Kvaliteedistandardid gotoAndPlay-s lõppkasutaja vaatenurgast

Viimati muudetud 27. mai 2022.

NB! Dokumenti sisu koostamine on veel pooleli

Juhend on mõeldud eelkõige gotoAndPlay poolt loodavate **veebilehtede kvaliteedistandardite defineerimiseks**. Loetletud nüansid on nn lõppkasutaja vaatenurgast ja valideerides enda tööd selle dokumendi vastu, saad olla kindel, tulemus on tipp-topp!

Kvaliteedistandardi seadmine jaguneb tinglikult kolme valdkonda:

1. Korrektsus
2. Ligipääsetavus / UX
3. Jõudlus
4. Turvalisus

NB! Tegemist on lõppkasutaja vaatenurgast sõnastatud nõuetega, mis tähendab, et see ei kata kõiki vajalikke gotoAndPlay kvaliteedistandardeid (nagu nt arendusprotsesside järgimine, code review, linterid, automaattestid jms).

Dokumendi sisukord:

TLDR; versioon

Nimekiri, mille vastu saab kiiresti valideerida (soovi korral *copy-paste* taski alla). Täpsem kirjeldus all tabelis.

- [ ] Korrektsus
  - [ ] Productionis ei tohi süsteemi veateateid lõppkasutajale näidata (PHP warnings & notices, console.log jms)
  - [ ] Arendus- ja stagingkeskkond ei tohi olla avalikult kättesaadav. Ligipääsu erandid üle IP / agentide.
  - [ ] Arendus- ja stagingkeskkond ei tohi olla indekseeritav robotite poolt.
  - [ ] Igal projektil peab olema projektile favicon

- [ ] Ligipääsetavus (mh ka *accessibility*)
    - [ ] Productionis peab olema otsingumootorid lubatud.
    - [ ] Productionis on paigaldatud võimalused hallata SEO seadistusi (WordPressi projekti puhul RankMath või Yoast SEO plugin).
    - [ ] Cross-browser & cross-device testide läbimine.
    - [ ] Productions kõik assetid kättesaadavad ilma paroolita (st urlid ei tohiks viidata staging ja/või arenduskeskkonda).
    - [ ] Lähtu gotoAndPlay veebilehtede ligipääsetavuse üldistest suunistest. Vastavalt projekti nõuetest võib-olla peab veenduma, et oleks täidetud WCAG 2.1 A või AA nõuded.
    - [ ] Veendu, et projektis ei esine enam levinud UX erroreid
- [ ] Jõudlus ehk PageSpeed Insights skoor 90+. Täpsemalt tähendab see prioritiseeritud järjekorras:
    - [ ] Lehte serveeritakse üle cache / SSR.
    - [ ] Kasutatakse gzip'i ja efektiivset cache poliitikat staatilistel assetite serveerimiseks (Cache TTL 30+ päeva, max aasta)..
    - [ ] Lehel kasutatud scripid (JS) ja CSS on uglified / minified.
    - [ ] Lehel kasutatakse scripte vastavalt vajadusele. Võimaluse korral defferred / lazyloaded.
    - [ ] Väldi skriptide poolt tekitatud renderduse blokki. Vajadusel loo CSS'iga vastavad laadimise olekud või kuvad, mis saab ehanced kui script laetud.
    - [ ] Kasuta responsive images'i.
    - [ ] Sisus veendu, et kasutatakse optimaalseid failiformaate: JPG vs PNG või SVG vs PNG.
    - [ ] Lazyloadi pildid ja videod (sh kolmanda osapoolse videoplayerid).
    - [ ] CDNi kasutamine, et kasutada next-gen failiformaate.
    - [ ] Font swap display
    - [ ] Kasuta vajadusel DNS prefetch'i / preload'i
- [ ] Turvalisus
    - [ ] Wordfence või muu WAF lahendus
    - [ ] Uploads kaustas PHP init keelata
    - [ ] HTTPS!

The long version...

Jrk	Kategooria	Kirjeldus
	Korrektsus	Productionis ei tohi süsteemiveateid (näiteks PHP warnings & notices või JavaScript console.log) lõppkasutajale näidata. See aga ei tähenda, et neid ei peaks koguma logidesse - “tulevane sina” tänab, kui on vaja nende põhjal debugida intsidente.
	Korrektsus	Arendus- ja stagingkeskkond ei tohi olla avalikult kättesaadavad ega indekseeritavad. See üheltpoolt võib kehvematel juhtudel põhjustada Google topelt indekseerimist ja originaallehe sisu eemaldamist otsingumootorist. Lisaks võib seeläbi jõuda lõppkasutajatena sisu, mis antud ajahetkel ei ole avalik. Ligipääsu piiramiseks piisab `.htaccess` piirangust. Ligipääsu erandid on võimalik luua üle IP! Näiteks gotoand.dev kaustades olev sisu üldiselt piiratud koos teatud eranditega.
	Korrektsus	Igal projektil peab olema favicon. Kui see puudu, siis küsi disainerilt või projektijuhilt. Defineeri vajalikud Faviconi suurused kasutades https://realfavicongenerator.net

	Ligipääsetavus / UX	Production keskkonnas tuleb veenduda, et robotitel on olemas ligipääs lehele ja võimalik seda indekseerida. Tüüpjuhul tähendab see, et robots.txt ei ole root (/) ligipääs keelatud ja lehe päises puudub robotite indekseerimist keelav meta tag (`<meta name="robots" content="NOINDEX, NOFOLLOW">`).
		
		WordPressi puhul näeb tüüpiliselt korrektne robots.txt selline:
		
		`User-agent: *`
		`Disallow: /wp-admin/`
		`Allow: /wp-admin/admin-ajax.php`

	Ligipääsetavus / UX	Production keskkonnas oleks hea, kui robots.txt sisaldaks viidet Sitemapile (WordPressis kasutades Yoast SEO või RankMath pluginat). Valideerimiseks veendu, et robots.txt faili lõpus oleks viide vastavale index failile. Näide:

	`Sitemap: https://playandnope.com/sitemap_index.xml`
Ligipääsetavus / UX	Projektis tuleks kokku leppida põhimõtted, mis ulatuses arendaja testib oma tehtud tulemust. Mis browserites jms. Näiteks võib-olla nõue, et enne Code Review-sse suunamist peab arendaja oma töö üle valideerima ühel mobiilsel seadmel (native telefon) ning teises op. süsteemi brauseris. Oluline testimisel - Inspect mode valideerimine mobiilse seadme emuleerimiseks on üldiselt petlik ja ei ole soovitatav! Päris seadmetega testimiseks kasuta meie device lab'i või browserstack.com teenust (Lastpassis kasutajakonto).
Ligipääsetavus / UX	Digiligipääsetavus võimaldab võimalikult paljudel kasutajatel veebisisu tarbida. Ligipääsetavus tähendab rohkem õnnelikke kliente, kvaliteetsemat veebilehte ning ka paremat leitavust otsingumootorites. Põhimõtted, mida tuleb alati järgida: https://accessibility.twn.ee/ -> **Üldised põhimõtted**. Soovitatav on olla kursis (ja võimalusel jälgida ka teisi peatükke). Juhul kui projektis on sõnastatud konkreetsed nõuded ligipääsetavusele, siis lähtu ametlikest juhenditest (nt WCAG 2.1 või EN 301 549 standarditest).
Ligipääsetavus / UX	Õige struktuuri märkimiseks kasuta vastavaid HTML5 struktuurielemente ja ARIA landmark rolle. Näiteid HTML stuktuurielementidest <header> päise esitamiseks, <nav> menüü jaoks, <footer> jaluse märkimiseks jne. Näiteid ARIA landmark rollidest: banner, search. jt. Loe lähemalt
Ligipääsetavus / UX	Lisaks soovitatav veenduda, et meil levinud nüansid oleks üle vaadatud: ➔ Input teksti suurus vähemalt 16px. Vastasel juhul tekib onFocus "sisse zoomimine" iOS seadmel. ➔ Veendu, et desktopis / mobiilis ei esineks horisontaalselt scrolli (erandiks on laiad tabelid, kaardid, pildid jms). Seda on eriti lihtne jätta märkamata OSX seadmetel hidden scrollbaridega. ➔ Peidetud elemendile tuleks anda CSS-is nii `display: none;` kui ka `visibility: hidden;` stiilid, et ükski ekraanilugeja ei näeks peidetud

	elementi. Ainult ekraanilugejate eest saab elemendi peita aria-hidden="true" atribuudiga. ➔ Igal lehel peaks olema täpselt üks <h1>, mis väljendab lehe põhisisu. ➔ Pealkirjadena <h1> kuni <h6> ei tohi esitada teksti, mis pole tegelikult pealkiri ➔ Vormid peavad olema klaviatuuriga (TAB'iga) navigeeritavad (loogilises järjekorras) ja vorm submititav. ➔ TODO: koguda veel mõtteid siia!
Jõudlus	Veebilehe laadimiskiiruse üheks suuremateks pudelikaelaks on andmebaasipäringute tegemine lehe renderdamise eel (mugav viis päringute monitoorimiseks üle Query Monitori plugina). Alati ei ole seda vaja teha ja võimalik selle asemel serveerida eelnevalt renderdatud vaadet. Seejuures on võimalik luua ka nö hübriidlahendusi, kus osa sisu on cached ja osa päritakse asünkroonselt pärast lehe kuvamist. Üldine rusikareegel on et server peab vastama < 200ms jooksul ja vältima üleliigseid HTTP suunamisi. WordPressi lehtedel saab selle enamasti lahendada WP Rocket Cache abil, kuid ilma cacheta ei tohiks ka lehe serveri request põhjendamatult kaua võtta! Reacti (avalikes veebides) tuleks kasutada Server Side Renderingu (ehk SSR). Üks paremaid moodusi selleks on kasutada Next.js Reacti frameworki. Loe edasi
Jõudlus	Kasuta http-cachet ja gzip-i, mis tuleks seadistada serveri konfiguratsiooni tasemel (Apache serveris saab kasutada selles `.htaccess` faili). WordPressi lehtedel kasutame WP Rocketi poolt loodud `.htaccessi` faili, teistel juhtudel tuleb veenduda, et see oleks mõistlikult lahendatud. Loe lähemalt siit ja siit
Jõudlus	Kasuta HTTP/2-i. Võimalik, et teenuspakkuja iseteenindusest on see vaja eelnevalt sisse lülitada (AWS jmt setupi korral suhtle DevOpsiga). Loe lähemalt: https://web.dev/performance-http2/
Jõudlus	Preloadi ressursid, mida sa tead, et on vaja (link rel=preload). Loe lähemalt.
Jõudlus	Kasuta optimaalseid meediaformaate ja suuruseid.

	On elementaarne, et veebilehel on kasutusel piltide renderdamiseks nn lazyload meetodit (eelista native `loading="lazy"`, mis fallbackib javascript meetodile), mis tähendab pildid & taustavideod väljaspool viewporti ei tohiks alla laadida enne kui neid on vaja. Seejuures ei tohi ka liiga agaraks minna – above the fold pilte ei ole alati mõtet *lazyloadida*! Video playerit tuleks initida alles vajutusel. Enne seda kuvada placeholder pilti ja play ikooni. Piltide renderdamiseks peab seadme ekraani põhiselt providema sobivad suurused (kasutades srcset ja picture meetodeid). Seejuures on oluline jälgida, et ei serveeriks pildi suurt originaalfaili. Kui võimalik, siis eelista JPG formaati PNG ees. JPG formaat on reeglina väiksem kui pildil on palju värve. PNG kasutus õigustab end ainult olukordades kus pildil on vähe värve ja on vaja läbipaistvat tausta. Eelista vektorgraafikat rastrile. Ikoonid / logod võiksid olla SVG formaadis. Kui võimalik, siis serveeri pilte next-gen formaadis (WebP). Võimalik, et on vaja kasutada kolmanda osapoole teenust selleks (CDN), vt järgmist punkti. Loe lähemalt siit ja siit
Jõudlus	Juhul kui eelmine punkt on lahendatud ja ei anna piisavalt võitu, siis kasuta (tasulist) CDN teenust. Cloudinary või Cloudimage, et serveerida pilte (ja staatilisi faile) optimaalsemalt. Loe lähemalt.
Jõudlus	Optimeeri tekstipõhised assetid (CSS, JS jne) kasutades content-specific minify-d & uglify-d. Loe lähemalt siit ja siit
Jõudlus	Veendu, et lehel ei ole katkiseid viiteid lehel. Mõistlik oleks seda teha läbi automaatse protsessi. WordPressi puhul saad jookustada Broken Link Checker pluginat. Seda võiks vähemalt korra enne LIVE teha.
Jõudlus	Väldi JavaScript ja CSS above-the-fold sisu renderdamise blokeerimist. Abiks on Critical CSS ja skriptide defer. Loe lähemalt WordPressi projektides on meil võimalik WP rocket plugina abil Critical CSS-i sisselülitamine. See vajaks eelnevalt aga testkeskkonnas valideerimist ja

		vajadusel kohendamist.
	Jõudlus	Võimalda teksti nähtavus ka hetkel, mil webfont veel laeb. Kasuta `font-display: swap;` Loe lähemalt
	Jõudlus	Vähenda kolmanda osapoole skriptide impacti lehe laadimisel. Eemalda need võimalusel või lae kolmanda osapoole skripte hiljem (async või defer), kui primaarsed asjad on laetud. Variant on ka kasutada link rel=preconnect või rel=dns-prefetch Loe lähemalt
	Turvalisus	Productionis on seadistatud WAF. Wordpressi puhul kasuta näiteks Wordfence pluginat. TODO: tegelikult peaks tulemüüri seadistust puudutama siin (mil.ee on hea näide)
	Turvalisus	Uploads kaustas ei tohi olla võimalik PHP-d jooksutada
	Turvalisus	Faili õigused TODO Esiteks tuleb jälgida, et konfiguratsioonifailid (nt .env) ei ole avalikult loetavad (õigustegrupp peaks olema 440).
	Turvalisus	Täna ei ole ühtegi mõjutav põhjust (peale laiskuse), miks FTP accessi oleks vaja.. Arenduskeskkonna ja LIVE keskkonna uuendamine läbi Bitbucket Pipelines-i. Kõikidel muudel juhtudel serveri access üle SSH, mis on tagatud võtmetega (mitte parooliga).
	Turvalisus	Lehel peab olema HTTPS sertifikaat. Tasuta Let's Encrypt on enamike hostingute poolt toetatud, devops oskab seda ka seadistada (vaja seadistada automaatne uuendamine).
Valideerimine		Kasuta Google Page Insights tööriista, et valideerida tulemust.

Desktop & mobiili performance skoor peab olema 90+. Mõjuval põhjusel (see peab olema kirjalik!) võib see olla alla 90, kuid kindlasti mitte alla 70! Vajadusel konsulteeri projekti PD-ga.

Nipid ja trikid. Kasulik lugemine.

Tõenäoliselt võiks olla selline sektsioon ka!

1. Konfigureeri oma IDE tegema asju, mida ei peaks ise manuaalselt tegema. Soovitatav on kasutada IDE-t, mida ka teised ümberringi kasutavad, et saada kasu nippidest ja trikkidest. Back-end / full-stack arendajatele soovitame PhpStormi ja front-enderitele Visual Studio Code-i.
    a. Spell check (et vältida näpukaid)
    b. Auto formatting
    c. Webhint (VisualStudio Code, PHPStorm)
2. Automatiseeri aspektid, mis on mõistlikud
3. Anna tagasisidet disainerile / projektijuhile arendaja vaatenurgast - kust on võimalik optimeerida koostöös osapoolte vahel.
4. Küsi nõu PD-lt ja/või #general chatis Slackis
5. Optimeerimisi peaks terve projekti vältel silmas pidama, mitte jätma kõik projekti lõppu ja avastada, et “noorena sai tehtud valesi otsuseid”.
6. Kasuta Hint-i (https://webhint.io)
7. Kasulikud juhendid ja selgitused high perfomance veebilehe jaoks: https://developers.google.com/speed
8. Google Lighthouse weighting logic (v3 ja v5)
9. Hea ja ülevaatlik eestikeelne veebi ligipääsetavuste suunised: https://accessibility.twn.ee/#
10. Me oleme veebimajutuse ametlik partner – kui klient ostab teenuse kasutades koodi **PLAY**, saavad nad seda 3 kuud tasuta kasutada.