

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Rene Kütt

Thonny logide analüüsi veebirakendus

Bakalaureusetöö (9 EAP)

Juhendajad: Marina Lepp, PhD
Heidi Meier, MSc

Tartu 2021

Thonny logide analüüsi veebirakendus

Lühikokkuvõte:

Programmeerimiskeele Python jaoks loodud Thonny arenduskeskkond võimaldab salvestada selles programmeerimiskeskkonnas tehtud tegevusi logifailidesse. Logifailide analüüsimine annab õpetajatele informatsiooni õpilaste ülesande lahendamisel esinenud raskustest, ülesande lahendamiseks kulunud ajast ning aitab veenduda, et õpilane on ise lahenduse kirjutanud jpm. Enne seda lõputööd oli mitmeid lahendusi Thonny logifailide analüüsimiseks, mis pakkusid erinevaid funktsionaalsuseid, kuid ükski neist ei võimaldanud terviklikku lahendust õpilaste logifailide kiireks ja efektiivseks analüüsiks. Selle lõputöö eesmärk oli luua veebirakendus, mis võimaldaks logifailide terviklikku analüüsi, kus õpetajad saaksid mugavalt õpilaste logifaile analüüsida.

Võtmesõnad:

Thonny, logifailid, analüüs, veebirakendus

CERCS: P175 Informaatika, süsteemiteooria, S281 Arvuti õpiprogrammide kasutamise meetodika ja pedagoogika

Web application for analysing Thonny log files

Abstract:

Programming language Python IDE Thonny provides the ability to save the actions performed during the development process into a log file. The analysis of aforementioned log files provides insights for the teacher like complications faced during development, how long it took and whether the student solved the task independently etc. Prior to this thesis, there was not a comprehensive application to analyse the log files quickly and efficiently. The aim of this thesis is to develop a web application for teachers to analyse Thonny log files more conveniently.

Keywords:

Thonny, log file, analysation, web application

CERCS: P175 Informatics, systems theory, S281 Computer-assisted education

Sisukord

Sissejuhatus	4
1. Programmeerimiskeskond Thonny ja logifailid	6
1.1 Thonny tutvustus	6
1.2 Thonny logifailid	6
2. Olemasolevad logifailide analüüsimise lahendused.....	8
2.1 Thonny taasesitaja	8
2.2 Age Roosi Thonny logifailide analüsaator	9
3. Nõuded veebirakendusele	11
3.1 Funktsionaalsed nõuded	11
3.2 Mittefunktsionaalsed nõuded.....	12
4. Thonny logide analüüsi veebirakendus	13
4.1 Veebirakenduses kasutatud tehnoloogiad.....	13
4.2 Veebirakenduse funktsionaalsused ning kasutamine	14
4.2.1 Logifailide sisestamine.....	14
4.2.2 Analüüsitud failide nimekirja grupeerimine	15
4.2.3 Analüüsi tulemused	16
4.2.4 Taasesitamine	17
4.2.5 Lisatud teksti pikkuse diagramm	18
4.3 Tagasiside veebirakendusele	18
4.4 Võimalikud edasiarendused.....	20
Kokkuvõte	21
Viidatud kirjandus	22
Lisad	24
I. Litsents	24

Sissejuhatus

Tartu Ülikoolis õpetatakse programmeerimise algkursuseid enamasti programmeerimiskeeles Python ning pakutakse ka mitmeid MOOCe (ingl *Massive Online Open Courses*), kus õpetatakse Pythonis programmeerimist algtasemel. Nendest kursustest võtab igal aastal osa tuhandeid inimesi [1]. Tartu Ülikoolis kasutatakse Pythoni programmeerimise algkursustel Aivar Annamaa väljatöötatud Thonny programmeerimiskeskonda. Põhjuseid selleks on mitmeid, aga üks neist on kindlasti see, et Thonny võimaldab luua logifaile, kuhu salvestatakse kasutaja tegevused programmeerimiskeskonnas. Logifailid on väga väärtuslikud õppejõududele, sest need võimaldavad analüüsida õpilaste õppeprotsessi. Selle tulemusena on võimalik iga õpilase kohta teada saada, kus asusid tema jaoks rasked kohad, kuidas õpilane ülesannet lahendas, mitu korda ta programmi käivitas, kas ta lahendas ülesandeid iseseisvalt jne. Kursusepõhiselt on võimalik logifailide analüüsi tulemusena korrigeerida aine ülesehitust või ülesandeid, et paremini toetada õpilaste õppimist.

Hetkel on võimalik logifaile analüüsida peamiselt kahel viisil. Esiteks on võimalik kasutada Thonnyse sisse ehitatud kasutussessiooni taasesitamise (ingl *replayer*) funktsionaalsust. Selle abil saab õpilase tehtud tööd programmeerimiskeskonnas samm-sammult läbi käia [2]. Selle lahenduse miinuseks on see, et ükshaaval logifailide läbimine on äärmiselt aeganõudev ja tülikas. Teiseks saab kasutada Age Roosi loodud Thonny logifailide analüsaatorit. Selle abil on võimalik logifailist kätte saada andmeid nagu käivitamiste arv, vigade arv ja tüübid, lahendamisele kulunud aeg jne [3]. Selle lahenduse miinuseks on see, et seda peab sageli kasutama koostöös taasesitamise funktsionaalsusega, et saada kätte terviklik pilt. Lisaks peab logifaile analüüsima ükshaaval ning igakord logifaili valimine on tülikas.

Selle bakalaureusetöö eesmärk on luua veebirakendus, millega on võimalik analüüsida logifaile. Veebirakendus võimaldab analüüsida nii üksikuid logifaile kui ka mitut, mis võivad asuda erinevates kaustades, mis võivad olla omakorda rekursiivselt tihendatud ZIP-failidesse. Veebirakendusel on samuti funktsionaalsus, mis laseb analüüsida kas üksiku õpilase logifaile või mitme õpilase logifaile korraga. Logifaili analüüsi tulemusena tagastab veebirakendus alustamis- ning lõpetamisaja; käivitamiste arvud; loodud, avatud ning käivitatud failide nimed; tekstilõigud, mis kopeeriti Thonnyse; ja vead, mis esinesid jne. Samuti võimaldab veebirakendus taasesitada logifaili salvestatud tegevusi, et kasutaja näeks programmi loomise käiku. Selle tarkvaralahenduse peamisteks kasutajateks on õpetajad ning õppejõud, kes võivad kasutada seda näiteks veendumaks, et õpilane kirjutas töö iseseisvalt,

milliste programmifailidega õpilane töötas logifaili jooksul, või välja selgitamaks, mis olid õpilaste jaoks kõige raskemad ülesanded ja kui palju õpilastel aega kulus jms.

Bakalaureusetöö esimeses osas tutvustatakse Thonnyt ja selle logifaile. Teises osas tutvustatakse olemasolevaid lahendusi. Kolmandas osas tuuakse välja rakendusele määratud nõuded enne selle arendamisega alustamist. Neljandas osas kirjeldatakse ning analüüsitakse valminud veebirakendust.

1. Programmeerimiskeskond Thonny ja logifailid

Selles peatükis käsitletakse programmeerimiskeskonda Thonny ning kirjeldatakse, kuidas on Thonny logifailid struktureeritud.

1.1 Thonny tutvustus

Thonny on Pythoni IDE (integreeritud programmeerimiskeskond), mis on suunatud alustavatele programmeerijatele [4]. Thonny eelised enamike teiste Pythoni arenduskeskkondade ees on näiteks selle kasutuslihtsus, lihtne silumine (ingl *debugging*), süntaksivigade esiletõstmise jpm. Kasutajasõbralikkust silmas pidades on Thonnyle Python sisse ehitatud, et vähendada ülesseadmise vaeva ning programmeerimiskeskonna liides on tehtud võimalikult lihtne, et alustavat programmeerijat mitte üle koormata. Thonnys on võimalik kasutada sisseehitatud silumise funktsionaalsust, mida on väga lihtne kasutada ka alustaval programmeerijal. Kasutajal on võimalik silurit kasutades liikuda läbi oma koodi samm-sammult ning näha, kuidas Python interpreteerib igat rida ja blokki [5]. Thonny üheks kasutajasõbralikuks funktsionaalsuseks on see, et käsurida ning koodiredaktor asuvad samas vaates. Integreeritud käsurida võimaldab õpilastel programmi käivitamisel lihtsasti näha selle väljundit ja veateateid käsureal.

1.2 Thonny logifailid

Thonnys on logifailide loomise funktsionaalsus, kus kirjutatakse programmeerimiskeskonnas toimunud sündmused logifailidesse JSON-vormingus (Joonis 1). Thonny logifaili struktuuris salvestub iga programmeerimiskeskonnas tehtud tegevus eraldi JSON-objektina, mis asuvad kõik peamises massiivis. Logifaili analüüsimiseks on tähtsamad kirjed sündmuste objektis *time*, *sequence* ja *text*. *Time*-kirje salvestab ajahetke, millal tegevus aset leidis. *Sequence* kirjeldab, mis sündmusega tegu on, näiteks *TextInsert* markeerib seda, et kasutaja sisestas teksti arenduskeskkonda. *Text*-kirje salvestab olenevalt sündmusest endasse infot, näiteks teksti sisestuse puhul sisestatud teksti ja veateate puhul veateate sõnumit jne.

```
C:\Users\Rene K\u00fctt\Documents\2020-09-30_21-03-58_0.txt - Notepad++
Fail Redigeeri Otsi Vaade Kodeering Keel S\u00e4tted T\u00f6\u00f6n\u00e4stid Makro K\u00e4ivitamine Pluginad Alken ?
2020-09-30_21-03-58_0.txt
159     },
160     {
161         "index1": "2.4",
162         "index2": "3.0",
163         "text_widget_id": 90525488,
164         "text_widget_class": "ShellText",
165         "trivial_for_coloring": true,
166         "trivial_for_parens": true,
167         "sequence": "TextDelete",
168         "text_widget_context": "shell",
169         "time": "2020-09-30T21:05:24.285104"
170     },
171     {
172         "index": "2.4",
173         "text": "%cd 'C:\\Users\\Rene K\u00fctt\\Documents\\didaktika\\thonny-logfile-analysation\\n%Debug katse.py\\n'",
174         "tags": "('magic', 'oplevel', 'command')",
175         "text_widget_id": 90525488,
176         "text_widget_class": "ShellText",
177         "trivial_for_coloring": false,
178         "trivial_for_parens": false,
179         "sequence": "TextInsert",
180         "text_widget_context": "shell",
181         "time": "2020-09-30T21:05:24.285104"
182     },
183     {
184         "index1": "4.0",
185         "index2": "4.0",
186         "text_widget_id": 90525488,
187         "text_widget_class": "ShellText",
188         "trivial_for_coloring": true,
189         "trivial_for_parens": true,
190         "sequence": "TextDelete",
191         "text_widget_context": "shell",
192         "time": "2020-09-30T21:05:24.285104"
193     },
194 }
```

Joonis 1. Thonny logifaili väljavõte.

Logifailidest on võimalik saada infot programmi(de) kirjutamise protsessi kohta. See info võib olla näiteks programmi kirjutamiseks kulunud aeg, programmi käivitamiste arv ning esinenud vigade arv. Veelgi enam, logifailide alusel on võimalik taasesitada programmi kirjutamine graafiliselt nii, et selle kasutaja näeb programmi valmimise käiku.

Thonny logifailidest on võimalik pikemalt lugeda Heidi Meieri magistritööst „Õppijate käitumismustrid programmeerimisülesande lahendamisel: logifailide analüüs“ [6].

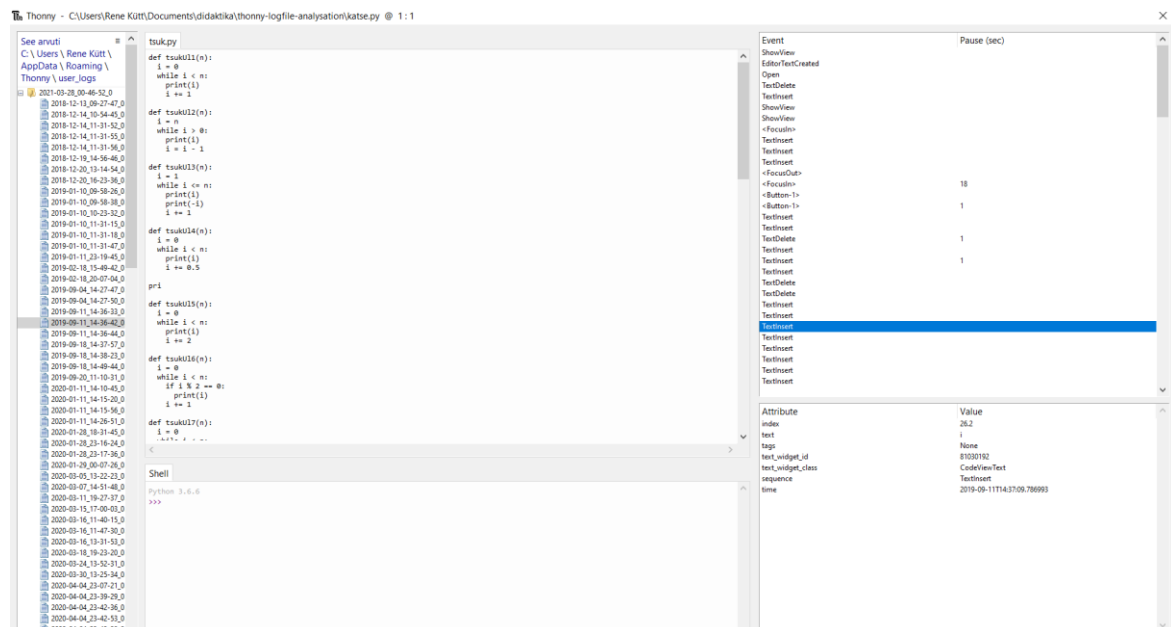
2. Olemasolevad logifailide analüüsimise lahendused

Järgnevat alapeatükki kirjeldatakse peamiselt kasutatavaid Thonny logifailide analüüsimise lahendusi, mis on valminud enne seda lõputööd.

2.1 Thonny taasesitaja

Thonny taasesitaja (ingl *replayer*) kasutamiseks tuleb Thonnys esmalt sisse lülitada *expert* kasutajaliidese režiim (*Tööriistad => Seaded... => Kasutajaliidese režiim*). Taasesitajat saab avada Thonny avaekraanilt (*Tööriistad => Ava Taasesitaja...*).

Taasesitaja programmiaken jaguneb viieks osaks: failipuu (vasakul), koodiredaktor (keskel üleval), käsurida (keskel all), sündmuste aken (paremal üleval) ja sündmuste atribuutide aken (paremal all) (Joonis 2).

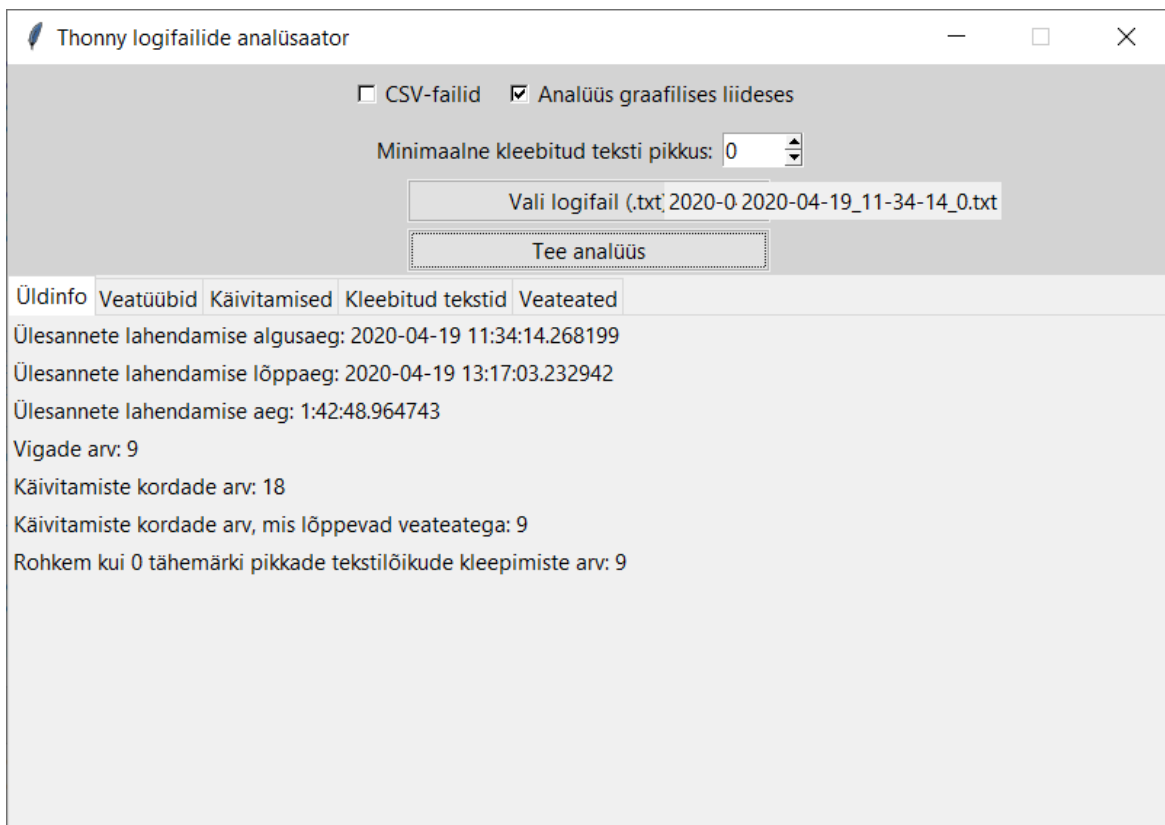


Joonis 2. Thonny taasesitaja.

Taasesitajas faili analüüsimiseks tuleb valida vasakul asuvast failipuust soovitud logifail ning teha sellele topeltklõps. Faili läbimängimiseks tuleb klõpsata sündmuste aknas suvalise sündmuse rea peale ja seejärel on võimalik klaviatuuri nooltega üles-alla liikuda. Sündmuste read on järjestatud sündmuste toimumisaja kasvamise järgi ülevalt alla. Faili läbi mängides on iga sündmuse rea peal olles vasakul all aknas nähtavad vastava sündmuse atribuudid. Vahetades sündmuse ridu, uueneb igakord koodiredaktori ning käsurea aken vastavaks sellele, mis oli kasutajal ekraanil selle sündmuse hetkel.

2.2 Age Roosi Thonny logifailide analüsaator

Age Roosi loodud Thonny logifailide analüsaator (edaspidi selles alapeatükis logifailide analüsaator) on loodud Tartu Ülikooli bakalaureusetöö raames aastal 2019. Programm on kirjutatud Pythonis ning selle kasutamiseks tuleb alla laadida lähtekood GitHubi repositooriumist. Programmi käivitamiseks tuleb käivitada Pythoni fail *eesliides*. Programmi käivitamisel avaneb kasutajale programmiaken. Programmiakna päises asuvate märkeruutudega on võimalik määrata, mida tehakse programmi väljundiga. Enamasti kasutatakse programmi väljundina *Analüüs graafilises liideses*, sest selle abil on kõige mugavam logifaile kiiresti analüüsida. Logifaili valimiseks on vaja klõpsata *Vali logifail (.txt)* nupule. Seejärel avaneb aken *File Explorer*, kus saab valida sobiva logifaili. See lahendus võimaldab analüüsida ühte logifaili korraga. Peale logifaili valimist tuleb klõpsata nupule *Tee analüüs*. Kui on programmi väljundiks valitud *Analüüs graafilises liideses*, siis täidetakse programmiakna allosas asuvad vaheaknad analüüsi tulemustega (Joonis 3).



Joonis 3. Thonny logifailide analüsaator.

Vahelehel *Üldinfo* näidatakse logifaili algus- ning lõppaega, lahendamise aega, vigade arvu, käivitamiste arvu, veateatega lõppenud käivitamiste arvu ning nende tekstilõikude kleepimiste arvu, mis ületasid kasutaja poolt määratud teksti pikkust (vaikimisi on 0). Vahelehel

Veatüübid on sektordiagramm, mis näitab, kuidas jagunesid veateadete tüübid. Vahelehel *Käivitamised* on ajateljel märgitud käivitamised ning on välja toodud, kas need lõppesid veaga või mitte. Vahelehel *Kleebitud tekstid* on tabelina välja toodud kõik kasutaja poolt koodiredaktorisse kleebitud tekstid koos toimumisajaga. Vahelehel *Veateated* on tabelina välja toodud vea toimumishetk, veatüüp ja veateate tekst.

3. Nõuded veebirakendusele

Funktsionaalsed nõuded veebirakenduse jaoks lepiti kokku juhendajatega enne veebirakenduse arenduse alustamist. Mittefunktsionaalsed nõuded on autori väljamõeldud, et veebirakendus oleks mugavalt kasutatav.

3.1 Funktsionaalsed nõuded

Funktsionaalsed nõuded kirjeldavad süsteemi funktsionaalsusi, kuidas andmeid salvestatakse, käideldakse, muundatakse ja edastatakse [7]. Veebirakendusele määratud funktsionaalsed nõuded on:

1. Veebirakendus peab suutma analüüsida üksikut logifaili.
2. Veebirakendus peab suutma analüüsida mitut logifaili korraga.
3. Veebirakendus peab suutma võtta sisendiks logifaile TXT-failivormingus.
4. Veebirakendus peab suutma võtta logifailide sisendiks ZIP-faili.
5. Veebirakendus peab suutma lugeda ZIP-faili sisu.
6. Veebirakendus peab suutma lugeda üksteise sees asuvaid ZIP-faile.
7. Veebirakendus peab suutma võtta logifailide sisendiks ZIP-faile ja/või logifaile sisaldavat kausta.
8. Veebirakendus peab suutma analüüsida kõiki sisendiks antud Thonny logifaile.
9. Veebirakendus peab suutma kuvada iga logifaili analüüsi tulemusi eraldi.
10. Veebirakendus peab analüüsi tulemusena näitama iga logifaili kohta väljasid:
 - 10.1. algusaeg
 - 10.2. lõppaeg
 - 10.3. kestus
 - 10.4. käivitamiste arv
 - 10.5. vigade arv
 - 10.6. kleepimiste arv
 - 10.7. silumiste arv
 - 10.8. loodud failide nimed
 - 10.9. käivitatud failide nimed
 - 10.10. avatud failide nimed
 - 10.11. kleebitud tekstid
 - 10.12. veateated
11. Veebirakendus peab suutma iga analüüsitud logifaili taasesitada.

3.2 Mittefunktsionaalsed nõuded

Mittefunktsionaalsed nõuded määratlevad, kui edukalt süsteem täidab enda oodatud funktsionaalsuseid [7]. Veebirakendusele määratud mittefunktsionaalsed nõuded on:

1. Veebirakendus peab olema kasutajasõbralik.
2. Veebirakendus peab suutma avakuva ära laadida 1 sekundiga.
3. Veebirakendus peab suutma 100 MB logifailide kogumiku analüüsida vähem kui 5 sekundiga.

4. Thonny logide analüüsi veebirakendus

Valminud Thonny logide analüüsi veebirakendus (edaspidi veebirakendus) asub veebiaadressil <https://progtugi.cs.ut.ee/thonny-log-analyser/>. Veebirakenduses kasutatavaks keeleks valiti juhendajatega inglise keel, sest see võimaldab veebirakenduse kasutamist laiemalt. Valminud veebirakendus on täielikult kliendipoolne, mis tähendab, et logifailide analüüsimine toimub kliendipool ja logifailidega seonduvat informatsiooni ei saadeta serveripoolle. Selles peatükis kirjeldatakse veebirakenduse arendamiseks kasutatud tehnoloogiaid. Samuti antakse ülevaade valminud veebirakenduse funktsionaalsustest ning tutvustatakse, kuidas seda kasutada. Lisaks analüüsitakse veebirakendusele saadud tagasisidet ning pakutakse välja võimalikud edasiarendused veebirakenduse jaoks. Valminud veebirakendus täidab kõiki sellele seatud nõudeid (ptk 3 Nõuded veebirakendusele).

4.1 Veebirakenduses kasutatud tehnoloogiaid

Veebirakenduse arendusel kasutati märgistuskeeltena HTMLi ja CSSi, ning programmeerimiskeelena JavaScripti. Veebirakenduse loomisel kasutati eelmainitud veebitehnoloogiaid, sest need on sobivad tööriistad vajalike funktsionaalsuste loomiseks ning veebirakenduse looja oli kõige kogenum nende veebitehnoloogiatega.

Veebirakenduse loomisel DOMi (ingl *Document Object Model*) lihtsamaks muutmiseks kasutati teeki jQuery ja raamistikku Bootstrap. jQuery on JavaScripti teek, mis võimaldab lihtsamalt DOMi läbida, seda muuta, sündmuseid käsitleda ning animeerida [8]. Bootstrap on HTMLi, CSSi ja JavaScripti kasutatav *front-end* raamistik, mis lihtsustab dünaamiliste veebilehtede loomist [9]. Veebirakendus kasutab jQuery minimeeritud versiooni 3.6.0 ja Bootstrap versiooni 4.5.2.

Valminud veebirakendus kasutab ZIP-failide lahtipakkimiseks teeki JSZip. See on JavaScripti teek, mis võimaldab ZIP-failide loomist, muutmist ja lugemist [10]. Veebirakendus kasutab JSZip versiooni 3.5.0.

Veebirakenduses kasutatakse logifaili taasesitajas programmikoodi esile tõstmiseks teeki highlight.js. See on JavaScripti kasutatav teek, mis võimaldab veebilehel näidatava programmikoodi esiletõstmist vastavalt sellele, mis programmeerimiskeelega tegu on [11].

See muudab veebilehel näidatava programmikoodi kergemini loetavaks. Veebirakendus kasutab highlight.js versiooni 10.6.0.

Veebirakenduses kasutatakse diagrammide loomiseks teeki Chart.js ja selle date-fns adapterit. Chart.js on JavaScripti teek, mis võimaldab veebilehel kuvada erinevat tüüpi diagramme [12]. Chart.js date-fns adapter võimaldab date-fns kasutamist [13]. date-fns on JavaScripti teek kuupäevadega töötamiseks [14]. Diagrammides ajaskaala kasutamiseks on vajalik teek kuupäevadega töötamiseks ja vastav adapter [15]. Veebirakendus kasutab Chart.js versiooni 3.0.2 ja date-fns adapter versiooni 2.0.0.

Veebirakendus kasutab märkeruudust lüliti tegemiseks Bootstrap 4 laienduse komponenti Custom switch. Custom switch võimaldab muuta HTMLi märkeruudud dünaamilisteks lülititeks [16]. Veebirakendus kasutab Custom switchi versiooni 1.1.2.

Veebirakendus kasutab ikoonide näitamiseks Font Awesome'it. See on fontide ning ikoonide riistakomplekt (ingl *toolkit*) [17]. Veebirakendus kasutab Font Awesome'i versiooni 4.7.0.

Veebirakenduses kasutati teekide ning raamistike haldamiseks CDNi ehk sisuedastusvõrku. Versioonihalduseks kasutati Bitbucketit¹.

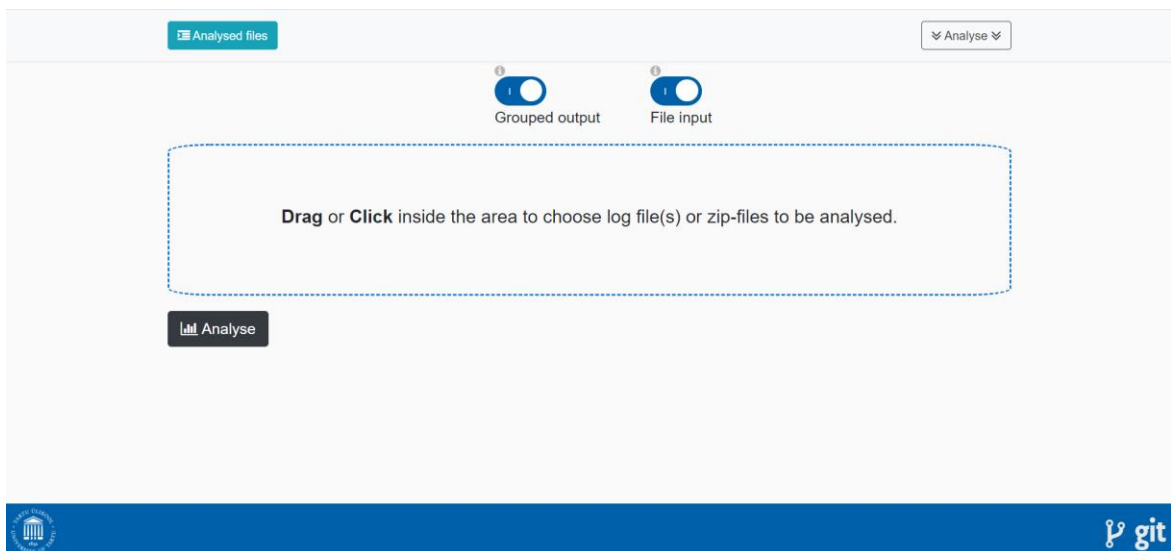
4.2 Veebirakenduse funktsionaalsused ning kasutamine

Järgnevates alapeatükkides kirjeldatakse valminud veebirakenduse peamised funktsionaalsused ning kuidas neid kasutada.

4.2.1 Logifailide sisestamine

Veebirakenduse avamisel (Joonis 4) on mitu võimalust logifailide sisestamiseks. Logifailide sisestuseks sobiliku tüübi valimiseks tuleb kasutada parempoolset lüliti (Joonis 4). Lüliti algolekus, kui lüliti all on kirjas *File input*, siis aktsepteerib lülite all olev märgistatud ala tekstifaile või üksikuid ZIP-faile. Lüliti peal klõpsates muudab lüliti olekut, misjärel aktsepteerib märgistatud ala üksikuid kaustasid, kus sees võib asuda tekstifaile ja/või ZIP-faile.

¹ Repositooriumi URL on <https://bitbucket.org/renekutt/thonny-log-analyser>.

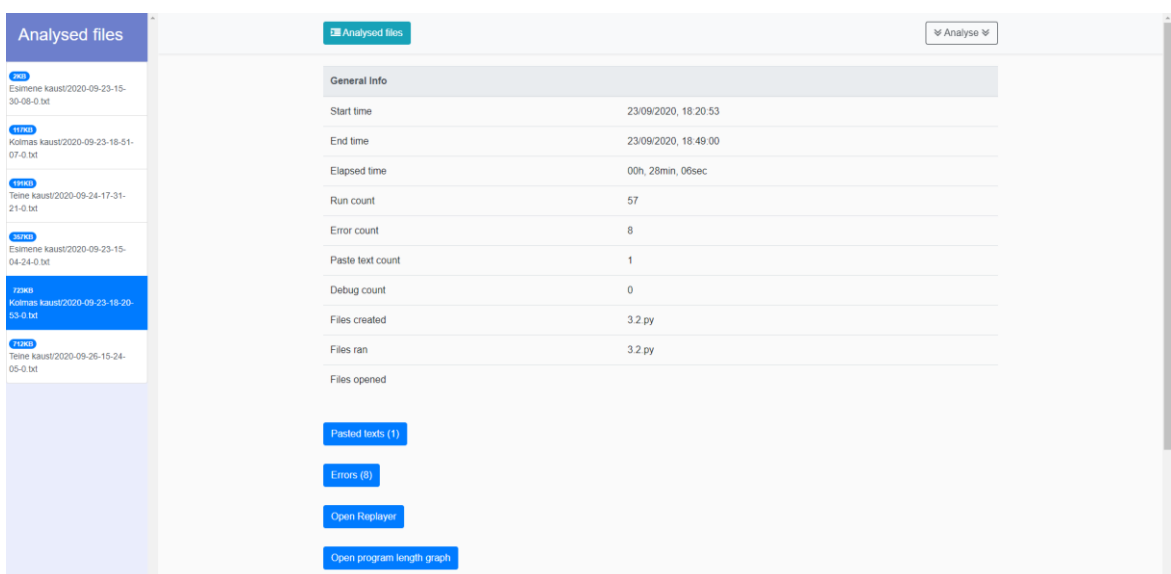


Joonis 4. Thonny logide analüüsi veebirakendus.

Logifaile on võimalik sisestada kahel viisil. Esiteks saab märgistatud alasse lohistada sobiliku faili. Teiseks saab klõpsata märgistatud alasse, misjärel avaneb aken *File Explorer*, kus saab valida sobivat tüüpi sisendi.

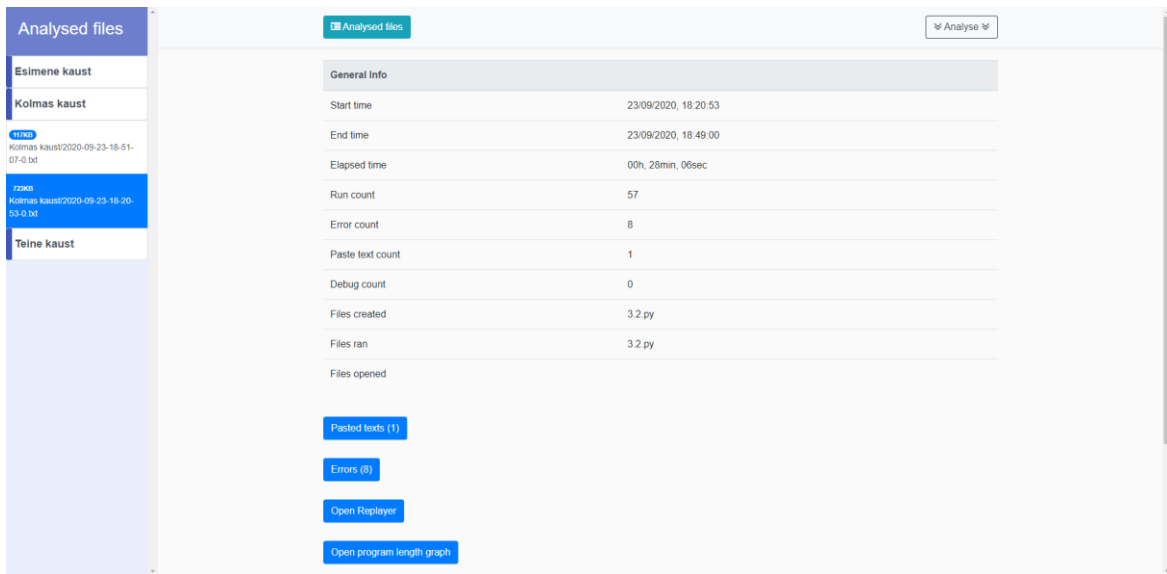
4.2.2 Analüüsitud failide nimekirja grupeerimine

Valimaks, kas analüüsitud logifaile grupeerida sisendi juurkaustas asuvate kaustade järgi, tuleb kasutada vasakpoolset lüliti (Joonis 4). Lüliti algolekus on analüüsitud failide nimekirja grupeerimine sisse lülitatud ning sellel klõpsates lülitatakse see välja. Kui analüüsitud failide nimekirja grupeerimine on välja lülitatud, siis kuvatakse analüüsitud failide nimekiri, kus kõik failid on üksteise järel ekraani vasakul asuvas paneelis (Joonis 5).



Joonis 5. Thonny logide analüüsi veebirakenduse analüüs grupeerimata faili nimekirjaga.

Kui analüüsitud failide nimekirja grupeerimine on sisse lülitatud, siis kuvatakse analüüsitud failide nimekiri, kus failid on grupeeritud juurkaustas asuvate kaustade järgi (Joonis 6).



Joonis 6. Thonny logide analüüsi veebirakenduse analüüs grupeeritud failide nimekirjaga.

Grupeeritud failide nimekirja puhul saab valida soovitud faili, avades selle faili vastava grupeeritud kausta, kas selle peale klõpsates või *Shift* klahviga liikudes.

Grupeerimise funktsionaalsuse lisamise põhjuseks on see, et faili nimekirja grupeerimisel on failide nimekiri kompaktsemalt ja struktureeritumalt paigutatud. Näiteks õpikeskkonnast Moodle saab õpilaste poolt esitatud logifailide allalaadimisel ZIP-faili mugavalt veebirakenduses analüüsida nii, et grupeeritud kaustade nimed oleksid vastavate õpilaste nimed.

4.2.3 Analüüsi tulemused

Selleks, et käivitada peale logifailide sisestamist analüüs, tuleb klõpsata logifailide sisestusala all olevat nuppu *Analyse* (Joonis 4). Peale seda avaneb automaatselt analüüsitud failide nimekiri ekraani vasakul asuvas paneelis, mida saab manuaalselt avada ning sulgeda nupuga *Analysed files*. Peale logifailide analüüsimist lähevad peitu logifailide sisestusala ning analüüsimise tüübi lülid, mida on võimalik manuaalselt avada ning sulgeda nupuga *Analyse* ekraani üleval paremal osas.

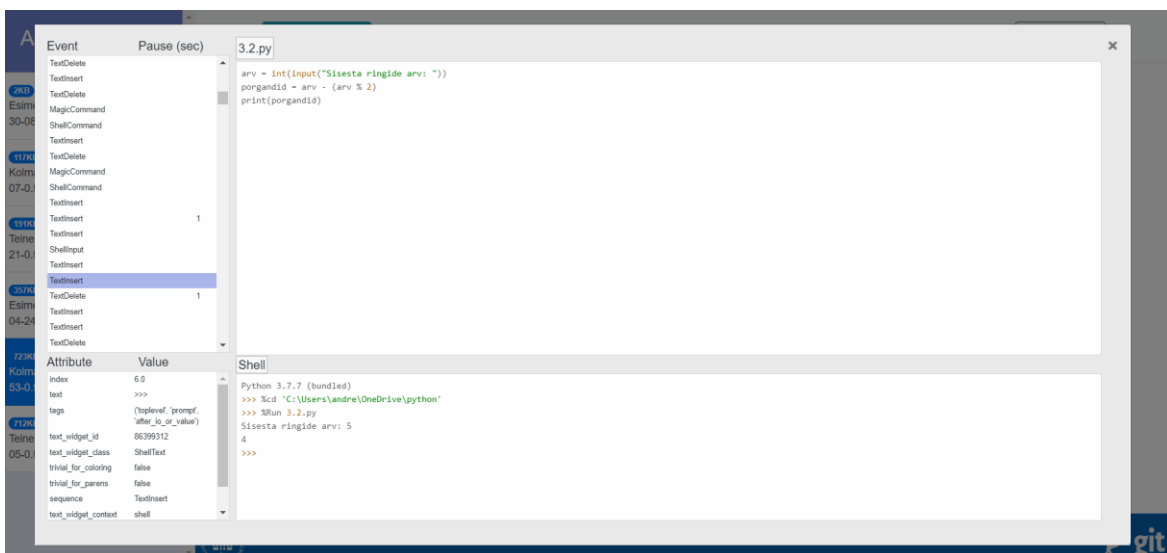
Peale logifailide analüüsimist ja vasakult logifailide nimekirjast sobiva faili valimist uueneb veebirakenduse analüüsi tulemuste aken ekraani keskosas (Joonis 5 ja Joonis 6). Analüüsi tulemuste aknas *General info* tabelis on järgmised andmed:

1. algusaeg
2. lõppaeg
3. kestus
4. käivitamiste arv
5. vigade arv
6. kleepimiste arv
7. silumiste arv
8. loodud failide nimed
9. käivitatud failide nimed
10. avatud failide nimed

Tabeli *General info* all asuvale nupule *Pasted texts* klõpsates avaneb selle all tabel, kus on näha kõik vastavas logifailis kleebitud tekstid ja toimumisajad. Kleebitud tekstide nupu all asub nupp *Errors*, millel klõpsates avaneb selle all tabel, kus on näha kõik logifailis esinenud veateated ja nende toimumisajad.

4.2.4 Taasesitamine

Veebirakenduse taasesitamise akna avamiseks peab analüüsi tulemuste akna allosas asuva nupu *Open Replayer* peal klõpsama. Taasesitaja aken jaguneb neljaks osaks: sündmuste aken (vasakul üleval), sündmuse atribuutide aken (vasakul all), koodiredaktor (keskel üleval) ja käsurida (keskel all) (Joonis 7).

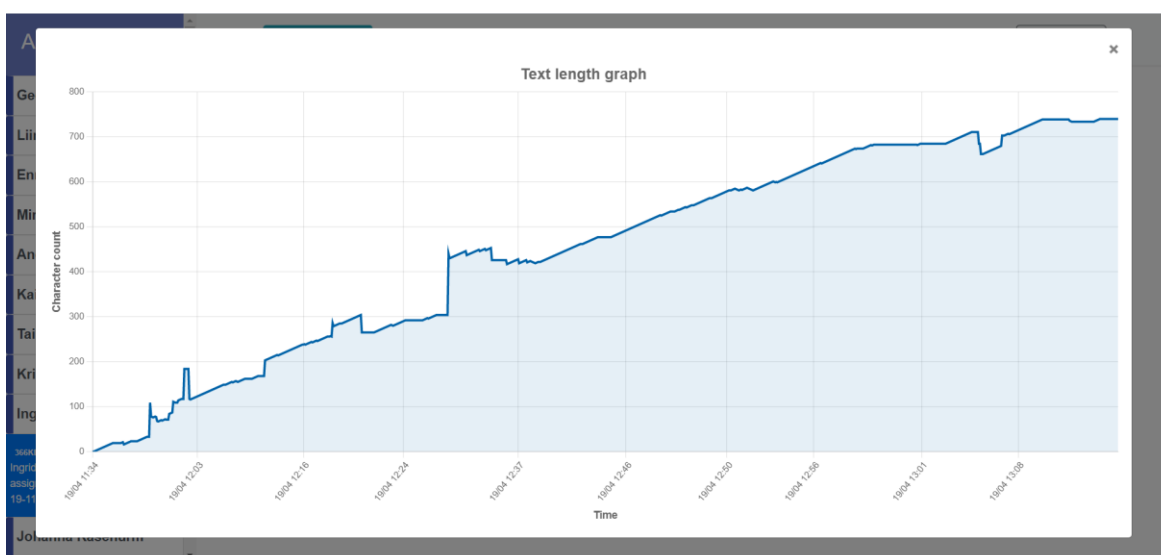


Joonis 7. Thonny logide analüüsi veebirakenduse taasesitaja.

Veebirakenduse taasesitaja läbimängimine käib samamoodi nagu Thonny taasesitaja läbimängimine peale logifaili valimist (ptk 2.1 Thonny tutvustus).

4.2.5 Lisatud teksti pikkuse diagramm

Lisatud teksti pikkuse diagrammi avamiseks peab analüüsi tulemuste akna allosas asuva nupu *Open program length graph* peal klõpsama. Lisatud teksti pikkuse diagramm näitab logifaili jooksul programmiaknas olevate programmide teksti pikkuste summa muutust ajas (Joonis 8).



Joonis 8. Thonny logide analüüsi veebirakenduse lisatud teksti pikkuse diagramm.

Selles diagrammis näitab y-telg programmide teksti tähemärkide summat ning x-telg aega. Liikudes kursoriga joondiagrammi kohale, avaneb vastavalt kursori asukohast joone peal tekstikast (ingl *tooltip*), kus on näha antud punkti täpset tähemärkide summat ning aega sekundi täpsusega. Lisatud teksti pikkuse diagrammi funktsionaalsus vajab veel edasiarendamist, et oleks võimalik vaadata logifaili jooksul töötatud programmifailide teksti pikkuste muutust eraldi diagrammidena.

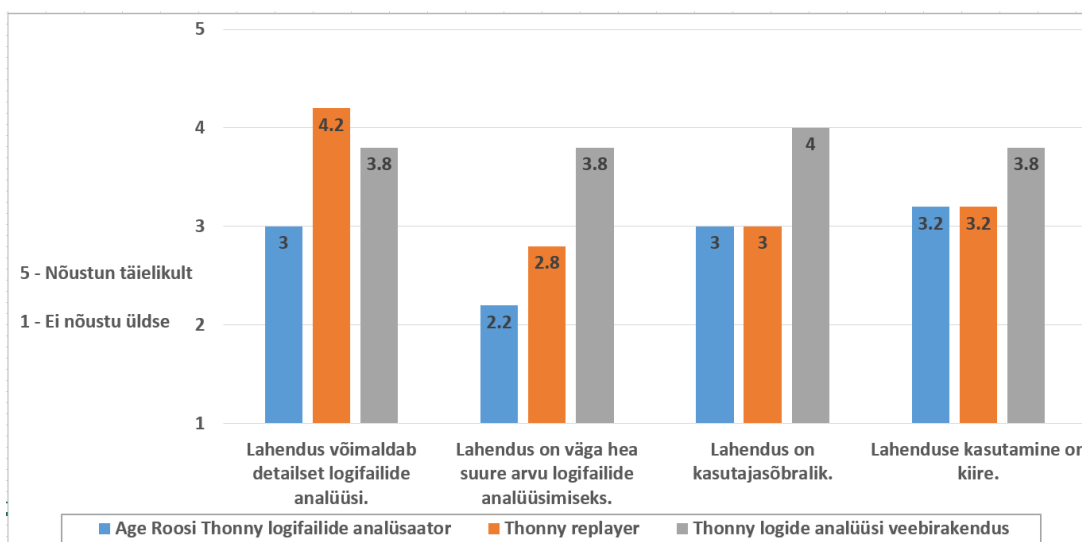
4.3 Tagasiside veebirakendusele

Veebirakenduse arendamise jooksul, kui valmis mõni kokkulepitud funktsionaalsus, siis juhendajad valideerisid ning tagasisidestasid tehtud täiendused. Kui valmis veebirakenduse versioon 0.4.0, siis kasutasid seda kursuste „Programmeerimise alused“ ja „Programmeerimise alused II“ viis õppejõudu, kes vastasid peale sellega töötamist küsimustikule. Versioon

0.4.0 sisaldas kõiki selles töös mainitud Thonny logide analüüsi veebirakenduse funktsionaalsuseid, välja arvatud lisatud teksti pikkuse diagrammi funktsionaalsust. Küsimustikus tuli õppejõududel hinnata väiteid Thonny taasesitaja, Age Roosi Thonny logifailide analüsaatori ja valminud veebirakenduse osas Likerti skaalal 1st 5ni, mis tähistavad vastavalt „Ei nõustu üldse“ ja „Nõustun täielikult“. Nendeks väideteks olid:

1. Lahendus võimaldab detailset logifailide analüüsi.
2. Lahendus on väga hea suure arvu logifailide analüüsimiseks.
3. Lahendus on kasutajasõbralik.
4. Lahenduse kasutamine on kiire.

Nende väidete keskmiste vastuste põhjal koostati tulpdiaagramm (Joonis 9).



Joonis 9. Thonny logide analüüsi lahenduste väidete keskmised vastused.

Küsimustiku väidete vastuste põhjal võib öelda, et valminud veebirakendus parendas olemasolevate lahenduste järgnevaid osasid: suure arvu logifailide analüüsi, kasutajasõbralikkust ja kasutuskiirust. Detailse logifailide analüüsi puhul parendas valminud veebirakendus Age Roosi lahendust, kuid ei parandanud Thonny taasesitaja võimekust.

Õppejõud vastasid väitele, „Kasutan Thonny log analyser veebirakendust ka tulevikus.“ 5-pallisel Likerti skaalal. Selle vastuse keskmiseks tulemuseks oli 4,6 ehk õppejõud hindasid, et kasutavad veebirakendust ka tulevikus väga suure tõenäosusega.

Õppejõud vastasid küsimustikus vabas vormis küsimusele, kuidas lahendust paremaks muuta. Selle küsimuse vastustest saadi palju ideid, kuidas lahendust parendada ning need

on välja toodud järgnevas alapeatükis. Üldine hinnang valminud veebirakendusele oli väga positiivne.

4.4 Võimalikud edasiarendused

Küsimustikust saadud veebirakenduse tagasiside, juhendajate tagasiside ning autori enda hinnangu põhjal on välja toodud järgnevad viisid, kuidas veebirakendust on võimalik parandada:

1. Veebirakenduse taasesitajale lisada järgnevaid täiendusi:
 - 1.1. Võimalus sündmuste läbimängimise kiirust muuta.
 - 1.2. Võimalus läbi mängida sündmuseid klahvi all hoidmata.
 - 1.3. Võimalus vahetada jälgitavat programmi koodiredaktoris.
 - 1.4. Võimalus vahetada analüüsitava logifaili.
 - 1.5. Parandada jõudlust suurte logifailide läbimängimisel.
 - 1.6. Koodiredaktori aken võiks kerida vastavalt sinna, kus tehakse muudatusi.
 - 1.7. Lisada kasutusõpetus.
2. Järjestada analüüsitud logifailide nimekiri aja järgi.
3. Lisada lahenduste võrdlemise võimalus.
4. Parandada veebirakenduse tagasiside andmist kasutajale faili ebaõnnestunud analüüsi korral.
5. Lisada analüüsitud logifaili tulemustele värvid tähistamiseks, kas logifaili omanik on teinud vastavat tegevust palju või vähe.
6. Täiendada lisatud teksti pikkuse diagrammi nii, et oleks võimalik vaadata logifaili jooksul tegutsetud programmide tekstide pikkuse muutust programmide kaupa.

Kokkuvõte

Selle bakalaureusetöö eesmärk oli luua veebirakendus, mis parendaks olemasolevaid Thonny logide analüüsimise lahendusi. Valminud veebirakenduse abil on nüüd võimalik analüüsida mitut logifaili korraga. Veebirakendusele saab logifailide sisendiks anda peale TXT-failiformaadis olevate logifailide veel kaustasid, mis võivad olla omakorda rekursiivselt tihendatud ZIP-failidesse. Analüüsi tulemusena on võimalik logifaili kohta näha üldandmeid ning kasutada taasesitajat ja lisatud teksti pikkuse diagrammi. Valminud veebirakendus täidab kõiki sellele seatud nõudeid. Valminud veebirakendus ühendas endasse olemasolevate Thonny logide analüüsimise lahenduste funktsionaalsused ning on terviklik lahendus õpilaste logifailide kiireks ja efektiivseks analüüsiks.

Thonny logide analüüsi veebirakendus on kättesaadav veebiaadressil <https://progtugi.cs.ut.ee/thonny-log-analyser/>.

Valminud veebirakenduse arenduse jooksul testisid lahendust peale autori töö juhendajad. Veebirakendust testisid kursuste „Programmeerimise alused“ ja „Programmeerimise alused II“ viis õppejõudu, kelle tagasiside näitab, et tööle seatud eesmärgid said täidetud. Saadud õppejõudude tagasiside, juhendajate tagasiside ja autori hinnangu põhjal tehti nimekiri veebirakenduse edasiarenduse võimalustest.

Viidatud kirjandus

- [1] Arvutiteaduse instituudi MOOC'id. <https://mooc.ee/> (28.12.2020)
- [2] Thonny logid. <https://courses.cs.ut.ee/2019/poemp/Main/Periood2Thonnylogid> (28.12.2020)
- [3] Roosi, A. (2019). Thonny logifailide analüüsi automatiseerimine (Bakalauresetöö), Tartu Ülikool, arvutiteaduse instituut. https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=66416&year=2019 (07.03.2021)
- [4] Annamaa, A. (2015). Introducing Thonny, a Python IDE for Learning Programming. Proceedings of the 15th Koli Calling Conference on Computing Education Research. <https://dl.acm.org/doi/10.1145/2828959.2828969> (02.04.2021)
- [5] Thonny. <https://thonny.org/> (28.12.2020)
- [6] Meier, H. (2018). Õppijate käitumismustrid programmeerimisülesande lahendamisel: logifailide analüüs (Magistritöö). Tartu Ülikool, arvutiteaduse instituut. https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=62175&year=2018 (02.04.2021)
- [7] Lauesen, S. (2002). Software Requirements: Styles and Techniques. Addison-Wesley. (pp. 14-15). <https://books.google.ee/books?id=6Yu7s6XOV8cC> (02.04.2021)
- [8] jQuery. <https://jquery.com/> (23.04.2021)
- [9] Bootstrap. <https://github.com/twbs/bootstrap> (23.04.2021)
- [10] JSZip. <https://stuk.github.io/jszip/> (23.04.2021)
- [11] Highlight.js. <https://github.com/highlightjs/highlight.js> (23.04.2021)
- [12] Chart.js. <https://www.chartjs.org/> (23.04.2021)
- [13] Chart.js adapter date-fns. <https://github.com/chartjs/chartjs-adapter-date-fns> (23.04.2021)
- [14] date-fns. <https://date-fns.org/> (23.04.2021)
- [15] Chart.js time axis. <https://www.chartjs.org/docs/3.1.1/axes/cartesian/time.html> (23.04.2021)
- [16] Custom switch. <https://github.com/haubek/bootstrap4c-custom-switch> (23.04.2021)

[17] Font Awesome. <https://fontawesome.com/> (23.04.2021)

Lisad

I. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, **Rene Kütt**

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

Thonny logide analüüsi veebirakendus

mille juhendajad on Marina Lepp ja Heidi Meier,

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.

Rene Kütt

07.05.2021