

Tallinna Ülikool
Digitehnoloogiaste instituut
Informaatika õppekava

GITHUBI PÕHISE ÕPIKESKKONNA ARENDAUS

Bakalaureusetöö

Autor: Andre Post
Juhendaja: Romil Rõbtšenkov

Autor:, 2018
Juhendaja:, 2018
Instituudi direktor:, 2018

Tallinn 2018

Autorideklaratsioon

Deklareerin, et käesolev bakalaureusetöö on minu töö tulemis ja seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, ning kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

.....

(kuupäev)

.....

(autor)

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina Andre Post (sünnikuupäev: 17.12.1993)

1. Annan Tallinna Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Githubi põhise õpikeskkonna arendus“, mille juhendaja on Romil Rõbtšenkov, säilitamiseks ja üldsusele kättesaadavaks tegemiseks Tallinna Ülikooli Akadeemilise Raamatukogu repositoorumis.
2. Olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õiguseid.

Tallinnas, _____

(digitaalne) allkiri ja kuupäev

Sisukord

Sissejuhatus	5
1 Õpikeskkond	6
1.1 Programmeerimisainetes kasutatavad LMS'id.....	6
2 Rakenduse kirjeldus	8
3 Valitud tehnoloogiad.....	9
4 Õpikeskkonna arendus	11
4.1 Nõuded	11
4.2 Arendusprotsess.....	11
4.3 Rakenduse arhitektuur	12
5 Valminud rakendus	14
5.1 Rakenduse testimine.....	20
Kokkuvõte.....	21
Kasutatud kirjandus.....	22
Summary	23

Sissejuhatus

Eestis suurimateks ülikoolideks, kus õpetatakse programmeerimist on Tallinna Ülikool, Tartu Ülikool ja TalTech. Lisaks ülikoolis õppimisele on suurenenud e-õpe. Rohkem on hakatud jagama materjale ja loenguid Internetis, mis on teinud programmeerimiskeelte õppimise rohkem kättesaadavamaks üle maailma.

Eestis näiteks on täiskasvanutel, kellel puudub kogemus IT valdkonnas, on võimalik osa võtta projektist „Vali IT!“. „Vali IT!“ on täiskasvanute ümberõppeprogramm, mille eesmärgiks on anda osalejatele tarkvaraarendaja alguskused (Vali-IT, 2018). StartIT, Eesti Infotehnoloogia ja Telekommunikatsiooni Liidu projekti poolt on loodud Tech Sistersi kogukond, mis soovib inspireerida ja jagada liikmete kogemust, et rohkem neide julgeks ja tahaks IT-teemadel diskuteerida ning ka ise selles valdkonnas tööle asuda (Eesti Infotehnoloogia ja Telekommunikatsiooni Liit, 2018).

Õppimise toetamiseks on vaja keskkonda, mis aitaks õpetajal hallata õpilasi ja õppetööd. Programmeerimiskeelte õpetamiseks aga ainult nendest ei piisa. Õppekeskkonnal on vaja lisafunktsionaalsust, mida tavalised õppekeskkonnad ei pruugi pakkuda. Õppekeskkond peab lubama jagada koodi faile ning võimaldama nende sisu näha. Lisaks on hea, kui saab jooksutada automaatsest jagatud töödel. Paljudel sellistel õppekeskkondadel on aga puudused, mida töö autor soovib lahendada, näiteks õpilaste tööde hindamine ja hinnete kuvamine.

Bakalaureusetöö eesmärgiks on luua rakendus, kus on võimalik lisada õppeaineid, näha õpilaste poolt esitatud kodutöid ning on võimalik hinnata esitatud töid. Õpetajal on võimalik valida kindlaid aineid mida ta saavib rakendusse lisada. Õpetaja saab näha ka tabelina andmeid esitatud tööde ja hinnatud tööde koguste kohta. Õpilane saab näha tema poolt esitatud tööde hindeid ja kommentaare.

Töö on jaotatud neljaks peatükiks. Esimeses peatükis tutvustab autor lühidalt, mis on õpikeskkond ja milliseid õpikeskkondi kasutatakse programmeerimisainetes. Teises peatükis kirjeldab autor rakendust, mida töö käigus arendatakse ning teisi sarnaseid rakendusi. Kolmandaks tutvustab autor arenduses kasutuses olnud tehnoloogiaid. Neljandas peatükis kirjeldab autor täpsemalt rakenduse arendust, arenduskeskkonna ülesehitust ja õpikeskkonna nõudeid. Viimasena antakse lõplik ülevaade töö käigus arendatud õpikeskkonna rakendusest.

1 Õpikeskkond

Õpikeskkonnad on viimastel aastatel populaarsust kogunud ning on saanud osaks meie õppetööst. Suurimaks teguriks on e-õppe suurenemine, kus põhiliseks vahendiks ongi õpikeskkond. Inglise keeles kasutatakse õpikeskkonna jaoks terminit Learning management system (LMS).

Õpikeskkonna roll oleneb kasutusala ja organisatsiooni vajadustest. Kõige populaarsem kasutusviis on e-õppe materjalide hoiustamine. See teeb materjalid lihtsasti kättesaadavaks õpilastele. Osadel juhtudel on õpikeskkonnas ka e-õppe funktsionaalsused olemas, mis võimaldavad hallata õpilasi, töid ja muud e-õppega seotud informatsiooni (Pappas, 2017).

Õpikeskkonda, LMS'i, saab ühe definitsioonina vaadata, kui suurt repositooriumit, kuhu on võimalik talletada ja jälgida informatsiooni. LMS'id saavad olla nii avalikud kui ka privaatsed, ehk ise hostitavad. Ise hostitud LMS'i jaoks peab olema kasutajal tarkvara installeeritud arvutisse. Avalike LMS'ide puhul on vaja ainult kasutajat, mis annab ligipääsu õpikeskkonnale (Pappas, 2017).

1.1 Programmeerimisainetes kasutatavad LMS'id

Kõige silmapaistvamaks tooteks on Codio¹, mis ei koosne ainult LMS'ist, vaid pakub ka muid lisafunktsionaalsusi programmeerimiskeelte õpetamiseks. Codio võimaldab teha virtuaalseid masinaid, mille peal on võimalik õpetajal õpilase kodutööd kohe käivitada ja testida tulemust. Lisaks programmeerimiskeeltele on võimalik ka õpetada erinevaid andmebaase. Võimalik on kasutada ka automaatset hindamissüsteemi programmeerimise kodutööde jaoks, mille abil server kontrollib edastatud kodutöö üle ja paneb hinde. Õpetajatel on võimalik teha ka lihtne plagiaarismi kontroll üle kõikide õpilaste ja nende tööde (Codio, 2018).

Kõige populaarsem vabavaraline LMS on Moodle², mida on igalühel võimalik ka paigaldada enda serverisse või arvutisse. Moodle on hästi paindlik ja võimaldab lisada funktsionaalsust teekide abil. Programmeerimisainete jaoks on olemas teegid, mis lisavad programmeerimisalast funktsionaalsust juurde. CodeRunner³ ja Virtual

¹ Codio - <https://codio.com/>

² Moodle - <https://moodle.com/>

³ CodeRunner - https://moodle.org/plugins/qtype_coderunner

programming lab⁴ on kaks Moodle teeki, mis annavad õpetajale funktsionaalsuse käivitada õpilase kood veebilehitsejas ning testida selle tulemust.

Autori poolt loodava õpikeskkonna eeliseks on see, et õpilane võib kasutada Github'i ka muudeks töödeks. Õpetaja saab seejuures parema ülevaate. Järgmises peatükis kirjeldatakse täpsemalt loodavat rakendust.

⁴ Virtual programming lab - https://moodle.org/plugins/mod_vpl

2 Rakenduse kirjeldus

Töö autori poolt arendatud rakenduse eesmärgiks on lihtsustada õpetajate tööd programmeerimisainetes. Õpikeskkonna kasutaja saab lisada õpetatavaid programmeerimisained rakendusse. Igale ainele on võimalik lisada kodutöid, mis on mõeldud õpilastele lahendamiseks. Õpilased esitavad oma kodutööd Github'i kaudu ning õpetaja näeb esitatud töid rakenduses. Õpetajal on võimalik hinnata esitatud töid. Õpilane saab näha enda esitatud töid ja hindeid.

Õpetaja saab esmalt lisada organisatsioone ainetena rakendusse. Kui ained on lisatud, saab õpetaja valida kodutööd ainetesse. Nii ained kui ka kodutööd salvestatakse andmebaasi. Õpilane saab esitada oma kodutöö Github'i kaudu peale mida lisatakse see andmebaasi rakenduse abil. Õpetaja saab seejärel näha tabelina esitatud kodutööde arvu, ning hinnata kodutöid.

Suurimaks konkurendiks on Githubi enda poolt loodud rakendus Github Classroom⁵. Classroom võimaldab sarnast funktsionaalsust nagu autori poolt loodud õpikeskkond. Miinuseks on tööde hindamine ja hinnete kuvamine. Classroom'is saab luua ainult ülesandeid, puudub võimalus luua repositooriumit näiteks loengu materjali jaoks. Sellise funktsionaalsuse jaoks tuleb koos Classroom'iga kasutada ka mõnda õpikeskkonda (Github, kuupäev puudub).

Autori poolt arendatud rakendus on töö loomise ajal mõeldud programmeerimisaineteid õppivatele õpetajatele ja õpilastele.

⁵ <https://classroom.github.com>

3 Valitud tehnoloogiad

Rakenduse loomisel said valitud Javascript'il põhinevad tehnoloogiad. Käesolevas peatükis tutvustatakse põhilisi raamistikke, mida kasutatakse rakenduse loomisel.

ReactJS⁶ (edaspidi React) sai valitud kasutajaliidese poolseks tehnoloogiaks, kuna on üks uuemaid tehnoloogiad. Kiiresti populaarsust ja kasutajaid kogunud JavaScript'i teek, mis võimaldab luua veebirakenduste kasutajaliideseid. React annab võimaluse luua rakendusi, mis on võimelised vastavalt vajadustele skaleeruma. Andmete muutusel laetakse ainult vastavad komponendid uuesti, mis on seotud andmetega (Facebook Inc, 2018). Lisaks sellele puudus töö autoril varasem kokkupuude React'iga ning töö arendus oli heaks esimeseks kogemuseks kuidas React'iga rakendusele kasutajaliidest arendada.

React'iga koostööks sai valitud kasutaja poolele **Ant Design**⁷ raamistik. Ant Design pakub React'ile kasutaja poolset kujundust ning lisafunktionaalsust. Lisa funktsionaalsused, mida Ant Design pakub, võimaldavad andmeid töödelda ja kasutaja vaadet muuta ilma, et peaks tegema lisapäringuid serveri poolele. Lisaks võimaldab Ant Design lihtsalt ja kiirelt luua vorme, tabeleid, jpm. Vormidega tuleb kohe kaasa ka valideerimisfunktsionaalsus, mille abil saab lihtsalt vormis erinevad sisendid ära seadistada (Ant Design Team, 2018). Ant Design on loodud Hinna ettevõtte AFX⁸ poolt ning tänu sellele on dokumentatsioon kohti, mis ei ole bakalaureuse töö kirjutamise ajal veel tõlgitud inglise keelde.

Serveri poolele sai valitud **NodeJS**⁹ (edaspidi Node) veebiserver, mis on lihtsasti seadistatav ja kasutab JavaScript'i serveri poolel. See annab koos React'iga võimaluse kasutada ühte keelt nii kasutajaliidese kui ka serveri poolel. Node on JavaScript käituskeskkond mis on üles ehitatud Chrome'i JavaScript mootoril. Node võimaldab luua veebiserverit kasutades JavaScript'i. Lisaks on Node'il kaasas paketi haldussüsteem *npm* millega on võimalik lisada veebiserverile juurde erinevaid pakette (Node.js Foundation, 2018).

⁶ ReactJS - <https://reactjs.org/>

⁷ Ant Design - <https://ant.design/>

⁸ AFX - <https://xtech.antfin.com/>

⁹ NodeJS - <https://nodejs.org/en/>

Express¹⁰ on paindlik Node'i veebirakendus raamistik, mis võimaldab lisada Node'i veebirakendustele lisafunktsionaalsusi (StrongLoop, 2018). Expressis toimub andmevahetus JSON formaadis ning tänu sellele on JavaScript'iga lihtne neid andmeid töödelda.

Andmebaasi valikuks sai **MongoDB**¹¹, NoSQL andmebaas, mis salvestab kirjed dokumentidena. Andmed on salvestatud JSON'i kujul, mis teeb ta ideaalseks valikuks, kuna andmete edastus rakenduses toimub ka JSON'i kujul. MongoDB skaleerub lihtsalt rakenduses suurenemisega ning on tasuta saadav ja avatud lähtekoodiga (MongoDB, 2018).

¹⁰ Express - <https://expressjs.com/>

¹¹ MongoDB - <https://www.mongodb.com/>

4 Õpikeskkonna arendus

Järgnevatel peatükkides on kirjeldatud rakenduse nõudeid ja arendusprotsessi. Seejärel antakse ülevaade rakenduse arhitektuurist.

4.1 Nõuded

Rakenduse kasutaja, õpetaja, peab saama lisada Githubi organisatsiooni rakenduse andmebaasi, mis käituvad kui õppeained. Vastavatele ainetele oleks võimalik lisada organisatsiooni repositooriumeid kodutöödena. Kasutaja peaks saama valida mis kodutööd on seotud ainega, kuna kõik repositooriumid ei pruugi olla mõeldud kodutöödena. Õpilaste poolt sooritatud tööd, Github'i failide hanke taotlused (ingl *pull request*), lisatakse vastava kodutöö juurde. Õpetaja saab näha infot ainete kohta: mitu tööd on, mitu on edastatud ja mitu on ta juba hinnanud. Igat esitatud kodutööd on võimalik õpetajal ka hinnata.

4.2 Arendusprotsess

Rakendus sai arendatud kokku nelja kuu jooksul ning koosnes kahest arendusetapist. Esmases etapis valminud rakendusel olid puudujäägid, mis parandati teises etapis. Õpilasel puudus võimalus näha enda kodutöid ning puudus ka õpetajal kodutööde hindamise võimalus.

Teises etapis vahetati rakenduses välja Bootstrap, kliendipoolne kujundusraamistik, Ant Design'i vastu, mis andis juurde rohkem funktsionaalsust ja paremt koostööd React'iga. Lisaks arendati juurde õpilase vaade, kus õpilane näeb enda esitatud kodutööde hindmeid, kodutööde hindamise võimalus jpm. Rakenduse kood on kättesaadav avalikult Github'i repositooriumis aadressil <https://github.com/AndreTLU/github-opikeskkond>.

Koodiredaktoriks kasutati Visual Studio Code'i¹². Lisa teekideks kasutati koodiredaktorist saadaval olevad npm¹³, DotENV¹⁴, TSLint¹⁵ ja npm intellisense¹⁶. Koodi talletamiseks ja versiooni halduseks kasutati Githubi¹⁷. Githubi koodi

¹² Visual Studio Code – <https://code.visualstudio.com/>

¹³ Npm - <https://marketplace.visualstudio.com/items?itemName=eg2.vscode-npm-script>

¹⁴ DotENV - <https://marketplace.visualstudio.com/items?itemName=mikestead.dotenv>

¹⁵ TSLint - <https://marketplace.visualstudio.com/items?itemName=eg2.tslint>

¹⁶ Npm Intellisense - <https://marketplace.visualstudio.com/items?itemName=christian-kohler.npm-intellisense>

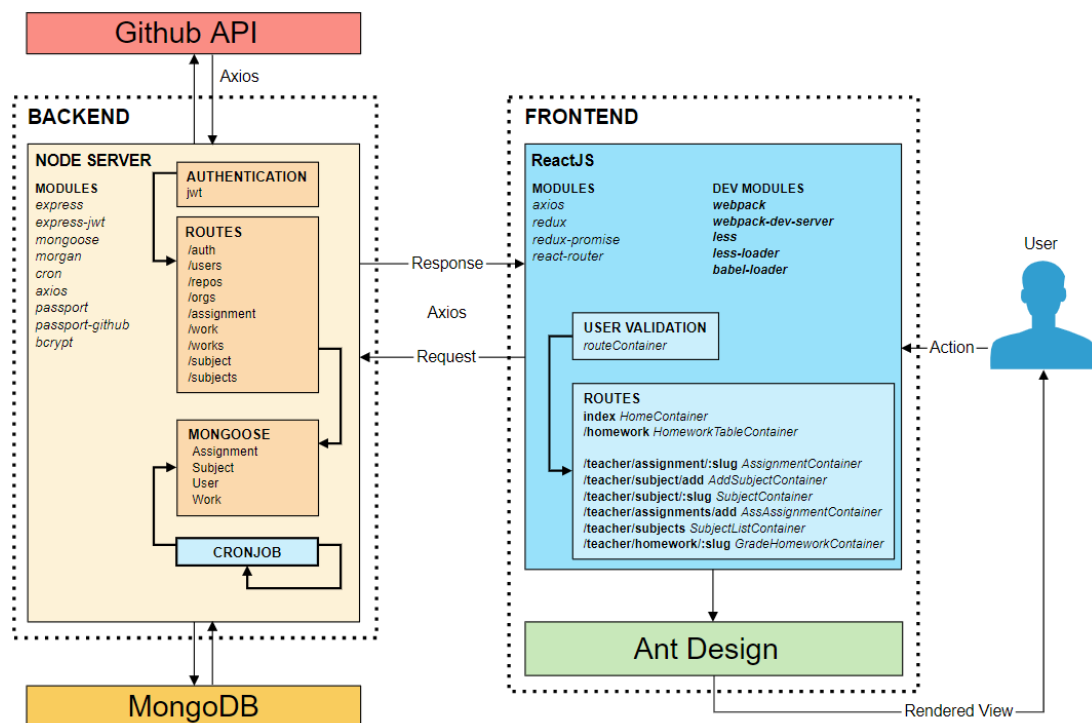
¹⁷ Github – <https://github.com/>

üleslaadimiseks ja erinevate versioonide heaks ülevaateks kasutati GitKraken'it¹⁸. GitKraken on GIT'i kasutajaliides, mis annab hea ülevaate versioonidest ja muudatustest.

4.3 Rakenduse arhitektuur

Arenduskeskkond on jagatud kaheks osaks. Kliendipoolseks osaks, *frontend*, ja serveripoolseks osaks, *backend*.

Serveripoolne osa kasutab erinevaid Node'i mooduleid, pärib Github'ist ja MongoDB andmebaasist andmeid ning saadab need edasi JSON kujul kliendipoolsele.



Joonis 1. Rakenduse ülesehitus

Kliendipoolne osa kasutab kujunduseks Ant Design'i raamistikku. Andmed kuvatakse Ant Design'i tabelite abil, mis annavad lihtsma üleehtuse ning suurema funktsionaalsuse toe, kui tavalised HTML tabelid. Andmete sisestamiseks kasutatakse sama raamistiku poolt pakutavaid *form*'ide lahendusi. Andmete pärimiseks ja saatmiseks kasutab rakendus Axios teeki nii kasutaja poolel, kui ka serveri poolel. Kasutaja poolel päritakse Axiose abil serverist andmeid. Serveri poolel kasutatakse

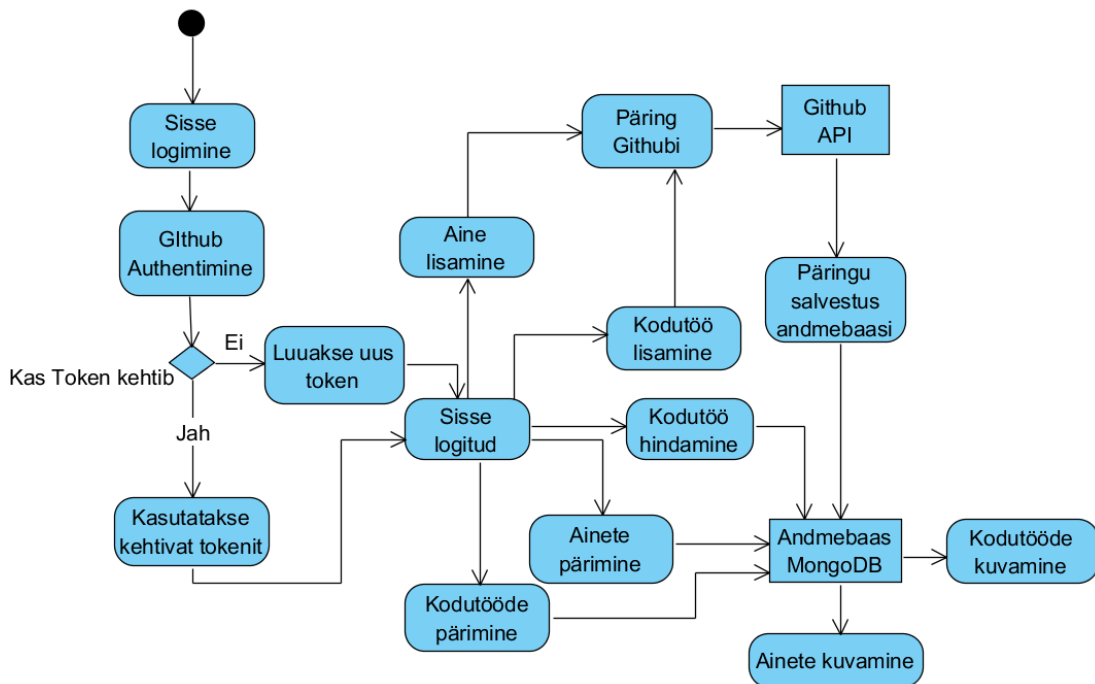
¹⁸ GitKraken - <https://www.gitkraken.com/>

seda Github'ist andmete saamiseks. Andmebaasiga suhtlus toimub läbi Node'i mooduli Mongoose.

Rakenduse arenduse käigus kasutati rakenduse testimiseks ja jooksutamiseks Docker'i keskkonda, kus rakendus töötas Linux'i keskkonnas. Rakendus oli jagatud neljaks *container*'iks, mis omavahel suhtlesid.

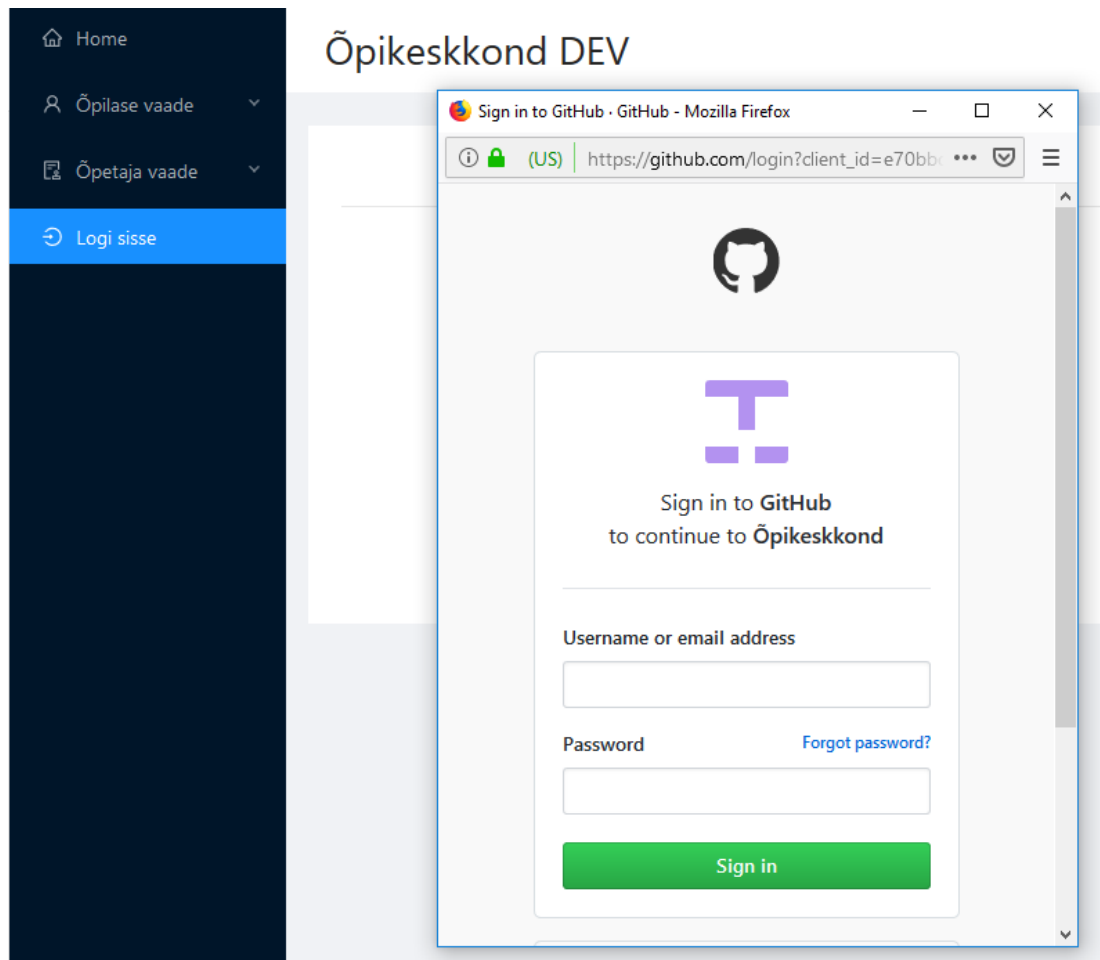
5 Valminud rakendus

Arenduse käigus valmis rakendus, mis vastab eelnevalt töös mainitud nõuetele (vt ptk 4.1 lk 11). Järgnev peatükk tutvustab valminud rakendust ning annab ülevaate rakenduse ülesehitusest.



Joonis 2. Rakenduse töö protsessi joonis

Kasutaja sisse logimiseks tuleb kasutada Github'i kontot. Logi sisse nupule vajutades tuleb ette väike aken, kus küsitakse Github'i konto andmeid, kasutajanime ja parooli (vt Joonis 3). Kui kasutaja on eelnevalt juba Github'i sisenenud mõnest muust akanast või Github'i enda lehelt, siis küsitakse millise Github'i kontoga soovitakse siseneda. Sellisel juhul kasutajanime ja parooli ei küsita.



Joonis 3. Rakendusse sisenemise vaade

Esmasel sisse logimisel on kasutajal ees õpetatavate ainete tabel andmeteta (vt. Joonis 4). Rakendus eeldab, et kasutaja on õpetaja rollis ning kuvab ainete tabeli. Menüüs on kategoriseeritud vaated õpetaja vaateks ja õpilase vaateks ning on võimalik muuta vaadet. Õpetaja näeb kõiki õpetatavaid aineid, kodutöid ning esitatud töid. Õpilane näeb enda aineid ja kodutöid.

Õpikeskkond DEV

Minu Ained (Õpetaja vaade)

Aine	Töid	Esitatud	Hinnatud
Unity test aine 2018 kevad	2	2	2
Angular test aine 2018 sügis	1	0	0

< 1 >

Joonis 4. Rakenduse esmane vaade

Menüüst „Lisa aine, saab õpetaja lisada aineid rakendusse (vt Joonis 5). Ained on Github’i organisatsioonid ning neile tuleb anda rakenduse ligipääsu õigus. Lisa aine lehel on link lehele, kus saab anda rakenduse ligipääsu õigused. Link avaneb uues aknas ning peale õiguste andmist tuleb õpikeskkonna leht taaslaadida. Peale *refresh*’i tekib alla tabelisse nimekiri ainetest, millele on rakendusel ligipääsu õigused ja mida on võimalik lisada rakenduse andmebaasi ainetena. Ained lisatakse rakenduse andmebaasi andmete lihtsamaks kättesaamiseks ning parema ülevaate andmiseks õpetajatele.

Õpikeskkond DEV

Lisa aine Githubist

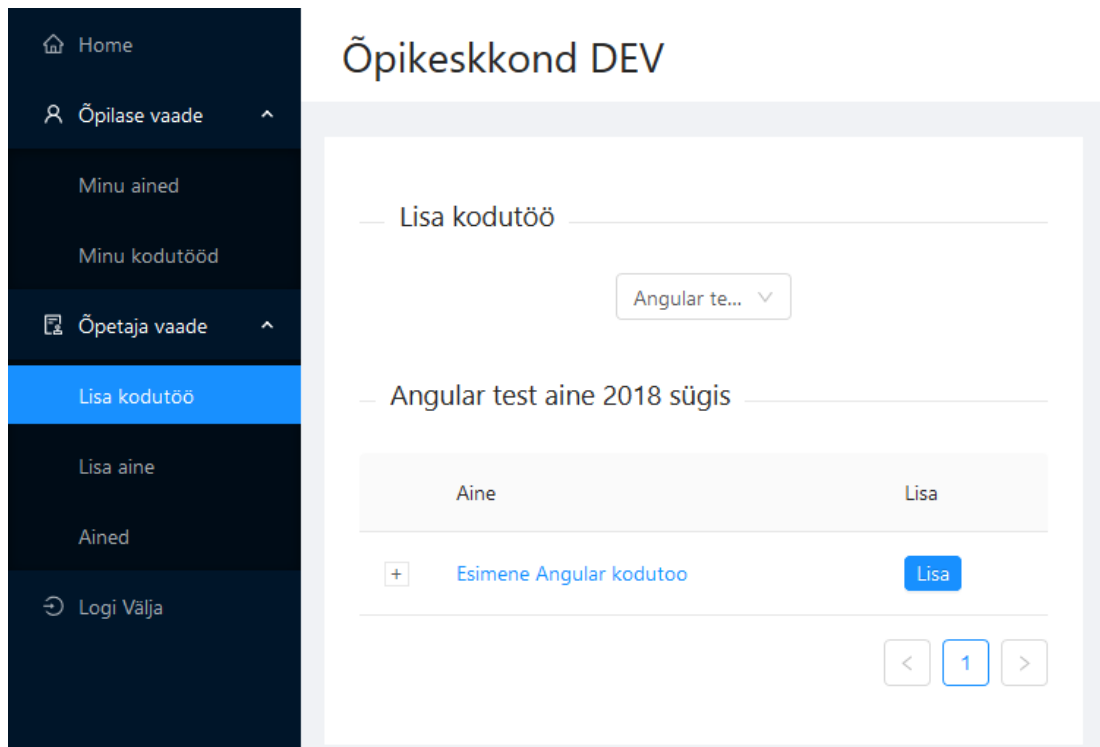
Githubi organisatsioonile tuleb anda õigused Githubist [SIIT](#)

Aine	Lisa
Unity test aine 2018 kevad	Lisa
Angular test aine 2018 sügis	Lisa

< 1 >

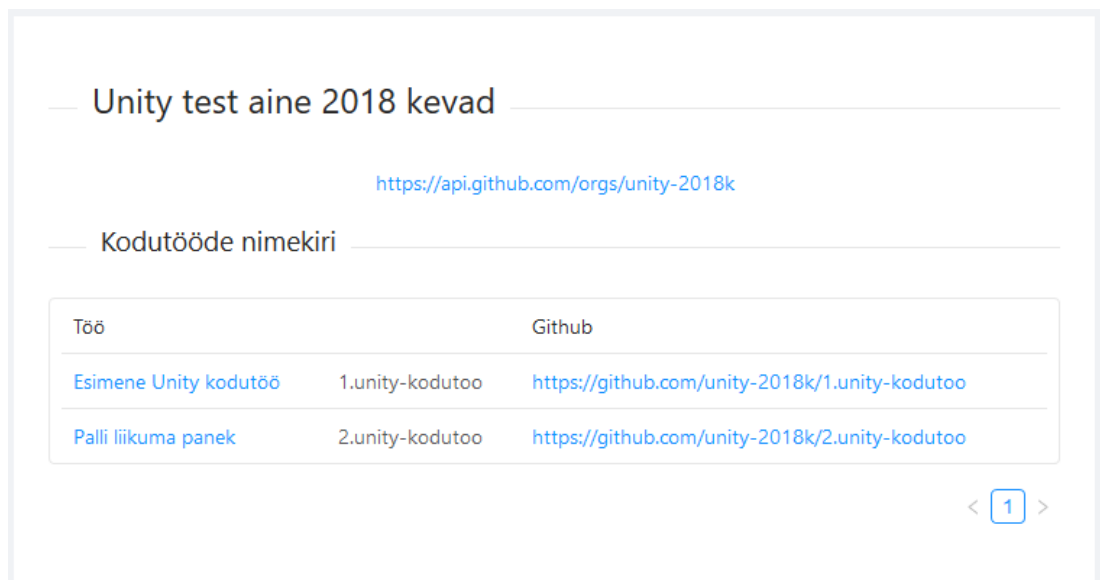
Joonis 5. Aine lisamise vaade rakenduses

Ainetele saab lisada juurde kodutöid Lisa kodutöö lehelt. Esmalt tuleb valida rippmenüüst aine ning seejärel kuvatakse tabelina kõik selle aine võimalikud kodutööd ehk repositooriumid (vt Joonis 6). Repositooriumid tuleb luua kasutaja poolt Github'i kaudu või git käskudega käsurealt. Kodutööd salvestatakse eraldi kirjetena andmebaasi ning viide lisatakse vastava aine juurde. Viiteid kasutatakse ainete tabeli vaates, et kuvada kodutööde arvu ühe andmebaasi päringuga.



Joonis 6. Ainele kodutöö lisamise vaade

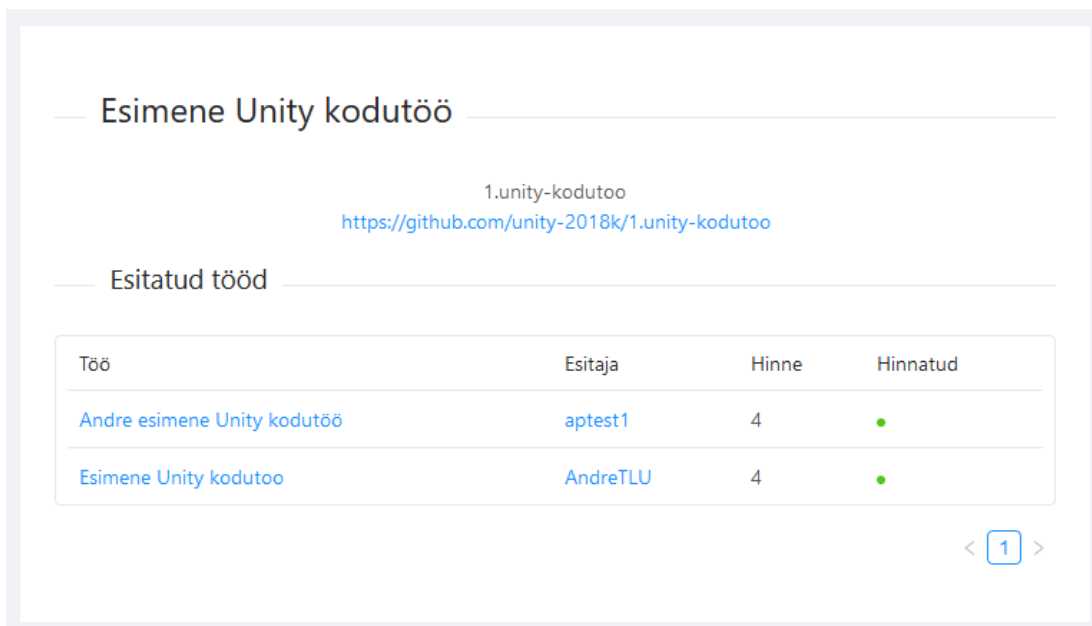
Vajutades ainete tabelis aine nimetuse peale kuvatakse kõik selle aine kodutööd, mis on kasutaja poolt ainele lisatud. Tabelis on välja toodud kodutöö pealkiri, Github'i lühend nimi ja link repositooriumile Github'is (vt Joonis 7).



Joonis 7. Aine kodutööde tabeli vaade

Kodutööde tabelist edasi minnes, vajutades töö pealkirja peale, saab näha kõiki esitatud töid vastavale kodutööle. Tabelis on välja toodud *pull request*'i pealkiri, mis

on õpilase poolt määratud. Näha on ka Github'i kasutajanime ja hinnet, kui töö on hinnatud (vt Joonis 8).

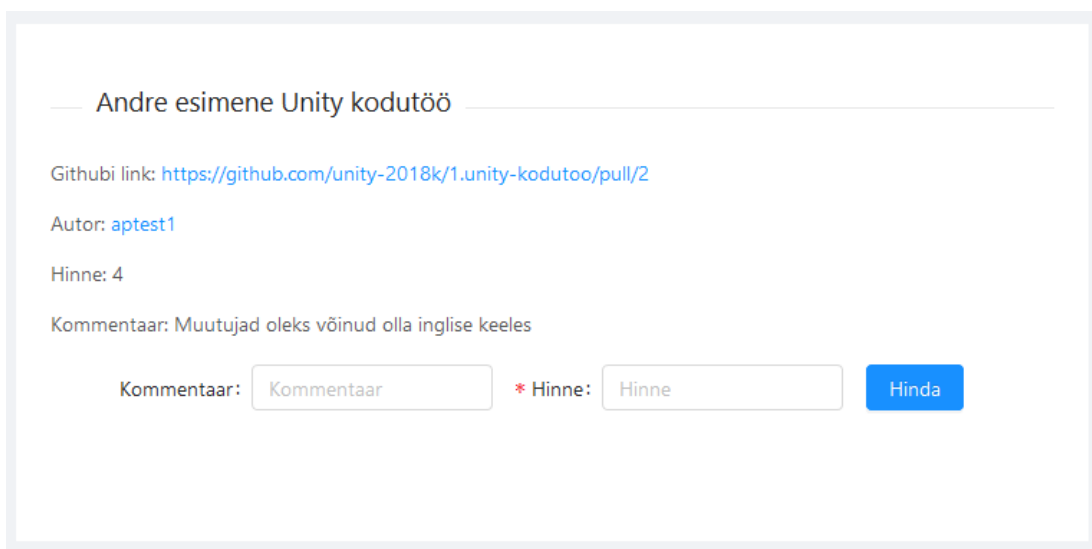


The screenshot shows a GitHub repository page for '1.unity-kodutoo'. Below the repository name, there is a table titled 'Esitatud tööd' (Submitted assignments). The table has four columns: 'Töö' (Assignment), 'Esitaja' (Submitter), 'Hinne' (Grade), and 'Hinnatud' (Graded). Two rows are visible in the table, both with a grade of 4 and a green dot in the 'Hinnatud' column, indicating they are graded. The first row is 'Andre esimene Unity kodutöö' by 'aptest1'. The second row is 'Esimene Unity kodutoo' by 'AndreTLU'. There is a pagination control at the bottom right showing '1' in a box with arrows.

Töö	Esitaja	Hinne	Hinnatud
Andre esimene Unity kodutöö	aptest1	4	●
Esimene Unity kodutoo	AndreTLU	4	●

Joonis 8. Kodutöö esitatud tööde tabel

Esitatud töid on võimalik õpetajal hinnata. Hinnatud töid saab ka uuesti hinnata ning lisada juurde kommentaar, mida õpilane näeb enda kodutööde tabelis. Hetkel Github'ist kommentaari ei näe, kuid on võimalik rakenduse edasi arendamise käigus lisada.



The screenshot shows a GitHub pull request page for 'Andre esimene Unity kodutöö'. It displays the GitHub link, the author 'aptest1', and the grade '4'. Below this, there is a comment section with the text 'Kommentaari: Muutujad oleks võinud olla inglise keeles'. At the bottom, there is a form for adding a comment and a grade, with a 'Hinda' button.

Githubi link: <https://github.com/unity-2018k/1.unity-kodutoo/pull/2>

Autor: [aptest1](#)

Hinne: 4

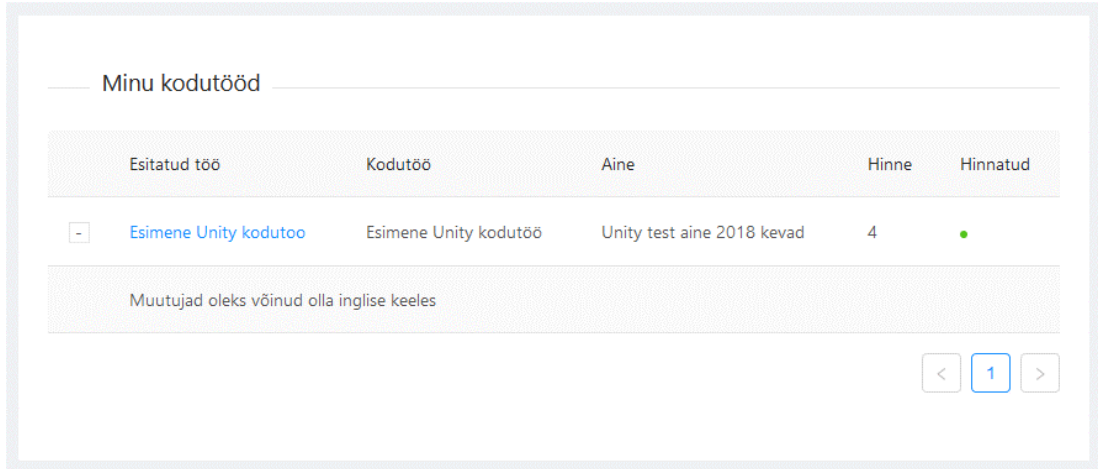
Kommentaari: Muutujad oleks võinud olla inglise keeles

Kommentaari: * Hinne:

Joonis 9. Esitatud töö hindamise vorm

Õpilane saab vasakult menüüst valida Minu kodutööd ning näha tabelina enda esitatud töid. Kui töö on õpetaja poolt hinnatud on tabelis näha ka töö hinne. Lisaks on igal real

vasakul nupp millega saab näha õpetaja poolt lisatud kommentaari esitatud tööle (vt Joonis 10).



Esitatud töö	Kodutöö	Aine	Hinne	Hinnatud
<input type="checkbox"/> Esimene Unity kodutöö	Esimene Unity kodutöö	Unity test aine 2018 kevad	4	•

Muutujad oleks võinud olla inglise keeles

< 1 >

Joonis 10. Õpilase kodutööde tabel

Iga 15 minuti tagant käib serveri poolel *cronjob* mis otsib Github'ist iga aine kohta *pull request*'e ning lisab need vajadusel esitatud tööde hulka. Cronjob, kasutatakse ka lühendit cron, on spetsiifilist ülesannet täitev programm millega saab käivitada töid perioodiliselt. Cronjob kasutab andmebaasis olevate kasutajate *accesstoken*'it päringute loomiseks. Kõikide andmebaasis olevate kodutöödega tehakse päring Github'i ja kontrollitakse kas vastuseks saadud kodutöö hanke taotlus (ingl *pull request*) on juba esitatud tööna salvestatud või mitte.

5.1 Rakenduse testimine

Valminud rakenduse kohta tagasiside saamiseks tegi autor koos Tallinna Ülikooli õppejõuga, Jaagup Kippariga, rakenduse põhitöökäigu läbi. Testimise käigus ilmnisid puudujäägid funktsionaalsustest, mis oleks hea edasiarenduses juurde lisada.

Põhiliseks puudujäägiks oli detailsem koondtabel, kus oleks võimalik näha aine kõiki õpilasi ja kodutöid. See annaks õpetajale parema ülevaate õpilastest ja millised kodutööd neil esitatud on. Teiseks soovis õppejõud, et oleks võimalik õpilase kodutöid juba enne esitamist märkida esitatud. Kolmandaks puudujäägiks oli õpilaste nimede muutmine rakenduse andmebaasi. Kuna kõik ei pruugi kasutada Github'is oma pärisnime siis peaks rakenduses olema võimalik muuta õpilase nähtavad Github'i nime.

Kokkuvõte

Bakalaureusetöö eesmärgiks oli luua õpikeskkonna veebirakendus, mida oleks võimalik kasutada programmeerimisainetes nii õpetajal kui ka õpilasel. Töö käigus tutvustati lühidalt õpikeskkonna mõistet ja milliseid õpikeskkondi kasutatakse programmeerimisainetes.

Bakalaureuse töö käigus arendatud õpikeskkonna rakendus võimaldab õpetajal lisada Github'i organisatsioone ainetena rakendusse. Igale ainele on võimalik lisada rakenduses kodutöid, mis on Github'is loodud repositooriumid vastavale organisatsioonile. Rakendus annab lihtsa ülevaate õpetaja ainetest, kodutööde arvust ja esitatud tööde arvust kasutades Ant Design'i tabeleid. Õpilane saab esitada kodutöö lahenduse Github'i *pull request*'i kaudu. Server pärib iga teatud aja tagant Github'i API'st uusi *pull request*'e ning lisab vajadusel need andmebaasi esitatud töödena. Õpetajal on võimalik näha esitatud töid ning vajadusel neid hinnata ja lisada kommentaar töö kohta. Õpilane saab näha tabelina enda esitatud töid, kui töö on hinnatud õpetaja poolt kuvatakse ka hinne ja kommentaar tabelis.

Arenduse käigus ilmunud raskuseks oli esitatud tööde saamine rakenduse andmebaasi. Kuna tööd saadetakse Github'i siis lahenduseks kasutatakse serveri poolel perioodiliselt töötavat programmi (ingl *cronjob*), mis iga teatud aja tagant pärib Github'ist kõik *pull request*'id ning kontrollib kas on uusi või mitte. Päringu tegemiseks on vaja Github'i kasutaja autentimist. Selleks kasutab server enda andmebasist juhusliku kasutaja *accessToken*'it ning pärib sellega andmed.

Töö autoril puudus eelnev kogemus React'iga veebirakenduse loomisest. Töös arendatud rakendus andis autorile hea ülevaate kuidas toimub rakenduste kasutajaliideste arendus kasutades React'i.

Loodud rakendust on võimalik ka edasi arendada ning töö käigus arendatud kood on Github'i repositooriumist kätte saadav. Esmalt võiks kasutajapoolse koodi struktuuri loogilisemalt üles ehitada. Rakenduses võiks olla kasutusel Service Worker, mis võimaldaks märguandeid edastada reaajas. Selle funktsionaalsusega saaks õpetajale saata sõnumi läbi veebilehitseja, et õpilane on töö esitanud. Õpilasele saaks saata, et tema töö on hinnatud. Õpilasele võiks kuvada rohkem informatsiooni tema tööde kohta. Hetkel on õpilasel näha ainult tema poolt esitatud tööd ja kas need on hinnatud.

Kasutatud kirjandus

Ant Design Team. (2. detsember 2018. a.). *Ant Design of React*. Allikas: Ant Design:
<https://ant.design/docs/react/introduce>

Codio. (1. detsember 2018. a.). *Infrastructure Teaching Code*. Allikas: Codio:
<https://codio.com/features>

Eesti Infotehnoloogia ja Telekommunikatsiooni Liit. (8. detsember 2018. a.). *Tech Sistersi kogukond*. Allikas: StartIT: <https://startit.ee/tech-sistersi-kogukond/>

Facebook Inc. (2018). *React*. Allikas: React: <https://reactjs.org>

MongoDB. (2. detsember 2018. a.). *What is MongoDB?* Allikas: MongoDB:
<https://www.mongodb.com/what-is-mongodb>

Node.js Foundation. (6. oktoober 2018. a.). *About Node.js*. Allikas: Node.js:
<https://nodejs.org/en/about/>

Pappas, C. (3. detsember 2017. a.). *What is A Learning Management System?*
Allikas: eLearning Industry: <https://elearningindustry.com/what-is-an-lms-learning-management-system-basic-functions-features>

StrongLoop. (24. september 2018. a.). *Express*. Allikas: Express:
<https://expressjs.com/>

Vali-IT. (8. detsember 2018. a.). *Projekt*. Allikas: VALI IT: <http://vali-it.ee/projekt/>

Summary

DEVELOPMENT OF GITHUB BASED LMS

The aim of bachelor's thesis was to create a Learning management system (LMS) web application that could be used in programming courses by both teachers and students. In the course of the work, the concept of LMS was introduced briefly and overview was made of what kind of Learning management systems are currently used in programming courses.

Web application developed in the work enables teachers to add Github organizations as subjects to the application. For each subject, a teacher can define Github repositories as assignments. Application gives simple overview of teacher's subjects, assignment count and the number of submitted tasks using Ant Design tables. Students can submit their homework using Github pull requests. Server requests all pull requests periodically at interval and checks if database contains entries with the same id's. Teachers can see all the submitted tasks, grade them and add an comment to each task. Students can see their submitted tasks and if they are graded or not. If a task is graded and has a comment then student can see it on his or her homework table.

The difficulty that emerged during the development of the Web application was getting submitted jobs to the application database. As the student tasks are sent to Github using pull requests, then the solution is to use a periodically running program, cronjob, on server side that requests all pull requests at interval. Retrieved pull requests are checked against application database and if there are new pull requests then they are added to the database. Github's user authentication is required to make these requests and the solution for that is to use random user from application database and make requests using that user's accessToken.

The author of the work had no previous experience in the process of developing web applications using React. The application developed in the work gave the author a good overview of how web application interfaces can be developed using React.

Developed application can also be further developed and the source code is available from the Github repository. First change would be to logically reconstruct frontend code structure. The application could also use Service Worker, which would allow push notifications to be sent. Server could send a message to a teacher, that a student

has submitted their homework. Students could receive a message, that their homework has been graded. Students should see more information about their submitted homework and their subjects. Right now they can only see their submitted homework.