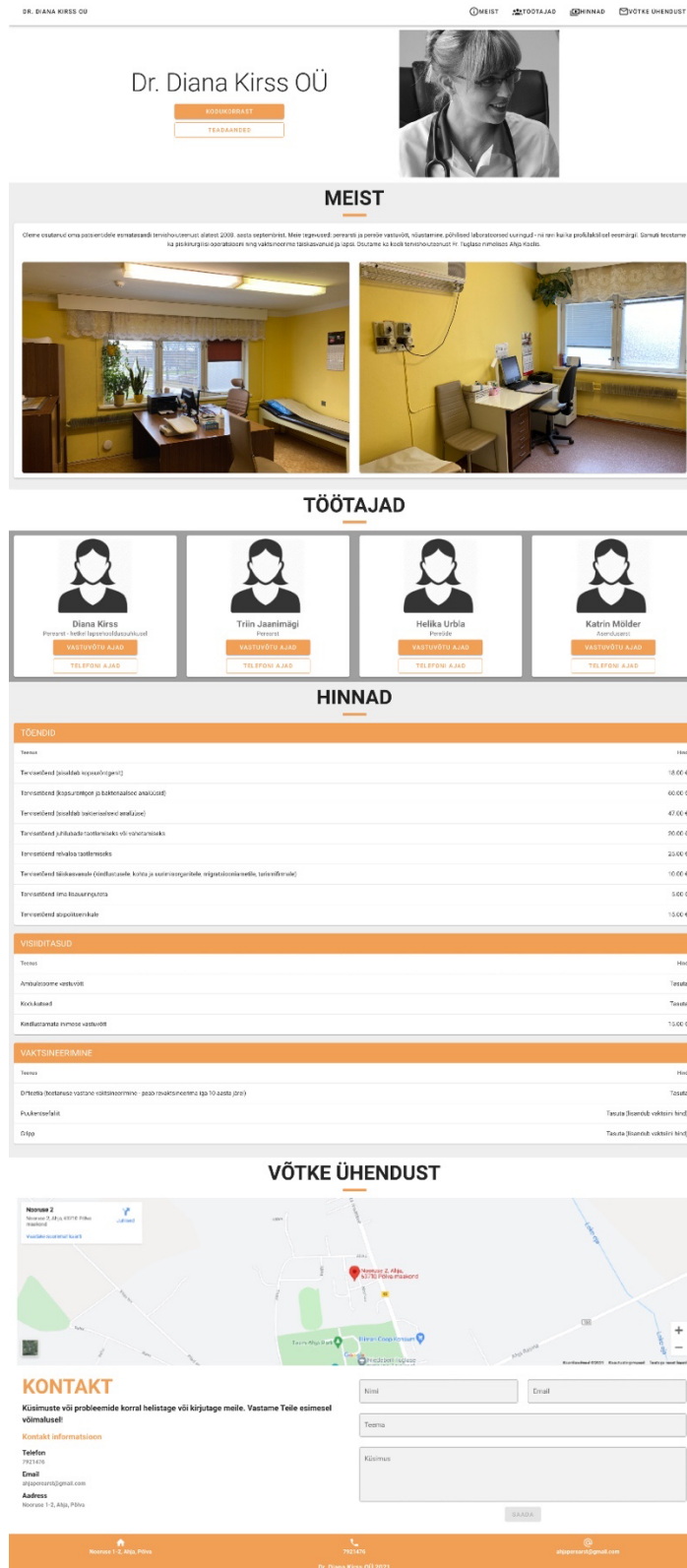
KODUKORRAST	
Vastuvõtule registreerimisest	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris vel interdum leo, vel gravida augue. Donec neque odio, interdum eget euismod quis, placerat eu leo. Phasellus mattis rhoncus velit, ut ullamcorper dolor dignissim ut. Curabitur et commodo massa. Cras vitae pellentesque eros, vel vestibulum neque. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. In hac habitasse platea dictumst. Pellentesque pharetra diam metus, ut ultrices orci porttitor ornare. Phasellus efficitur, purus at gravida euismod, felis nisl consequat libero, dapibus dictum massa quam et urna. Vestibulum suscipit, eros at pulvinar pharetra, mauris velit laoreet justo, ut auctor sem felis sed mi. Vestibulum finibus odio enim, a elementum nunc euismod at. Morbi luctus maximus tellus semper scelerisque.
Retseptide tellimisest	
Töövõimetus- ja hoolduslehtedest	
Telefoniajad	
Analüüside võtmine	
Nimistusse registreerimine	Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Proin facilisis diam a ex iaculis, a efficitur orci congue. Integer tempor eros in nisi venenatis viverra. Quisque pretium sem in quam pharetra interdum. Fusce consectetur molestie elementum. Etiam eu justo viverra, facilisis velit vitae, vulputate risus. Sed semper nulla at neque vehicula, a vulputate ipsum porta. Mauris ultrices enim ut orci efficitur pellentesque. Phasellus et lorem eu justo euismod molestie sed quis odio. Mauris eu turpis ac neque congue venenatis. Vestibulum at elit a massa tincidunt molestie sit amet pharetra eros. Duis quis enim mattis, efficitur risus vitae, convallis velit. Nullam vehicula fermentum tincidunt. In dolor diam, molestie et ante sed, iaculis gravida diam. Sed pharetra posuere vulputate.
Juhilubade taotlemisest	
Väljaspool perearsti tööaega erakorraliste probleemid	
Muud kontaktnumbrid	

Address Nooruse 1-2, Ahja, Põlva vald	Telefon 7921476	Email email@email.com
---	---------------------------	---------------------------------

Dr. Diana Kirss OÜ, 2021

Joonis 42. Kodukorrrast disain

III. Uue veebisaidi pildid



Joonis 43. Uue veebisaidi avaoleht

Dr. Diana Kirss OÜ

KODUKORRAST

TEADAANDED



KODUKORRAST NING KASULIKUD LINGID

KODUKORRAST

VASTUVÕTULE REGISTREERIMISEST

RETSEPTIDE TELLIMISEST

TÖÖVÕIMETUS- JA HOOLDUSLEHTEDEST

TELEFONIAJAD

ANALÜÜSIDE VÕTMINE

NIMISTUSSE REGISTREERIMINE

JUHILUBADE TAOTLEMISEST

VÄLJASPOOL PEREARSTI TÖÖAEGA
ERAKORRALISTE PROBLEEMIDEGA SAATE
PÕRDUDA

MUUD KONTAKTNUMBRID

Palume perearsti vastuvõtule alati registreerida! Erakorraliste probleemidega on võimalik eelnevalt registreerimata pöörduda pereõe vastuvõtule. Ägeda haigestumise korral teeb esmase nõustamise pereõde ja kiiret lahendamist vajava terviseprobleemi puhul registreerib õde Teid vajadusel arsti vastuvõtule samaks päevaks! Vastuvõtule registreerib õde telefoni teel või koha peal, registreerimisel palume Teil lühidalt probleemi kirjeldust. Ühele patsiendile on ette nähtud 20 minutit läbivaatuseks. Juhul, kui soovite rääkida mitmest probleemist või teiega on kaasas pereliige, kes vajab ka arsti poolset konsultatsiooni - palun registreeriga kaks aega. Kui te ei saa registreeritud ajal vastuvõtule tulla, palun andke meile sellest teada, siis saame oma tööd paremini organiseerida.

LINGID

Link

[Eesti haigekassa](#)[Terviseamet](#)[Patsiendiportaal](#)[Ravijuhendid](#)[Informatsioon toitumise kohta](#)[Informatsioon diabeedi kohta](#)[Informatsioon vaimse tervise kohta](#)[Informatsioon vaktsineerimise kohta](#)[Informatsioon suitsetamisest loobumise kohta](#)

1-9 of 18

Nooruse 1-2, Ahja, Põlva

7921476

ahjaperearst@gmail.com

Dr. Diana Kirss OÜ 2021

Joonis 44. Kodukorrasst vaheleht



MEIST

Normal 

Oleme ositanud oma patsientidele eemaltasandi tervishoiuteenust alates 2008. aasta septembrist. Meie tugevused: perearsti ja pereõe vastuvõtt, nõustamine, põhilised laboratoorsed uuringud - nii ravi kui ka profülaktilisel eesmärgil. Samuti teostame ka piskirugilisi operatsioone ning vaktsineerimise täiskasvanud ja lapsi. Osutame ka kooli tervishoiuteenust Fr. Tugläse nimelises Ahja Koolis.





























TÖÖTAJAD

LISA UUS

Töötaja nimi	Ametikoht	Muuda/Kustuta
Diana Kirss	Perearst - hetkel lapsehoolduspuhkuses	 
Triin Jaanimägi	Perearst	 
Helika Ubbila	Pereõde	 
Katrin Mölder	Asendustarst	 





































HIINNAD

LISA UUS

Teenuse nimi	Hind	Klass	Muuda/Kustuta
Terviseteend (sisaldab kopsuröntgenit)	18.00 €	Tõendid	 
Terviseteend (kopsuröntgen ja bakteriaalsed analüüsid)	60.00 €	Tõendid	 
Terviseteend (sisaldab bakteriaalseid analüüse)	47.00 €	Tõendid	 
Terviseteend juhulabade taotlemiseks või vahetamiseks	20.00 €	Tõendid	 
Terviseteend rehvala taotlemiseks	25.00 €	Tõendid	 
Terviseteend täiskasvanule (kindlustusete, kohtu ja suurimorganitele, migratsiooniametile, turismifirmale)	10.00 €	Tõendid	 
Terviseteend ilma lisauringuteta	5.00 €	Tõendid	 
Terviseteend abipõltsenikule	15.00 €	Tõendid	 
Ambulatoorne vastuvõtt	Tasuta	Visiiditasud	 
Kodukutsed	Tasuta	Visiiditasud	 
Kindustamata inimese vastuvõtt	15.00 €	Visiiditasud	 
Diftetia (teetase vastane vaktsineerimine - peab reaktiveerima iga 10 aasta järel)	Tasuta	Vaktsineerimine	 
Puukentefaliti	Tasuta (lisandub vaktsiini hind)	Vaktsineerimine	 
Gripp	Tasuta (lisandub vaktsiini hind)	Vaktsineerimine	 

LINGID

LISA UUS

Järjekor	Nimi	Link	Muuda/Kustuta
01	Eesti haigekassa	https://www.haigekassa.ee	 
02	Terviseamet	https://www.terviseamet.ee	 
03	Patsiendiportaal	https://www.digiugu.ee	 
04	Ravijuhendid	https://www.ravijuhend.ee	 
05	Informatsioon tootumise kohta	https://www.toitumine.ee	 
06	Informatsioon diabeedi kohta	https://www.diabeet.ee	 
07	Informatsioon väimse tervise kohta	https://www.peaasi.ee	 
08	Informatsioon vaktsineerimise kohta	https://www.vaktsineeri.ee	 
09	Informatsioon sisetamistest loobumise kohta	https://www.tubakainfo.ee	 
10	Seksuaaltervis	https://miljõpohjust.hiv.ee	 
11	Naisetele rinnavähist	https://rinnavahk.ee/	 
12	Urinipidamatus	https://www.kuivaks.ee	 
13	Meestele	https://www.klinikum.ee/meestekliinik	 
14	Meestele- eesnäärmest	http://www.eesnaare.ee	 
15	Peavalu	https://www.peavalu.ee	 
16	Südameinfarkt	https://www.sudameinfarkt.ee	 
17	Ligeesed	https://www.liiges.ee	 
18	Narko	https://www.narko.ee	 

KONTAKTANDMED

Address	Telefon	Email
Nooruse 1-2, Ahja, Põlva	7921476	ahjaperearst@gmail.com



















UUDISED

LISA UUS

D-vitamiini vajalikkus	 
Uus veebileht valmis	 

KODUKORD

LISA UUS

VASTUVÕTULE REGISTREERIMISEST	 
Palume perearsti vastuvõtule alati registreerida! Erakorraliste probleemidega on võimalik eelnevalt registreerimata pöörduda pereõe vastuvõtule. Ägeda haigestumise korral teeb esmase nõustamise pereõde ja kliiret lähendamist vajava tervisprobleemi puhul registreerib Sõde Teid vigadusel arsti vastuvõtule-saamaks päevakas! Vastuvõtule registreerib Sõde telefoni teel või kohtu peal registreerimisel palume Teil lähidalt probleemi kirjeldada. Ühele patsiendile on ette nähtud 20 minutit läbivaatuseks. Juhul, kui soovite rääkida mitmet probleemist või teiega on kaasas pereliige, kes vajab ka arsti poolset konsultatsiooni - palun registreerige kaks aega. Kui te ei saa registreeritud ajal vastuvõtule tulla, palun andke meile sellest teada, siis saame oma tööd paremini organiseerida.	
RETSEPTIDE TELLIMISEST	 
TÖÖVÕMETUS- JA HOOLDUSLEHTEDEST	 
TELEFONIJAD	 
ANALÜÜSIDE VÕTMINE	 
NIMISTUSSE REGISTREERIMINE	 
JUHULABADE TAOTLEMISEST	 
VÄLJASPOOL PEREARSTI TÖÖAEGA ERAKORRALISTE PROBLEEMIDEGA SAATE PÖÖRDUDA	 
MUUD KONTAKTNUMBRID	 

Joonis 45. Administraatori vaheleht

IV. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, **Janar Kallson**,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose **Veebisaidi loomine Dr. Diana Kirss OÜ-le**, 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'i litsentsiga CC BY NC ND 3.0, 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.

Janar Kallson

30.04.2021