

Tallinna Ülikool
Digitehnoloogiate instituut

Veebisaidi arendus sisuhaldussüsteemile WordPress

Seminaritöö

Autor: Ain Arend
Juhendaja: Romil Rõbtšenkov

Tallinn 2017

Autorideklaratsioon

Deklareerin, et käesolev bakalaureusetöö 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)

Sisukord

Sissejuhatus	5
1. Veebisaidi koostamine	7
1.1. Veebimajutus	7
1.2. Domeen.....	7
1.3. Kohanduv veeb	9
1.4. Sisuhaldussüsteemid.....	9
2. Disain ja arendus.....	11
2.1. Ülevaade projektist.....	11
2.2. Domeen ja veebimajutus	12
2.3. Sisuhaldussüsteemi paigaldamine	13
2.4. Lehtede struktuur	14
2.5. Kujundusmalli loomine	15
2.6. Turvalisuse tõstmine.....	16
2.6.1. Liidesed ja kujundusmallid	17
2.6.2. Kasutajaõigused ja admin kasutaja	18
2.6.3. Spämm.....	18
2.6.4. Tegevusplaan.....	19
2.7. Otsingumootoritele ja sotsiaalmeediale optimeerimine	20
2.7.1. Otsingumootoritele optimeerimine	21
2.7.2. Sotsiaalmeediale optimeerimine.....	22
2.8. Puhverdamine	23
2.9. Varundamine	23
2.10. Uuendamine	24
3. Testimine.....	26
Kokkuvõte	27

Kasutatud kirjandus..... 28

Sissejuhatus

Autorit motiveeris antud teemal kirjutama isiklik huvi veebiarenduse ja WordPressi vastu. Lisaks veel ka kogemus klientidega suhtlemisel ning mõistmine, millest enamasti jääb väiksemate ettevõtete kodulehti või isiklike veebe tellides kliendil vajaka, et saada tõesti nii hea koduleht kui soovitakse.

Qmee poolt väljatoodud statistika põhjal tekkis 2014 aastal Internetti iga minuti tagant 571 uut kodulehekülge (Qmee, 2014). See on väga kiire kasv, mille järgi tekib ühe tunni jooksul $571 \cdot 60$ kodulehekülge ning ühes päevas $571 \cdot 60 \cdot 24$ kodulehekülge, mis teeb kokku 822 240 uut kodulehte iga päev. Sellise koguse uute lehtede kõrval peab tagama enda kodulehel kvaliteedi, et seda leitakse Internetist üles ning inimesed sooviksid seda ka edaspidi kasutada.

Järjest populaarsemaks muutuvad ka teenused, mis lubavad lihtsasti omale tasuta kodulehte luua, mille üheks näiteks on teenus nimega Wix.com¹. Samas eksisteerivad endiselt paljude selliste teenuste tasuta versioonidel teatud piirangud, mis ei luba täielikku vabadust ning kodulehe täpselt enda vajaduste järgi seadistamist. Käesoleva tööga tutvustatakse, mida on vaja vabavaralisel sisuhaldussüsteemil WordPress² (lühend WP) põhineva kodulehe loomiseks.

Antud seminaritöö võiks olla kasulikuks lugemiseks inimestele, kel on vajadus kodulehe järele või kes on varem loonud kodulehti mõne teise sisuhaldussüsteemiga. Sel juhul võiks töö anda kasulikke teadmisi, soovitusi ning suuniseid, millele mõelda või kuidas oma probleemile WP keskkonnas läheneda. Kuna WP on aga esmapildul üsnagi lihtne süsteem ja toimiva lahenduse saab suhteliselt kerge vaevaga tööle, siis paraku tihti sellest punktist edasi ei minda ning jäetakse kodulehe kogu potentsiaal kasutamata.

Töö eesmärgiks on anda teoreetilised teadmised selle kohta, mida on vaja ühe kodulehe jaoks, millele tuleks mõelda omale kodulehte WordPressi sisuhaldussüsteemile luues või tellides ning teha läbi ka praktilisteks näideteks ühe konkreetse kodulehe arendus WordPressil.

Kuigi WordPressi kohta leiab palju erinevat materjali Internetist, siis pole tegelikult ühest dokumenti, mis sisaldaks juhiseid selle kohta, kuidas teha omale korralik koduleht otsast lõpuni. Näiteks on 2011. aasta aprillis ilmunud ajakirja Digi 72. numbris³ artikkel selle kohta,

¹ <http://www.wix.com/>

² <https://wordpress.org/>

³ <http://www.digi.ee/wp-content/uploads/2016/04/ajakiri-digi-4-2011.pdf>

kuidas paigaldada WP-i versiooni 3.1. Veel leiab Internetist otsides palju artikleid⁴⁵, mis on õpetused selle kohta, kuidas WP-iga kodulehte luua. Kuid enamasti piirduakse paigaldamise, kujunduse muutmise, lehtede lisamisega ning paremal juhul soovitustega mõne, tegelikult WP-iga kodulehe loomisel äärmiselt olulise, liidese (ingl *plugin*) paigaldamist. Eestis on ka väga palju kompetentseid kodulehe valmistamise teenusepakkujaid, kuid enne nende poole pöördumist on kasulik uurida, kas kodulehe ise ära tegemine tunduks liiga keeruline, ning kui siiski otsustada teenusepakkuja kasuks, siis tuleks teada millele mõelda kodulehe tellimisel, mida üdse tellida ning mis kujul.

Esimeses peatükis avatakse teoreetist tausta selle kohta, mida on vaja teada, et omale kodulehte luua. Seda nullist alates – kuidas valida omale sobiv veebimajutus, millele peaks kujunduse juures mõtlema ning miks kasutada sisuhaldussüsteeme. Teises peatükis kirjeldatakse WP põhiselt, kuidas luua korralikku kodulehte ning praktilisteks näideteks luuakse ka üks konkreetne WordPressil põhinev kliendi nõuetele vastav koduleht. Lisaks analüüsitakse kliendi nõudeid, et mõista millele peaks veel mõtlema, kui oma kodulehte teha või tellida. Viimases peatükis kirjutatakse lahti testimise olulisust ning viiakse läbi ka üks kasutajakogemuse testimine, et selgitada testimise olulist kodulehe arendamisel.

⁴ <http://rahafoorum.ee/kuidas-teha-wordpressi-koduleht/>

⁵ <https://www.veebimajutus.ee/wordpress-koduleht/>

1. Veebisaidi koostamine

Antud peatükis antakse ülevaade sellest, mida on vaja veebisaidi töötamiseks tänapäevases Interneti keskkonnas korrektselt – leht oleks Internetist kõikidele kättesaadav, töötaks eri suurustega ekraanidel ja oleks lihtsasti hallatav.

1.1. Veebimajutus

Enne, kui saab alustada tööd kodulehega, on vaja kohta, kuhu koduleht paigutada ning inimestele lihtsasti meeldejäävat ligipääsupunkti sellele. Selleks, et loodav kodulehekülg oleks Internetist leitav, tuleb kasutada veebiserverit. Tegu on lihtsamalt öeldes arvutiga, mis majutab kodulehekülje jaoks vajalikke faile ning annab neile ligipääsu üle Interneti, kasutades selle jaoks enamasti domeeninime (vt ptk 1.2).

Veebimajutuse jaoks saab, kas ise endale veebiserveri püstitada või osta seda teenusena mõnelt firmalt. Teenusepakkuja käest veebimajutuse ostmine on eelistatum, kuna seljuhul jääb kogu veebiserveri füüsiline haldamine ning vajadusel ka mahtude suurendamine ja muu taoline teenusepakkuja hallata. Lisaks pakuvad paljud firmad virtuaalserverit, mis võimaldab serverit kergemalt vastavalt oma vajadustele seadistada ning kohandada.

Omale veebimajutust valides, tuleks otsustada kuidas ja/või kellelt serveriruum hangitakse. Selle jaoks tuleks lähtuda oma vajadustest püstitatud eesmärgi täitmisel – kui palju on vaja kettaruumi, milliseid tehnoloogiaid kavatsetakse kasutada, milliseid kiiruseid on vaja jne.

Veel tuleks mõelda ka sellele, millist veebiserverit vajatakse. Enamlevinumad vabavaralised lahendused on Apache ja Nginx (Ellingwood, 2015). Neil süsteemidel on ülesehituslikud erinevused ning sellest tulenevad ka võimaluste ja kiiruste erinevused. Täpsemat võrdlust saab lugeda DigitalOceani kodulehel olevast artiklist „Apache vs Nginx: Practical Considerations“⁶.

1.2. Domeen

Interneti domeeninimeserverite süsteem (ingl *Domain Name System*, lühend *DNS*) lubab kasutada inimsõbralikke nimesid veebiserveritele – domeeninimesid. Samal ajal peidavad domeeninimed enda taga IP-aadresse, mis viitavad arvutivõrgus olevatele seadmetele (Gil,

⁶ <https://www.digitalocean.com/community/tutorials/apache-vs-nginx-practical-considerations>

2016). Domeen on firmade puhul enamasti nende ärinimi või kaubamärk ja eraisikutel puhul, nende poolt valitud nimi, mis peaks täitma püstitatud eesmärgi.

Domeeninimi jaguneb astmeteks. Domeeninime lõppu, peale viimast punkti tulevat osa, nimetatakse tippastme või esimese astme domeeninimeks, mis koosneb punktist ja tähtedest. Riikide puhul on igale riigile määratud kahetäheline tippastme domeeninimi – Eesti puhul näiteks .ee. Kuid tippastme domeenide puhul on kasutusel ka erinevaid kombinatsioone – näiteks .info, .samsung või .xyz. Kogu nimekiri võimalikest variantidest on leitav Internet Assigned Numbers Authority⁷ kodulehelt. Tallinna Ülikoolile kuuluva Tlu.ee puhul on tegu juurdomeeniga või teise astme domeeniga (kus .ee on tippaste), ois2.tlu.ee puhul on tegu kolmanda astme domeeniga jne. Veel tuleb mõelda ka sellele, et erinevate tippastme domeenidel on erinev registreerimise hind.

Omale domeeni ostes tuleb arvestada sellega, et praeguseks on registreeritud juba üle 300 miljoni erineva domeeninime (Verisign, 2016). Seega on enamus lihtsaid ning häid nimesid juba võetud. Kuid kuna firma nimed ning kaubamärgid peavad olema unikaalsed, siis on neile ka kergem domeeninime saada. Kaubamärgi omamine annab ainuõiguse selle kasutamiseks ning sarnase nime kombinatsioonide kasutamise keelamise. Seega tuleks alati esmalt registreerida kaubamärk ning seejärel domeen. Eraisikutel on lihtsaim registreerida personaalse kodulehe tarbeks oma nimeline domeen, kuid alates 2014 aastast on võimalik kasutada rohkemaid tippastme domeene (Roesler, 2015), mis lubavad domeeninime registreerimisele läheneda loomungulisemalt, moodustades nime kasutades tippastme domeenime – näiteks <http://www.toureiffel.paris/> .

Tänapäeval on võimalik registreerida ka täpitähti sisaldavaid domeeninimesid, kuid ainult sellise domeeni omamisel on mõningaid puudusi. Nimelt ei ole kõikidel klaviatuuridel täpitähti ning täpiähed tekitavad segadust inimestele, kes ei oska eesti keelt. Sel puhul on lahenduseks kahe domeeni registreerimine. Nii täpitähti sisaldav domeen kui ka neid mitte sisaldava domeeni kasutamine – nt jõulud.ee ja joulud.ee. Sellisel juhul suunaks jõulud.ee domeen joulud.ee domeenile ja tänu sellele saab kasutada nii eestlastele suupärast domeeni kui ka lehele pääseb ligi kasutades klaiatuuri, millel puuduvad täpitähed (ilma, et täpitähti peaks Internetist üks haaval kopeerima hakkama). .

⁷ <http://data.iana.org/TLD/tlds-alpha-by-domain.txt>

1.3. Kohanduv veeb

Kohanduv veeb (ingl *responsive web*) on modernse veebiarenduse standard. Kuna asjade Internet on reaalsus ning järjest enamatel seadmetel on Interneti ühendus, siis peavad ka veebisaidid olema nendel seadmetel kasutatav. Seda nii teksti kujul kui ka visuaalselt ja funktsionaalsuselt.



Joonis 1. Veebisaidi kujundus kohandub vastavalt ekraani suurusele⁸.

Lähtudes veebisaidi disainimises ja arenduses kohanduva veebi printsiibist, peaks leht vastama kasutaja seadmele ja kasutamisele – põhinedes ekraani suurusel, platvormil ja orientatsioonil. Kui kasutaja lõpetab veebilehe kasutamise arvutis ning jätkab telefonis, peaks leht automaatselt vastavat resolutsiooni, pildisuurust ja muid võimalusi kasutama. Ehk veebilehel peaks olema kasutusel tehnoloogiad, mis võimaldavad kasutaja eelistustele kohanduda (vt Joonis 1). Seda printsiipi kasutades välditakse vajadust igale erinevale seadmele eraldi disainimist ja arendamist (Smashing Magazine, 2011).

1.4. Sisuhaldussüsteemid

Sisuhaldussüsteemid (ingl *Content Management Systems*, lühend *CMS*) loovad kodulehele või mõnele muule süsteemile struktureeritud sisuhalduse võimalused. Näiteks dokumentide, piltide, tekstide või muu sisu haldamise võimalused (Built With). CMS-e on palju ning erinevaid, seda nii programmeerimiskeelelt, mida on kasutatud süsteemi loomisel, kui ka

⁸ <https://media.licdn.com/mpr/mpr/p/8/005/08e/3f4/3fd054a.jpg>

võimalustelt, mida süsteem pakub. CMS-e kasutatakse enim, et lihtsustada kodulehe haldamist, võimaldades tavakasutajatel lihtsamalt oma kodulehel oleva sisuga tegeleda.

Veebis kasutatavatest CMS-ideest on enim levinud WordPress, Joomla ja Drupal, mis vastavalt W3Techile omavad turuosadest 58,7%, 7,1% ja 4,8%-st osa (W3Techs, 2017). Kõik kolm on vabavaralised sisuhaldussüsteemid, kuid ometigi omab WordPress teistega võrreldes kordades suuremat turuosa. Süsteemi nii suur populaarsus põhineb selle lihtsusel – seda on tavakasutajal väga lihtne hallata ja liidestega täiendada ning välimust muuta (CommonPlaces, 2013). Näiteks piisab kontaktivormi loomiseks sobivaima liidese allalaadmisest, aktiveerimisest ja õigele lehele lisamisest. Põhjalikum ülevaade sellest, millised on peamised erinevused nimetatud kolme süsteemide vahel, on leitav Website Setupi artiklist „[WordPress vs Joomla vs Drupal](#)“⁹.

WordPressi üheks eeliseks konkurentide ees on alates versioonist 4.4 WP tuuma juurde kuuluv kohanduvate piltide funktsionaalsus. See tähendab, et kui läbi WP administraatori paneeli laetakse serverisse üles uus pilt, siis WP teeb sellest automaatselt erineva suurusega pildid. Kui seda pilti kasutatakse kodulehel, siis lisatakse kaasa kõik suurused ja brauser saab valida endale õige suurusega pildi ning teisi eirata. Misläbi on võimalik, et vähendatakse andmeedastusmahtu ja suurendatakse lehe laadimise kiirust (McGill, 2015). Ühe miinusena võib WordPressi välja tuua mitmekeelsuse puudumise tuumafunktsionaalsusest. Selleks, et luua mitmekeelset veebi, on vaja kasutusele võtta mõni liides.

WordPress sobib nii lihtsamate kodulehtede jaoks kui ka keerukamate projektide jaoks, pakkudes peale värsket paigaldust koheselt üsna paljusid võimalusi – rollidega kasutaja süsteemi, failihaldurit, menüüde süsteemi ning artiklite ja lehtede loomise võimalust, koos otsingumootori jaoks optimeeritud veebiaadressidega ning kommentaariumiga. Lisaks tuleb kaasa ka vaikimisi kohanduva kujundusega kujundusmall, mida on võimalik kohandada omale sobivamaks. Samas on võimalik paigaldada ka uus kujundusmall, mille saab välja valida tuhandete valmis mallide seast WordPressi kodulehelt¹⁰. Just oma sobivuse ning lihtsuse pärast sai WordPress valitud ka loodava projekti sisuhaldussüsteemiks.

⁹ <https://websitesetup.org/cms-comparison-WordPress-vs-joomla-drupal/>

¹⁰ <https://WordPress.org/themes/>

2. Disain ja arendus

Antud peatükis kirjeldatakse esmalt kliendipoolseid nõudeid (vt Tabel 1) ja analüüsitakse neid. Teises alampeatüks räägitakse üldisemalt domeeni registreerimisest, serveritest ning Wordpressi paigaldamisest. Kolmandas alampeatüks kirjeldatakse kirjeldatakse lehtede struktuur ning luuakse põhivaated. Neljas alampeatükk sisaldab kujundusmalliga töötamise erinevaid variante ning meetodeid. Viendas alampeatüks antakse ülevaade sellest, kuidas optimeerida kodulehte otsingumootoritele ning sotsiaalmeediale. Viimasena kirjeldatakse üldiselt klientide või tavakasutaja poolt enim tähelepanuta jäävaid, kuid äärmiselt olulisi teemasid – süsteemi uuendamist ning varundamist.

2.1. Ülevaade projektist

Projektiks on luua kliendi nõuetele ja soovidele vastav koduleht, mis oleks vastavuses ka modernse veebi nõuetega. Klient on Ettevõtte esindaja, kes soovis omale lihtsat kodulehte. Täpsemad kliendipoolsed kodulehele esitatud nõuded on nähtavad Tabel 1, kus on ära kirjeldatud iga kliendi poolt esitatud nõue ning sellele probleemile vastav kasutusele võetav lahendus.

Tabel 1 Kliendipoolsed nõuded kodulehele ning kasutusele võetavad lahendused.

Nõue	Lahendus
Domeeni registreerimine	Veebimajutuseja domeeni tellimine Zone ¹¹ kaudu
Lihtsasti brauserist hallatav	Sisuhaldussüsteemi Wordpressi kasutamine
Töötab kõikidel ekraanidel	Kohanduv kujundusmall
Võimalik lehti ja postitusi lisada	Wordpressi sisseehitatud funktsionaalsus
Lehtedel peab olema kindel struktuur	Wordpressi sisseehitatud funktsionaalsus
Kodulehel peab olema kontaktivorm	Contact Form 7 ¹² liidese paigaldamine

Kliendipoolsetest nõuetest (Tabel 1) on näha, et klient pole IT inimene, ega oma ka varasemat kogemust või teadmisi kodulehe omamises ja haldamises. See tuleb välja sellest, et tegelikult on tähelepanuta jäänud väga paljud olulised aspektid, millele peab omale kodulehte tellides või ehitades mõtlema ning tähelepanu pöörama – turvalisus, otsingumootoritele ja

¹¹ <https://www.zone.ee/et/>

¹² <https://et.wordpress.org/plugins/contact-form-7/>

sotsiaalmeediale optimeerimine, süsteemi uuendamine ning sisu ja tarkvara varundamine (ingl *back up*).

On küll ära kirjeldatud lihtsamad kodulehe funktsionaalsused, mis peavad ühel modernsel kodulehel olema, kuid ainuüksi nendest ei piisa, et tagada omale kvaliteetne koduleht. Sisuliselt saaks nõuetele vastava kodulehe tehtud WordPressi ning ühe liidese paigaldamisega. Lisaks peab mõistma ka seda, et enamasti ei piisa oma kodulehe tegemisel ainult domeeni registreerimisest, vaid juurde on vaja veebimajutust (on olemas ka teenuseid, millest sai põgusalt sissejuhatuses räägitud, mis pakuvad tasuta veebimajutust ning kodulehe loomist ja lubavad suunata mõnele muule registreeritud domeenile, kuid enamasti ekisteerivad neil erikujulised piirangud).

Nendest probleemidest lähtuvalt kasvavad välja ka üldisemad probleem IT arenduses. Nimelt ei oska tavakasutaja või klient näha kogu spektrit, mis käib kodulehe arenduse juurde ning enamasti piirduakse oma kodulehe loomisega või tellimisel pinnapealse poolega. Internetis leiduvaid kodulehti vaadates, leitakse mõned endameelest head lahendused ning võetakse need oma kodulehe aluseks. Siin tulevadki mängu teadmised ja kogemused veebiarenduses, sest pealt näha ilus ja korralik kodulehekülg ei tähenda paraku alati korralikku ning head veebilehte. Teadmised veebiarendusest aitavad selliseid probleeme vältida ning ennetada.

Lisaks ei suudeta läbi teadmiste puudumise, hinnata ka tööle kuluvat aega ja töömahtu. Näiteks kuulub veebiarenduse alla veel ka üks oluline faktor – testimine. Testimine tagab, et kui koduleht avaldatakse, toimib see kasutajatele mõistetavalt ning korrektselt (täpsemalt vt ptk 3 Testimine). Samas jäetakse see tihti omale kodulehte tellides või arendades tähelepanuta ja tulemuseks on vigane ja/või poolik koduleht.

Nagu näha, on kliendi lihtsatel soovidel palju puudusi, millega tegelikult tuleb arvestada, kui soovetakse omale korrektset kodulehte. Omale kodulehte tellides peab seljuhul lootma, et arendaja või agentuur, kelle poole pöörduakse, on professionaalne ning viitab ja suunab klienti õiges suunas, mitte ei aja ainult raha taga. Järgmistes peatükkides antakse ülevaade sellest, kuidas omale kodulehte luues väljatoodud probleemidele läheneda ning millised võiksid olla lahendused.

2.2. Domeen ja veebimajutus

Omale domeeni ja veebimajutust ostes on aspekte, millele mõelda palju. Domeeni poolelt, tuleb valida nimi, mis seostub loodava kodulehaga kõige paremini, aga nagu peatükis 1.2

Domeen räägitud, siis ei pruugi see tänapäeval enam kõige lihtsam olla. Veebimajutusel tuleb mõelda sellele, millised on loodava kodulehe arhitektuurilised nõuded ehk mida peab veebiserver toetama, et koduleht töötaks korrektselt. Lisaks peaks olema ka HTTPS protokoll tugi, tagamaks süsteemi turvalisust (Basques, 2017). Veel peaks mõtlema ka kasutuses olevatele rahalistele ressursidele, nii domeeni registreerimise kui veebimajutuse hinnad on vastavalt teenuspakkujale erinevad.

Antud projekti puhul kasutatakse WordPressi, mille viimase versiooni 4.7.2 soovitatavad nõuded veebiserverile on järgnevad (Wordpress):

- PHP versioon 7 või kõrgem;
- MySQL versioon 5.6 või kõrgem VÕI MariaDB versioon 10.0 või kõrgem;
- HTTPS tugi.

Nendest nõuetest ning kliendipoolsest nõudest registreerida domeen (sisult tegelikult ka veebimajutus), sai registreeritud kliendile sobiv domeen ning tellitud virtuaalserver Zone.ee teenuspakkujalt. Nende virtuaalserveri pakett I¹³ sobib edukalt antud nõuetele – omades piisavalt ketta- ja piiramatult andmeedastusmahtu, pakkudes tasuta SSL-i ja emaili kontosid. Lisaks on ka paketi hind vastavuses pakutava teenusega, makstes aastase arveldusperioodi korral 4,80€ kuus.

Samas tuleks omale kodulehte arendades mõelda, kas kohe on vaja osta domeen ja veebimajutus. Alguses saab kodulehte edukalt arendada ka lokaalses arvutis ning selle jaoks on olemas erinevaid programme erinevate operatsioonisüsteemide jaoks. Lokaalselt kodulehe valmis tegemine ning seejärel produktsiooni versiooni Interneti ülespanek on ka veebiarenduse standard. Täpsemalt WordPressi jaoks lokaalse keskkonna ülespaneku ning selle vajalikkusest saab lugeda nende kodulehelt¹⁴.

2.3. Sisuhaldussüsteemi paigaldamine

Sisuhaldussüsteemi (vt ptk 0) WordPress paigaldamiseks on mitmeid viise. Enamus teenusepakkujaid võimaldavad WordPressi paigaldada serverisse mõne üheklikki paigaldaja (ingl *one click installer*), nagu näiteks Installatron¹⁵, Fantastico¹⁶ või Softaculous¹⁷ abil, mis

¹³ <https://www.zone.ee/et/teenus/virtuaalserver/hinnad/>

¹⁴ <https://developer.wordpress.org/themes/getting-started/setting-up-a-development-environment/>

¹⁵ <http://installatron.com/>

¹⁶ <https://netenberg.com/>

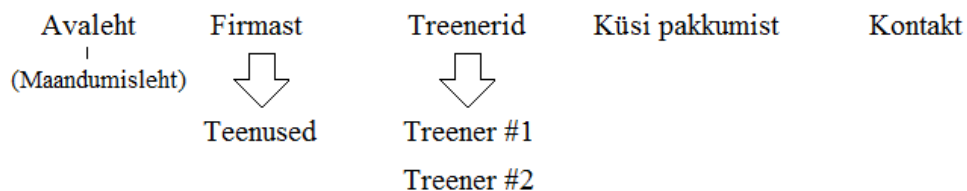
¹⁷ <https://www.softaculous.com/>

teevad taustal vajalikud tööd nagu failide serverisse paigutamine, andmebaaside loomine ja seadistamine vastavalt etteantud andmetele ise ära. Ka Zone pakub WordPressi jaoks automaatset paigaldamist ning seda ka antud projekti puhul kasutati.

Lisaks saab WordPressi paigaldada ka käsitsi. WordPressi käsitsi paigaldamise lihtsus oli samuti üks põhjuseid, miks on tegu tänapäeval niivõrd populaarse sishaldussüsteemiga. WordPress nimetab seda viie minuti paigalduseks (Wordpress) ning enne automaatseid tarkvara paigaldajaid oli niivõrd lihtne ja kiire süsteemi paigaldamine selge eelis konkurentide ees. Selleks, et käsitsi WP-i paigaldada, tuleb süsteem allalaadida ning lahti pakituna serverisse õigesse kohta paigutada. Seejärel tuleb luua andmebaas ning seadistada konfiguratsiooni fail. Viimasena tuleb jooksutada WP installeerimiskript. Detailsem õpetus, kuidas manuaalselt WordPressi paigaldada, on leitav nende dokumentatsioonist¹⁸.

2.4. Lehtede struktuur

Kliendi üks nõuetest oli, et koduleht peab vastama kindlale struktuurile. See on kirjeldatud Joonis 2, kus on näha et kodulehel on viis erinevat põhilehte ning kahel neist on olemas ka alamlehed. Ehk süsteem peab toetama lehtede hierarhiat, mis on WordPressi sisseehitatud funktsionaalsus ning saavutatav läbi Lehtede (ingl *Pages*) mooduli.



Joonis 2. Lehtede struktuur.

WordPressiga tuleb kohe kaasa ka teine oluline moodul Postitused (ingl *Posts*), kuid siinkohal on oluline saada aru nende kahe mooduli funktsionaalsest ja eesmärgilisest erinevusest. Postituste moodul võimaldab luua näiteks uudiste või blogi süsteemi, mida on võimalik siduda Siltidega (ingl *Tags*) ning Rubriikidega (ingl *Categories*), mis omakorda võimaldavad filtreerimist ning postituste grupeerimist. Kasutades Lehe moodulit uue lehe loomisel, on see konkreetse teema või sisuga lehekülg, mis võib küll omakorda jaguneda vastava teemaga seotud alamlehtedeks. Näiteks on Joonis 2. näha, kuidas konkreetset teemat sisaldav leht

¹⁸ https://codex.wordpress.org/Installing_WordPress

Treenerid jaguneb omakorda alamlehtedeks, kus iga alamlehega tutvustatakse mõnd treenerit lähemalt.

Kliendipoolsetest nõuetest (vt Tabel 1) selgus, et lehel peab olema ka kontaktivorm ning selle jaoks planeeriti kasutada Contact Form 7 liidest. Tegu on WordPressi ühe populaarseima liidesega, omades üle 3 miljoni aktiivse paigalduse (Miyoshi). Tänu oma populaarsusele on süsteemile loodud ka palju erinevaid laiendusi, mis lisavad erinevaid funktsionaalsusi – nt mitmejärgulise vormi loomis võimalus või PayPal'i tugi. Miinuseks võib lugeda kasutajaliidese keerukust, ehk esmakasutamisel tundub süsteem keeruline ning vajab süvenemist, et õigesti ära kasutada kõiki võimalusi, mida pakutakse. Samas pole õpifaktor nii suur, et see segaks süsteemi kasutamist ning sellele viitab ka kasutamise populaarsus.

Lisaks kuvatakse ka maandumislehel (milleks on Avaleht) oma tegemiste kohta uudiseid. Selle funktsionaalsuse saavutamiseks kasutatakse ära Postituste moodulit ning kuna WordPress pakub avalehe jaoks erinevaid lahendusi (staatiline leht või viimased postitused), ei pea funktsionaalsuse implementeerimiseks muud tegema, kui avalehe kujundusmallis enda soovide järgi kujundama.

2.5. Kujundusmalli loomine

Kujundus on kodulehe üks olulisemaid osi, moodustades koostöös lehel oleva sisuga ühe terviku, millena kodulehte kasutajatele presenteeritakse. Kujundus on üks neist faktoreist, mille põhjal luuakse kodulehest esmamulje ning tehakse otsus, kas soovitakse jätkata kodulehe kasutamist. WordPress pakub oma kodulehe endale sobivaks muutmiseks mitmeid võimalusi.

Nagu peatükis 1.4 Sisuhaldussüsteemid öeldud, on WP tuhandeid tasuta või tasulisi kujundusmalle, mille seast endale sobivaim valida. Olemasolevate kujunduste kasutamine on populaarseim valik omale kodulehte tehes, kuna paljud kujundused lubavad vabat kasutamist ehk selle saab omale vajalikuks kohandada. See tähendab, et leides oma soovidele vastava kujunduse ning tehes mõningad muudatused, saab tavakasutaja lihtsa vaevaga endale soovitud välimusega kodulehe.

Olemasolevate kujunduste täiendamisel tuleb meeles pidada, et kui kujundusmallis tahetakse koodi juures midagi muuta, siis tuleks luua kujundusmallist tütar-teema (ingl *child theme*), millega saab algse kujundusmalli faile üle kirjutada. Kuidas seda täpsemalt teha, saab lugeda

WordPressi dokumentatsioonist¹⁹. Selline lähenemine säilitab võimaluse algset kujundust uuendada, kui sellest on tulnud välja uus versioon, ilma kaotamata endapoolseid täiendusi.

Kuid nagu ka iga liidese puhul, mis kodulehele paigaldatakse, lisab ka kolmanda osapoole poolt loodud kujundusmall võimalikke turvariske, millest räägitakse lähemalt peatükis 2.6.1 Liidesed ja kujundusmallid. Alati tuleks tutvuda ka kujundumalli litsentsiga, et teada kas ja mida ikkagi võib muuta ning millised on näiteks kaasa tulevate piltide kasutustingimused. Sellise käitumisega välditakse võimalikke tulevaid probleeme ning lisa kulusid, kui litsentsi vastu on eksitud.

Teine variant omale soovitud kujunduse saamiseks, on see ise luua. Seda varianti kasutati ka kliendile kodulehe tegemisel. WordPressile on palju trafaretseid (ingl *boilerplate*), oma kujunduse loomiseks mõeldud, kujundusmalle. Nende seast saab valida omale parima, vastavalt selle kujundusmalli pakutavatele ning endal vaja minevate võimalustele. Samas ei pea aluseks võtma ühtegi kujundusmalli ning võib luua kõik ise nullist, kuid enamasti kiirendavad trafaretsed kujundusmallid arendust.

Kuna ka kliendi soov (vt Tabel 1) oli, et koduleht toimiks kõikidel ekraanidel, siis selle jaoks võeti aluseks Understrap kujundusmall²⁰, mis sisaldab endas WP kujundusmallide loomiseks paljukasutatud Underscore ning kohanduva veebi ja mobiil-esimesena lähenemise populaarseimat raamistikku Bootstrap (Understrap). Understrapil on kohe olemas ka tütar-teema, seega on selle peale väga mugav ehitada oma kujundusmalli.

Ise kujundusmalli luues, tuleks kindlasti seda ka testida, et tagada selle õigesti toimimine populaarseimatel ekraanisuurustel ning brauserites. Sellest räägitakse lähemalt peatükis 3. Testimine. Kuid nagu ka olemasolevate kujundusmallide kasutamise puhul, tuleb ise loodud kujundusmalli puhul tagada selle turvalisus.

2.6. Turvalisuse tõstmine

Üldisemalt sisuhaldussüsteemide vastu kasutatavatest ründamismeetoditest ning populaarseimate CMS-ide turvalisusest, on kirjutatud 2014 aastal Tallinna Ülikoolis kaitstud Mikk Lillese seminaritöös „Populaarsete veebi sisuhaldussüsteemide turvalisus“²¹ ja sama autori 2015 aastal kaitstud bakalaureusetöös „Sisuhaldussüsteemide turvalisus Harjumaa

¹⁹ https://codex.wordpress.org/Child_Themes

²⁰ <https://understrap.com/>

²¹ http://www.cs.tlu.ee/teemad/get_file.php?id=272

üldhariduskoolide veebilehtede näitel²². Mõlemad tööd sisaldavad peatükke ka WordPressi kohta – seminaritöös kirjeldatakse WP versioon 3.8 ja vanemad, bakalaureusetöös käsitletakse WP versiooni 4.1.2 ja vanemad, ning kirjeldatakse WP-is esinevatid vigu ja turvaauke. Tööd on head täiendused antud seminaritööle, kuna käsitlevad üldisemalt teemasid, mis jäävad antud seminartöö skoobist välja, kuid on siiski olulised mõistmaks üldiselt, milliseid rünnakuid tehakse sisuhaldussüsteemide vastu ning millised turvaauke neis esineb.

Alampeatükis 2.4 Lehtede struktuur, lisati kodulehele kergekäeliselt juurde üks liides ning saadi sellega edukalt täidetud üks kliendinõue. Tegelikult tuleks aga iga arenduse tegemisel või liidese paigaldamisel mõelda läbi ka selle turvakaalutus. Lisaks tuleks mõelda ka kasutajate ning nende õiguste peale, süsteemile ligipääsu piiravate rünnakute ning spämmi vastu. Viimaseks tuleks paika panna ka tegevusplaan selle kohta, mis juhtub siis, kui kõikidest pingutustest hoolimata koduleht kompromiteeritakse.

2.6.1. Liidesed ja kujundusmallid

Liideste juures tuleks mõista millist lisafunktsionaalsust antud laiendus pakub ning milline on tegelik vajadus selle paigaldamisel. Liideste puhul tuleks vaadata selle ülesehitust ning koodi, et mõista kui turvaline on antud lahendus. Samas puuduvad enamasti tavakasutajal, kes tahab saada lihtsalt tema jaoks töötavat lahendust, selle jaoks vajalikud teadmised ning ka aeg, et süveneda täpsemalt koodi. Kõikvõimalike vigade või turvaaukude vastu ei aita ka väga hästi kirjutatud kood ning siit tuleb ka järgmine aspekt millega arvestada – kui tihti liides saab uuendusi ning kui aktiivselt seda kasutatakse. Mida suurem on kasutajaskond ning aktiivsem autor, seda tõenäolisem on ka see, et leitud vead või turvaaugud saavad vajalikud parandused. Seega tuleks kasutada ainult aktiivselt kasutuses olevaid liideseid.

Olemasolevate kujundusmallide kasutamise juures on samuti heaks märgiks, kui kujundusmalli aktiivselt uuendatakse, kuid kujundusmallid enamasti ei saa nii tihti uuendusi kui liidesed. Samas tuleks nii liidese kui ka kujundusmalli juures vaadata, kuidas tagatakse andmete kasutamisel ning kuvamisel turvalisus. Andmekäsitluse juures tuleks lähtuda reeglist, et kõik andmed on ebausaldusväärsed ning nii sisend- kui väljundandmeid peab valideerima

²² http://www.cs.tlu.ee/teemad/get_file.php?id=357

ja puhastama (Langston, 2012). Oma kujundusmalli luues, tasuks lugeda läbi ka WordPressi dokumentatsioon²³ kujundusmallide turvalisuse kohta.

2.6.2. Kasutajaõigused ja admin kasutaja

Liidesed on aga ainult üks osa WordPressi kodulehest ning turvalisuse juures on veel palju, mille peale mõelda. Üks enam levinumaid vigu, mis WordPressi installeerimisel tehakse, on esimese ehk administraatori rollis oleva kasutaja nimeks „admin“ määramine. Õnneks on WordPress lõpetanud selle soovitamise vaikeväärtusena installeerimise protsessis²⁴, kuid siiski on just see üks enam levinud meetodeid jõuvõtte rünnaku puhul, kuidas WordPressi lehele ligipääseda (Nicholson, 2016). Ehk siis, selle meetodi vastu esmane pääsetee on esimese kasutaja (ID 1) nime ära muutmise või veel parem oleks uue kasutaja loomine, kellele administraatori õigused antakse ning esimeselt kasutajalt nende eemaldamine. Lisaks tuleks paigaldada WordPressi liides nimega WP Limited Login Attempts²⁵, mis tagab selle, et lühikese aja jooksul ei saa tuhandeid kordi järjest sisselogimist proovida. See aitab 90% selliste rünnakute vastu.

Parim meetod administraatori ligipääsupunkti rünnakute vastu on nende teistkordne parooliga kaitsmine. Eestis pakub veebiserverite teenuspakkujatest seda vaikimisi näiteks Radicenter (Radicenter). Selle puhul lisatakse WP administraatori paneeli sisselogimisele veel üks autentimiskiht, millega tagatakse ka see, et jõuvõtetel sissepääsu üritades ei kasutata ära serveri ressursse ning omakorda ei tekitata teenuse keelamise rünnakut (ingl *Denial of Service Attack*). Täpsem õpetus, kuidas adminisitraatori ligipääsu punkte parooliga kaitsta, on leitav WPBeginner kodulehelt²⁶.

2.6.3. Spämm

Peamine lähtepunkt, kuskohast WordPressi kodulehti spämmitakse on kommentaariumid. Peale värsket paigaldust, on WP lehel kommenteerimine lubatud ning siinkohal tulebki mõelda, kas see funktsionaalsus kodulehel on vajalik ja millist väärtust see kodulehele lisab.

Juhul kui tegu on näiteks blogiga, uudistega või mõne muu sarnase süsteemiga, kus siiski kommenteerimist vaja on, pakub WordPress ka lahendusi, kuidas spämmi vastu võidelda.

²³ <https://developer.WordPress.org/themes/theme-security/>

²⁴ <https://core.trac.WordPress.org/ticket/24078>

²⁵ <https://WordPress.org/plugins/wp-limit-login-attempts/>

²⁶ <http://www.wpbeginner.com/wp-tutorials/how-to-password-protect-your-WordPress-admin-wp-admin-directory/>

WordPressi kaasa tulevad, süsteemsed lahendused, on võimalus määrata, kui palju linke ühes kommentaaris on lubatud, sõnade musta nimekirja loomine ja erinevad modereerimise seaded.

Lisaks tuleb värske WP paigaldusega kohe kaasa ka spämmi vastu võitlev liides Akismet, kuid see pole vaikimisi aktiveeritud. Akismet kasutab keerukat algoritmi koostöös kogukonna poolt loodava andmebaasiga, tuvastamaks millised kommentaarid kvalifitseeruvad spämmi alla ning millised mitte (Wordpress). Akismetile lisaks on WP-l veel palju spämmi filtreerimiseks mõeldud liideseid, mis võimaldavad rohkem enda postituste tüübile vastavamat kommentaariumit seadistada.

Veel üks aspekt, millele tasuks mõelda, kui tegeletakse kommentaariumi ja spämmi probleemiga, on pilvelahendusi kasutavad kommentaarumid, nagu nt Disqus²⁷. See tähendab, et kommentaarid ei ole salvestatud lokaalselt enda veebiserveris, vaid asuvad pilves. Selleks pead oma kodulehe siduma vastava teenusepakkuja API-ga ning seejärel toimib kommentaarium vastavalt teenusepakkuja süsteemile. See tähendab, et spämmiga võitlemine, kommentaaride hoiustamine ja kuvamine jääb välise süsteemi hallata. Samas luuakse oma kodulehele üks sõltuvus välise süsteemiga.

Teine punkt, kuidas WordPressi lehti spämmitakse on veebilehele linkide lisamine, mille eesmärgiks on levitada pahavara või saavutada paremaid kohti otsigumootori tulemuste lehtedel. Sellist rünnakut on võimalik läbi viia tänu ebaturvaliste liideste, kujundusmallide kasutamisele, aga ka omades ligipääsu serverile või administraatori kontole. Sellisel puhul ei tea 90% veebisaidi omanikest, et nende koduleht on kompromiteeritud, kuna lingid lisatakse veebilehel loogilistesse kohtadesse või lisatakse nad tavakasutaja jaoks nähtamatult (Paulick, 2012). Sellise rünnaku vastu aitab eelmistes alampeatükkides välja toodud administraatori kasutaja, liideste, kujundusmalli ning ka FTP kontode turvalisuse tagamine. Lisaks saab oma kodulehte kontrollida Google „Safe Browsing Site Status“²⁸ kontrollis, kus on näha, kas Google robotid on leidnud kodulehelt pahavara.

2.6.4. Tegevusplaan

Olles omalt poolt teinud kõik, et tagada oma kodulehe turvalisus, saab küll kergendatult hingata, kuid ei saa öelda, et koduleht on 100% turvaline ja kunagi midagi ei juhtu. Alati jääb võimalus, et inimliku vea, tehnoloogiate arengu või mõne muu faktori tulemusena koduleht kompromiteeritakse, sest kõike ei ole võimalik ennetada. Sellisel juhul on oluline omada

²⁷ <https://wordpress.org/plugins/disqus-comment-system/>

²⁸ <https://www.google.com/transparencyreport/safebrowsing/diagnostic/index.html>

tegevusplaani, et oma koduleht võimalikult kiirelt tagasi tööle saada ning seda minimaalsete kaotustega.

Tegevusplaani juures on oluline mõista, kus on kodulehe suurimad väärtused ning need prioritseerida. Vastavalt nendele prioriteetidele peaks toimima ka info varundamine. Oma kodulehe varundamisest ja selle varukoopiast taastamisest räägitakse lähemalt peatükis **Error! Reference source not found. Error! Reference source not found.** Samas on oluline ka teada märkidest ja signaalidest, mis viitavad sellele, et kodulehte on rünnatud. Veel tuleks mõelda selle peale, kuidas kompromiteeritud koduleht karantiini panna ning hiljem, peale rünnaku meetodi tuvastamist, kuidas võimalik truvaauk või pahavara eemaldada. Täpsemalt saab lugeda erinevatest kompromiteerituse indikaatoritest ning järgmistest sammudest, mis tuleks ettevõtta, WordPressi kodulehelt²⁹.

Antud seminaritöö raames loodud koduleht näitab, et tegevusplaan ei pea olema väga keeruline. Kuna kodulehele väga tihti uut sisu ei looda, piisab regulaarsetest varundustest, et säilitada viimast loodud sisu. Lisaks paigaldati ka WordFence³⁰ liides, mis lisab teatud turvameetmeid ning võimaldab ka rakendusetasandil kodulehe skanneerimist. Veel pandi paika ka konkreetse kodulehe ning veebimajutuse spetsiifilisemad sammud, nagu teenusepakkujaga suhtlemine, kodulehe karantiin ja taastamine. Kindlasti pole väljatoodud näited kõikehõlmavad ning tegevusplaani koostadas peab lähtuma konkreetsest kodulehest ning selle vajadustest.

2.7. Otsingumootoritele ja sotsiaalmeediale optimeerimine

Robotid moodustasid 2015 aastal Globaalse Robotite Liiklusraporti põhjal 48,5% kogu Internetis toimuvast liiklusest (Zeifman, 2015). Ehk on oluline mõista, et robotid on suur osa kodulehe veebiliiklusest ning ka neile on vaja koduleht arusaadavaks teha, et tagada omale parem tulemus erinevates otsingutes ning ilusam kuvamine linkide jagamisel. Põhjus, miks peaks tegelema nii otsingumootorile kui sotsiaalmeediale optimeerimisega, on lihtne – tuua oma kodulehele liiklust.

²⁹ https://codex.wordpress.org/FAQ_My_site_was_hacked

³⁰ <https://wordpress.org/plugins/wordfence/>

2.7.1. Otsingumootoritele optimeerimine

Otsingumootoritele optimeerimine (ingl *Search Engine Optimisation*, lühend *SEO*) on eraldi veebiarenduse haru. See sisaldab endas erinevaid strateegiaid, mille eesmärkideks on, nagu nimi ütleb, kodulehe paremad tulemused erinevate otsingumootorite süsteemides (Wishpond).

See, mis lehti millised robotid võivad külastada on kirjeldatud robots.txt failis, mis peaks enamasti asuma veebiserveri avaliku kausta juurikataloogis. Samas on siin oluline mõista, et halvad robotid, kes näiteks võivad otsida kodulehele nõrkusi või emaili aadresse spämmimiseks, võivad robots.txt faili ignoreerida ning seega ei ole robotite liikumise piiramine kodulehel väga kerge (Robotstxt.org). Enamus otsingumootorite roboteid, mis indekseerivad kodulehti, et määrata nende otsitulemusi, on aga head robotid ning järgivad ja austavad robots.txt-sse kirjutatud reegleid.

Seega, olles määranud ära, milliseid osasid kodulehel saab indekseerida, tuleb tegeleda sellega, et need oleksid otsingumootorite silmis võimalikud head ning väärksid kõrgemat kohta tulemuste lehel. Selle jaoks peaksid kodulehel olevad lingid olema lihtsasti loetaval kujul – mitte `veebinimi.ee/otsing.php?q=saabas` vaid näiteks `veebinimi.ee/otsing/saabas`. Lisaks peaks kodulehel olema sisukaart ning iga leht peaks sisaldama kindlat teemat. Lehel olevad pildid peaksid olema õige nimega, omama pealkirja ning alltekste, faili suurus peaks olema viidud miinumumi ning pilt peaks olema seotud lehel oleva tekstiga. Veel tuleb kasuks, kui teised otsingumootorite poolt usaldusväärseteks märgitud lehed lingivad kodulehele. Samas ei tohiks üritada ka otsingumootorite roboteid ülekavaldada, lisades kodulehele väga palju märksõnu või kasutades lingifarme (Robotstxt.org). Kuid on veel väga palju aspekte, millele tuleks mõelda SEO-ga tegelemisel ning väga hea artikkel WordPressile SEO tegemiseks on kirjutatud Yoasti tiimi poolt³¹.

WordPressi süsteem on SEO mõistes peale värsket paigaldust üsnagi võimekas. Kuid siiski on kohti, mida saab järele aidata ning projekti raames loodava kodulehel paigaldati SEO raames WP liidesed Yoast SEO³² ja Simple Sitemap³³. Veel optimeeriti kodulehe piltide kasutamist, mille kohta saab täpsemalt lugeda samuti Yoasti kodulehel avaldatud artiklist³⁴. Lisaks tuleb mõista ka seda, et mobiilitoega veebilehed, saavad Google otsingus kõrgema tulemuse (Google Support). Seega on kohanduva veebi printsiip (vt ptk 1.3 Kohanduv veeb), millest

³¹ <https://yoast.com/WordPress-seo/>

³² <https://et.WordPress.org/plugins/WordPress-seo/>

³³ <https://et.WordPress.org/plugins/simple-sitemap/>

³⁴ <https://yoast.com/image-seo/>

lähtuti antud kodulehe kujundusmalli loomisel, ka otseselt SEO tööriist, muutes kodulehe mobiili sõbralikuks.

Lisaks paigaldati ka kodulehele WP liides Google Analytics³⁵, mis võimaldab kodulehel Google Analyticsi (lühend GA) kasutamist. GA võimaldab saada informatsiooni kodulehe kasutajate koht – kuidas jõuavad kasutajad koduleheni ning nende käitumist kodulehel. Veel pakub GA erinevat statistikat ja raporteid lehe külastajate kohta, mis võimaldab analüüsida kodulehe efektiivsust.

2.7.2. Sotsiaalmeediale optimeerimine

Kuna sotsiaalmeedia osakaal inimeste Interneti tarbimises ja käitumises, on järjest kasvamas, on ka oluline kodulehe sotsiaalmeediale optimeerimine (ingl *Social Media Optimization*, lühend *SMO*). SMO võib olla osa SEO-st, aga on ka iseseisvalt oluline osa sellest, kuidas jõuab liiklus koduleheni ning kui teadlik ollakse kodulehe eksisteerimisest. SMO eesmärk on luua sidet brändi (klient või konkreetne koduleht) ja potentsiaalsete klientide vahel ning tugevdada brändi, olles nähtav erinevates võimalikes kliendi poolt tarbitavates sotsiaalkanalitest, millega omakorda kasvatatakse kodulehe liikulust ning müüki (Wishpond).

Projekti jaoks loodi ka ettevõttele Facebooki leht ning koduleht seoti sellega läbi FB-i Like Buttoni³⁶, mis lisab kodulehele FB-i lehe oma FB-i konto meeldivate lehtede sekka lisada. Lisaks võimaldab WP-i Yaost SEO liides seadistada, millisena kuvatakse kodulehe linkide jagamist erinevates sotsiaalmeedia kanalites. Veel anti kliendile suuniseid, kuidas siduda koduleheküljel toodetavat sisu FB-i postitustega ning kuna teha postitusi.

Kindlasti tuleks siinkohal oma kodulehte luues mõelda, millistele sotsiaalmeedia kanalitele fokuseerida, kuna erinevaid võimalusi on palju ning lihtsalt seisvad ja tühjad kontod sotsiaalmeedias, ei täida SMO eesmärki, pigem isegi kahjustavad seda näidates brändi ebaaktiivsena. See, milliseid sotsiaalmeedia kanaleid valida, sõltub valdkonnast, teemadest ning sihtgruppidest ja ka olemasolevast ressursist (tööjõud ning aeg), mida on võimalik sotsiaalmeedia haldamise alla paigutada – nt valiti selle kodulehe puhul ainult Facebooki kasutamine.

³⁵ <https://wordpress.org/plugins/googleanalytics/>

³⁶ <https://developers.facebook.com/docs/plugins/like-button>

2.8. Puhverdamine

Vahemälu (ingl *cache*) on mälu, mida kasutatakse sagelikasutatava ning hiljutise admete salvestamiseks. Kodulehe brauseris puhverdamine (ingl *caching*) tähendab lehel olevate piltide, sisu ja failide salvestamine kasutaja brauseri vahemällu. Mis omakorda tähendab, et kui kasutaja avab tihti kasutatud lehe või faili, on see juba brauserisse salvestatud ning lokaalsest mälust lugemine toimub kiiremine kui lehe uuesti kokkupanek WordPressi poolt (Ansari, 2016).

Veel on olemas ka serveri poolne puhverdamine, mille puhul lihtsamad süsteemid puhverdavad serveri lokaalsel kettal, kuid keerulisemad süsteemid kasutavad mitmeid eraldi vahemälu servereid. Selline vahemälu kasutamine on kasutusel veebisaitidel, millel on suurem liiklus, süsteemide keeruka implementeerimise tõttu (WordPress).

WordPressil pole vahemälu kasutamine sisseehitatud tuuma võimalustes, vaid see tuleb saavutada kasutades liideseid. Loodud kodulehe puhul võeti kasutusele W3 Total Cache³⁷ liides, millel on palju erinevaid puhveradmise ja seadistuse võimalusi.

2.9. Varundamine

Varundamine (ingl *backup*) on kodulehtede puhul äärmiselt oluline aspekt, mis paraku pahatihti ei saa piisavalt tähelepanu planeerimise ja teostuse faasis. Mistahes, kui stabiilne ja hästi on seadistatud riistvara või tarkvara, tuleb ikka ette vigu ja seda nii inimlikul kui tehnilisel tasandil. Misläbi jääb alati oht, et olulised andmed lähevad kaotsi (Toova, 2013). Lihtne varunduse süsteem tagab ka kindluse, kodulehe turvaliseks uuendamiseks.

Kahjuks ei ole WordPressi tuumas mingilgi kujul varundamine esindatud, seega tavaliselt mõistetakse varunduse olulisust alles siis, kui juba on hilja ning kahju on juba tehtud. Selleks, et seda vältida tuleks kodulehe planeerimisel ka mõelda sellele, milliseid andmeid varundada, kuhu seda teha ning kui tihti. WordPressi puhul on kuldreegliks, et varundusi tuleks teha regulaarselt, vastavalt sellele, kui tihti luuakse uut sisu, ning kindlasti enne igat uuendust (WordPress).

Varunduse tegemiseks on erinevaid viise – manuaalselt, automaatselt, ainult mingitest kindlatest osadest koossev, täielik või ainult andmebaasi varundus. Wordpressi puhul peab tegema varunduse nii failidest kui andmebaasist, et kogu süsteem oleks kaetud. Veel üks

³⁷ <https://et.wordpress.org/plugins/w3-total-cache/>

oluline aspekt varunduse puhul on see, kuhu loodud koopia salvestatakse. Kui seda tehakse samale kettaruumile, kus asub ka varundatav süsteem, siis jääb oht, et keegi pääseb kettaruumile ligi ning kustutab süsteemi, koos loodud varundustega ja ollakse tagasi nullpunktis. Samas, kui andmed salvestatakse nt pilve, siis tuleks mõelda, milliseid andmeid sinna salvestatakse, sest oma anded usaldatakse kellegi kolmanda hoolde. Sellest, kuidas täpsemalt WordPressi varundamist tagada, saab täpsemalt lugeda selle kohasest WP dokumentatsioonist³⁸.

Veel tuleks läbi mõelda ning teha läbi ka varundusest taastamise protsess, et näha kas tegelikult ka kõik vajalik saab varundatud ja kas taastamisel esineb mingeid tõrkeid. Kui loodud varundused on vigased ja neid peaks vaja minema, aga keegi neid katsetanud pole, ollakse sisuliselt taaskord nullpunktis, kus on küll olemas kasutatud koopiad, kuid kodulehte taastada ei anna (ehk siiski suurema tahtmise puhul annab neist midagi kätte saada, seega seis on juba siiski natukene parem kui olla täitsa ilma varukoopiateta).

Loodud kodulehel paigaldati WP liides UpdraftPlus WordPress Backup Plugin³⁹, mis võimaldab luua varundusi mõlemaist, nii failidest kui andmebaasist ning seda salvestada pilve ja lisaks ka kodulehte varukoopiast taastada. Antud juhul salvestatakse varukoopia Dropboxi⁴⁰ pilveteenusesse ning süsteem loob iga kolme päeva tagant uue varukoopia versiooni. .

2.10. Uuendamine

WordPressi puhul on erinevaid komponente, mida uuendatakse palju. Aga nagu eelmises alampeatükis öeldud, tuleb enne igat uuendust luua varukoopia – seda nii andmebaasist kui failidest. Võib küll tunduda, et koduleht toimib praegu ilusti ning miks peaks üldse riskima uuendustega ning millegi ära lõhkumisega?

Võimalikud komponendid, mille uuendamise peale tuleks kodulehe ehitamisel mõelda, on WordPressi tuum, liidesed ja kujundusmallid. Põhjuseid, miks neid uuendatakse on erinevaid. Alates kõge olulisematest turvauuendustest (mis tuleks paigaldada nii kiirelt kui võimalik) kuni edasi arenduste ning vigade eemaldamiseni välja. Seega, et hoida oma kodulehte turvalisena ning funktsionaalsena, tuleb seda paratamatult uuendada.

Veel tuleb tagada, et uuendustega ei kirjutata üle või kaotata vajalikku funktsionaalsust, mis on võimalik kui arendades on kasutatud valesid võtteid. Lisaks kui kodulehel on kasutusel

³⁸ https://codex.wordpress.org/WordPress_Backups

³⁹ <https://wordpress.org/plugins/updraftplus/>

⁴⁰ <https://www.dropbox.com/>

puhverdamine (vt ptk 2.8 Puhverdamine), tuleks enne uuenduse tegemist see välja lülitada, sest see mõjutab kodulehe töötamist. Nii WP tuuma kui liideste ning kujundusmallide uuendamine on tehtud iseenesest väga lihtsaks ning seda saab teha läbi WP administraatori paneeli, täpsem juhend selle kohta on leitav themeisle artiklist „How to update WordPress safely“⁴¹. Peale vajalike uuenduste tegemist, tuleks puhverdamine tagasi sisse lülitada ning kontrollida, kas kõik töötab nii nagu varem (siin tuleb kasuks ka automaattestide olemasolu). Juhul kui kõik töötab korrektselt ning mingeid muudatusi pole vaja teha, tuleks teha süsteemist ka uus varukoopia, et varunduses oleks olemas ka uusim versioon.

Projekti raames loodud kodulehe arendamisel peeti uuendamist meeles, kui loodi ise kujundusmalli, kasutades Understrapi tütarmaat. See võimaldab emateemat (ingl *parent theme*) uuendada ilma kaotamata kodulehe jaoks loodud funktsionaalsust ning välimust.

⁴¹ <https://www.elegantthemes.com/blog/tips-tricks/how-to-update-wordpress-safely-every-time>

3. Testimine

Testimine on üks olulisemaid osi tarkvara arenduse protsessist. Testimisel võib olla erinevaid eesmärke – leida süsteemist vigu, valideerida süsteemi nõuete põhiste töötamist või teada saamine, kas süsteem toimib nii, nagu kasutajad seda eeldaksid (ISTQB Exam Certification).

Käesoleva seminaritöö raames loodud kodulehel toimus arenduse käigus pidevalt funktsionaalsuse testimine autori ning kliendi poolt. Samas ei piisa edukaks testimisest ainult sellest, kui süsteemi katsetavad selle arendusega seotud inimesed, sest nemad teavad, kuidas süsteem toimib ning kuidas peaks seda kasutama, et saavutada soovitud tulemust. Lisaks ei loodud süsteemi toimise kontrollimise tagamiseks ka ühtegi automaattesti.

Kodulehel viidi läbi arenduse käigus mitte kaasatud inimestega kasutajakogemuse testimine, milles osales seminaritöö kirjutamise ajal kaks inimest. Läbi viidud testimise miinuseks on valimi väiksus, mille põhjuseks on autori ajalise ressursi puudus, kuid siiski leiti testimise käigus kohti, kus kodulehte parandada ning täiendada, mis muutis testimise edukaks.

Osalejatele anti ette kolm stenaariumid, mille alusel nad pidid teostama kodulehel teatud tegevused või leidma vajaliku info. Testimise põhjal selgus, et koduleht on üldiselt mõistlikult ülesehitatud ning kasutaja jaoks arusaadav. Samas leiti mõningaid vigu lehtede toimises kui ka sisu poolest tekstilisivigu:

- Avalehel olev Teenuste nupp suunas valesti;
- Teenuste lehel kirjavead;
- Ühe teenuse kirjeldav pilt tekistas segadust.

Kõik leitud vead ja parandused dokumenteeriti ning võetakse arvesse edasiarendustes.

Omale kodulehte tellides või arendades tuleb arvestada ka testimisele kuluva ajalise ja materiaalse ressursiga, mis kulub selle peale. Kuidas kirjutada ja milline näeb välja üks hea testilugu, on kirjeldatud TestLodge kodulehel avaldatud artiklis „How to Write Test Cases for Software (with a sample)”⁴². Antud testimise korral läks koos kolme stenaariumi selgitamise, tegevuse jälgimise ja dokumenteerimise peale autoril ühe inimese kohta ligikaudu 25 minutit. Ajakulu sõltub süsteemist, testlugudest ning inimesest, kes süsteemi testib. Seega tuleb igale testi korrale läheneda indivuaalselt, sest igal testimisel peab olema süsteemi arengu mõistes väärtust lisav põhjus.

⁴² <https://blog.testlodge.com/how-to-write-test-cases-for-software-with-sample/>

Kokkuvõte

Kuigi WordPress on äärmiselt populaarne sisuhaldussüsteem, puudub siiski konkreetne dokument, mis sisaldaks nii teooriat kui praktilisi näiteid WordPressile korraliku lehe loomiseks. Käesoleva töö eesmärgiks oli anda teoreetilist teadmist selle kohta, mida on vaja kodulehe tegemiseks, kuidas ning miks kasutada WordPressi sisuhaldussüsteemi ja tutvustada, millised on praktilised võtted süsteemi kasutamises kodulehe arendamisel.

Kliendinõuete analüüsist peatükis 2.1 Ülevaade projektist, selgus et omale kodulehte tehes või tellides, omades selleks puudulikke teadmisi, on loogiliseks tulemuseks poolik kodulehekül. Seega kui peale antud seminaritöö läbi lugemist jääb tunne, et WordPressiga korraliku kodulehe tegemine käib siiski üle jõu ning poolik lahendus ei ole aksepteeritav (mida ta ei tohiks peaaegu kunagi olla), on mõistlik võtta ühendust mõne professionaaliga, kes tagab töö kvaliteedi ning kodulehe hoolduse ka peale selle valmimist. Kuid kindlasti annab töö läbi lugemine kindlust asumaks kliendi poolele selles suhtes ning nõudmaks omale kvaliteetset kodulehte.

Töö esimeses peatükis antakse teoreetiline ülevaade sellest, mida on vaja ühe kodulehe tegemiseks – domeeni, veebimajutust ning soovitavalt sisuhaldussüsteemi kasutamist, koos kohanduva kujundusega. Teises peatükis kirjeldatakse konkreetse kliendi soovide baasil, kuidas luua WordPressil põhinevat head kodulehte. Antakse ülevaade sellest, millele tuleks mõelda ning millised on võimalikud WP põhised lahendused. Kolmandas peatükis kirjeldatakse lühidalt üldiselt testimist ning ka loodud kodulehe testimiseks tehtud testikorda ja selle tulemusi.

Antud temaatikal võiks edasi uurida, millised on võimalused WP-le automaatsete loomiseks liideste ja kujundusmallide puhul. Veel võiks täpsemalt uurida erinevaid puhverdamise võimalusi, kuna vahemälu kasutamine loob märgatava efekti kodulehe laadimiskiirusele.

Kasutatud kirjandus

- Ansari, R. (17. June 2016. a.). *6 Best WordPress Caching Plugins Compared*. Allikas: DesignBombs: <https://www.designbombs.com/top-wordpress-caching-plugins-compared/>
- Basques, K. (9. February 2017. a.). *Why HTTPS Matters*. Allikas: Google Developers: <https://developers.google.com/web/fundamentals/security/encrypt-in-transit/why-https>
- Built With. (kuupäev puudub). *CMS Usage Statistics*. Allikas: Built With: <https://trends.builtwith.com/cms>
- CommonPlaces. (20. June 2013. a.). *Why is WordPress so popular?* Allikas: CommonPlaces: <http://www.commonplaces.com/blog/why-is-wordpress-so-popular/>
- Ellingwood, J. (28. January 2015. a.). *Apache vs Nginx: Practical Considerations*. Allikas: DigitalOcean: <https://www.digitalocean.com/community/tutorials/apache-vs-nginx-practical-considerations>
- Gil, P. (16. November 2016. a.). *What Exactly Is a 'Domain Name'? How Does It Work?* Allikas: LifeWire: <https://www.lifewire.com/what-is-a-domain-name-2483189>
- Google Support. (kuupäev puudub). *Make sure your site's ready for mobile-friendly Google search results*. Allikas: Google Support: <https://support.google.com/adsense/answer/6196932?hl=en>
- ISTQB Exam Certification. (kuupäev puudub). *What is Software Testing?* Allikas: ISTQB Exam Certification: <http://istqbexamcertification.com/what-is-a-software-testing/>
- Langston, M. (30. October 2012. a.). *Secure Your WordPress Theme*. Allikas: ThemeShaper: <https://themeshaper.com/2012/10/30/secure-your-wordpress-theme/>
- McGill, J. (10. November 2015. a.). *Responsive Images in WordPress 4.4*. Allikas: WordPress Make: <https://make.wordpress.org/core/2015/11/10/responsive-images-in-wordpress-4-4/>
- Miyoshi, T. (kuupäev puudub). *Contact Form 7*. Allikas: Wordpress: <https://et.wordpress.org/plugins/contact-form-7/>

- Nicholson, J. (30. December 2016. a.). *WordPress wp-login.php brute force attack*. Allikas: Inmotion Hosting: <http://www.inmotionhosting.com/support/edu/wordpress/wp-login-brute-force-attack>
- Paulick, C. (22. September 2012. a.). *How to Identify and Fix a Hacked WordPress Website*. Allikas: ManageWP: <https://managewp.com/how-to-identify-and-fix-a-hacked-wordpress-website>
- Qmee. (8. July 2014. a.). *Online in 60 seconds [Infographic] – A Year Later*. Allikas: Qmee: <http://blog.qmee.com/online-in-60-seconds-infographic-a-year-later/>
- Radicenter. (kuupäev puudub). *Kodulehekülgede turvalisus*. Allikas: Radicenter: <https://www.radicenter.ee/et/Kodulehekulgede-turvalisus>
- Robotstxt.org. (kuupäev puudub). *About /robots.txt*. Allikas: Robotstxt.org: <http://www.robotstxt.org/robotstxt.html>
- Robotstxt.org. (kuupäev puudub). *How do I get the best listing in search engines?* Allikas: Robotstxt.org: <http://www.robotstxt.org/faq/bestlisting.html>
- Roesler, P. (5. January 2015. a.). *Will New Top Level Domains Matter in 2015?* Allikas: Inc: <http://www.inc.com/peter-roesler/will-new-top-level-domains-matter-in-2015.html>
- Smashing Magazine. (12. January 2011. a.). *Responsive Web Design: What It Is And How To Use It*. Allikas: Smashing Magazine: <https://www.smashingmagazine.com/2011/01/guidelines-for-responsive-web-design/>
- Zeifman, I. (9. December 2015. a.). *2015 Bot Traffic Report: Humans Take Back the Web, Bad Bots Not Giving Any Ground*. Allikas: Imperva Incapsula: <https://www.incapsula.com/blog/bot-traffic-report-2015.html>
- Toova, T. (15. July 2013. a.). *Andmete varundamine ja taastamine*. Allikas: www.cs.tlu.ee/windows/Windows_op_systeemid_konspekt_12_loeng.pdf
- Understrap. (kuupäev puudub). Allikas: Understrap: <https://understrap.com/#page>
- W3Techs. (2. March 2017. a.). *Usage of content management systems for websites*. Allikas: W3Techs: https://w3techs.com/technologies/overview/content_management/all

- Verisign. (19. July 2016. a.). *Internet Grows to 326.4 Million Domain Names in the First Quarter of 2016.* Allikas: Verisign:
<https://investor.verisign.com/releaseDetail.cfm?releaseid=980215>
- Wishpond. (kuupäev puudub). *SEO vs. SMO: What's the Difference, and Why Should You Care?* Allikas: Wishpond: <http://blog.wishpond.com/post/67575277462/seo-vs-smo-whats-the-difference-and-why-should-you>
- Wordpress. (kuupäev puudub). *Combating Comment Spam.* Allikas: Wordpress:
https://codex.wordpress.org/Combating_Comment_Spam
- Wordpress. (kuupäev puudub). *Installing WordPress.* Allikas: Wordpress:
https://codex.wordpress.org/Installing_WordPress
- Wordpress. (kuupäev puudub). *Requirements.* Allikas: Wordpress:
<https://wordpress.org/about/requirements/>
- WordPress. (kuupäev puudub). *WordPress Backups.* Allikas: WordPress:
https://codex.wordpress.org/WordPress_Backups
- WordPress. (kuupäev puudub). *WordPress Optimization/Caching.* Allikas: WordPress:
https://codex.wordpress.org/WordPress_Optimization/Caching