

Tallinna Ülikool
Digitehnoloogiaste Instituut

Semantic UI õppematerjal

Seminaritöö

Autor: Erki Sidron

Juhendaja: Jaagup Kippar

Autor:””2016

Juhendaja:.....“”2016

Tallinn 2016

Sisukord

Sissejuhatus	4
1 Semantic UI tutvustus	5
1.1 Semantic UI õppematerjalid	5
1.2 Semantic UI dokumentatsioon	5
1.3 Muud õpetused	6
2 Semantic UI komponendid	7
2.1 Konteiner	7
2.2 Võrestik	7
2.3 Vahesein	9
2.4 Tüpoograafia	10
2.5 Nupud	11
2.6 Ikoonid	12
2.7 Navigeerimismenüü	12
2.8 Leivapuru	13
2.9 Tabelid	14
2.10 Akordion	15
2.11 Vormid	16
2.12 Vormi valideerimine	16
2.13 Dialogiaken	17
2.14 Meedia	18
3 Semantic UI komponentide kood	20
3.1 Semantic UI ülesseadmine	20
3.2 Konteiner	21
3.3 Võrestik	21
3.4 Vahesein	22
3.5 Tüpoograafia	23

3.6 Nupud	23
3.7 Icoonid	23
3.8 Navigeerimismenüü	24
3.9 Leivapuru	24
3.10 Tabelid.....	25
3.11 Akordion.....	25
3.12 Vormid	26
3.13 Vormi valideerimine	27
3.14 Dialoogiaken	29
3.15 Meedia.....	30
4 Ülesanded	31
4.1 Ülesanne 1	31
4.2 Ülesanne 2	32
4.3 Ülesanne 3	33
5 Semantic UI näidisleht	36
6 Tagasiside.....	37
Kokkuvõte	38
Kasutatud kirjandus.....	39

Sissejuhatus

Siinse seminaritöö teemaks on Semantic UI veebiraamistiku õppematerjal. Autor on motiveeritud kirjutama sellel teemal, sest hetkel puudub sellele järjest arenevale ja populaarsemaks muutuvale veebiraamistikule eestikeelne õppematerjal. Antud seminaritöö põhieesmärgiks on tutvustada Semantic UI veebiraamistikku ning läbi selle õpetuse anda ülevaade ja eeldused pärast õppematerjali läbitöötamist iseseisvalt veebilehti kujundada ja luua kasutades Semantic UI veebiraamistikku.

Õppematerjal on mõeldud eelkõige neile, kellel puudub varasem kokkupuude Semantic UI veebiraamistikuga ning kellele võiks sellest abi olla. Õppematerjali edukaks läbitöötamiseks on vajalik HTML ja CSS tundmine heal tasemel ning JavaScripti tundmine algtasemel. Antud töös on pandud rohkem rõhku komponentidele, mis töötavad HTML ja CSS baasil.

Õppematerjali on lisatud autori arvates kõige vajalikumad Semantic UI komponendid. Semantic UI õpetus on kirjutatud toetudes enamasti Semantic UI dokumentatsioonile ja Semantic UI ametlikul veebilehel asuvale informatsioonile.

1 Semantic UI tutvustus

Semantic UI on *responsive front-end* raamistik, mis on täielikult *responsive* ning kõikide elementide suuruste defineerimiseks on kasutatud *em* ja *rem* ühikuid, tänu millele kõik osad toimivad hästi igal ekraanisuurusel. Koodi kirjutamiseks kasutatakse lihtsaid üldkasutatavaid sõnu klasside defineerimiseks ja võimaldab luua *responsive* kujunduse lihtsalt ja mugavalt. Lisaks kasutatakse JavaScripti väljakutsumiseks ja seadistamiseks lihtsaid fraase, millest on kõigil lihtne aru saada. Semantic UI loojaks on Jack Lukic ning esimene versioon anti välja 2013 aastal, hetkel on saadaval 2.1 versioon.

Semantic UI raamistik kasutab HTMLi, CSSi ja JavaScripti lehtede loomisel ja on avatud lähtekoodiga, mida on võimalik muuta ja kohandada vastavalt vajadusele ning omab MIT litsentsi. Semantic UI töötab kõikidel populaarsematel brauseritel ning samuti on kohandatud nii, et toimiks ka mobiilsetes seadmetes. Semantic UI sisaldab endas uusi komponente, mida teistel veebiraamistikel pole. Näiteks vaadetest pakub Semantic UI *feed* ja *comment* vaateid, moodulidest on lisandunud *sidebar* ja *shape*. Lisaks on võimalus veebikonsoolis jälgida komponentide tööd ja näha kas kõik komponendid töötavad korralikult. (Ivaylo, 2014)

1.1 Semantic UI õppematerjalid

Semantic UI kohta on saadaval peamiselt veebipõhised ingliskeelsed õpetused ja videoid, kuid kuna Semantic UI on alles arenev ja uus veebiraamistik, siis õpetusi on saadaval vähe ning kõige parema ülevaate saab Semantic UI veebiraamistikust kasutades Semantic UI dokumentatsiooni, mis sisaldab endast palju näiteid ning põhjalikku ülevaadet. Sellest lähtuvalt kasutan käesoleva õppematerjali koostamiseks peamiselt Semantic UI dokumentatsiooni.

1.2 Semantic UI dokumentatsioon

Dokumentatsioon on Semantic UI kohta koostatud väga põhjalikult ning sisaldab endas näiteid ning võimalusi, kuidas erinevaid elemente kasutada. Kõikidel näidetel on olemas

koodinäide ning näited, mis sisaldavad JavaScripti, mida saab otse dokumentatsioonis katsetada. Semantic UI dokumentatsioon on jaotatud neljaks põhiliseks osaks. Esimene osa on elemendid, kus kõikidel elementidel on üks funktsioon, näiteks nupud. Teine osa on kollektsioonid, mis sisaldab endas erinevaid elemente, mida kasutatakse tavaliselt koos, näiteks menüü. Kolmas osa on vaated, mis võimaldab esitada veebisaidi sisu ühtse osana ja sisaldab endas mitmeid sisu esitamise elemente, mis on omavahel kokku pandud, näiteks meedia. Neljas osa on moodulid, mis sisaldab endas definitsiooni, kuidas need ilmuvad ja käituvad veebiraamistikus, näiteks dialoogiaken. Lisaks on dokumentatsioonis olemas erinevad kavandid, mida on hea veebilehe loomise põhjana kasutada.

1.3 Muud õpetused

Muid õpetusi, peale Semantic UI enda dokumentatsiooni, veebiraamistiku kohta eriti pole, mis oleks tasuta kättesaadavad ja sisaldaks hulgaliselt õpetusi funktsioonide kohta, mis oleks sisukad. Siiski on olemas artikleid ja õpetusi erinevate üksikute elementide kohta koos erinevate kasutusnäidetega. Lisaks võib YouTubes leida mitmeid erinevaid õppevideoid, kuidas luua veebilehekülgi kasutades Semantic UI-d ning õppevideoid erinevate elementide kasutamise kohta. Põhiliselt leidub õppematerjale ja infot rohkem keerukamate komponentide kohta.

2 Semantic UI komponendid

2.1 Konteiner

Konteiner on element, mis kohandab elementide maksimaalset suurust vastavalt sellele, milline on kasutaja ekraani suurus. Konteinereid on mitut tüüpi, mida on võimalik kasutada. Nendeks on *ui container*, mis on tavaline konteiner, *ui text container*, mida saab kasutada teksti jaoks ning on alternatiiv kasutamaks ühe tulbaga võrestikku. Olemas on veel konteinerid, millega on võimalik paigutada tekst vasakule, kasutades *ui left aligned container*, teksti paremale paigutamiseks konteineris tuleb kasutada *ui right aligned container*, teksti keskele paigutamiseks tuleb kasutada *ui center aligned container* ning *ui fluid container*, mille puhul kasutatakse lehe maksimaalset laiust. Kõik elemendid kirjutatakse konteineri sisse, et elemendid kuvataks vastavalt ekraani suurusele.

2.2 Võrestik

Võrestik ehk *grid* jagab lehe 16 võrdseks veeruks, tänu millele on veebileht igas seadmes *responsive*. Veergude loomiseks tuleb kasutada klassi *row*, mille sisse on vaja tekitada vastav arv veerge, veergude koguväärtus peab kokku tulema 16. Näiteks on võimalik lehekülge jagada kaheks võrdseks veeruks, kaheks erineva suurusega veeruks või lihtsalt üheks suureks veeruks. Igale reale on võimalik lisada juurde lisaparameetreid, millega saab määrata mitmeks osaks jaotatuna lehekülge ilmub erinevates seadmetes või millise seadme puhul mingid veerud peidetakse.



Joonis 1 – Võrestik süsteemi veergudeks jagunemise visuaalne näide

Võrestik sisaldab nelja erinevat klassi, telefoni jaoks kasutatakse *(x)wide mobile*, tahvelarvutite jaoks kasutatakse *(x) wide tablet*, arvutite jaoks kasutatakse *(x) wide computer* ja suurte monitoride jaoks *(x) wide large monitor*, x asemel tuleb lisada, mitut veergu soovitakse kasutada, näiteks *sixteen wide mobile*.

Semantic UI võrestik	Telefon	Tahvelarvuti	Arvuti	Suur monitor
Maksimaalne laius	100%	723px	933px	1127px
Klass	(x) wide mobile	(x) wide tablet	(x) wide computer	(x) wide large monitor
Seadme laius	alla 768px	768px - 991px	992px-1199px	üle 1200px

Tabel 1 – Võrestik süsteemi kasutamine erinevatel seadmetel

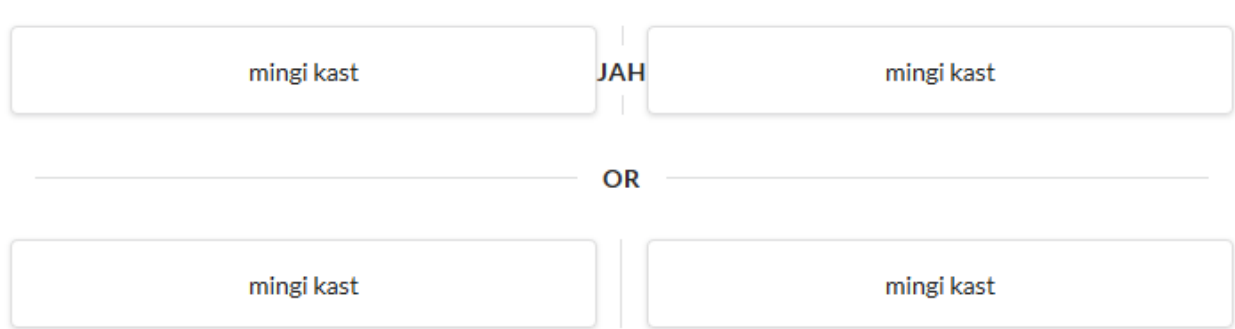
Samuti saab võrestikul nähtavust reguleerida, milliseid veerge näeb telefonis, tahvelarvutis, lauaarvutis või kombineeritult. *Responsive* lahenduse jaoks saab lisada, et mobiilivaates näitaks kahte veergu kõrvuti või kuhjaks veerud üksteise peal.

Klass	Kirjeldus
mobile only	Nähtav ainult seadmetel, kus laius jääb alla 768px, suurematel ekraani laiustel peidetud
tablet only	Nähtav ainult seadmetel, kus laius jääb 768px-991px vahele, teistel ekraani laiustel peidetud
computer only	Nähtav ainult seadmetel, kus laius on üle 992px vahele, väiksematel ekraani laiustel peidetud
stackable	Nähtav ainult seadmetes, kus laius jääb alla 768px, lisab veerud üksteise peale
doubling	Nähtav ainult seadmetes, kus laius jääb alla 991px, lisab veerud kahekaupa kõrvuti

Tabel 2 – Võrestik süsteemi kuvamine erinevatel seadmetel ja nende klassid koos kirjeldusega

2.3 Vahesein

Vahesein ehk *divider*, mille abil on võimalik tekitada vahesein ning jagada asju horisontaalselt või vertikaalselt gruppidesse, tekitades vahele joone. Joone keskele on võimalik lisada tekst või tekitada lihtsalt joon. Vaheseina kasutades on võimalik tekitada erinevaid seksioone, lisada peidetud vahesein või lisada joon pealkirjade alla.



Joonis 2 – Vaheseina visuaalne näide

2.4 Tüpoograafia

Semantic UI kasutab pealkirjades *em* mõõtühikut, kus objekti suurus sõltub otseselt tema *parent* objekti suurusest. Teksti suurus on vaikimisi 1 *em* ja kasutatavad kirjatüübid on järgmised: *Lato*, *Helvetica Neue*, *Arial*, *Helvetica*, *Sans-Serif*. Tavalise teksti kirjutamiseks tuleb tekst lisada `<p>` märkide vahele. Teksti väiksemaks muutmiseks on võimalik lisada tekst `<small>` märkide vahele, teksti rõhutamiseks tuleb tekst lisada `` märkide vahele. Pealkirjadel on 5 erinevat suurust ning pealkiri tuleb märkida `<h1>` kuni `<h5>` märkide vahele, vastavalt sellele, millise suurusega pealkirja soovitakse. Pealkirjale on võimalik juurde lisada alampealkiri, pealkiri on vaikimisi musta värvi ning alampealkiri halli värvi.

H1 pealkiri

Alampealkiri

H2 pealkiri

H3 pealkiri

H4 pealkiri

H5 pealkiri

Joonis 3 – Tüpoograafia visuaalne näide

2.5 Nupud

Semantic UI raamistikul on palju erinevaid variante nuppudest. Kõige tavalisem on *ui button* nupp, mis on standardne ja vaikimisi halli värvi tumehalli tekstiga. Nupp *ui primary button* on sinist värvi, mida kasutatakse nupu olulisuse edasi andmiseks, sama funktsiooni täidab ka *ui secondary button*. Lisaks on veel *ui basic button* nupp, kus on eemaldatud kõik üleliigne kujundus.



Joonis 4 – Nupu liikide visuaalne näide

Samuti on võimalik nuppudele lisada lisaklassid, millega on võimalik muuta nuppude suurust ja värvi. Nuppude suuruse defineerimiseks on olemas järgmised klassid: *mini*, *tiny*, *small*, *large*, *big*, *huge* ja *massive*.



Joonis 5 – Nupu suuruste visuaalne näide

Võimalik on muuta ka nuppude värve, lisades lisaklassi, näiteks punase nupu lisamiseks *ui red button*. Kasutades vaikimisi nuppu on tekst valge ning nupp soovitud värvi. Kasutades *basic* nuppu, on nupp seest valge ning ääred ja tekst soovitud värvi.



Joonis 6 – Nupu värvide visuaalne näide

2.6 Ikoonid

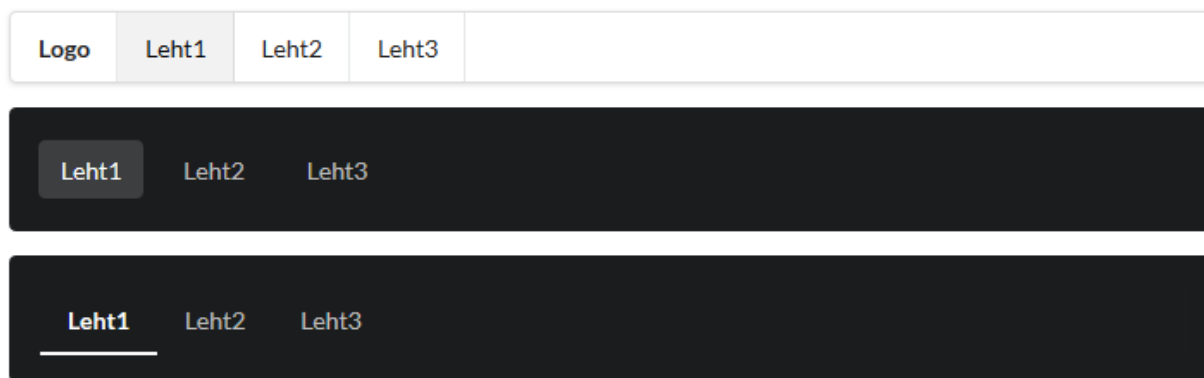
Semantic UI sisaldab endas palju erinevaid ikoone, mida on võimalik kasutada koos teiste elementidega, näiteks nuppudele juurde lisada. Lisaks ikoonidele saab lisada ka enamus riikide lippe. Ikoonide kasutamiseks tuleb kasutada `<i>` elementi ning määrata soovitud ikooni elemendi klass. Ikoonidel on võimalik ümber lisada kastike või ring ning kasutada erinevaid värve, ikoonide suurus on võimalik määrata lisades klassile juurde *mini*, *tiny*, *small*, *large*, *big*, *huge* või *massive*.



Joonis 7 – Ikoonide visuaalne näide

2.7 Navigeerimismenüü

Semantic UI navigeerimismenüü pakub järgnevaid menüü tüüpe: *menu*, *secondary*, *pointing*, *tabular* ja *vertical*. Kõik menüüd on *responsive* ning kohanduvad ekraani suurusega. Menüü lisamiseks kasutame klassi *ui menu*. Menüüd on võimalik fikseerida, et see käiks lehega koguaeg kaasas, määrates menüüle juurde rolli. Fikseerimiseks üles tuleb kasutada *ui top fixed menu*, fikseerimaks alla *ui bottom fixed menu*. Väiksemate ekraanide jaoks on võimalik menüü lisada *ui stackable menu*, mis mobiilivaates paneb menüü üksteise peale virna. Lisaks on menüüdel võimalik määrata suurem suurus, kasutades *ui large menu* või väiksem suurus, kasutades *ui small menu*.



Joonis 8 – Navigeerimismenüü visuaalsed näited



Joonis 9 – Stackable navigeerimismenüü visuaalne näide mobiilivaates

2.8 Leivapuru

Leivapuru ehk *breadcrumb* kasutatakse lehel navigeerimiseks. Leivapuru märgistab ära teekonna, kuidas on antud lehele jõutud ning mille kaudu on võimalik tagasi liikuda tulnud teed. Leivapuru eraldamiseks pakub Semantic UI mitmeid erinevaid võimalusi. Kõige tavalisemad on kaldjoon, mida on võimalik vaheseina abil tekitada ning teisena on võimalik kasutada sektsioone, mis tekitab noolekese. Lisaks on veel erinevaid lisavõimalusi, kasutamaks ikoone või kombineeritult erinevaid elemente. Veel pakub Semantic UI võimalust määrata leivapuru suurust, lisades klassile juurde *small*, *large* või *huge*.

Leht1 / Leht2 / Leht3

Leht1 > Leht2 > Leht3

small > small2 > small3

large > large1 > large2

huge > huge1 > huge2

Joonis 10 – Leivapuru visuaalne näide erinevate klasside ja suurustega

2.9 Tabelid

Semantic UI tabelite puhul on võimalik kasutada mitmeid variante, kõige tavalisema tabeli kasutamiseks on vaja kasutada *ui table*, millel eraldab joon ridu. Tabel ise on ümbritsetud joonega ja veeru pealkirjad on hallika taustaga. Kasutatava klassi *ui celled table* puhul näeb tabel samasugune välja nagu tavalisel tabelil, kuid kogu tabelit eraldavad jooned. Tabelite klassidel *ui basic* ja *ui very basic* puhul on mõlemal taust valge ning ridu eraldavad jooned, ainuke erinevus antud tabelitel on see, et esimest nimetatud tabelit ümbritseb äärejoon, kuid teisel tabelil mitte. Lisaks on veel klass *ui selectable table*, kus hiirega vastavale reale liikudes muutub rea värv hallikaks ning klass *ui striped table*, millel on reavärvid läbi kriipsutatud kahes erinevas värvitoonis, üks rida on hele ja teine tume.

Person	Pealkiri1	Pealkiri2	Pealkiri3
Rida1	Sisu1	Sisu2	Sisu3
Rida2	Sisu1	Sisu2	Sisu3
Rida3	Sisu1	Sisu2	Sisu3
Rida4	Sisu1	Sisu2	Sisu3

Person	Pealkiri1	Pealkiri2	Pealkiri3
Rida1	Sisu1	Sisu2	Sisu3
Rida2	Sisu1	Sisu2	Sisu3
Rida3	Sisu1	Sisu2	Sisu3
Rida4	Sisu1	Sisu2	Sisu3

Joonis 11 – Vasakul on klassi *table* näide ja paremal klassi *basic table* näide

Person	Pealkiri1	Pealkiri2	Pealkiri3
Rida1	Sisu1	Sisu2	Sisu3
Rida2	Sisu1	Sisu2	Sisu3
Rida3	Sisu1	Sisu2	Sisu3
Rida4	Sisu1	Sisu2	Sisu3

Person	Pealkiri1	Pealkiri2	Pealkiri3
Rida1	Sisu1	Sisu2	Sisu3
Rida2	Sisu1	Sisu2	Sisu3
Rida3	Sisu1	Sisu2	Sisu3
Rida4	Sisu1	Sisu2	Sisu3

Joonis 12 – Vasakul on klassi `very basic table` näide ja paremal klassi `striped table` näide

2.10 Akordion

Akordion ehk *accordion* on horisontaalselt paiknevad read, mida kasutatakse sisude kuvamiseks, kuhu peale vajutades kuvab vastava rea sisu. Semantic UI pakub kahte tüüpi akordione, esimene on standard akordion, kus puudub kujundus, mille kasutamiseks tuleb kasutada klassi `ui accordion`. Teine akordioni tüüp on kujundatud akordion, kus igale reale on lisatud kast ümber, mis säilib ka akordioni lahtirullumisel ning mille kasutamiseks tuleb kasutada klassi `ui styled accordion`.

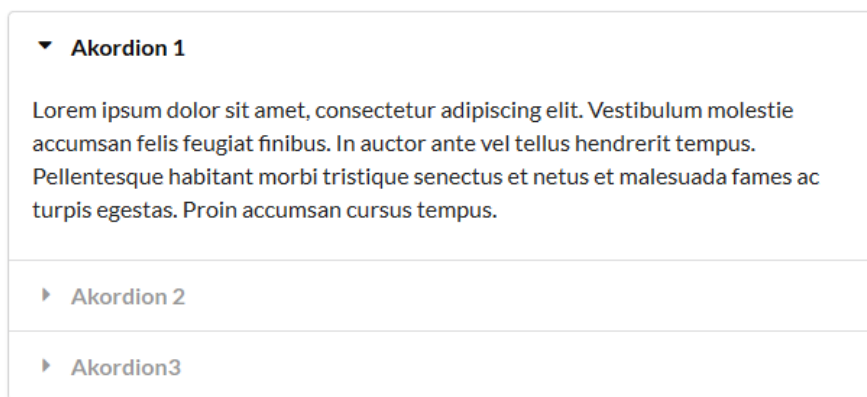
▼ Akordion 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum molestie accumsan felis feugiat finibus. In auctor ante vel tellus hendrerit tempus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin accumsan cursus tempus.

▶ Akordion 2

▶ Akordion3

Joonis 13 – Akordioni näide kasutades klassi `ui accordion`



Joonis 13 – Akordioni näide kasutades klassi ui styled accordion

2.11 Vormid

Semantic UI vormide koostamiseks tuleb kogu vormi sisu panna `<form class="ui form"></form>` klassi vahele. Vormide loomiseks kasutatakse järgmisi elemente: sisestusväljad, valikukastid ja rippmenüü. Vormidele saab juurde panna olekuid, laadimise(*loading*) oleku, mis näitab, et vorm on laadimisolekus, edukas(*success*) olek tekitab rohelise kasti koos kirjaga ja annab teada, et kõik väljad on korras, vea(*error*) olek tekitab punase kasti koos kirjaga ja annab märku, et kuskil on viga, hoiatus(*warning*) olek tekitab pruunika kasti ning annab hoiatuse, sisestusvälja vea(*field error*) olek näitab individuaalselt, kui kuskil väljal on viga. Lisaks saab sisestusväljasid deaktiveerida ja muuta need ainult lugemiseks. Vormidel saab määrata klassi juurde *small*, mis teeb teksti väiksemaks või *large*, mis teeb teksti suuremaks. Sisestusväljade laiust on võimalik määrata samamoodi nagu võrestikul laiusi määratakse, näiteks `<div class="six wide field">`. Kõik vormid on *responsive* ning alla 768px ekraanidel liiguvad kõik vormi elemendid üksteise alla.

2.12 Vormi valideerimine

Semantic UI sisaldab endas peale vormide loomise ka vormide valideerimist. Vormi valideerimisega on võimalik ennetada ebakorreksete andmete sisestamist ning on lihtsaim viis andmete kontrolliks, kui pärast sisestust andmeid kontrollida, mil andmed on juba

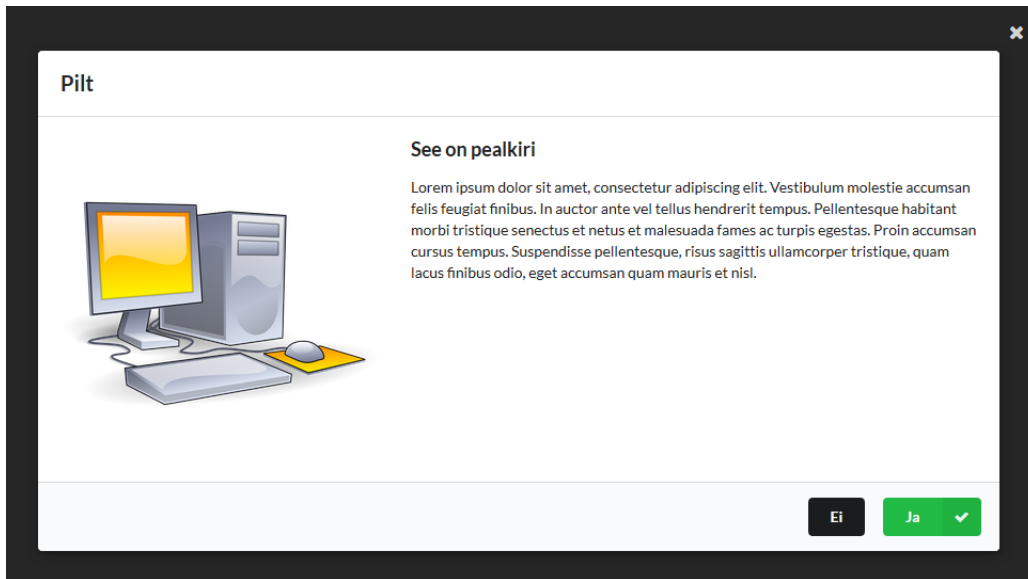
andmebaasi saadetud. Vormi valideerimine näitab, millises vormi osas viga tekkis, muutes välja punaseks ning kuvab alla veateate. Lihtsamate asjade puhul, kus pole vajalik veateate näitamine, saab muuta sisestusvälja punaseks ilma veateateta. Valideerimine kasutab JavaScripti ning sisaldab endas rohkelt võimalusi, mis kujul tekst peab olema. Näiteks võib lisada teksti minimaalse pikkuse, maksimaalse pikkuse, väli peab täidetud olema või sisaldama ainult numbreid.

The image shows a registration form with several fields, each with a red border indicating a validation error. The fields are: Eesnimi (First name) with the value 'Nimi'; Perekonnanimi (Last name) with the value 'Perekonnanimi'; Kasutajanimi (Username) with the value 'Kasutajanimi'; Parool (Password) with three dots; Email with the value 'email@'; and Isikukood (ID number) with the value '123a'. Below the fields is a checkbox labeled 'Nõustun tingimustega' (I agree with the terms) which is unchecked. A blue button labeled 'Sisesta' (Sign up) is located below the checkbox. At the bottom of the form, there is a red-bordered box containing a list of error messages in Estonian: 'Palun sisesta korrektne email', 'Palun sisesta korrektne isikukood', 'Palun sisesta oma perekonnanimi', 'Palun sisesta kasutajanimi', 'Parool peab olema vähemalt 6 tähte', and 'Tingimustega peab nõustuma'.

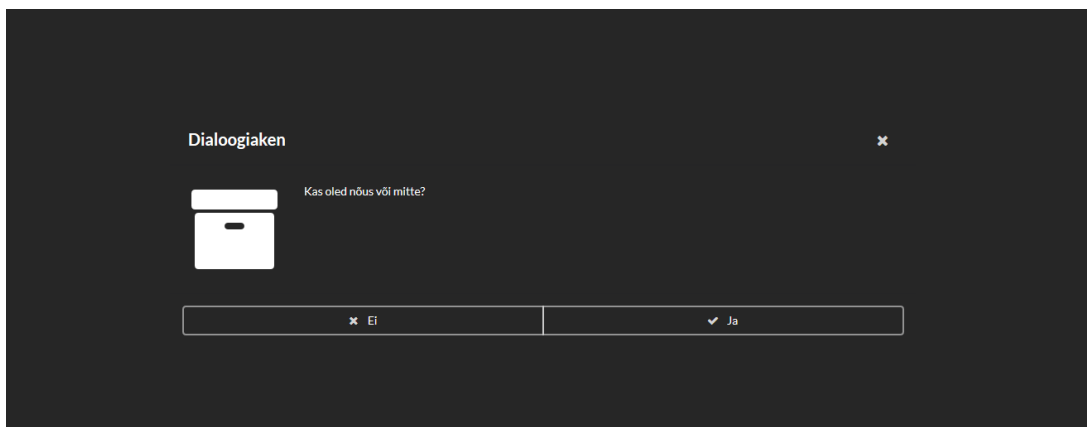
Joonis 14 – Vormide visuaalne näide koos valideerimisega

2.13 Dialoogiaken

Dialoogiakent ehk *modal* kasutatakse peamiselt kasutajatele hoiatuse edastamiseks, küsides kas ta soovib midagi teha ja on selles kindel. Samuti võib dialoogiakent kasutada info kuvamiseks. Semantic UI pakub kahte tüüpi dialoogiaknaid, tavaline dialoogiaken, milleks tuleb kasutada *ui modal* klassi, millega kuvatakse tavaline dialoogiaken. Kasutades klassi *ui basic modal* tekib üle terve ekraani tumeda taustaga dialoogiaken.



Joonis 15 – Dialoogiakna visuaalne näide klassi ui modal kasutamisega



Joonis 16 – Dialoogiakna visuaalne näide klassi ui basic modal kasutamisega

2.14 Meedia

Meedia on vaade, kus pannakse kokku erinevad elemendid ning näidatakse neid koos ühe vaadena. Selle jaoks tuleb kasutada *ui items* klassi, mille alla tuleb tekitada eraldi *ui item* klassid, kuhu on võimalik lisada pilt koos pealkirja ja tekstiga, pilt on vasakul ning sisu paremal või vastupidi ja millele on juurde võimalik lisada linke või erinevaid nuppe. Meedia on *responsive* kujundusega ning mobiilivaates läheb pilt automaatselt üles ja sisu alla.



Pealkiri

Kirjeldus

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum molestie accumsan felis feugiat finibus. In auctor ante vel tellus hendrerit tempus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin accumsan cursus tempus.

Midagi lisaks

Joonis 17 – Meedia visuaalne näide

3 Semantic UI komponentide kood

Siinses peatükis on välja toodud koodinäited komponentide kohta, mida eelmises peatükis on kirjeldatud. Lisaks veel juhised, kuidas Semantic UI üles seada ja kasutada.

3.1 Semantic UI ülesseadmine

Semantic UI pakub erinevaid võimalusi veebiraamistiku ülesseadmiseks, tarkvara lihtsalt allalaadimiseks, installimiseks kasutades NodeJS, gulp või kasutades CDN linke. Semantic UI puhul on võimalik kasutada ainult mõne kindla elemendi CSS ja JavaScripti koodi, eeldusel, et rohkem elemente pole vaja. Alustuseks tuleb Semantic UI kasutamiseks luua HTML fail.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1"/>
  <meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0">
  <title>Semantic UI õppematerjal</title>
</head>
<body>
</body>
</html>
```

Koodinäide 1 – HTML faili loomine

Kõige lihtsam viis Semantic UI rakendamiseks on kasutada CDN linke, mis tuleks lisada `<head></head>` märkide vahele ning lisada ka JavaScript(jQuery) CDN, kuna mõned funktsioonid kasutavad JavaScripti. Tähelepanu tuleks pöörata sellele, et JavaScript oleks lisatud enne Semantic UI JavaScripti.

```
<link rel="stylesheet" type="text/css"
href="https://oss.maxcdn.com/semantic-ui/2.1.8/semantic.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/
jquery.min.js"> </script>
<script src="https://oss.maxcdn.com/semantic-
ui/2.1.8/semantic.min.js"></script>
```

Koodinäide 2 – Semantic UI ja jQuery lisamine

3.2 Konteiner

Kasutades tavalist konteinerit, on laius määratud ja kasutades fluid konteinerit, on laius määramata ning kasutatakse maksimaalselt laiust.

```
<div class="ui container"></div>
<div class="ui fluid container"></div>
```

Koodinäide 3 – Konteineri loomine

3.3 Võrestik

Koodinäites on näha erinevad võimalused, kuidas jagada võrestik osadeks ning kuidas kuvada teatud osad erinevatel ekraanidel ja määrata erinevate võrestike suurus. Kindlasti tuleb jälgida, et võrestiku laius kokku oleks 16 osa ja võrestiku laiuse määramata jätmisel jagatakse automaatselt lehekülj võrdseteks osadeks vastavalt veergude arvule.

```
<div class="ui grid">
  <div class="row">
    <div class="column">1</div>
    <div class="column">1</div>
    <div class="column">1</div>
    <div class="column">1</div>
    <div class="column">1</div>
    <div class="column">1</div>
    <div class="column">1</div>
    <div class="column">1</div>
    <div class="column">1</div>
    <div class="column">1</div>
    <div class="column">1</div>
    <div class="column">1</div>
    <div class="column">1</div>
    <div class="column">1</div>
    <div class="column">1</div>
    <div class="column">1</div>
    <div class="column">1</div>
  </div>
  <div class="four column doubling row">
    <div class="column">4</div>
    <div class="column">4</div>
    <div class="column">4</div>
    <div class="column">4</div>
  </div>
  <div class="row">
    <div class="twelve wide column">12</div>
    <div class="four wide column">4</div>
  </div>
  <div class="mobile only row">
    <div class="eight wide column">8 mobile</div>
    <div class="eight wide column">8 mobile</div>
```

```

</div>
<div class="mobile tablet only row">
  <div class="twelve wide column">12</div>
  <div class="four wide column">4</div>
</div>
<div class="three column row">
  <div class="computer only column">arvuti</div>
  <div class="tablet mobile only column">Tahvel ja mobiil</div>
  <div class="column">Kõik ekraanid</div>
  <div class="column">Kõik ekraanid</div>
</div>
</div>

```

Koodinäide 4 – Võrestik süsteemi loomine

3.4 Vahesein

Koodis on näha, kuidas erinevad võrestik süsteemiga tekitatud kastid on vaheseinaga eraldatud vertikaalselt ja horisontaalselt ning kuidas on kasutatud tekstiga vaheseina ja ilma tekstita. *Segment* klass tekitab kastikese veeru ümber ning aitab visuaalselt paremini aru saada, kus võrestik asub ja kuidas ta vaheseinaga jagatud on.

```

<!-- Osadeks jagamine vertikaalselt koos tekstiga -->
<div class="row">
  <div class="eight wide column">
    <div class="ui segment">
      mingi kast
    </div>
  </div>
  <div class="ui vertical divider">
    OR
  </div>
  <div class="eight wide column">
    <div class="ui segment">
      mingi kast
    </div>
  </div>
</div>
<!-- Horisontaalselt pooleks jagamine koos tekstiga -->
<div class="ui horizontal divider">
  Or
</div>
<!-- Osadeks jagamine vertikaalselt jagamine ainult joonega -->
<div class="divided row">
  <div class="eight wide column">
    <div class="ui segment">
      mingi kast
    </div>
  </div>
  <div class="eight wide column">
    <div class="ui segment">
      mingi kast
    </div>
  </div>

```

```
    </div>
  </div>
</div>
```

Koodinäide 5 – Vaheseina kasutamine

3.5 Tüpograafia

Esimeses pealkirja näites, kus on kasutatud pealkirja suurust H1, on lisatud alampealkirja kasutusnäide. Juhul kui alampealkirja ei kasutata, tuleb H1 pealkirja kasutada samamoodi nagu järgnevad pealkirjad on näidatud.

```
<h1 class="ui header">
  H1 pealkiri
  <div class="sub header">Alampealkiri</div>
</h1>
<h2>H2 pealkiri</h2>
<h3>H3 pealkiri</h3>
<h4>H4 pealkiri</h4>
<h5>H5 pealkiri</h5>
```

Koodinäide 6 – Pealkirjade loomine

3.6 Nupud

Koodinäites 7 on välja toodud erinevat tüüpi nuppude koodinäited ning viimasena on toodud näide, kuidas kasutada nuppu koos ikooniga.

```
<button class="ui button">Vaikimisi</button>
<button class="ui primary button">Põhiline</button>
<button class="ui secondary button">Teisejärguline</button>
<button class="ui basic button">Tavaline</button>
<button class="ui compact button">Kompaktne</button>
<button class="ui labeled icon button">
  <i class="heart icon"></i>
  Ikooniga
</button>
```

Koodinäide 7 – Nuppude loomine

3.7 Ikoonid

Koodinäites 8 on näidatud erinevad ikoonide suurused, ikoon ringiga, ikoon ruuduga ja omavahel kombineeritult 2 ikooni.

```
<i class="circular home icon"></i>
<i class="large bordered home icon"></i>
<i class="huge disabled home icon"></i>
<!-- Kaks ikooni omavahel kokku pandud -->
<i class="huge icons">
  <i class="big thin circle icon"></i>
  <i class="user icon"></i>
</i>
```

Koodinäide 8 – Icoonide loomine

3.8 Navigeerimismenüü

Koodinäites 9 kasutatud menüü on *stackable* ehk mobiili ja tahvelarvuti vaates liigub menüü üksteise alla.

```
<div class="ui stackable menu">
  <div class="header item">Logo</div>
  <a class="active item">
    Leht1
  </a>
  <a class="item">
    Leht2
  </a>
  <a class="item">
    Leht3
  </a>
</div>
```

Koodinäide 9 – Navigeerimismenüü loomine

3.9 Leivapuru

Järgnevalt on koodinäide vaheseinaga eraldatud leivapurust ning koodinäide ikooniga eraldatud leivapurust. Tähelepanu tuleks leivapuru kasutamisel pöörata sellele, et lehel millel ollakse, tuleks määrata klass *active section*.

```
<div class="ui breadcrumb">
  <a class="section">Leht1</a>
  <i class="right angle icon divider"></i>
  <a class="section">Leht2</a>
  <i class="right angle icon divider"></i>
```



```

    <div class="active section">Leht3</div>
</div>
<div class="ui breadcrumb">
  <a class="section">Leht1</a>
  <i class="right angle icon divider"></i>
  <a class="section">Leht2</a>
  <i class="right angle icon divider"></i>
  <div class="active section">Leht3</div>
</div>

```

Koodinäide 10 – Leivapuru loomine

3.10 Tabelid

Tabelite koodinäites toon välja läbi kriipsutatud tabeli näite, kus tabeli read on kordamööda erinevat värvi. Tabeli tüübi muutmiseks tuleb lihtsalt tabeli klass ära muuta.

```

<table class="ui right aligned striped table">
  <thead>
    <tr>
      <th class="left aligned">Person</th>
      <th>Pealkiri1</th>
      <th>Pealkiri2</th>
      <th>Pealkiri3</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td class="left aligned">Rida1</td>
      <td>Sisul</td>
      <td>Sisu2</td>
      <td>Sisu3</td>
    </tr>
    <tr>
      <td class="left aligned">Rida2</td>
      <td>Sisul</td>
      <td>Sisu2</td>
      <td>Sisu3</td>
    </tr>
  </tbody>
</table>

```

Koodinäide 11 – Tabelite loomine

3.11 Akordion

Akordioni tüübi muutmiseks tuleb lihtsalt klassi nimi muuta ning klasside *active title* ja *active content* abil saab määrata, millist akordioni lehe avamisel kuvatakse automaatselt.

```

<div class="ui styled accordion">
  <div class="active title">
    <i class="dropdown icon"></i>
    Akordion 1
  </div>
  <div class="active content">
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. </p>
  </div>
  <div class="title">
    <i class="dropdown icon"></i>
    Akordion 2
  </div>
  <div class="content">
    <p>tekst2</p>
  </div>
</div>

```

Koodinäide 12 – Akordioni loomine

3.12 Vormid

Vormi loomiseks tuleb kogu vormi sisu panna `<form class="ui form"></form>` vahele, näites on lisatud klassile juurde *segment*, mis tekitab kogu vormile kasti ümber. Pärast seda tuleb tekitada klassid nimega *field*, kuhu sisse tuleb vormi märgistus ning sisestuskast. *Placeholderit* kasutatakse teksti kuvamiseks, mida kasutaja näeb enne teksti sisestust. Lisaks tuleks igale vormile märkida *name*, mis on vajalik hiljem vormi valideerimisel. Kindlasti tuleks meeles pidada, et kui soovitakse veateksti kuvada, tuleb vormi lõppu lisada *ui error message* klass.

```

<form class="ui form segment">
  <div class="two fields">
    <div class="field">
      <label>Eesnimi</label>
      <input placeholder="Eesnimi" name="eesnimi" type="text">
    </div>
    <div class="field">
      <label>Perekonnanimi</label>
      <input placeholder="Perekonnanimi" name="perekonnanimi"
type="text">
    </div>
  </div>
  <div class="two fields">
    <div class="field">
      <label>Kasutajanimi</label>
      <input placeholder="Kasutajanimi" name="kasutajanimi" type="text">
    </div>
    <div class="field">
      <label>Parool</label>

```

```

        <input name="parool" type="password">
    </div>
</div>
<div class="two fields">
    <div class="ten wide field">
        <label>Email</label>
        <input placeholder="Email" name="email" type="text">
    </div>
    <div class="six wide field">
        <label>Isikukood</label>
        <input placeholder="Isikukood" name="isikukood" type="text">
    </div>
</div>
<div class="inline field">
    <div class="ui checkbox">
        <input name="tingimused" type="checkbox">
        <label>Nõustun tingimustega</label>
    </div>
</div>
<div class="ui blue submit button">Sisesta</div>
<div class="ui error message"></div>
</form>

```

Koodinäide 13 – Vormi loomine

3.13 Vormi valideerimine

Vormi valideerimiseks tuleb lisada juurde skript HTML faili või eraldi skripti fail, mis tuleks sisse lugeda *head* märkide vahel. *Identifier* määrab ära, millise vormi osa kohta antud valideerimine käib ehk *identifier* peaks olema sama mis *input*-is määratud *name* on. *Rules* all määratakse valideerimise reeglid ning millist veateadet soovitakse kuvada, reegleid võib mitu olla.

```

<script>
    $('ui.form')
        .form({
            fields: {
                eesnimi: {
                    identifier: 'eesnimi',
                    rules: [
                        {
                            type: 'empty',
                            prompt: 'Palun sisesta oma eesnimi'
                        }
                    ]
                },
                email: {
                    identifier: 'email',
                    rules: [
                        {

```

```

        type : 'email',
        prompt : 'Palun sisesta korrektne email'
    }
]
},
isikukood: {
    identifier : 'isikukood',
    rules: [
        {
            type : 'integer',
            prompt : 'Palun sisesta korrektne isikukood'
        }
    ]
},
perekonnanimi: {
    identifier: 'perekonnanimi',
    rules: [
        {
            type : 'empty',
            prompt : 'Palun sisesta oma perekonnanimi'
        }
    ]
},
kasutajanimi: {
    identifier: 'kasutajanimi',
    rules: [
        {
            type : 'empty',
            prompt : 'Palun sisesta kasutajanimi'
        }
    ]
},
parool: {
    identifier: 'parool',
    rules: [
        {
            type : 'empty',
            prompt : 'Palun sisesta parool'
        },
        {
            type : 'minLength[6]',
            prompt : 'Parool peab olema vähemalt 6 tähte'
        }
    ]
},
terms: {
    identifier: 'tingimused',
    rules: [
        {
            type : 'checked',
            prompt : 'Tingimustega peab nõustuma'
        }
    ]
}
}
})
;
</script>

```

Koodinäide 14 – Vormi valideerimine JavaScriptiga

3.14 Dialoogiaken

Dialoogiakna kasutamiseks tuleb lisada koodile juurde skript või siis lugeda antud koodiosa failist, mis käivitab dialoogiakna. Dialoogiakna kasutamiseks tuleb nupul määrata *id*, mida kasutada skriptis ning samuti tuleb lisada *id* juurde ka *modal* klassile. HTML kood asub koodinäites 13 ja JavaScript koodinäide asub koodinäites 14.

```
<button class="ui button" id="modal">Ava dialoogiaken</div>
<div class="ui basic modal" id="modaldiv">
  <i class="close icon"></i>
  <div class="header">
    Dialoogiaken
  </div>
  <div class="image content">
    <div class="image">
      <i class="archive icon"></i>
    </div>
    <div class="description">
      <p>Kas oled nõus või mitte?</p>
    </div>
  </div>
  <div class="actions">
    <div class="two fluid ui inverted buttons">
      <div class="ui red basic inverted button">
        <i class="remove icon"></i>
        Ei
      </div>
      <div class="ui green basic inverted button">
        <i class="checkmark icon"></i>
        Ja
      </div>
    </div>
  </div>
</div>
```

Koodinäide 15 – Dialoogiakna loomine

```
<script type="text/javascript">
$(document).ready(function(){
  $('#modal').click(function(){
    $('#modaldiv').modal('show');
  });
});
</script>
```

Koodinäide 16 – Dialoogiakna kuvamine JavaScriptiga

3.15 Meedia

Mitme meedia eraldi esitamiseks paremini on võimalik kastid ümber lisada meediale, kasutades *ui segment* klassi, kuhu sisse lisada *item* klass või eraldada vaheseinaga, kui lisada *items* klassile juurde lisaklass *divided*.

```
<div class="ui items">
  <div class="item">
    <div class="image">
      
    </div>
    <div class="content">
      <a class="header">Pealkiri</a>
      <div class="meta">
        <span>Kirjeldus</span>
      </div>
      <div class="description">
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
      </div>
      <div class="extra">
        Midagi lisaks
      </div>
    </div>
  </div>
</div>
```

Koodinäide 17 – Meedia loomine

4 Ülesanded

Semantic UI õppematerjali läbinu jaoks on autor välja mõelnud kolm praktilist ülesannet, mida teoreetilise osa läbinu peaks läbi lahendama.

4.1 Ülesanne 1

Esimeses ülesandes tuleb luua esialgu tavaline index.html leht ning lisada Semantic UI jaoks vajalikud CDN-id. Lähemalt tutvumiseks võiks lisada kõigepealt *stackable* navigeerimismenüü ning katsetada seda ka mobiilivaates. Teisena tuleks lisada vahesein tekstiga „Ülesanne 1“ ning meedia koos pildiga, pealkirjaks panna enda nimi ning lisada väike tekst juurde. Ülesanne 1 kättesaadav <http://sidron.ee/semantic/ylesanne1.html> leheküljel.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0">
    <title>Semantic UI õppematerjal</title>
    <link rel="stylesheet" type="text/css"
href="https://oss.maxcdn.com/semantic-ui/2.1.8/semantic.min.css">
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js">
</script>
    <script src="https://oss.maxcdn.com/semantic-
ui/2.1.8/semantic.min.js"></script>
  </head>
  <body>
    <div class="ui container">
      <div class="ui stackable menu">
        <div class="header item">Logo</div>
        <a class="active item">
          Leht1
        </a>
        <a class="item">
          Leht2
        </a>
        <a class="item">
          Leht3
        </a>
      </div>
      <div class="ui horizontal divider">Ülesanne 1</div>
      <div class="ui segment">
        <div class="ui items">
          <div class="item">
```



```

<div class="computer only column">
  <div class="ui segment">
    <button class="ui basic button">Arvuti aind</button>
  </div>
</div>
<div class="tablet only column">
  <div class="ui segment">
    <button class="ui inverted blue button">Tahvel aind</button>
  </div>
</div>
<div class="mobile only column">
  <div class="ui segment">
    <button class="ui yellow button">Mobiil aind</button>
  </div>
</div>
</div>
</div>

```

Koodinäide 19 – Ülesande 2 lahendus

4.3 Ülesanne 3

Kolmandas ülesandes esimese asjana lisa tavaline vahesein ning loo sisselogimise vorm, kus on võimalus sisestada kasutajanimi ja parool ning alla kaks nappu, kus ühel on kirjas „Sisene“ ja teisel „Registreeri“. Siis tekita dialoogiaken, kus vajutades nappu registreeri, mis kuvab ekraanile dialoogiakna koos registreerimisvormiga, millel võiks olla 2-3 sisestuskasti. Sellele juurde tuleb lisada valideerimine koos veateadetega. Ülesanne 3 kättesaadav leheküljel <http://sidron.ee/semantic/ylesanne3.html>. Ülesanded on kokkupanduna ühel leheküljel on kättesaadavad lehel <http://sidron.ee/semantic/ylesanded.html>.

```

<div class="ui divider"></div>
<div class="ui middle aligned center aligned grid">
  <div class="eight wide column">
    <h2>
      <div class="content">
        Sisselogimine
      </div>
    </h2>
    <form class="ui large form" id="vorm1">
      <div class="ui stacked segment">
        <div class="field">
          <input type="text" name="kasutajanimi"
placeholder="Kasutajanimi">
        </div>
        <div class="field">
          <input type="password" name="parool" placeholder="Parool">
        </div>
        <div class="ui primary submit button">Logi sisse</div>
      </div>
    </form>
  </div>
</div>

```

```

        <div class="ui secondary button"
id="registreeri">Registreeri</div>
    </form>
</div>
</div>
</div>
<div class="ui modal" id="modaldiv">
    <i class="close icon"></i>
    <div class="header">
        Registeerimine
    </div>
    <form class="ui form segment" id="vorm2">
        <div class="eight wide field">
            <label>Kasutajanimi</label>
            <input placeholder="Kasutajanimi" name="kasutajanimi" type="text">
        </div>
        <div class="eight wide field">
            <label>Parool</label>
            <input name="parool" type="password">
        </div>
        <div class="eight wide field">
            <label>Email</label>
            <input placeholder="Email" name="email" type="text">
        </div>
        <div class="inline field">
            <div class="ui checkbox">
                <input name="tingimused" type="checkbox">
                <label>Nõustun tingimustega</label>
            </div>
        </div>
        <div class="ui blue submit button"
id="registreeri2">Registreeri</div>
        <div class="ui error message"></div>
    </form>
</div>

```

Koodinäide 20 – Ülesande 2 lahendus HTML osa

```

<script>
    $('#vorm2, #vorm1')
        .form({
            fields: {
                kasutajanimi: {
                    identifier: 'kasutajanimi',
                    rules: [
                        {
                            type: 'empty',
                            prompt: 'Palun sisesta kasutajanimi'
                        }
                    ]
                },
                email: {
                    identifier: 'email',
                    rules: [
                        {
                            type: 'email',
                            prompt: 'Palun sisesta korrektne email'
                        }
                    ]
                },
                parool: {
                    identifier: 'parool',

```

```

rules: [
  {
    type : 'empty',
    prompt : 'Palun sisesta parool'
  },
  {
    type : 'minLength[6]',
    prompt : 'Parool peab olema vähemalt 6 tähte'
  }
]
},
tingimused: {
  identifier: 'tingimused',
  rules: [
    {
      type : 'checked',
      prompt : 'Tingimustega peab nõustuma'
    }
  ]
}
})
;
</script>

```

Koodinäide 21 – Ülesande 2 lahendus JavaScript vormi valideerimiseks

```

<script type="text/javascript">
  $(document).ready(function() {
    $('#registreeri').click(function() {
      $('#modaldiv').modal('show');
    });
  });
</script>

```

Koodinäide 21 – Ülesande 2 lahendus JavaScript dialoogiakna jaoks

5 Semantic UI näidisleht

Näidislehestikuna lõi autor informatiivse veebilehe, kus on kasutatud erinevaid Semantic UI komponente. Näidislehestikus on kasutatud järgmisi Semantic UI komponente:

- Navigatsioonimenüü
- Vahesein
- Võrestik
- Dialoogiaken
- Meedia
- Nupud
- Icoonid
- Vormid
- Vormi valideerimine

Näidislehestik on jagatud kolme osasse: päis, sisu ja jalus. Päises on kasutatud menüül klassi `stackable secondary menu`, millele on lisatud paremale poole 2 nuppu ja vasakule on lisatud nupp, mis avab küljemenüü. Logi sisse nupule on lisatud juurde dialoogiaken, millele vajutades see avaneb, kus on sisselogimiseks tekitatud vorm. Sisu osas on kasutatud võrestikku, mis jagab lehekülje kolmeks võrdseks osaks, kuhu sisse on lisatud ikoonid, mida ümbritseb ring ning pealkirjad ja eraldamiseks on kasutatud vaheseina. Nende alla on lisatud kaks meedia kasti, kus pildid asetsevad vasakul ja tekst koos pealkirjaga paremal pool. Veel on lisatud vorm, pealkirja jaoks on kasutatud horisontaalset vaheseina. Vormi väljadele on juurde lisatud valideerimine. Veebilehekülg on tehtud responsive kujul ning kasutatav igas seadmes. Näidisleht on kättesaadav <http://sidron.ee/semantic/naiteleht.html> veebilehel.

6 Tagasiside

Kõigi inimestega viis autor läbi ühe tunnise õppetunni, mis koosnes Semantic UI raamistiku ülevaatest, komponentide tutvustusest ja õppematerjali läbivaatamisest. Kõik lahendasid tunni aja jooksul näidisülesandeid iseseisvalt ning küsimuste korral autor vastas küsimustele ja abistas.

Kõik 5 inimest suutsid ülesanded ühe tunni jooksul ära lahendada ning kõik vajalikud komponendid tööle saada, kuna varasem kokkupuude sarnaste veebiraamistikega oli olemas. Probleemid tekkisid enamasti kolmandas ülesandes valideerimisel, kus olid identifikaatorid puudu või ei kuvanud veateksti, kuna seda polnud vormile juurde lisatud. Lisaks tekkis probleeme hooletusest, kus kood polnud korrektselt kirjutatud või mõni märk ununes lisada.

Siinse õppematerjali läbinud inimesed olid varasemalt kokku puutunud Bootstrapi ja Foundation 5 veebiraamistikega, kuid varasem kokkupuude neil Semantic UI veebiraamistikuga puudus. Kõigile õppematerjali läbinuile meeldis lihtsus klasside määramise puhul ja Semantic UI visuaalne kujundus elementidel. Lisaks mainiti, et Semantic UI dokumentatsioon on põhjalikum võrreldes Bootstrapi dokumentatsiooniga ja sisaldab endas palju näiteid.

Ülesannetest valmistas autorile kõige rohkem probleeme kolmas, kus dialoogiaknasse tuli lisada vorm ning veateate kuvamine vormi valideerimisel. Esimesena tekkis probleem, kuidas dialoogiaknasse vorm paigutada ja kuidas seda kuvada korrektselt dialoogiaknas. Teise probleemina ei saanud algselt kuvama veateadet vormide valideerimise kohta, kuna identifikaatorid olid valesti määratud või määramata jäänud.

Kokkuvõte

Siinse seminaritöö eesmärgiks oli tutvustada Semantic UI veebiraamistiku ja luua eestikeelne õppematerjal selle kasutamiseks, kus on olemas erinevate komponentide kirjeldused ja koodinäited. Õppematerjal on eelkõige kavandatud neile õpilastele, kellel varasem kokkupuude Semantic UI veebiraamistikuga puudub.

Töö sissejuhatamiseks kirjutab autor, milliseid oskusi õpilane õppematerjali läbimiseks omada võiks, et õppematerjalist paremini aru saaks ja lihtsam läbida oleks.

Töö teises osas tutvustas autor lühidalt veebiraamistikku ja selle tähtsamaid komponente autori arvates. Komponentidel on välja toodud erinevaid klasse ja mis nad endast kujutavad ning milliseid lisaklasse nendele juurde on võimalik lisada. Kõikidele komponentidele on juurde lisatud illustreerivad pildid, et paremini komponentide olemust paremini mõista.

Töö kolmandas osas kirjutab autor, millised on võimalused veebiraamistiku kasutamiseks. Lisaks on lisatud kõikide komponentide jaoks koodinäited, mille põhjal komponendid töötavad. Koodinäidetes on osadel komponentidel toodud välja mitme erineva klassiga koodinäiteid, et oleks parem aru saada, kuidas erinevaid klasse kasutada saab.

Töö viimases osas koostas autor ülesanded, mida õpilased läbi võiks lahendada. Ülesannetele on juurde lisatud autori enda poolsed lahendused. Lisaks on ülesannete lahendused lisatud autori tehtud näidislehestikule, mille koostamisel autor kasutas siinses õppematerjalis välja toodud komponente.

Kasutatud kirjandus

Semantic UI 2016. Semantic UI dokumentatsioon, allikas <http://semantic-ui.com/> (20.03.2016)

Semantic UI 2016. Semantic UI õpetused, allikas <http://learnsemantic.com/> (20.03.2016)

Semantic UI 2016. Semantic UI kasutajaliidese näited, allikas <http://semantic-ui.com/kitchen-sink.html> (20.03.2016)

Gerchev, Ivaylo 2014. Introducing: Semantic UI Component Library, allikas <http://www.sitepoint.com/introducing-semantic-ui-component-library/> (20.03.2016)

Neup, Bibo 2014. Semantic-UI, allikas <https://www.youtube.com/watch?v=pSKGEob1sQo> (20.03.2016)

Semantic UI 2016. Semantic UI võrestik näited, allikas <http://semantic-ui.com/examples/grid.html> (20.03.2016)