

Tallinna Ülikool

Digitehnoloogiate instituut

Vue.js raamistiku õppematerjal

Seminaritöö

Autor: Fred Korts

Juhendaja: Romil Rõbtšenkov

Autor:”” 2018

Juhendaja:”” 2018

Instituudi direktor:”” 2018

Tallinn 2018

Autorideklaratsioon

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

.....

(kuupäev)

.....

(autor)

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina Fred Korts (sünnikuupäev: 14.04.1991)

1. Annan Tallinna Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose “Vue.js õppematerjali koostamine”, mille juhendaja on Romil Rõbtšenkov, säilitamiseks ja üldsusele kättesaadavaks tegemiseks Tallinna Ülikooli Akadeemilise Raamatukogu repositooriumis.
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 õigusi.

Talinnas, _____

(digitaalne) allkiri ja kuupäev

Sisukord

Sissejuhatus.....	5
1. Vue.js ülevaade	6
1.1. Võrdlus teiste raamistikega	6
2. Olemasolevad õppematerjalid	8
3. Õppematerjali koostamine	8
3.1. Informatsiooni kogumine	9
3.2. Materjali struktuur	10
Kokkuvõte	11
Kasutatud kirjandus.....	12

Sissejuhatus

Uute tehnoloogiate voog on viimaste aastatega ainult suurenenud, ning sellega järge hoida on järjest keeruline paljudele arendajatele. Mõned otsustavad keskenduda tehnoloogiale mis lahendab võimalikult palju probleeme nende töös, ning defineerivad oma ametit selle järgi. Teised hoiavad veebiarendus maastiku muutustel silma peal, kuid eelistavad oma lahendusi.

See seminaritöö keskendub ühele tehnoloogiale, Vue.js. Vue.js on JavaScript raamistik, kuid ta pole ainulaadne. Raamistikud pakuvad struktuuri ja reeglistiku mis aitab vähendada segadust koodis.

Selle seminaritöö eesmärgiks on luua Vue.js 2 (edaspidi Vue.js) raamistiku õppematerjal mida saab pakkuda teise aasta DTI informaatika bakalaureuseõppe üliõpilastele sissejuhatuses JavaScript raamistikke teemal. See materjal sobib just neile, kes on alles astumas veebiarendus maailma, kuid kes ei ole veel lähemalt uurinud raamistikke või ei tea kust alustada.

Antud teema ei eelda varasemat kokkupuudet sarnaste raamistikutega, ning piisab vaid kesktasemel HTML, CSS ja JavaScript oskustest, et omandada antud materjal.

Seminaritöö on jaotatud kolmeks peatükiks. Esimene peatükk annab ülevaate Vue.js'ist, ning võrdleb seda teiste sarnaste raamistikega. Teine peatükk tutvustab olemasolevaid õppematerjale, ametliku dokumentatsiooni, ning kirjeldab autori oma kogemusest antud teema omandamisel. Kolmas peatükk kirjeldab õppematerjali sihtrühma, kuidas autor kogus informatsiooni, ning miks otsustati sellise formaadi kasuks.

1. Vue.js ülevaade

Viimase kümne aasta jooksul on veebilehed kasvanud palju ressursinõudlikumaks ja dünaamilisemaks. Kood mis oli varem serveri poolel on toodud kasutaja lehitsejasse, ning see on tekitanud tuhandeid ridu JavaScript koodi mis ühildub HTML ja CSS koodiga ilma korraliku struktuurita. Sellel põhjusel on välja arendatud raamistikud mis annavad meile parema organiseeritavuse.

Vue.js autor on Evan You, endine Google arendaja. Selle idee põhineb AngularJS'il¹ mida You kasutas paljudes projektides omal ajal Google'is. Talle meeldisid selle võimalused, nagu andmete sidumine DOM'iga, aga selle keerukus tundus talle ebavajalik. Tal oli vaja ainult pooli võimalusi² mida AngularJS pakkus, seega ta hakkas uurima kuidas see raamistik oli üles ehitatud, ning lõi oma vähendatud versiooni (Cromwell, 2016).

Vue.js oli loodud et lahendada väga spetsiifilist probleemi – andmete sidumine DOM'is ning komponendid, seega selle raskusaste on madal võrreldes teiste sarnaste raamistikega nii, et igaüks kellel on HTML, CSS ja JS oskused saab selle päevaga selgeks teha. Just sellel põhjusel on selle populaarsus viimastel aastatel kasvanud hüppeliselt (Cromwell, 2016).

1.1. Võrdlus teiste raamistikega

Esialgse kirjelduse järgi võib tekkida kahtlus, et Vue.js on rohkem nagu demo versioon võrreldes oma konkurentidega. Selle peatüki raames toob välja nii raamistiku tugevused ja nõrkused, ning miks ta on viimaste aastate jooksul tõusnud tõsiseks konkurendiks veebiarendus maastikul.

Võtame võrdluseks React ja Angular'i. Angular'i, sest Vue.js on sellel põhinev ja React kuna ta on hetkel üks kõige populaarsemaid raamistikke (Npmcharts, kuupäev puudub). Antud kolmiku sarnasus esineb selles, et nad kõik kasutavad komponente. Komponent selle töö kontekstis on iseseisev osa koodist millel on oma HTML struktuur ning andmed ja funktsioonid mida võib välja kutsuda rakenduse ulatuses.

Väljaspool eelmainitud sarnasust on tegemist kolme väga erineva raamistikuga, kuigi nad täidavad sama eesmärki: eesrakenduse arendus. Suurim neist on koodi ülesehitus. React kasutab JSX'i (JavaScript süntaksi laiendus mis lubab kasutada HTML koodi), mis tähendab, et selle kood on JavaScript ja HTML'i segu. Selle eelis on, et kõik rakenduse kood on ühes kohas, mis teeb koodi vea leidmise kergemaks kompileerumise faasis.

Angular kasutab HTML malle mis on täiustatud tema enda loogikaga. Võrreldes React'iga võib väita et see on just keerulisem, sest Angular sunnib sind uusi teadmisi omandama erinevalt React'ist kus piisab JavaScript'i tundmisest (Neuhaus, 2017).

¹ Populaarne front-end raamistik
² Andmete sidumine ja virtuaalne DOM

Vue.js ülesehitus võib olla natuke loomulikum noorematele arendajatele kuna meil on eraldatud mallid, JavaScript loogika, ning stiilid failide siseselt. Siin on muidugi see sarnasus, et Vue.js kasutab Angularile sarnast täiustatud HTML'i.

Kiiruse poolest on Vue.js ja React kõige sarnasemad, sest nad mõlemad kasutavad virtuaalset DOM'i, mis peaks teoreetiliselt tõstma rakenduse jõudlust võrreldes Angulariga. Lisaks mängib faili suurus suurt rolli raamistiku kiiruses. Angular on nendest kolmest kõige mahukam (143k), ning Vue just kõige väiksem (23k) (Neuhaus, 2017).

Selle võrdluse põhjal võiks eeldada, et kõik kasutavad Vue.js või on hetkel üle minemas sellele tehnoloogiale. Tegelikult ei ole asi nii lihtne. Raamistiku populaarsus põhineb rohkemal kui ainult selle tehnilistes saavutustes. Kuigi individuaalselt võivad arendajad eelistada Vue.js'i, firma siseselt otsustakse tehnoloogia suuna sellistel parameetritel nagu: hind, aeg, ühilduvus jms.

Samas on paljud firmad³ otsustanud Vue.js kasuks viimastel aastatel. Siinkohal tuleks välja tuua, et Facebook ei ole Vue.js peal ehitatud, see au on endiselt React'i käes. Kuid selle uudistevoog (ingl *newsfeed*) on ehitatud puhtalt Vue.js peal (Bilejczyk, 2017).

³ Facebook, Alibaba, EuroNews, Grammarly, Gitlab, Laracast

2. Olemasolevad õppematerjalid

Vue.js on hetkel üks enim arutatud raamistikest interneti foorumites, ning arendajate vahelistes gruppides. See järgneb tihti vastusena küsimusele: „Millist JavaScript raamistiku on mul kõige mõtekam kasutada oma projektis?“. Kuna tegemist on suhteliselt uue tehnoloogiaga, siis suurem osa õppematerjale on inglise keeles. Eestikeelne materjal on olematu välja arvatud lühikesed kokkuvõtted mõnedes bakalaureuse- ja magistritöös.

Vue.js dokumentatsioon on väga kompetentselt kokku pandud, ning juhul kui järgida õiges järjestuses, siis saab juba esimese peatüki lõpuks omandatud kõige olulisemad funktsionaalsused ja ehitada keskpärane Vue.js rakendus. Ametliku dokumentatsiooni teisest peatükist alates tutvustatakse lugejale Vue.js keerulisemad mõisted juhul kui teil tekib huvi või vajadus.

Autor luges ametliku dokumentatsiooni mitu korda läbi, kuid otsustas täiendavalt uurida teisi allikaid, et paremini omandada selle raamistiku omapärasused. Autor eelistas kasutada veebist leitavaid video kursuseid, ning proovis kahte seeriat. Üks Laracast⁴ ja teine Udemy lehel⁵. Laracast pakkus oma seeriat tasuta, kuid selle miinuseks oli ülesannete puudus. Udemy maksis 11,99€ terve kursuse eest, ning pakkus ülesandeid peatükkide siseselt kui ka suuremate väljakutsete näol.

Autori arvamus on, et Laracast on kõige mõtekam nii finantsiliselt kui ka ajaliselt (Laracast kursus kestis neli tundi võrreldes Udemy'ga mis võttis 21 tundi), ning mistahes augud teadmistes võib täita ametliku dokumentatsiooniga. Autor ei kahetse otsust järgida Udemy kursust, kuid tunnistas et praktilisem oleks olnud kasutada alternatiivi.

Seni kõige kompaktsem õpetus Vue.js jaoks leidub just Tutorialspoint lehel (Tutorialspoint, kuupäev puudub). Kuid kompaktsem ei tähenda parim. Kasutajale seletatakse kõik tähtsamad mõisted lahti, ning selle materjali lõpuks on teil piisavalt teadmisi, et panna oma rakendus tööle iseseisvalt. Näiteid tehakse jooksvalt, ning see annab enamvähem hea arusaama mida Vue.js on suuteline tegema. Selle variandi miinus on, et ei pakuta lugejale väljakutset, et paremini talletada teadmised.

Õppematerjali koostamisel tuli autoril valida millistel teemadel keskenduda ja kuidas neid kõige paremini talletada. Ametlikus dokumentatsioonis olid head koodinäited, kuid teooria osa jäi nõrgaks. Selle õppematerjali koostamisel tuli leida tasakaal teooria ja praktika vahel.

3. Õppematerjali koostamine

Järgnev peatükk annab ülevaate autori kogemusest selle õppematerjali kokkupanekul. Autor kirjutab lähemalt selle õppematerjali eesmärgist, kuidas ta kogus informatsiooni, ning materjali ülesehitusest. Kõige mahukam osa sellest peatükist kirjeldab õppematerjali struktuuri. Autor seletab miks ta keskendus antud teemadel ja mis jäi välja.

⁴ Learn Vue2: Step By Step

⁵ Vue JS 2 – The Complete Guide (incl. Vue Router & Vuex)

Käesoleva materjali läbimise eelduseks on hea arusaam HTML ja CSS koodist, ning mõningane kokkupuude JavaScript'iga. Koodi ei ole vaja kompileeruda, ning piisab tavalisest teksti redaktorist⁶. Vue.js ei ole piiratud operatsioonisüsteemi poolt, kuid autor kirjutas ja testis koodi Windows masinal.

Autori eesmärk antud õppematerjaliga on anda lugejale arusaama Vue.js raamistiku põhiosast. Selle materjali omandamisel peaks lugeja suutma luua kuni keskmise mahuga dünaamilisi rakendusi või tõsta olemasolevate rakenduste võimalusi. Lisaks loodab autor et lugeja saab parema perspektiivi JavaScript raamistikest, ning kuidas nende rakendamine nii säästab arendaja aega kui ka lahendab paljud probleemid veebiarendus projektides.

Selle töö eesmärk on luua eestikeelne õppematerjal mida võiks pakkuda Digitehnoloogiate instituudi teise kursuse õpilastele sissejuhatuseks JavaScript raamistikke teemal. See materjal sobib just neile, kes on alles astumas veebiarendus maailma, kuid kes ei ole veel lähemalt uurinud raamistikke või ei tea kust alustada. Autor arvab, et Vue.js raskusaste on ideaalne algajatele, ning kuna ta pakub paljuid samu võimalusi mida pakkuvad sellega konkureerivad tehnoloogiad on ta ideaalne hüppelaud õpilaste arengule.

Seminaritöö jooksul koostatud õppematerjal ja ülesanded on kättesaadavad veebiaadressilt <https://github.com/fredkorts/seminaritoo/wiki>.

3.1. Informatsiooni kogumine

Autori oma teadmised põhinevad eelnevalt mainitud Udemy kursuse video seeriast. Enne seda oli autor ainult tutvunud ametliku dokumentatsiooniga Vue.js kodulehel, ning uurinud koodinäiteid.

Siinkohal peaks mainima, et esialgne huvi oli puhtalt professionaalne, ning selle õppematerjali sisu põhineb autori algsetel märkmetel. Alles peale nende teadmiste omandamist tekkis autoril idee, et see oleks ideaalne tehnoloogia mida õpetada noorematele huvilistele.

Kuna tegemist on seminaritööga tuli otsustada millist informatsiooni tuleb selle raames käsitleda ilma et töö paisuks liiga suureks. Mitmed teemad nagu: *Routing*, *State Management* ja *Server-Side Rendering* tuli sellest tööst kõrvale jätta, kuid autor arvab et õppematerjalis käsitletud teemad annavad hea ülevaate Vue võimalustest.

Valituks said järgnevad teemad:

- vue.js üles seadmine;
- andmete sidumine;
- direktiivid;
- klassi ja stiilide sidumine;
- komponendid;
- pesa (ingl *slots*);

⁶ Notepad++ või Atom

- vormid.

Informatsiooni kogumise faasis tuli esile mure, et ametlik dokumentatsioon pakub ainult miinimumi teooriat, ning paar koodi näidet iga teema juures. Kogenud arendajale see ei tekitaks probleeme, aga algajal võib kindlasti tekkida veel mitmeid küsimusi.

Autoril tuli lisaks dokumentatsioonile tuua esile selgitusi mitmetest veebi põhistest artiklitest, ning paaril juhul ka uurida teisi õpetusi nagu Tutorialspoint, et paremini seletada mõisteid või lappida auke oma teadmistes.

3.2. Materjali struktuur

Autori esialgne plaan oli teha terviklik materjal PDF dokumendi kujul, kuid juhendaja soovitusel otsustas autor panna sisulise osa õppematerjalist GitHub'i. Selle eelis seisneb kolmes osas:

1. õpilasel on kergem navigeerida materjali teemade vahel Wiki kujul;
2. koodinäited ning ülesanded on eraldatud omaette kaustadesse;
3. tulevikus on võimalik jätkata õppematerjali arengut koostöös huvilistega.

GitHub on ideaalne koht kus hoida projekte või dokumente mis läbivad mitmeid iteratsioone. See näitab koodi muudatuste ajalugu, ning igaüks kellel on ligipääs projektile võib panustada selle arengusse. Juhul kui autor plaanib õppematerjali kunagi tulevikus edasi arendada või teha kellegagi koostööd siis see on ideaalne keskkond selle jaoks. Igaüks saab tekitada hargversiooni ja seda täiendada.

Autor üritas luua õppematerjali struktuuri mis voolab ühest peatükist teise loomulikult moel. See tähendab, et teoreetiline ja praktiline osa on läbi põimunud, et pakkuda lugejale praktilisi näiteid antud teemast. Selline struktuur on levinud internetist leitavatel materjalidel, ning on autori arvamusel kõige efektiivsem viis omandada veebiarendus vallas uusi teadmisi.

Teoreetiline osa õppematerjalist põhineb ametlikul dokumentatsioonil, ning on täiendatud autori oma teadmistega. Autor arvab, et ametlik dokumentatsioon annab kogenud arendajale hea ülevaate raamistikust, kuid vähem kogenud arendaja võib jääda hätta.

Praktiline osa õppematerjalist sisaldab koodinäiteid mida leiab iga teema all, ning kaks ülesannet õppematerjali lõpus. Autor kirjutas suuremad koodinäited sellisel kujul, et lugeja võib selle kopeerida oma masinasse ning kohe hakata katsetama omal käel antud teemaga.

Viimased kaks ülesannet on koostatud sellisel kujul, et lugeja saaks neid lahendada mitut erinevat moodi, ning läbi proovida kõik selle õppematerjali jooksul tutvustatud teemad. Autor pakub lugejale kaks versiooni: tühi ülesanne, kus struktuur on juba valmis ehitatud, ning autori oma lahendus ülesandele mida võib uurida kui jääb hätta lahendusega.

Kokkuvõte

Seminaritöö eesmärgiks oli luua Vue.js õppematerjal, mida võib pakkuda teise aasta DTI informaatika bakalaureuseõppe üliõpilastele. Autor otsustas Vue.js kasuks, sest ta tundis, et selle maht ja keerukuse tase on ideaalne algajatele, kes soovivad laiendada oma teadmisi selles valdkonnas.

Käesolevas töös tutvustati Vue.js raamistiku tausta. Vue.js on JavaScripti raamistik, mis põhineb laialdasemalt kasutuses oleval Angulari raamistikul. Selle raskusaste on oma eellasega võrreldes tunduvalt madalam ning see on sobilik algajatele, kes soovivad uurida JavaScripti raamistikke.

Anti ülevaade õppematerjali kavandamise etapist, tuues välja selle põhilised printsiibid, ning kuidas see tõstab kasutaja rakenduse dünaamilisust. Töö raames koostati õppematerjal, mis on saadaval aadressil <https://github.com/fredkorts/seminaritoo/wiki>. Materjalis on välja toodud ülesanded, mis panevad proovile lugeja teadmised. Juhul kui nendega hakkama ei saa, siis on võimalik uurida autoripoolseid lahendusi ülesannetele.

Materjalist jäid välja järgmised teemad: marsruutimine (ingl *routing*), oleku juhtimine (ingl *state management*) ja serveripoolne vaate loomine (ingl *server-side rendering*). Autori arvates on need teemad hetkel liiga keerulised algajatele, kuid neid võiks tutvustada tulevikus eraldi.

Autor on tulemusega rahul ning arvab, et antud õppematerjal on piisav Vue.js raamistiku põhiosaga tutvumiseks. See annab lugejale vajalikud teadmised, et hakata katsetama ja arendama oma ette. Edasiarenduse poolest on võimalik tulevikus koostada õppematerjal välja jäetud teemadel.

Kasutatud kirjandus

Bilejczyk, B. (2017) Will Vue.js Become a Giant Like Angular or React? Loetud aadressil <https://10clouds.com/blog/vuejs-angular-react/>

Cromwell, A. (2016). *Evan You*. Loetud aadressil <https://betweenthewires.org/2016/11/03/evan-you/>

Gore, A. (2017) What's The Deal With Vue's Virtual DOM? Loetud aadressil <https://Vue.jsdevelopers.com/2017/02/21/vue-js-virtual-dom/>

Logrocket. (2017) An imperative guide to forms in Vue.js. Loetud aadressil <https://blog.logrocket.com/an-imperative-guide-to-forms-in-vue-js-7536bfa374e0>

Neuhaus, J. (2017) Angular vs. React vs. Vue: A 2017 comparison. Loetud aadressil <https://medium.com/unicorn-supplies/angular-vs-react-vs-vue-a-2017-comparison-c5c52d620176>

Npmcharts. (Kuupäev puudub) Compare react, @angular/core, vue download trends. Loetud aadressil <https://npmcharts.com/compare/react,@angular/core,vue>

Tutorialspoint (Kuupäev puudub) VueJS Tutorial. Loetud aadressil <https://www.tutorialspoint.com/vuejs/index.htm>

Vue.js. (Kuupäev puudub) What is Vue.js? Loetud aadressil <https://Vue.js.org/v2/guide/>