Pärnu Rääma Põhikool

XXXXX XXXXX

8.X klass

OTSINGUMOOTORI LOOMINE

Praktiline loovtöö

Juhendaja Mart Kimmel

Pärnu 2019

SISUKORD

SISSEJUHATUS

Valisin selle teema, sest see tundus mulle ülesanne, mis pole liiga kerge ega ka liiga raske. Soovisin teada saada kuidas otsingumootorid töötavad, nagu nt Google Otsing.

Otsingumootor on programm, mis otsib mingile kriteeriumile vastavaid andmeid. Eriti populaarsed on nn veebiotsingu saidid, mis võimaldavad otsida vajalikku infot kogu veebist. [1]

Loovtöö eesmärgiks on luua oma otsingumootor, mida saab külastaja oma arvutist või nutiseadmest kasutada. See peaks olema mugav ning küsimuste ja teadmiste kättesaamine peaks olema kiire ning tõhus. Selles loovtöös piirasin veebi "eestikeelseteks".
Tegemist on praktilise loovtööga, mille käigus:

1) luuakse oma programm, mis surfab interneti, et leida uusi veebilehti ("Ämblik");
2) luuakse andmebaas, kuhu info, mis Ämblik veebilehtede kohta leiab sisestatakse;
3) luuakse veebileht, kus saab kasutaja sisestada, mida ta soovib otsida.

Kuidas Ämblik peaks töötama? Enne kui kasutaja otsib otsingumootoris, korjavad veebiämblikud internetist infot ning organiseerivad seda andmebaasi. See algab veebiaadresside nimekirjast. Veebiämblikud külastavad neid lehti, ning kasutavad seal leiduvaid linke, et leida uusi lehti. [2]

Otsingumootor asub aadressil: https://siit.eileiamidagi.ee [3]

TÖÖ KÄIK

Töö käigu võib jagada järgmisteks etappideks:

I etapp ehk “Andmebaasi ning virtuaalarvutite ülesseadistamine”;

II etapp ehk “Ämbliku programmeerimine”;

III etapp ehk “Veebilehe ülesseadistamine, disainimine”:

IV etapp ehk “Kõikide osade kokkupanek”.

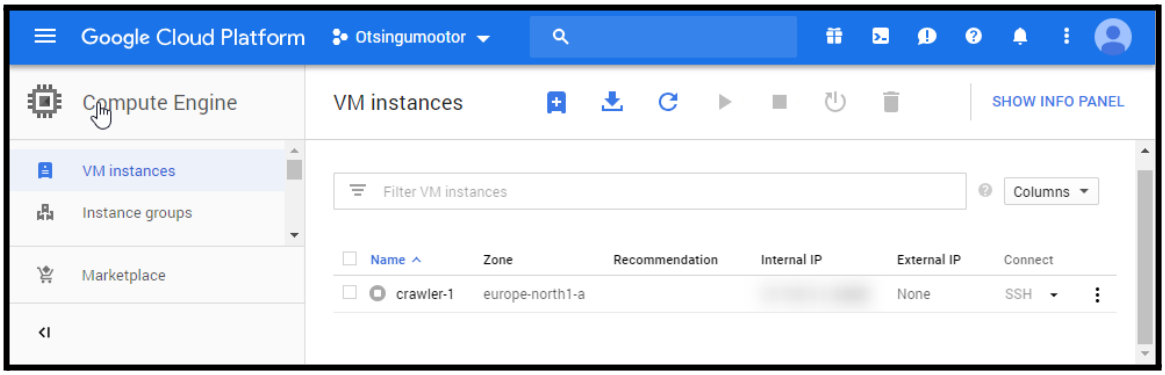
I etapp oli “Andmebaasi ning virtuaalarvutite ülesseadistamine”, kus lõin Google Cloud Platformis (“GCP”) andmebaasi ja virtuualarvuti. Virtuaalarvutisse paigaldasin enda poolt programmeeritud Ämbliku. Andmebaasi jaoks kasutasin GCP “Cloud SQL” funktsiooniga, mis loob ise andmebaasi virtuaalarvuti. Lõin selles siis andmebaasi MySQL-iga. MySQL on kõige populaarsem andmebaasi juhtimissüsteem. Seda arendab, levitab ning toetab Oracle Corporation. [4]

Algul lõin virtuaalarvuti (vt **Joonis 1**) Ämbliku jaoks, ning valisin selle asukohaks Põhja-Euroopa. Kuid hiljem andmebaasi luues polnud valikut luua teda Põhja-Euroopas, nii et pidin valima Lääne-Euroopa. Oleks meeldinud, kui oleks ta olnud ka Põhja-Euroopas, aga see midagi massiivselt ei muuda, ainult seda, et mõlemad virtuaalarvutid peavad infot jagama üle pikema distantsi (mis maksab natukene rohkem.) Andmebaasi näed **joonisel 2**.

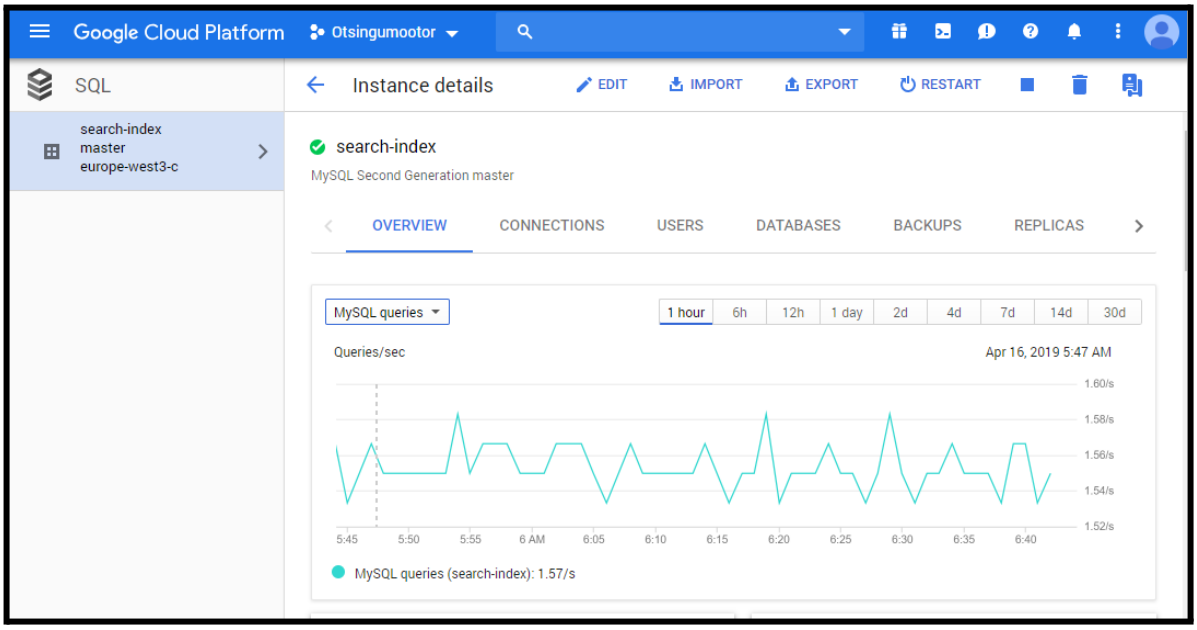
Andmebaasi seadistasin allolevalt.

1. Lõin andmebaasi “otsingumootor”.
2. Lisasin sinna tabeli “site_queue” ehk eesti keeles “veebilehtede järjekord”. Sinna tabelisse lisab Ämblik read linkide kohta, mida ta on leidnud, ning plaanib külastada.
3. Lisasin ka tabeli “visited_sites” ehk eesti keeles “külastatud veebilehed”. Sinna lisab Ämblik read linkide kohta, mida ta on külastanud ning läbi uurinud.

Ämbliku virtuaalarvuti palju seadistamist ei vajanud, ainuke asi mis selles vajas seadistamist oli ämbliku programm.



Joonis 1. Ämbliku veebiserver GCP's, autor: XXXXX XXXXX.



Joonis 2. Andmebaas GCP's, autor: XXXXX XXXXX.

II etapp oli "Ämbliku programmeerimine", kus kasutasin programmeerimiskeelt Python, et programmeerida oma enda Ämblik.

Ämbliku töökäigu seadistasin allolevalt.

1. Järgmise veebilehe valimine. Ämblik valib millist veebilehte lugeda. Ta vaatab, milliselt domeenilt ta viimati luges, ning valib ühe veebilehe järjekorrast või külastatud veebilehtede tabelist. See valik oleneb sellel, kas kell on peale 12:00 või enne. Ta valib veebilehe, mida vaadeldi selles tabelis kõige rohkem aega tagasi.
2. Enne valitud veebilehte valimist, tuleb ennem kontrollida, kas veebilehe haldur lubab minu Ämblikul külastada veebilehte. Selleks on loodud "robots.txt" fail, mida peaksid veebirobotid nagu minu Ämblik või Google'i Ämblik jälgima. Loen selle faili läbi, ning teen kindlaks kas minu Ämblikul on lubatud jätkata. Kui mitte, siis kustutatakse link andmebaasist ning minnakse järgmisega edasi.
3. Kui aga lubatakse siis hakatakse veebilehte külastama. Veebilehte lugedes paneb robot kirja uued lingid, mida andmebaasis veel kirjas pole. Kui aga on vana link, pannakse see ka kirja. Pannakse kirja, et seda on veebilehes märgatud.
4. Ämblik teeb ka vahepeal kindlaks, kas veebileht on ikka eestikeelne. Kuna ma ei soovi ingliskeelseid või muu võõrkeelseid veebilehti andmebaasi sisestada, siis olen võtnud järgmised "kui?" küsimused kasutusele. Kui üks neist on tõsi, siis võtab Ämblik tõdeda, et veebileht on eestikeelne.
    a. Kui veebileht annab teada, et ta on eestikeelne, siis ta on eestikeelne.
    b. Kui domeen lõppeb ".ee"-ga, siis ta on ka eestikeelne.
    c. Kui tekst veebilehes on Google Translate API arust eestikeelne, siis ta on eestikeelne.
5. Kui ükski Kui-küsimus ei ole tõene antud veebilehe kohta, mis olid punktis 4, siis on arvatavasti tegemist võõrkeelse veebilehega. See lisatakse blokeeritud domeenide tabelisse. Kui see on lisatud blokeeritud domeenide tabelisse, siis see tähendab seda, et kui Ämblik leiab uue lingi, mis läheb sellele domeenile, siis ta seda lihtsalt kirja ei panegi.

6. Vahepeal tehakse kindlaks ka veebilehe nimi ning kui olemas siis ka ta kirjeldus.
7. Kui kõik läheb hästi, siis lisatakse veebileht külastatud veebilehtede tabelisse, mida näed hiljem **Jooniselt 5**, ning hakatakse peatselt järgmise veebilehega pihta.

Ämblikuprogramm kasutab peale Python'isse sisseehitatud funktsioonide ka teiste inimeste/organisatsioonide funktsioone ning programme, mis on esitatud allpool.

1. PyMySQL - tänu sellele programmile saan ühenduse Ämbliku ja andmebaasiga. Sellega saab siis luua ühendus andmebaasiga ning andmebaasi lisada ridu, tabeleid, ja igasugust muud, mida saab andmebaasiga teha.
2. Google Cloud Translation API - kasutades seda programmi, saan kindlaks teha millise keelega on tekst, mis on veebilehel. See põhineb ka GCP'l, seda sama kasutab ka Google enda tõlkeprogrammis. [5]

Kokkuvõttes on ämblikuprogrammis 380 rida koodi ning 13916 tähte.

III etapp oli "Veebilehe ülesseadistamine, disainimine", kus seadistasin üles veebilehe. Kasutasin selleks endal kodus juba ettevalmistatud serverit. Koduserveri arvutiks on "Raspberry Pi" miniarvuti ning selles on veebiserveri programm "Nginx." Raspberry Pi on väike ja odav arvuti, mida kasutatakse programmeerimise õppimiseks. [6]. Nginx on avatud lähtekoodiga tarkvara, mis võimaldab 400 miljonit veebilehel eksisteerida. [7]. **Joonisel 3** näed näidet ühe otsingu kohta Otsingumootori veebilehel.

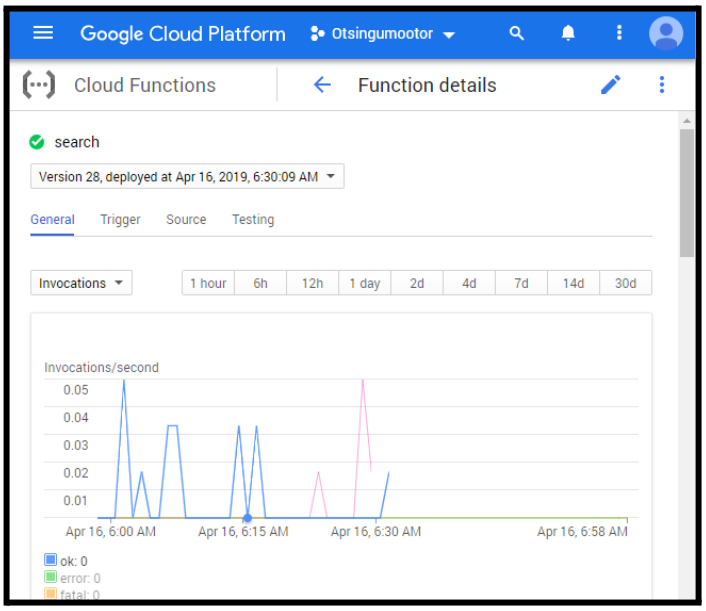
Disainimisel järgisin allolevaid kriteeriume.

1. Tahtsin kindel olla, et veebileht "mahub" kõikidesse nutiseadmetesse ära. Seda nimetatakse **Responsive Web Design**'iks. [8]
2. Jätsin veebilehe disaini üsnagi lihtsaks, sest tegemist pole miskisuguse veebilehega, mis vajab palju asju.

Joonis 3. Otsing märksõnale “Vikipeedia”, autor: XXXXX XXXXX.

IV etapp oli “Kõikide osade kokkupanek”, kus ühendan andmebaasi ning veebilehe. Et veebilehte ning andmebaasi ühendada, kasutan GCP’s kasutatavat nö teenust nimega “Google Cloud Functions.” Google Cloud Functions on koodi läbiviimis programm, mis ei vaja ühtegi serverit. Serverid on Google’i poolt. Ainuke asi, mida on vaja on programmi kood. [9]

Minu funktsioon Cloud Functions’is “search,” mida näed **Joonisel 4**, vajab ühte sõne, milleks on otsing. Siis minu programmeeritud kood loob läbi PyMySQL ühenduse andmebaasiga, ning sealt otsib otsingule vastust. Minu veebileht, kui vajutatakse punasele “Otsi!” nuppu peale, kutsub seda funktsiooni, ning annab vastu vastuse. Funktsiooni kutsumine käib läbi ühe veebilehe külastamise. Kuid minu veebileht teeb seda veebilehe külastamist tagataustal, ning toob selle vastuse siis ette.



Joonis 4. Funktsioon "search" "Google Cloud Function'is", autor: XXXXX XXXXX.

```
*************************** 82. row ***************************
       link: https://et.wikipedia.org/wiki/Ingliskeelne_Vikipeedia
     domain: et.wikipedia.org
      title: Ingliskeelne Vikipeedia – Vikipeedia
description:
 visited_at: 1555329918
   mentions: 1
82 rows in set (0.01 sec)
```

Joonis 5. Andmebaasi tabelisse "visited_sites" viimane sisestatud rida, autor: XXXXX XXXXX.

VIKIPEEDIAST ALUSTAMISE PLUSSID, MIINUSED

Kui oled mõne otsingu teinud, siis võib tunduda, et kõik vastused tulevad Vikipeediast. See on sellepärast, et kõige esimene veebileht, kus valisin, et Ämblik alustaks ning edasi läheks oli eestikeelse Vikipeedia koduleht. Kui vaadata seda lehte, siis on näha, et seal on rohkesti linke, mis peaks olema ju hea Ämblikule, mis vajab linke. Miinus on aga see, et paljud lingid on Vikipeedia-sisesed, see tähendab seda, et väljaspoole Vikipeediat satub Ämblik alguspäevil harva. Seepärast võibki tunduda, et kõik vastused tulevad Vikipeediast.

Kuid õnneks on mõndade Vikipeedia artiklite all internetiviited, mis viivad uuteni veebilehtedeni.

KOKKUVÕTE

Töö tegemise tugevad küljed olid järgmised:

1) mul oli piisavalt hea arusaam mida ma teen ja kuidas ma peaksin tegema;

2) mul oli piisavalt head oskused programmeerimises, ning oskasin oma lahendused ellu viia;

3) ma oskasin siis kui ei osanud mõnda kohta, kergesti internetist abi/infot otsida, et kiiresti edasi töötada.

Raskused töös olid järgmised:

1) soovisin luua veel funktsioone otsingumootorile; raske oli ka leida kohta, kus alustada töö tegemisega; kuna GCP's on valik funktsioonide ning kasutusalade jaoks lihtsalt nii suur, siis alguses ma ei teadnudki millist asja kasutada;

2) kuid mõndade kohtade juures ei osanud mõnda programmeerimis punkti, võib hakkas Ämblik jälle midagi muud tegema; vahest uuris Ämblik samat veebilehte mitukümmend korda, kuni sain pihta miks ta seda tegi; vahest ta lihtsalt ei teinudki midagi. "Cloud Functions'i" juures oli ühendamisega probleeme.

Kui ma saaks seda loovtööd uuesti teha, siis teeksin teisiti järgmist:

1) otsiksin ning kasutaksin teistsuguseid ning targemaid algoritme, mida kasutada otsingumootoril; vaadates tagasi, siis mulle tundub, et mõnda osa saab kergesti ära kasutada, et nt. pahatahtlikult saada kõrgematele positsioonidele otsingu tulemustes; otsing hetkel tegelt ei tundugi nii mugav, peab teadma kuidas otsida.

2) proovinud rohkem fokuseerida programmile endale: võib-olla oleks siis saanud ka rohkem funktsioone lisada otsingumootorile.

Õppisin loovtöö tegemisel:

1) sain proovida uut platvormi (GCP), mida polnud ennem kasutanud;

2) sain mõnevõrra targemaks selle kohta, mida otsingumootor tegelikult teeb ja mida aastate kaupa töö tehtud nagu firmal Google võimaldab saavutada.

3) õppisin ja leidsin uuse materjale, kust teema kohta edasi teada saada.

KASUTATUD ALLIKAD

[1] http://vallaste.ee/ Otsing: otsingumootor

[2] https://www.google.com/search/howsearchworks/crawling-indexing/

[3] https://siit.eileiamidagi.ee

[4] https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html

[5] https://cloud.google.com/translate/

[6] https://www.raspberrypi.org

[7] https://www.nginx.com

[8] https://www.w3schools.com/html/html_responsive.asp

[9] https://cloud.google.com/functions/docs/concepts/overview