

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Informaatika õppekava

**Ahti Ankro**

**Veebiliidesega infosüsteemi loomine liikumiseks  
Tartu Ülikooli hoonete vahel**

**Bakalaureusetöö (9 EAP)**

Juhendaja: Vambola Leping

Tartu 2023

# **Veebiliidesega infosüsteemi loomine liikumiseks Tartu Ülikooli hoonete vahel**

## **Lühikokkuvõte:**

Töö eesmärk on luua veebiliidesega infosüsteem, kus on võimalik saada juhiseid Tartu Ülikooli hoonete vaheliseks liikumiseks. Töös vaadatakse olemasolevaid süsteeme ja pannakse kirja loodava süsteemi nõuded. Kirjeldatakse kasutatuid tehnoloogiaid ning valminud süsteemi ja selle testimist. Vaadatakse ka võimalikke edasiseid arendamise võimaluseid.

## **Võtmesõnad:**

Andmebaas, Veebileht, MySQL, React

**CERCS:** P175 Informaatika

# **Creating an information system with a web interface for moving between the buildings of the University of Tartu**

## **Abstract:**

The goal of the thesis is to create an information system with a web interface, where it is possible to get instructions for movement between the buildings of the University of Tartu. This thesis contains information about already existing systems and requirements for the system that is being created. Also it contains descriptions for used technologies, the created system and the testing of the system. In addition, it also discusses what are the possible future development possibilities.

## **Keywords:**

Database, Website, MySQL, React

**CERCS:** P175 Informatics

# Sisukord

<b>Sissejuhatus.....</b>	<b>4</b>
<b>1. Sarnased süsteemid ja tehnoloogiad.....</b>	<b>5</b>
<b>2. Loodava süsteemi nõuded ja kasutajalood.....</b>	<b>7</b>
<b>3. Kasutatud tehnoloogiad.....</b>	<b>8</b>
3.1 Andmebaasi ja tagateenuste loomise tehnoloogiad.....	8
3.2 Kasutajaliidese loomisel kasutatud tehnoloogiad.....	9
3.3 Muud tehnoloogiad.....	10
<b>4. Süsteemi kirjeldus.....</b>	<b>11</b>
4.1 Andmed ja andmebaas.....	11
4.2 Veebiliidese kirjeldus.....	11
4.3 Marsruudi leidmine.....	13
<b>5. Testimine.....</b>	<b>17</b>
<b>6. Edasised arendused.....</b>	<b>19</b>
<b>7. Kokkuvõte.....</b>	<b>20</b>
<b>8. Viidatud kirjandus.....</b>	<b>21</b>
<b>LISAD.....</b>	<b>23</b>
1. Väljavõtted koodist.....	23
2. Litsents.....	25

## Sissejuhatus

On olemas süsteeme, milles on võimalik planeerida marsruute erinevate asukohtade vahel, kasutades mitmeid transpordivõimalusi. Eestis on olemas platvorm peatus.ee[1], kus on võimalik leida ühistranspordi liinid valitavate peatuste või asukohtade vahel. Veel on kasutatav ka Google Maps[2].

Käesoleva bakalaureusetöö eesmärk on luua veebiliidesega infosüsteem, kus on võimalik saada juhiseid Tartus asuvate ülikooli hoonete vaheliseks liikumiseks, mille eelis juba olemasolevate süsteemidega oleks kasutamise lihtsus, kuna selles süsteemis pole vaja sisestada aadresse või neid peast teada, vaid valida lihtsalt kaks hoonet ja transpordimeetod.

Töö koosneb kuuest peatükist. Esimese peatükis vaadatakse juba olemasolevaid sarnaseid süsteeme ja tehnoloogiaid mida nad kasutavad. Teises peatükis tuuakse välja kasutajalood ning loodava süsteemi nõuded. Kolmandas peatükis kirjeldatakse tehnoloogiaid, mida on kasutatud selle süsteemi loomiseks. Neljandas peatükis kirjeldatakse valminud süsteemi. Viiendas peatükis antakse ülevaade testimisest. Kuuendas peatükis arutatakse võimalikke edasiseid arendusi.

# 1. Sarnased süsteemid ja tehnoloogiad

On olemas mitmeid süsteeme, kus on ühistranspordi või ka lihtsalt erinevad liikumisvõimalused kohtade vahel ning nende marsruudid. Selline süsteem on olemas ka Eestis ning mitmel pool mujal, siin on välja toodud kuidas osad neist töötavad ja nende poolt kasutatavad tehnoloogiad.

Eestis on olemas platvorm peatus.ee[1], kus on võimalik valida peatus, liin või asukoht. Valides peatuse näidatakse kõiki sealt väljuvaid bussi- või rongiliine ning liini valides kuvatakse see kaardil. Asukoha valides näidatakse lähedal olevatest peatustest väljuvaid liine. Lisaks on võimalik valida kaardil kaks punkti, mille puhul kuvatakse vastavalt valitud transpordiliigle (ühistransport, kõndimine, jalgratas) marsruut, valitud punktide vahel liikumiseks. Peatus.ee platvormi puhul on peatuste ja liinide andmed saadud Maanteeameti Ühistranspordiregistrist ning kaarditeenusena kasutatakse OpenStreetMap'i[3]. Lisaks kasutatakse ka Digitransit[4] platvormi, mis on marsruutimisplatvorm, mille on välja töötanud HSL (Helsingi Regiooni Transpordiamet) ja Traficom[5].

Kaarditeenus OpenStreetMap on kogukonnapõhine kaardisüsteem, kuhu on võimalik kasutajatel panustada oma andmetega, et kaarte paremaks teha. Lisaks on kõik andmed avalikult kättesaadavad ja vabalt kasutatavad, mille tõttu on see ka kasutusel paljudes rakendustes.

Andmed, mida peatus.ee kasutab on saadud Maanteeametilt, kes on koondanud ühistranspordi info kokku ühistranspordi infosüsteemi (ÜTRIS)[6]. Seal olevad andmed on avaandmed ja kättesaadavad kõigile. Need andmed on GTFSi (*General Transit Feed Specification*) formaadis, mis on ühistranspordi sõidugraafikute ja nendega seotud geoandmete formaat, mille on välja töötanud Google.

Väljaspool Eestit on olemas UNIBUS[7], mis tegutseb Suurbritannias Bournemouth regioonis. See on suunatud sealse ülikooli tudengitele ja töötajatele, aga ka laiemale avalikkusele. Sellel platvormil on samuti võimalik valida peatused, millest näidatakse sealt väljuvaid bussiliine või valida kaks punkti või peatust, peale mida kuvatakse nende vahel liikumiseks liinid ja marsruudid. Samuti on sellel süsteemil ka lisavõimalus, kus on võimalik

vaadata busside asukohti reaalsajas, et oleks võimalik paremini planeerida, kui buss peaks hilinema. Kaarditeenusena kasutatakse UNIBUS puhul samuti OpenStreetMap'i.

Eelmise kahe näite puhul ei ole nende veebilehtedel välja toodud, kuidas selline süsteem detailsemalt toimib. Sellest on võimalik rohkem infot saada S. C. Pun-Chengi artiklist[8], kus on välja toodud täpsemalt, kuidas sarnane ühistranspordi süsteem toimib Hong Kongis kasutuses oleva ühistranspordi planeerimise süsteemi (PTES) põhjal. Hong Kongis kasutuses olev ühistranspordi koosneb kolmest alamsüsteemist. Ühistranspordi alamsüsteem, kus on järgmised andmebaasid:

- 1) Kõigi ühistranspordi vormide (rong, tramm, buss, praam) marsruutide nimed ning kas see liin on ühe- või kahe-suunaline.
- 2) Piletihindade andmebaas.
- 3) Geograafiline andmebaas, peatuste asukohtade hoidmiseks.
- 4) Muu kasulik info, näiteks peatuste nimed, aadressid, piirkonnad.

See info tuleb transpordi firmadelt ja Transpordi Ametilt. Teiseks on aadressi alamsüsteem, kus on vaatamisväärsuste ja ostukeskuste asukohad, mille info saadakse Hong Kongi turismi agentuurilt ning muude huvipunktide asukohad. Nende andmete hoidmiseks on ka geograafiline andmebaas. Viimaseks on kaartide alamsüsteem, mis kasutab geoinfosüsteemi (GIS).

GIS on arvutisüsteem geograafiliste andmete hoidmiseks ja kuvamiseks. GIS võib ühel kaardil näidata palju erinevat infot, näiteks tänavad, hooned ja taimkate[9]. GIS süsteemi kasutamise eelised on välja toonud C. Keechoo ja J. Wonjae oma artiklis[10], nendest peamine on, et aluskaart, mis on saadaval tavaliselt väikese tasu eest on kasutatav otse. Võrreldes tavalise meetodiga, mida kasutatakse selle süsteemi loomisel on GIS täpsem, järjepidevam ja nõuab vähem tööd.

Lisaks on ka olemas üldisem teenus Google Maps[2], mis on kaarditeenus, milles on võimalik planeerida marsruute, seal on auto ja jalgsi liikumisele lisaks võimalik ka leida bussiliinid, selle töö käigus valminud süsteemi jaoks on ka seda kasutatud marsruutide kuvamiseks.

Tuginedes vaadatud sarnastele süsteemidele ja algsele ideele on järgnevalt pandud kirja süsteemi nõuded ja kasutajalood.

## **2. Loodava süsteemi nõuded ja kasutajalood**

Süsteemi loomiseks on välja toodud kasutajalood ning kirja pandud nõuded, et oleks ülevaade, mida süsteem peab suutma teha.

### **Kasutajalugu 1: Üliõpilane**

Üliõpilasena soovin ma valida kaks ülikooli hoonet, ühe alguspunktina ja teise lõpppunktina. Peale seda tahan ma valida transpordi meetodi ning saada juhised kuidas nende kahe hoone vahel liikuda.

### **Kasutajalugu 2: Administraator**

Administraatorina, soovin ma lisada süsteemi uusi ülikoolihooneid või muuta nende andmeid, et hoida süsteem ajakohasena.

### **Nõuded**

Funktsionaalsed nõuded on:

- Kasutajal peab olema võimalik valida kaks ülikooli hoonet ja transpordi tüüp nende vahel.
- Peale hoonete ja transpordi tüübi valiku kinnitamist, peab kasutaja saama näha marsruuti teenuses Google Maps [2].
- Kasutajal peab olema võimalik muuta valitud ülikooli hoonet ja transpordi meetodit.
- Kasutajale peab olema näha ülikooli hoone nimi ja aadress.
- Kasutajal peab olema võimalus navigeerida eri vaadete vahel.
- Administraatoril peab olema võimalik oma kasutajaga sisse logida.
- Administraatoril peab olema võimalik lisada uusi ülikooli hooneid süsteemi.
- Administraatoril peab olema võimalik muuta ja kustutada olemasolevaid ülikooli hooneid.

Mittefunktsionaalsed nõuded on:

- Veebiliidest peab olema võimalik kasutada ka nutitelefonis.
- Veebiliidese kasutamine peab olema lihtne ja arusaadav.
- Süsteem peab olema kasutatav peamistes veebilehitsejates: Google Chrome, Firefox, Microsoft Edge, Safari.

Nende nõuete põhjal on valitud tehnoloogiad, mida kasutatakse süsteemi loomiseks. Nendest saab ülevaate järgmises peatükis.

### 3. Kasutatud tehnoloogiad

Süsteemi loomisel on kasutatud mitmeid erinevaid tehnoloogiaid, millest saab selles peatükis ülevaate, kus on välja toodud, milleks neid kasutatud on ja miks on valitud just need tehnoloogiad.

#### 3.1 Andmebaasi ja tagateenuste loomise tehnoloogiad

##### MySQL

Andmebaasi haldamiseks on kasutatud süsteemi MySQL[11], mis on avatud lähtekoodiga relatsiooniliste andmebaaside haldamise süsteem. Relatsiooniline andmebaas hoiustab andmeid eraldi tabelites, mille vahel on võimalik reeglitega luua ühendused. MySQL on valitud, kuna:

- See on üks kõige laialdasemalt kasutatav andmebaaside haldamise süsteem, tänu millele on ka selle kasutamiseks palju juhiseid ja materjale.
- On lihtne seadistada ja kasutada.
- Andmete lugemine on kiire, mis on loodava süsteemi juures oluline.

##### Node.js

Node.js[12] on avatud lähtekoodiga JavaScript'i[13] käituskeskkond. Node.js'i on kasutatud veebirakenduse tagateenuste (ingl. k. *back-end*) loomiseks. Selle juures on kasutatud ka lisaks Node.js raamistikku Express[14], mis teeb oma pakutavate võimalustega Node.js kasutamise lihtsamaks.

Node.js üks peamisi eelised on, et ta põhineb JavaScript'il, seega ei ole vaja kasutada tagateenuste loomiseks mingit teist programmeerimiskeelt, kuna JavaScript on ka kasutusel React raamistikus.

##### Axios

Axios[15] on lubaduste põhine Node.js HTTP (*Hypertext Transfer Protocol*) klient brauserite jaoks. Axios aitab teha HTTP taotlusi Node.js'is. Neid taotlusi kasutatakse veebirakenduse tagateenustega suhtlemiseks.



## Bcrypt

Bcrypt[16] on räsifunktsioon, mis on loodud Niels Provos ja David Mazières poolt ning seda kasutatud paroolide kodeerimiseks. Bcrypt'i teevad turvaliseks omadused:

- Bcrypt nõuab alati soola (juhuslike andmete rea) kasutamist, mis raskendab saadud räsi ära arvamist.
- Kuna see räsifunktsioon on loodud paroolide kodeerimiseks on see ka aeglane, mis vähendab ründajal parooli arvamise võimalusi.

## 3.2 Kasutajaliidese loomisel kasutatud tehnoloogiad

Kasutajaliidese loomisel on kasutatud keeli:

- HTML (*Hypertext Markup Language*), mida on kasutatud veebilehe märgendamiseks[17].
- CSS (*Cascading Style Sheets*), mida on kasutatud veebilehe kujundamiseks[18].
- Programmeerimiskeel JavaScript, mida on kasutatud veebilehe funktsionaalsuse loomiseks[13].

## React

React[19] on programmeerimiskeele JavaScript raamistik kasutajaliideste loomiseks. React'i on kasutatud kuna tal on mitmed eelised[20]:

- Kuna React on komponendipõhine, mis teeb kasutajaliidese loomise ja selle osade haldamise lihtsamaks ning kiiremaks.
- Kõrge jõudluse tase, kuna React haldab ja otsustab, millal on vaja lehe sisu uuendada ja värskendada, mis teeb dünaamilise sisuga veebirakenduste laadimise kiireks.
- Lihtne õppida ja kasutada, kuna React ei tutvusta nii palju uusi arendamise viise võrreldes puhta JavaScript'iga ega sunni järgima, kindlaid arendamise meetodeid.

Veel on kasutatud ka React'i enda raamistikku React Hook Form[21], mida on kasutatud vormide valideerimisel.

## Bootstrap

Bootstrap[22] on HTML-il, CSS-il ja JavaScript-il põhinev raamistik, mida on kasutatud kasutajaliidese loomisel, selle komponentide kujundamiseks. Bootstrap'i kasutamise eelised on[23]:

- Seda on lihtne kasutada, kuna ei ole vaja sügavaid HTML'i ja CSS'i teadmisi.
- See lihtsustab mobiilisõbralike veebirakenduste loomist
- See ühildub enamike veebibrauseritega: Chrome, Firefox, Internet Explorer, Safari, Opera jne.

Veebirakenduse loomisel on kasutatud Bootstrapi dokumentatsioonis leiduvaid juba valmis komponente.

### **3.3 Muud tehnoloogiad**

#### **Google Maps**

Google Maps[9] on kaarditeenus, mida on kasutatud ülikooli hoonete vaheliste marsruutide kuvamiseks. On olemas ka teisi kaarditeenuseid, kuid Google Maps on valitud, kuna see on kõige levinuim kaarditeenus, mida on lihtne kasutada nii brauseris, kui ka nutitelefonis. Lisaks oskab Google Maps ise leida ka bussiliinide marsruudid.

#### **Visual Studio Code**

Visual Studio Code (VS Code)[24] on tekstiredaktor, mida on kasutatud koodi kirjutamiseks. See on valitud, kuna sellel programmil on võimalik integreerida, mitmeid erinevaid programmeerimiskeeli ja tehnoloogiaid. Lisaks on võimalik kasutada palju lisavõimalusi, mis teeb koodi kirjutamise lihtsamaks ja kiiremaks.

Neid tehnoloogiaid kasutades valminud süsteemi tööpõhimõtet ja veebiliidese välimust kirjeldatakse järgnevalt. Lisaks on võimalik näha näiteid nende tehnoloogiate kasutamisest Lisas 1.

## 4. Süsteemi kirjeldus

Käesolevas peatükis kirjeldatakse süsteemi ja tema tööd ning andmeid, mida kasutatakse. Lisaks ka kust on kasutatavad andmed saadud.

### 4.1 Andmed ja andmebaas

Selle süsteemi kasutatavas andmebaasis on kolm tabelit:

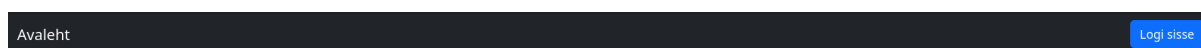
- Hooned, kus on iga hoone nimi, aadress ja koordinaadid.
- Kasutajad, mis sisaldab kasutajate e-maile, koos krüpteeritud paroolidega.
- Rattaringluse parklad, kus on iga parkla aadress ja koordinaadid.

Süsteemis kasutatavate rattaringluse parklate andmed on saadud Tartu linna geoHUB'i avaandmetest[25]. Sealt on võetud parklate aadressid ja nende koordinaadid, mis on salvestatud andmebaasi eraldi tabelisse. Tartu Ülikooli hoonete andmed on saadud Tartu Ülikooli kinnisvaraosakonnalt[26].

### 4.2 Veebiliidese kirjeldus

#### Päis

Päises on vasakul link „Avaleht”, mis viib avalehele ning paremal on sisselogimise nupp, mis viib lehele, kus on võimalik administraatoril sisse logida. Päis on olemas igal lehel.



Joonis 1. Veebilehe päis

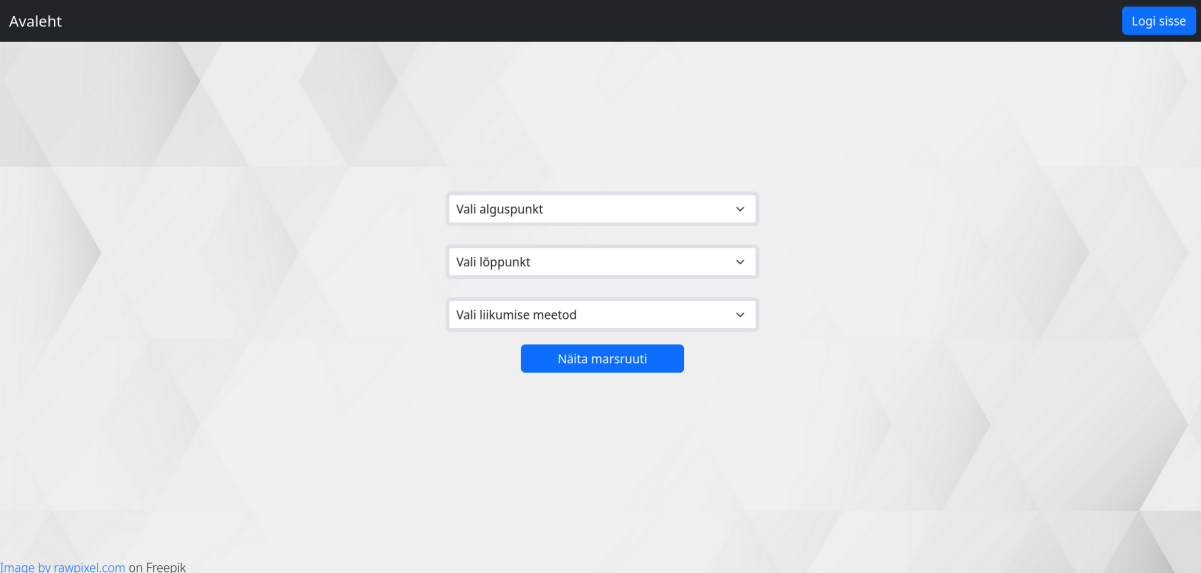
#### Avaleht

Peamine leht, kus on võimalik menüüst valida alguspunktiks ja lõpp-punktiks ülikooli hooned, mille vahel soovitakse liikuda. Veel on võimalik valida ka järgmiste transpordi meetodite vahel:

- Kõndimine
- Buss
- Auto
- Jalgratas (enda ratas)

- Jalgratas (rattaringlus)

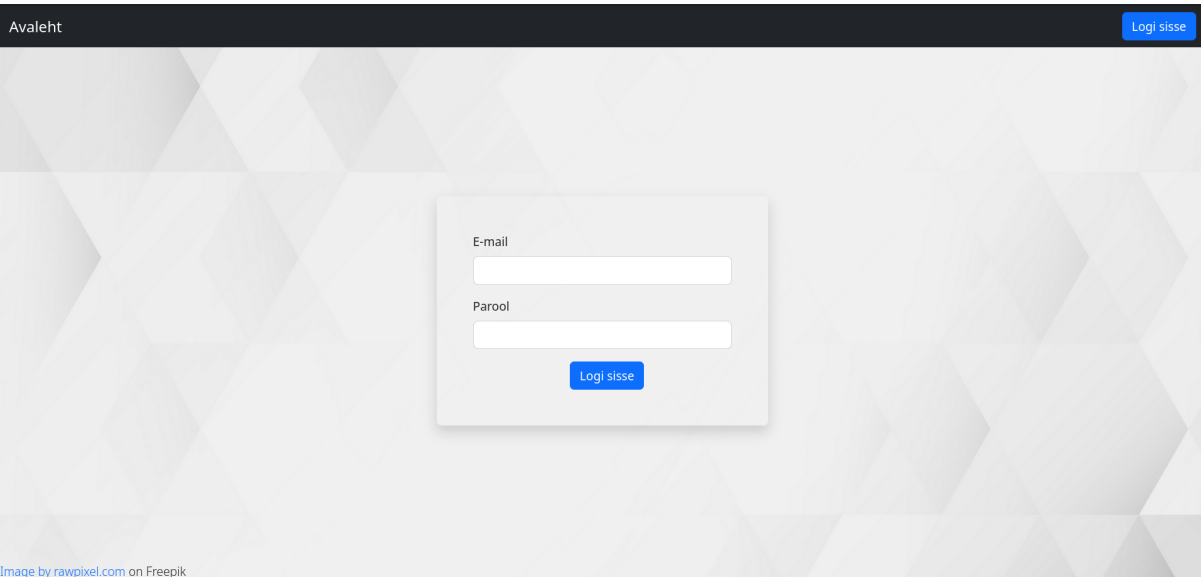
Peale hoonete ja transpordi meetodi valimist, vajutades nupule „Näita marsruuti” kuvatakse marsruut kaarditeenuses Google Maps[9].



Joonis 2. Veebilehe avaleht.

## Sisselogimine

Sisselogimise lehel on võimalik administraatoril sisse logida kasutades e-maili ja parooli. Peale edukat sisselogimist viiakse kasutaja administraatori vaate lehele.



Joonis 3. Sisselogimise leht.

## Administraatori vaade

Administraatori vaates asendatakse päises olev „Logi sisse” nupp „Logi välja” nupuga ning päisesse ilmub ka veel link, mis „Hoonete muutmine”, mis viib administraatori vaatesse. Veel kuvatakse loetelu ülikooli hoonetest, kus on võimalik iga hoone andmeid muuta, vajutades „Muuda” nupule ning kustutada vajutades „Kustuta” nupule. Lisaks on võimalik lisada uus hoone „Lisa hoone” nupuga.

Avaleht

Hoonete muutmine

Logi välja

Hoone nimi

Aadress

Pikkuskraad

Laiuskraad

Lisa Hoone

ID	Nimi	Aadress	Pikkuskraad	Laiuskraad	Operatsioonid
1	Delta	Narva mnt 18, Tartu	26.72534	58.38508	<div>Muuda</div> <div>Kustuta</div>
2	Physicum	W. Ostwaldi 1, Tartu	26.69023	58.36618	<div>Muuda</div> <div>Kustuta</div>

Joonis 4. Administraatori vaade.

Peale „Muuda” nupule vajutamist muutub rida, kus hoone andmed on kuvatud, muudetavaks ning tekivad „Salvesta” ja „Katkesta” nupud, mis vastavalt, kas salvestavad või katkestavad tehtud muudatused.

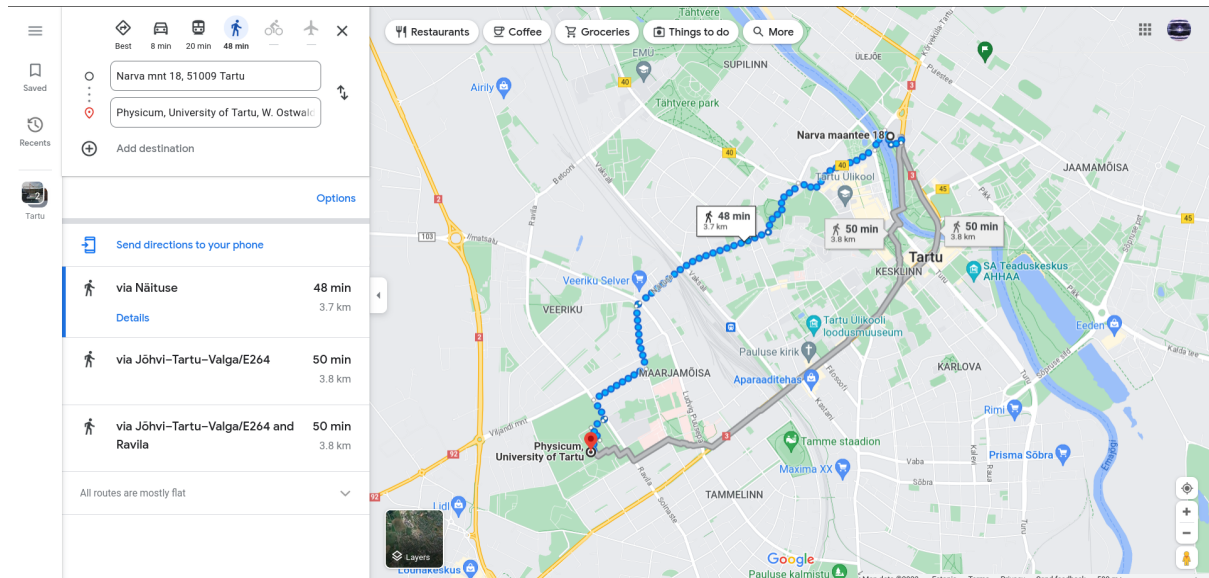
ID	Nimi	Aadress	Pikkuskraad	Laiuskraad	Operatsioonid
1	Delta	Narva mnt 18, Tartu	26.72534	58.38508	<div>Muuda</div> <div>Kustuta</div>
2	<div>Physicum</div>	<div>W. Ostwaldi 1, Tartu</div>	<div>26.69023</div>	<div>58.36618</div>	<div>Salvesta</div> <div>Katkesta</div>

Joonis 5. Administraatori vaates hoone andmete muutmine.

## 4.3 Marsruudi leidmine

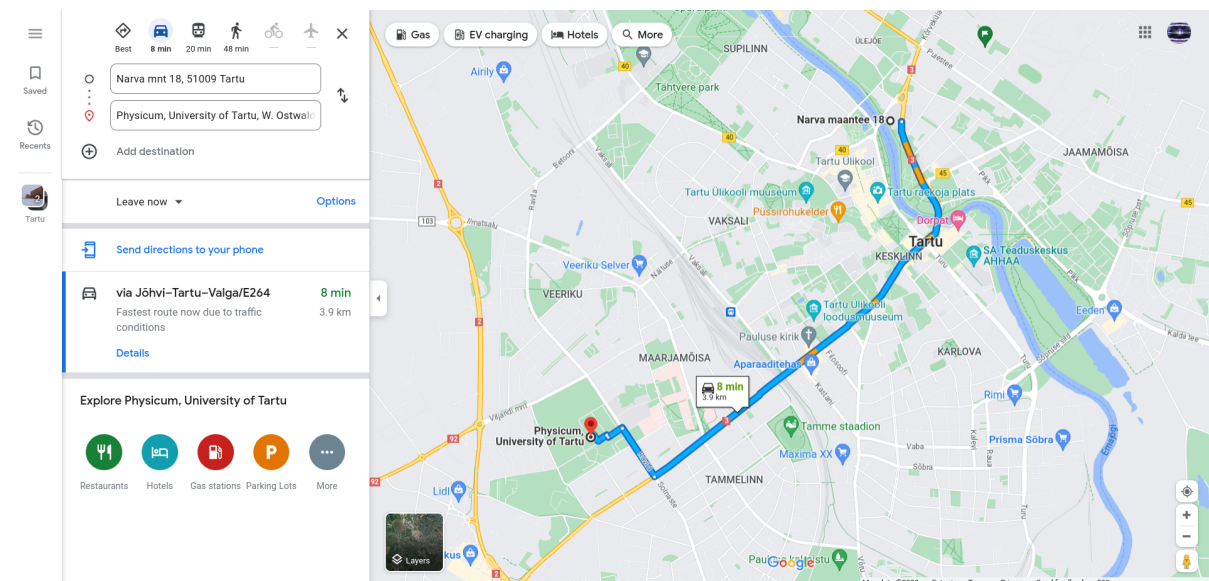
Kui kasutaja on valinud hooned ja transpordi meetodi ning vajutanud „Näita marsruuti” nuppu (joonis 2), siis kuvatakse vastavalt valitud transpordi meetodi põhjal see marsruut Google Maps’is.

Kui valitud transpordi meetod on kõndimine, siis saadab süsteem Google Maps'ile hoonete aadressid ja liikumise viisi, milleks sellel juhul on kõndimine. Nende andmete põhjal kuvab Google Maps vastava marsruudi.



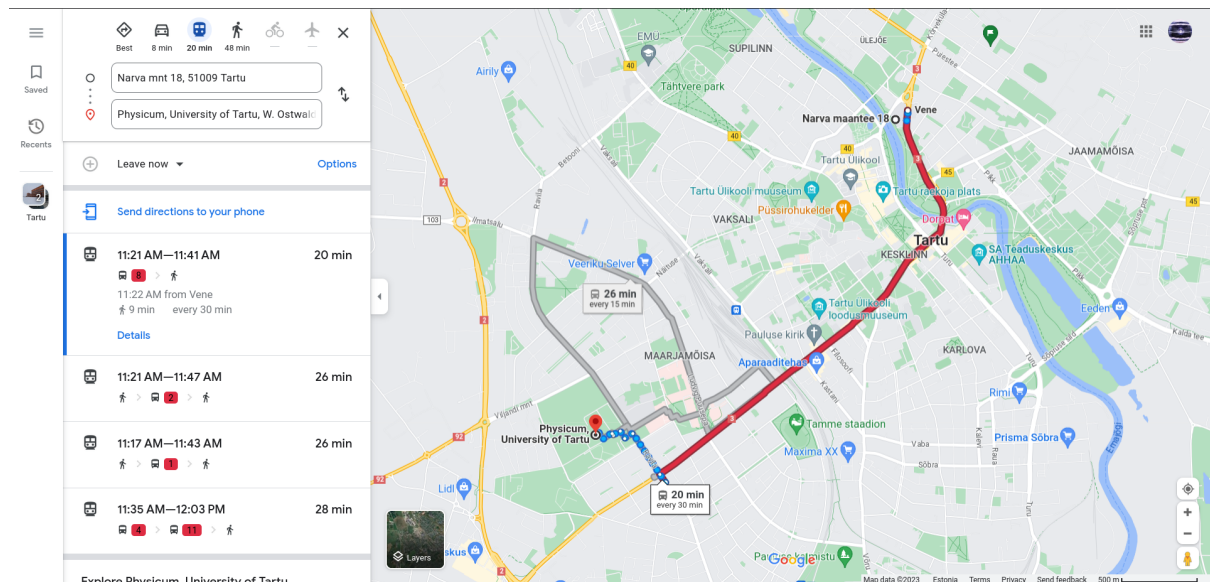
Joonis 6. Kõndimise marsruut Füüsika instituudi ja Delta maja vahel.

Kui on valitud transpordi meetodiks auto, jalgratas (enda ratas), siis toimib süsteem sama põhimõttega, kui kõndimise korral, selle erinevusega, et saadatud liikumise viis on sõitmine. Jalgratta puhul kasutatakse marsruudi kuvamiseks autoga samat teekonda, kuna Google Maps ei võimalda Tartus kasutada eraldi jalgratta marsruudi planeerimist.



Joonis 7. Auto ja enda jalgratta marsruut Füüsika instituudi ja Delta maja vahel

Kui valitud transpordi meetod on buss, siis on varasemaga erinevus jällegi liikumise meetodis, milleks see kord on buss. Sellisel juhul kuvab Google Maps erinevad busside liinid, koos kellaaegadega, millega on võimalik valitud hoonete vahel liikuda.

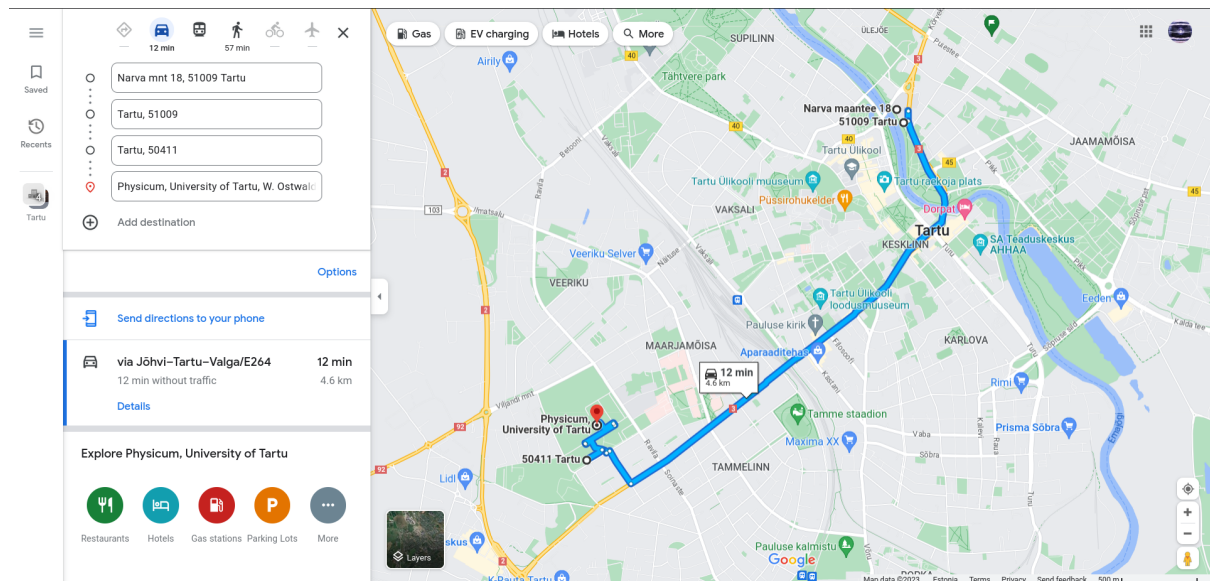


Joonis 8. Bussi marsruut Füüsika instituudi ja Delta maja vahel.

Rattaringluse puhul arvutatakse algselt linnulennult kõige lähimad rattaringluse punktid nii alguspunktile kui ka lõpp-punktile. Selleks kasutatakse järgmist valemit:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Selles valemis on  $x_1$ ,  $x_2$ ,  $y_1$  ja  $y_2$  vastavate koordinaatide laius- ja pikkuskraad. See valem ei ole täpne kera peal, aga kuna loodav süsteem arvutab vahemaid ainult Tartu linnas, kus vahemaad on väikesed, siis selle valemiga arvutamise tulemus on piisavalt täpne. Peale lähimate arvutust saadetakse Google Maps'ile lisaks hoonetele, ka leitud jalgrattaringluse punktide koordinaadid, mis lisatakse marsruudile vahepeatustena.



Joonis 9. Rattaringlusega marsruut Füüsika instituudi ja Delta maja vahel.

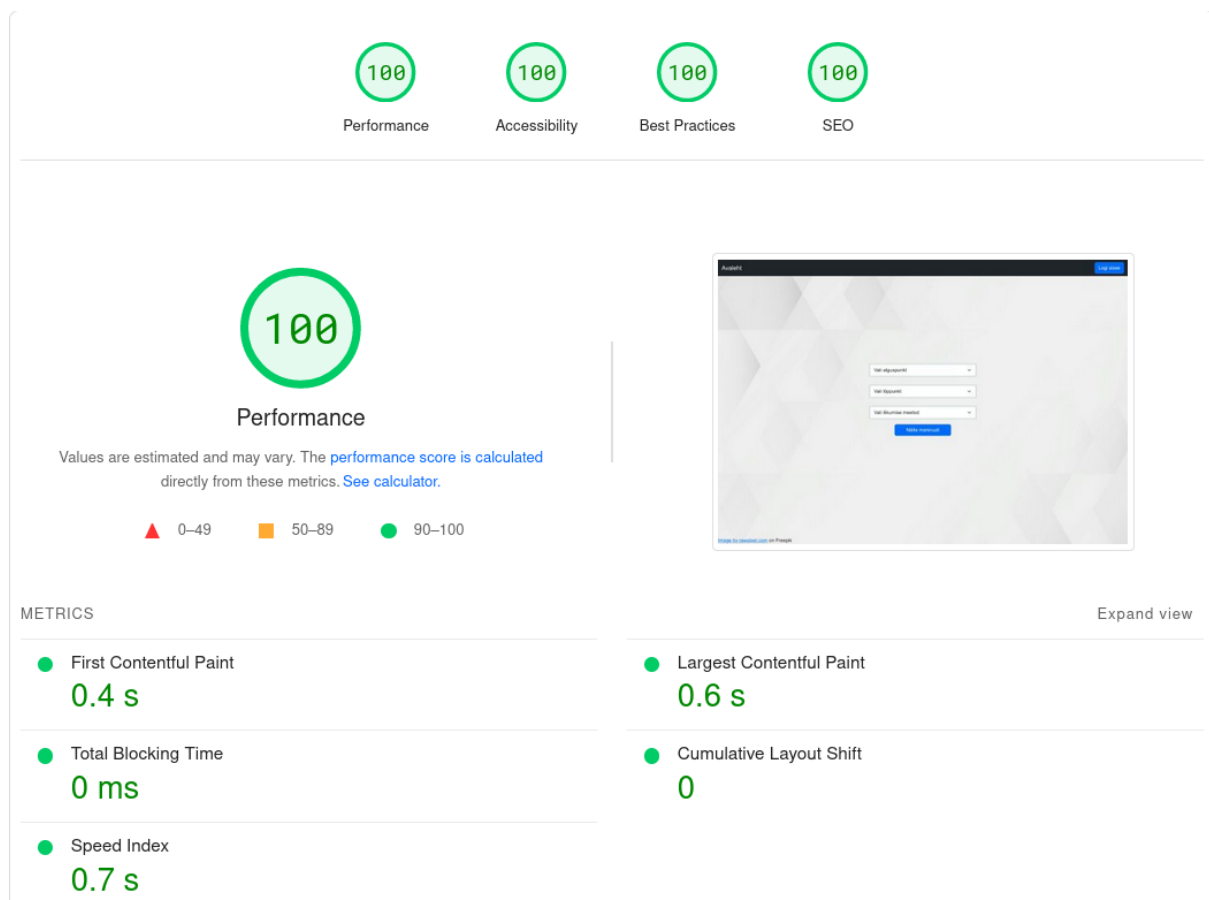
Valminud süsteemi on ka testitud erinevate testidega, millest saab ülevaate järgnevalt.



## 5. Testimine

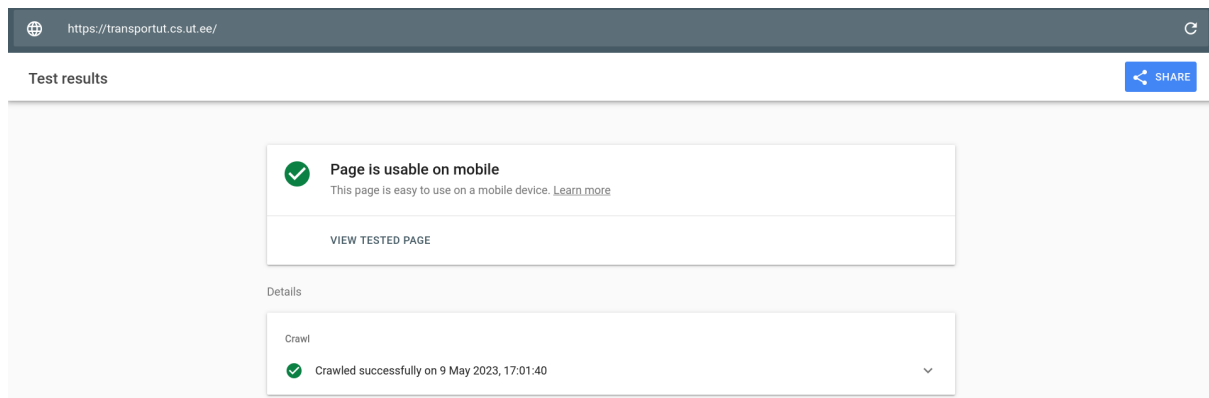
Arenduse käigus on kõikide funktsionaalsuste kontrolli jooksvalt käsitsi tehtud. Lisaks on veel kõik läbi testitud ka lõppversioonis ning selle käigus oli näha, et kõik funktsionaalsused töötavad.

Veebilehe testimiseks on ka kasutatud teenust PageSpeed Insights[27]. Selle testi käigus testiti jõudlust, ligipääsetavust ja ka parimaid tavaid. Veebileht läbis kõik selle teenuse poolt pakutavad testid edukalt (joonis 10).



Joonis 10. Testimise tulemused teenusega PageSpeed Insights.

Nutitelefonis kasutamist prooviti käsitsi läbi ning ka kasutades Google'i mobiilisõbralikkuse testi[28], mis oli samuti edukas (joonis 11).



Joonis 11. Google'i mobiilisõbralikkuse testi tulemus.

Veel kasutati ka teenust BrowserStack[29], millega testiti süsteemi tööd erinevates veebilehitsejates. Selle testimise käigus testiti süsteemi tööd veebilehitsejates: Google Chrome, Firefox, Microsoft Edge ja Safari. Kõikides nendes süsteem töötas.

Kuigi süsteem on valmis ning läbinud kõik testid, on seda võimalik veel edasi arendada. Erinevaid võimalusi on arutatud järgmises peatükis.

## **6. Edasised arendused**

Loodud süsteemi on võimalik veel täiendada ja mitmes suunas edasi arendada või laiendada.

Üks võimalus süsteemi laiendamiseks lisada ka väljaspool Tartut olevad Tartu Ülikooli hooned süsteemi ning pakkuda võimalust ka nende vaheliste marsruutide leidmiseks. Veel oleks ka võimalik pakkuda seda teenust teistele ülikoolidele või asutustele, millel on mitmeid hooneid.

Kuna praegu on veebiliides ainult eestikeelne oleks ka üks lisaarendus veebiliidese inglise või ka muudes keeltes kasutamise võimaluse loomine.

Lisaks on võimalik rattaringluse punktide korral ka kontrollida, kas selles parklas, kuhu süsteem juhatab on olemas vabu rattaid ning kui ei ole, siis juhatatakse kasutaja järgmisse lähimasse parklasse.

## 7. Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli luua veebiliidesega infosüsteem, mis oleks abiks Tartu Ülikooli hoonete vahel liikumiseks Tartu linnas, kasutades eri transpordi viise. Nendeks transpordi viisideks on kõndimine, buss, auto ja jalgratas. Jalgratta puhul on võimalik valida enda ratta ja rattaringluse ratta vahel.

Töös uuriti juba olemasolevaid süsteeme ja nende tööpõhimõtteid. Sarnast süsteemi on leida ka Eestis, kui ka mujal maailmas. On nii üldisi, kui ka kindla regiooni või ülikooli jaoks loodud variante.

Pandi kirja üliõpilase, kes on selle süsteemi peamine kasutaja, ja administraatori kasutajalood. Lisaks ka veel funktsionaalsed nõuded ja mittefunktsionaalsed nõuded, mille põhjal süsteem loodi.

Kirjeldati, mis allikatest on saadud kasutatavad andmed, milleks olid ülikooli hoonete aadressid ja rattaringluse punktid. Need andmed saadi vastavalt Tartu Ülikooli kinnisvaraosakonnalt[26] ja Tartu linna geoHUB'i avaandmetest[25]. Veel kirjeldati ka veebiliidest ja selle kasutamise võimalusi nii ülipilasel, kui ka administraatoril. Anti ka ülevaade, kuidas toimub marsruudi leidmine valitud transpordi meetodi põhjal.

Testiti ka süsteemi tööd nii töö käigus, kui ka valminud süsteemil, kasutades erinevaid teste. Veenduti, et kõik funktsionaalsed ja mittefunktsionaalsed nõuded oleks täidetud.

Viimasena vaadati ka võimalusi edasiseks arendamiseks, mille hulka kuuluvad nii süsteemi laiendamine väljapoole Tartut või teistele ülikoolidele, kui ka olemasoleva täiendamine ning lisafunktsionaalsuse lisamine.

Loodud süsteemi saab kasutada aadressil: <https://transportut.cs.ut.ee/>.

## 8. Viidatud kirjandus

- [1]Peatus.ee, teenusest <https://web.peatus.ee/tietoja-palvelusta> (10.12.2022)
- [2]Google Maps <https://www.google.com/maps> (10.12.2022)
- [3]OpenStreetMap <https://www.openstreetmap.org/about> (10.12.2022)
- [4]Digitransit <https://digitransit.fi/en/> (10.12.2022)
- [5]Traficom <https://www.traficom.fi/fi> (03.06.2023)
- [6]Transpordiamet, Ühistranspordi infosüsteem  
<https://www.transpordiamet.ee/uhistranspordi-infosusteeem> (10.12.2022)
- [7]UNIBUS <https://www.unibuses.co.uk/> (10.12.2022)
- [8]Lilian S. C. Pun-Cheng, An Interactive Web-Based Public Transport Enquiry System With Real-Time Optimal Route Computation, *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 13, NO. 2, 2012* , pg 954-955.
- [9]National Geographic, Resource library, GIS (Geographic Information System)  
<https://education.nationalgeographic.org/resource/geographic-information-system-gis>  
(10.12.2022)
- [10]Choi Keechoo, Jang Wonjae, Development of a transit network from a street map database with spatial analysis and dynamic segmentation, *In Transportation Research Part C*, 2000 8(1), pg 144.
- [11]MySQL, What is MySQL <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>  
(17.03.2023)
- [12]Node.js <https://nodejs.org/en> (17.03.2023)
- [13]MDN Web Docs, JavaScript <https://developer.mozilla.org/en-US/docs/Web/JavaScript>  
(03.05.2023)
- [14]Express <https://expressjs.com/> (17.03.2023)
- [15]Axios, Getting started <https://axios-http.com/docs/intro> (04.05.2023)
- [16]Dan Arias, Hashing in Action: Understanding bcrypt, *auth0*, 2021  
<https://auth0.com/blog/hashing-in-action-understanding-bcrypt/> (04.05.2023)
- [17]MDN Web Docs, HTML <https://developer.mozilla.org/en-US/docs/Web/HTML>  
(03.05.2023)
- [18]MDN Web Docs, CSS <https://developer.mozilla.org/en-US/docs/Web/CSS> (03.05.2023)
- [19]React <https://react.dev/> (17.03.2023)
- [20]KnowledgeHut, What are the Pros and Cons of React, 2022  
<https://www.knowledgehut.com/blog/web-development/pros-and-cons-of-react> (17.03.2023)

- [21]React-hook-form <https://react-hook-form.com/> (17.03.2023)
- [22]Bootstrap <https://getbootstrap.com/docs/5.1/about/overview/> (17.03.2023)
- [23]Javatpoint, What is Bootstrap <https://www.javatpoint.com/what-is-bootstrap> (17.03.2023)
- [24]Visual Studio Code <https://code.visualstudio.com/> (05.05.2023)
- [25]Tartu Linnavalitsus, LI rattaringluse parklad avaandmed, *Tartu linna geoHub*  
<https://geohub.tartu.lv.ee/datasets/Tartu:li-rattaringluse-parklad-avaandmed/about>  
(18.03.2023)
- [26]Tartu Ülikool, Kinnisvaraosakond <https://ut.ee/et/kinnisvaraosakond> (20.04.2023)
- [27]PageSpeed Insights <https://pagespeed.web.dev/> (09.05.2023)
- [28]Google, Mobile-Friendly Test <https://search.google.com/test/mobile-friendly>  
(09.05.2023)
- [29]BroswerStack <https://www.browserstack.com/> (09.05.2023)

# LISAD

## 1. Väljavõtte koodist

```
17 });
18 //Võta Andmebaasist hooned
19 app.get("/buildings", (req, res) => {
20   db.query("SELECT * FROM buildings", (err, result) => {
21     if (err) {
22       console.log(err);
23     } else {
24       res.send(result);
25     }
26   });
27 });
28
29 //Kasutaja parooli kontroll
30 app.post("/users", (req, res) => {
31   try {
32     //Leia andmebaasist kasutaja e-maili põhjal
33     db.query("SELECT * FROM users WHERE email = ?", [req.body.loginEmail], (err, result) => {
34       if (err) {
35         req.send({err: err});
36       } else {
37         //Võrdle parool
38         bcrypt.compare(req.body.loginPassword, result[0].password).then(function(re) {
39           console.log(re)
40           if (result.length > 0 && re) {
41             res.send(result);
42           } else {
43             res.send({message: "Vale e-mail või parool"})
44           }
45         });
46       }
47     });
48   } catch(e) {
49     console.log(e)
50   }
51 });
52 //Võta andmebaasist jalgrattapeatused
53 app.get("/bicyclestops", (req, res) => {
54   db.query("SELECT * FROM bicyclestops", (err, result) => {
55     if (err) {
56       console.log(err);
57     } else {
58       res.send(result);
59     }
60   });
61 });
```

Joonis 12. Näited, andmebaasist andmete võtmisest.

```
33 useEffect(() => {
34   //Kontroll, et lehe algsel laadimisel ei hakkaks seda osa tegema
35   if (startBuilding) {
36     //Lähima rattaringluse peatuse kaugus võrdlemiseks
37     let closestDistToStart = Number.MAX_VALUE;
38     //Lähim rattaringluse peatus
39     let startStop
40
41     let closestDistToEnd = Number.MAX_VALUE;
42     let endStop
43     bicycleStops.forEach(stop => {
44       //valem kauguse arvutamiseks  $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ 
45       //Arvuta rattaringluse peatuse kaugus algus ja lõpppunktist
46       let distToStart = Math.sqrt((startBuilding[0].x_coordinate - stop.x_coordinate)**2 + (startBuilding[0].y_coordinate - stop.y_coordinate)**2);
47       let distToEnd = Math.sqrt((endBuilding[0].x_coordinate - stop.x_coordinate)**2 + (endBuilding[0].y_coordinate - stop.y_coordinate)**2);
48
49       //Võrdle peatuse kaugust ja kui on lähemal uuenda peatus
50       if (distToStart < closestDistToStart) {
51         closestDistToStart = distToStart;
52         startStop = stop
53       };
54
55       if (distToEnd < closestDistToEnd) {
56         closestDistToEnd = distToEnd;
57         endStop = stop
58       };
59     });
60     if (startStop && endStop) {
61       window.open('https://www.google.com/maps/dir/?api=1&origin=${startBuilding[0].address}&destination=${endBuilding[0].address}&travelmode=${t
62     }');
63   }
64 }, [bicycleStops, startBuilding, endBuilding, travelMethod])
65
66 const onSubmit = data => {
67   //Kui transpordimeetod on driving2 ehk jalgratas, kasutades rattaringluse punkte
68   if (data.method === "driving2") {
69     getBicycleStops();
70
71     //Leia valitud hoonete koordinaadid ja salvesta need
72     const startLoc = buildings.filter(function (elem) {
73       return elem.address === data.startLocation;
74     });
75     const endLoc = buildings.filter(function (elem) {
76       return elem.address === data.endLocation;
77     });
78   }
```

Joonis 13. Marsruudi leidmine rattaringluse punktide korral.

```

94 | // "Muuda nupu vajutus"
95 | const handleEditButtonClick = (e, building) => {
96 |   e.preventDefault();
97 |   setEditRowId(building.id)
98 |
99 |   const formData = {
100 |     name: building.name,
101 |     address: building.address,
102 |     x_coordinate: building.x_coordinate,
103 |     y_coordinate: building.y_coordinate
104 |   };
105 |
106 |   setEditBuildingFormData(formData);
107 | };
108 | // "Katkesta nupu vajutus"
109 | const handleCancelButtonClick = () => {
110 |   setEditRowId(null);
111 | };
112 | // "Kustuta" nupu vajutus
113 | const handleDeleteButtonClick = (e, buildingId) => {
114 |   const newBuildings = [...buildings]
115 |   const index = buildings.findIndex((building) => building.id === buildingId);
116 |
117 |   newBuildings.splice(index, 1);
118 |
119 |   setBuildings(newBuildings);
120 |
121 |   Axios.post("http://localhost:3001/deleteBuilding", {
122 |     id: buildingId,
123 |   })
124 |     .then(function (response) {
125 |       console.log(response);
126 |     })
127 |     .catch(function (error) {
128 |       console.log(error);
129 |     });
130 | };

```

Joonis 14. Administraatori vaates nuppudele vajutamise funktsioonid.



## 2. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Ahti Ankro,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose „Veebiliidesega infosüsteemi loomine liikumiseks Tartu ülikooli hoonete vahel”, mille juhendaja

on Vambola Leping, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, alates 09.05.2023 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.