

Opinnäytetyö (AMK)

Tietojenkäsittely

2021

Axel Lahti

WEB-PALVELUN KEHITTÄMINEN TV-SARJOJEN JAKSOARVIOINTIIN WORDPRESSILLÄ

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietojenkäsittely

2021 | 48 sivua

Axel Lahti

WEB-PALVELUN KEHITTÄMINEN TV-SARJOJEN JAKSOARVIOINTIIN WORDPRESSILLÄ

Tämä opinnäytetyö käsittelee ja seuraa yhteisöllisen WordPress-alustalla luodun palvelun prototyypin kehittämistä. Opinnäytetyönäni loin palvelun, jossa käyttäjät voivat jakaa suosituksia siitä, mitkä jaksot kannattaa ohittaa ajan säästämiseksi. Tällaisen palvelun kehittäminen on ajankohtaista, koska suoratoistopalvelut ovat vaikuttaneet tapaan, jolla sarjoja kulutetaan.

WordPress on avoimeen lähdekoodiin pohjautuva verkkosivujen kehitysalusta, jolla yleisemmin toteutetaan kotisivuja ja blogeja, jotka päivittyvät manuaalisesti ylläpitäjien avulla. Tässä opinnäytetyössä kehitettiin dynaamisesti päivittyvä ja yhteisöllisyyttä tukeva palvelu. Palvelu tukee sarjojen tehokasta kuluttamista ja säästää käyttäjien aikaa.

Opinnäytetyössä toteutettu palvelu kehitettiin käyttäen WordPress-alustaa ilman ohjelmointia tai koodia. Sivuston toiminnallisuudet toteutettiin WordPressin lisäosilla ja kaikki muokkaus tapahtui verkkoselaimessa pelkkiä visuaalisia käyttöliittymiä käyttäen. Erilaisia WordPressin lisäosia ja käyttöliittymiä käytettiin, joista käytetyin oli Elementor sen visuaalisen sivunmuokkaimen monipuolisuuden takia.

Työn lopputuloksena saatiin teknisesti toimiva verkkopalvelun prototyyppi, jossa on kaikki tarvittavat toiminnallisuudet, joita tämän kaltaisessa sivustossa tulisi olla. Prototyyppiä voi käyttää jo sellaisenaan, mutta sen ensisijainen tarkoitus on näyttää, että tämän kaltainen palvelu on toteutettavissa.

ASIASANAT:

Web-suunnittelu, www-suunnittelu, verkkosuunnittelu, ohjelmointi, prototyypit, WordPress, verkkokehitys, WordPress- lisäosat.

BACHELOR'S / MASTER'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information Technologies

2021 | 48 pages

Axel Lahti

DEVELOPMENT OF TV-SERIES EPISODE EVALUATION SERVICE USING WORDPRESS

The objective of this thesis was to develop a prototype for community driven website created with the WordPress platform. As my thesis I created a service where users can share what episodes of a TV-show can be skipped to save time. Such service is relevant as streaming has affected the way series are consumed.

The service created for this thesis is developed using the WordPress platform. One of the goals was to refrain from writing code during the development of the site. Functional features on the site are created with WordPress plugins and all modifications on the site are to be made with frontend visual editor.

The result is technically functional webservice prototype which provides all required functionalities that this kind of site should have. The prototype is functional and useable as is, but primary function of the site is to demonstrate that this kind of service is doable with WordPress and without coding.

KEYWORDS:

Web Design, Programming, Prototypes, WordPress Development, Web Development, WordPress Plugins.

SISÄLTÖ

KÄYTETYT LYHENTEET	5
1 JOHDANTO	6
2 ONGELMAN MÄÄRITTELY	8
3 WORDPRESS-ALUSTAN JA TOTEUSTYÖKALUJEN ESITTELY	10
3.1 Elementor	11
3.2 Tietokanta	11
4 SIVUSTON TOTEUTUS	13
4.1 Visuaalinen suunnittelu	13
4.2 Verkkosoitteen tekeminen ja WordPressin asentaminen	16
4.3 WordPress-sivuston kehittäminen vaiheittain	18
4.4 Sivujen kehittämisen aloitus ja etusivun kehittäminen	19
4.5 Sarjakohtaisen sivuston kehittäminen	22
4.6 Yläpalkki ja alapalkki	25
4.7 Käyttäjien tuottaman sisällön mahdollistaminen	27
5 TESTAUS JA TEKNISTEN ONGELMIEN KORJAAMINEN	33
6 LOPPUPÄÄTELMÄT	42
6.1 Asioita, joita olisin voinut tehdä toisin	43
6.2 Muuta pohdintaa ja jatkokehitysideoita	44
LÄHTEET	46

KÄYTETYT LYHENTEET

BW	Binge Watching (Merikivi ym. 2019)
CSS	Cascading Style Sheets (W3Schools 2016)
Fomo	Fear of Missing Out (Milyavskaya ym. 2018)
HTML	Hyper Text Markup Language (W3Schools 2016)
HTTPS	Hyper Text Transfer Protocol Secure (digicert 2021)
OLED	Organic Light Emitting Diodes (OLED-info 2021)
SSL	Secure Sockets Layer (digicert 2021)
WP	WordPress (WordPress.org 2021)

1 JOHDANTO

Vuonna 2021 suuri osa television ja kaapeli-TV:n tarjonnasta on siirtynyt verkkoon. Palvelut, kuten Netflix, Amazon Prime ja HBO Nordic, ostavat lisenssejä sarjoihin ja näyttävät niitä palveluissaan omien tuotantojensa ohella. Tämä on mahdollistanut sen, että suurin osa merkittävistä tai ainakin suosituimmista TV-sarjoista on katseltavissa jossakin suoratoistopalvelussa (JustWatch 2021). Suoratoistopalvelut ovat mahdollistaneet myös monien sarjojen nousun kulttimaineeseen. Sarjat, kuten *Twin Peaks* (Cohen 2017) tai *Arrested Development*, eivät olleet alkuvaiheessa tuottavia, mikä johti niiden peruuttamiseen. Nyt molemmat sarjat ovat saaneet uuden elämän ja uusia kausia on tuotettu, kiitos suoratoistopalveluiden (Walters 2021).

DVD-settien ja suoratoistopalveluiden noustessa suosioon on kehittynyt uusi tapa katsoa sarjoja. Maratoonaaminen tai bingettäminen eli ”Binge watching” (BW) tarkoittaa sarjan ”ahmimista” eli sarjan monen jakson katsomista yhdellä tai muutamalla sessiolla (Merikivi ym. 2019, s. 3—7). Kuten John Jurgensen (2012) kertoo *Wall Street Journalin* artikkelissaan, tämä tapa on muuttanut sen, miten ihmiset katsovat ja kokevat sarjoja.

Koska sarjat ovat suoratoistopalveluiden ansiosta helposti saavutettavissa, ne ovat myös populaarikulttuurin keskeinen osa. Niin sosiaalisessa kuin perinteisessäkin mediassa sarjoista jaetaan videopätkiä, meemejä ja viittauksia, jotka vain teoksen nähneet ymmärtävät (Cheded 2017). Tämä voi aiheuttaa FOMO (fear of missing out) -reaktiota eli ulkopuoliseksi jäämisen pelkoa sosiaalisesta kanssakäymisestä ja ajankohtaisista aiheista (Milyavskaya ym. 2018). Monet sarjat vaikuttavat populaarikulttuuriin, ja niihin viitataan usein esimerkiksi sosiaalisessa mediassa ja meemeillä. Mikäli näitä viittauksia ei henkilö pysy yhdistämään, voi se aiheuttaa heissä FOMO-reaktiota.

Ulkopuolisuuden tunnetta välttääkseen monet ihmiset, itseni ja kaveriporukani mukaan lukien, kokevat, että ajankohtaisten ja suosittujen sarjojen seuraaminen on tärkeää. Tämän työn tekee raskaaksi se, että varsinkin monta vuotta käynnissä olleet sarjat ovat pitkiä ja niiden seuraaminen vie paljon aikaa. Tässä opinnäytetyössä tarkoitukseni onkin luoda sivusto, joka vähentää sarjojen ja elokuvien kuluttamiseen käytettyä aikaa.

Opinnäytetyössä luon WordPress-alustalle rakennetun verkkosivuston. Verkkosivuston luon käyttäen ainoastaan WordPressin tukemia visuaalisia sivuston muokkaustyökaluja.

Tätä sivustoa kehitettäessä kehittäjä ei syötä sivustolle koodia, vaan kaikki sivustolle tulevat koodit tulevat suoraan visuaalisten muokkaimien ja lisäosien kautta.

Sivuston tarkoitus on nopeuttaa sarjojen kulutusnopeutta. Tätä varten sivustolle kirjautuneet käyttäjät voivat lisätä oppaita, joissa on listana sarjakohtaisesti ohitettavat jaksot. Sivuston muut kirjautuneet käyttäjät voivat arvioida oppaita ja kommentoida niitä.

Luvussa 2 alustetaan kyseisen palvelun ratkaisevaa ongelmaa sekä kerrotaan tarkemmin sen sisällöstä ja toiminnoista. Luvussa 3 esitellään työkalut ja teknologiat, joita käytetään sivuston kehittämisessä. Luvussa myös perustellaan työkalujen valinnat. Luku 4 etenee päiväkirjamaisesti kronologisessa järjestyksessä. Luku 4 käsittelee sivuston kehittämistä suunnitteluvaiheesta sivuston teknisen toteutuksen valmiiksi saattamiseen. Luku 5 keskittyy testaamiseen ja sivuston viimeistelyyn. Luvussa käsitellään sivuston testaamista, korjaamista sekä viimeistelyä selaimille yhteensopivaksi. Luvussa 6 käsitellään sivuston valmistuksessa opittuja asioita ja mitä projektissa olisi voinut tehdä toisin. Luku 7 sisältää opinnäytetyön lopetuksen ja sivuston mahdollisen jatkokehityksen suunnitelman.

2 ONGELMAN MÄÄRITTELY

Monet tv-sarjat ovat rakenteeltaan sellaisia, että osa jaksoista ei ole merkittäviä sarjan tarinan kululle (Marinescu ym. 2014). Näitä jaksoja kutsutaan täytejaksoiksi tai filleriksi. Esimerkiksi yhdessä maailman suosituimmista sarjoista, Game of Thronesissa, osa jaksoista on arvioitu Rotten Tomatoes -verkkopalvelussa huomattavasti muita huonommaksi. Esimerkiksi viidennen kauden kuudes jaksot on arvioitu noin 40 % huonommaksi kuin kauden muut jaksot (Rotten Tomatoes 2021). Tämän jakson kesto on 54 minuuttia, kun vastaavia jaksoja tulee enemmän ja kausia on useita, alkaa jaksojen summa olla jo monissa tunneissa.

Vaikka sarjojen arviointipalveluita ja -sivustoja on olemassa, en ole kokenut niitä hyödylliseksi, kun aloitan uuden sarjan katselun. Useat palvelut ohjaavat arvioimaan jaksoja laadun perusteella. Näissä palveluissa käyttäjä ei pysty erottamaan, onko kyseinen jaksotarinallisesti tarpeellinen. Mikäli koko sarja on arvioitu heikoksi tai tarinaan liittymätön jaksot on hyvin vastaanotettu, ei palvelun käyttäjä voi arvion perusteella päätellä, onko jaksotarinallisesti tärkeä.

Ratkaisuni edellä mainittuun ongelmaan on palvelu, joka helpottaa sarjojen binge watchingin nopeutta ja laatua. Rakennan selainpohjaisen palvelun, johon rekisteröidyt käyttäjät voivat itse lisätä listan sarjojen jaksoista, jotka edistävät tarinaa. Palvelu on avoin kaikille, mutta vain rekisteröidyt käyttäjät voivat listata sisältöä.

Palvelun tarkoituksena on toimia oppaana, joka tarjoaa käyttäjille kausikohtaisena listana jaksot, jotka käyttäjä voi ohittaa uhraamatta katselukokemusta. Palvelun tavoitteena on olla mahdollisimman helppolukuinen ja -käyttöinen. Tämän lisäksi sivustolla tulee sarjakohtaisesti lukea säästettävä aika. Kun käyttäjä etsii sarjaa, jota haluaisi alkaa katsomaan, hän voi avata sarjan sivun, nähdä kaikki ohitettavat jaksot ja yhteenlasketun minuutti- tai tuntimäärän ajasta, jonka hän voi säästää seuraamalla sivustolle luotua opasta.

Palveluun voi rekisteröityä ja tehdä oppaan sarjoista ja niiden jaksoista. Oppaiden tulisi olla arvioitavissa, jotta käyttäjäkunta voi määrittää, mitkä oppaat ovat vartenotettavia. Listat tulisivat arvioitavaksi yhdelle tai useammalle ylläpitäjälle.

Sivuston yleiset määrittelyt ovat seuraavat:

- käyttäjille toimivat sivut työpöydällä ja mobiilissa
- helppolukuinen ja selkeä visuaalinen ilme
- mahdollisimman relevantti hakuominaisuus
- oikean sarjan etsiminen vaivatonta
- mahdollisuus luoda tili ja kirjautua
- kyky hallinnoida omaa tiliä
- modernin näköinen
- uusimmat oppaat etusivulle
- ohjeet ja kuvaus sivustosta.

Sarjojen katsomisen listauksen ohjeet ovat seuraavat:

- rekisteröitynyt käyttäjä voi luoda oppaan jaksoista
- käyttäjän luoma opas siirtyy arvioitavaksi ylläpitäjälle
- ylläpitäjä voi hylätä tai hyväksyä oppaan
- käyttäjät ja vierailijat voivat arvioida oppaita
- käyttäjät ja vierailijat voivat kommentoida listauksia
- listauksissa on mainittava säästettävä tuntimäärä
- listauksissa on mainittava jaksot, jotka voi ohittaa
- sarjan oppaassa tuntimäärä, jonka säästää, kun seuraa kyseistä opasta.

3 WORDPRESS-ALUSTAN JA TOTEUTUSTYÖKALUJEN ESITTELY

Pohjistan tässä luvussa teknologian, jota käytän palvelun toteuttamiseen. Koska palvelun voisi toteuttaa monella eri tavalla, avaan ja selvennän, miksi päädyin toteuttamaan palvelun WordPress-julkaisualustalla ja kuinka se eroaa tavanomaisesta verkkokehittämisestä. Useimmiten verkkosivut luodaan HTML-, CSS- ja JavaScript-koodin avulla (W3C 2016). Nämä koodikielot vastaavat sivuston rungosta, visuaalisuudesta ja toiminnallisuudesta. Kyseisiä kieliä voi kirjoittaa käsikäyttöisillä koodinkirjoitusohjelmilla kuten Notepad++:lla, Visual Studio Codella ja lukuisilla muilla koodinkirjoitusohjelmilla. Useat koodinkirjoitusohjelmat tukevat myös muita kieliä HTML:n, CSS:n ja JavaScriptin lisäksi. Nämä ohjelmat myös auttavat kehittäjää eri tavoin koodinkirjoittamisessa. Ohjelmat esimerkiksi värikoodaavat koodin osia, rivittävät ja sisentävät koodia tarvittaessa ja ilmoittavat, mikäli ne huomaavat epäselvää tai ongelmallista koodia. Vaikka tällaiset ohjelmat auttavat ohjelmoijaa koodin formatoinnissa ja sen tuottamisessa, on kehittäjän silti osattava ja ymmärrettävä valittua ohjelmointikieltä hyvin, jotta lopputuloksesta tulisi toimiva.

Koodikielen opettelemisessa on omat haasteensa. Jotta kehittäjä voisi luoda pelkkää tekstiä sisältävää sivustoa monimutkaisemman sivuston, tulee kehittäjän ymmärtää koodin logiikkaa ja kieltä (Ivanova ym. 2020). Tämä on joillekin haastavampaa kuin toisille, sillä suurin osa koodinmuokkausohjelmista on tekstipohjaisia ja ne vaativat kielen opettelua.

WordPress on avoimen lähdekoodin ohjelmisto, joka mahdollistaa nettisivujen luomisen lähes kokonaan ilman koodaamista. WordPress (WP) perustettiin vuonna 2003. Se oli alun perin tarkoitettu blogien kehittämiseen, mutta nykyisin sillä pyöritetään yli 42 % koko netin sivustoista. WP on rakennettu PHP ja MySQL teknologioiden päälle, mutta se on täysin käyttäjien muokattavissa (WordPress.org 2021).

WordPressin toimivimpia ominaisuuksia on mahdollisuus ladata tarpeellisia lisäosia, jotka muuttavat tavan, joilla nettisivuja voidaan rakentaa. WordPress toimii teemojen avulla. Teemat määrittävät sivuston perusparametreja sekä alustavat pohjaa. Teemaan voi lisätä sivuja, jotka koostuvat sisältölohkoista. Vaikka WordPress on jo itsessään

melko monimuotoinen ja muokattava, puuttuu siitä vapaus, johon monet graafiset suunnittelijat ovat esimerkiksi Photoshopia ja Figmaa käyttäessään tottuneet. Tähän avuksi saapuu ladattava ilmainen lisäosa: Elementor.

3.1 Elementor

Kuten Elementor-nimestä voi päätellä, se koostuu elementeistä. Näitä voidaan sijoittaa sivulle mielivaltaisesti ja juuri sinne, minne halutaan. Tämä kaikki tapahtuu elementtejä liikuttelemalla sivun pohjan sisällä. Elementeille voidaan määrittää parametrit eli tietyt arvot, joita ne noudattavat ja niihin voidaan lisätä lähes mitä tahansa sisältöä (Elementor 2021).

Sivulle voi myös syöttää omatekoista CSS-koodia, mikäli jokin toiminallisuus ei ole täysin toimiva sellaisenaan. Elementor on melko joustava sivuston rakentamisen suhteen. Ainoa haittapuoli Elementorissa tällä hetkellä on se, että monet Elementorin toiminnoista ovat maksullisen Elementor Pron takana. Koska WordPress on avoimeen lähdekoodiin perustuva, monet toiminnot voidaan korvata ilmaisilla ladattavilla lisäosilla.

Elementor siis mahdollistaa hienojen ja monimutkaisten nettisivujen toteutuksen ilman, että koodia tarvitsee kirjoittaa yhtäkään riviä. Elementor kyllä kirjoittaa koodia, mutta se on piilotettu lisäosan visuaalisen puolen konepellin alle. Koodia ei tarvitse itse muokata, ellei niin halua.

3.2 Tietokanta

Kuten aikaisemmin on mainittu, WordPress käyttää tietokannassaan MySQL-nimistä tietokannan hallintajärjestelmää. MySQL on avoimeksi lähdekoodiksi tehty tietokannan hallinnan järjestelmä (MySQL 2021),

MySQL:n käyttö WordPressin kanssa on helppoa, sillä käyttäjän tai web-suunnittelijan ei tarvitse normaalitilanteessa koskea siihen. Palveluissa, jotka tukevat WordPressin asennusta on usein MySQL jo asennettu.

MySQL rakentuu siis projektin mukana WordPressin muokkauksien pohjalta. Tämä mahdollistaa sen, web-suunnittelijat ja kehittäjät voivat keskittyä enemmän sivuston front en-

diin eli visuaaliseen ja käyttäjälle näkyvään toiminnalliseen puoleen. Mikäli WordPressissä hajoaa jotakin eikä esimerkiksi sivujen linkitys tai rakenne enää toimi, voi sen käydä manuaalisesti muokkaamassa sivuston MySQL-tiedostosta. Koska tietokantaan tallentuu kaikki sivuston tiedot, voi sinne myös tallentaa tietoa, joka on sivuston peruskäyttäjiltä piilossa. Tämä mahdollistaa sivun kevyemmän selaamisen teknisesti ja visuaalisesti.

4 SIVUSTON TOTEUTUS

4.1 Visuaalinen suunnittelu

Aloitan suunnittelun visuaalisesta ilmeestä, sillä visuaalinen ensivaikutelma on isossa roolissa, kun ihmiset muodostavat mielipiteensä sivustosta ensimmäisten 50 millisekunnin aikana (Lindegraad ym. 2006). Sivuston alustavan ”layoutin” tai ”leiskan” eli sivuston pohjan suunnittelen käyttäen Figma-suunnittelualustaa. Se mahdollistaa verkkosivujen elementtien kaltaisten osien rakentamisen ja liikuttamisen hiiren avulla.

Olen kerännyt huomioita ja esimerkkejä visuaalisesta ilmeestä seuraavilta vastaavilta arvostelusivuilta. Sivuja, IMDb, Rotten Tomatoes, Episodi ja Roger Ebert vertaillessani huomasin, että monet modernit sivut tukevat tai ovat kokonaan siirtyneet niin sanottuun ”Dark Theme” tai ”Dark mode” -teemaan eli siihen, että suurin osa sivustosta on joko mustaa, tumman harmaata tai muita tummia sävyjä. Web-kehityksen asiantuntijoiden kuten Mobietamin mukaan (Mobiteam 2020) tämä trendi on yleistynyt ohjelmoijien myötä, kun ohjelmointieditoreihin on lisätty Dark Mode. Dark Mode on hieman kevyempi silmille verrattuna kirkkaaseen valkoiseen valoon. Tätä kautta toiminnallisuus on todennäköisesti myös siirtynyt nettisivuille ja muihin palveluihin. Trendin lisäksi Dark Modella on sanottu olevan vaikutusta laitteiden akun kulutukseen, mikäli laitteessa on OLED-näyttö tai sitä vastaava näyttö, joka osaa tarkasti säädellä yksittäisten pikselien tai pikseliosuuksien valon määrää (Chamary 2021). Vertailussa IMDb oli ainoa, jolla tummat sivut olivat oletusasetuksena.

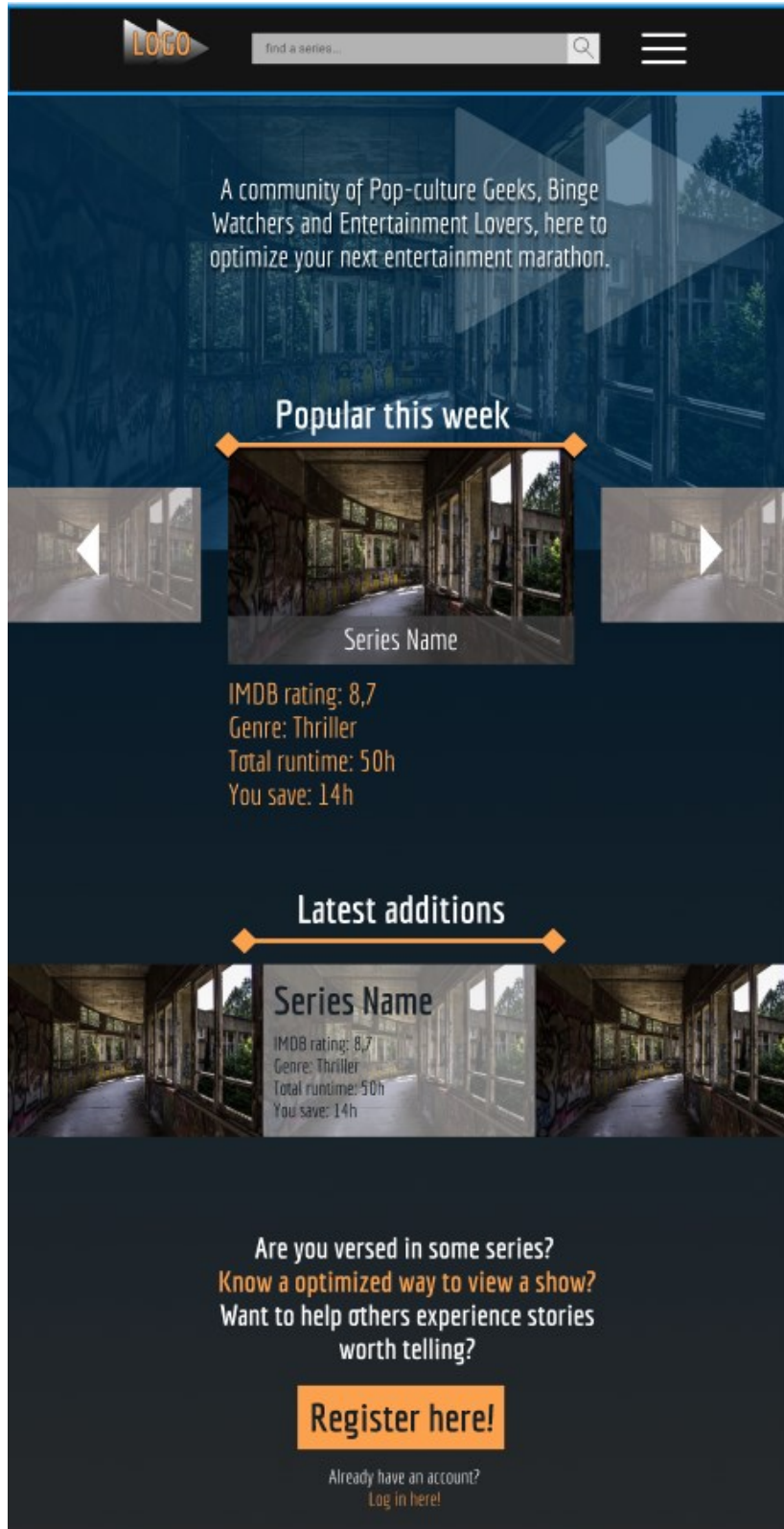
Yleisesti sivustojen etusivuilla on arvosteluja ja artikkeleja uusimmista elokuvista ja sarjoista. Vaikka sivustojen etusivut nostavat samoja asioita ja aiheita etusijalle, ovat verratut sivut melko erinäköisiä keskenään ja painottavat etusivulla hieman eri asioita. Ne sisältävät otsakkeen, jossa on hakualue, navigaatio ja sen jälkeen sivuston keskeisimmät nostot.

Vaikka sivustojen sisältö on samankaltaista, eroavat ne visuaalisesti toisistaan kuten IMDb:n tumma tausta ja yleinen olemus, jonka helposti yhdistää nykyaikaiseen elokuva-teatteri elämykseen. Roger Ebertin sivut ovat vertailun pelkistetyimmät.

Kaikkien sivustojen etusivuilla on paljon arvosteluja, listoja ja artikkeleita, jotka vaikeuttavat sivuston nopeaa silmäilyä.

Vertailun tuloksena sain selkeämmän suunnan sivustoni visuaalisesta ilmeestä. Sivustolla tulee olla otsakkeessa logo tai palvelun nimi, hakualue sekä navigaatio. Otsakkeen alle tulee visuaalisesti sävyttävä kuva, josta ilmenee palvelun kuvaus tai tavoite. Tämän jälkeen tulee osio, johon uusimmat katselulistat ilmestyvät. Uusimpien listojen jälkeen tulee osio, johon nostetaan sillä hetkellä sivuston suosituimmat ja klikatuimmat listat. Näiden kohtien jälkeen lisään syvemmän kuvauksen sivuston toiminnasta sekä painikkeen, josta voi liittyä jäseneksi. Sivuston etusivu (kuva 1) tulee tummalle taustalle, johon lisään yksi tai kaksi korostusväriä. Muut elementit ovat valkoisen ja mustan sävyisiä. Sarjojen listojen yhteyteen voi liittää mainoksen, jos sarjalle löytyy mainosvideo.

Sivuston keskeinen osa on sarjojen oppaat. Ne kertovat, miten monta jaksoa sarjassa on, mitkä niistä ovat katsomisen arvoisia ja kuinka monta tuntia käyttäjä säästää seuraamalla opasta. Sarjoja koskevat esittelyt muodostuvat haitarimaisesti, jolloin yhden kauden jaksot näkyvät valittaessa kerralla. Jotta opas pysyisi mahdollisimman lyhyenä ja nopeasti luettavana tulisi oppaan laatijan merkata ainoastaan ohitettavat jaksot ja mahdollinen lyhyt selitys jakson tapahtumista. Kullakin sarjan jaksolla tulisi olla oma osio, jossa sitä voi kommentoida tai arvostella.



Kuva 1. Suunnitelma verkkosivun asettelusta.

4.2 Verkkosoitteen tekeminen ja WordPressin asentaminen

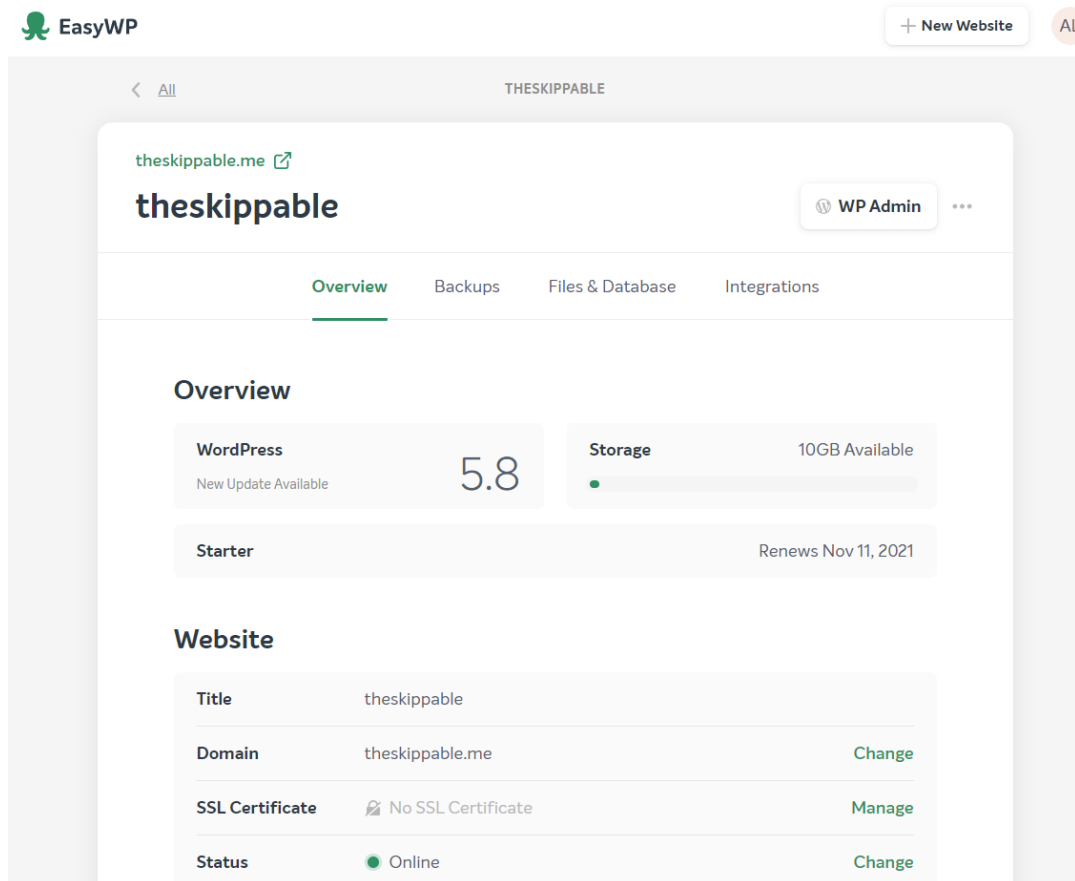
Aloitin sivun teknisen ja toiminnallisen toteutuksen etsimällä projektiin sopivan verkkosivustopalvelun. Luin muutamia artikkeleita ja päädyin valitsemaan namecheap.com palvelun. Valitsin kyseisen palvelun, sillä korkeakouluopiskelijana minulla oli mahdollisuus aktivoida GitHub.com-palvelun ilmainen opiskelijapaketti. Pakettiin kuuluu Name Cheapin ilmainen 12 kuukauden tilaus verkkotunnukseksi. Name Cheap tukee helppoa WordPressin asennusta ja on vertailun mukaan kilpailijoita nopeampi (Fitton 2019).

GitHub.com on erittäin suosittu kehitysalusta ja koodisäilö palvelu, jota käytti vuonna 2020 jopa yli 40 miljoonaa käyttäjää ja ohjelmistokehittäjää (Johnson 2020) GitHub.com ja sen työpöytäversio GitHub Desktop tarjoaa visuaalisen käyttöliittymän avoimen lähdekoodin Git-versiohallintajärjestelmälle. GitHub pyrkii tukemaan kehittäjiä ja opiskelijoita tarjoamalla oppilaitosten opiskelijoille ja opettajille useita ilmaisia ohjelmia, jotta heillä olisi pääsy parhaisiin kehitystyökaluihin (GitHub 2021). Koin järkeväksi hyödyntää tätä pakettia, jotta pystyin luomaan WP-sivuston suoraan verkkoon. Näin sain ilmaiseksi nopean sivuston, jossa on SSL-suojaus eli Secure Sockets Layer valmiina. Mikäli olisin alkanut kehittämään sivustoa paikallisesti, en voisi testata sen aitoa nopeutta. Myös SSL-suojaus olisi pitänyt lisätä myöhemmin manuaalisesti. Yksinkertaisuudessaan SSL-suojaus vaaditaan, jotta selaimet voivat todeta sivuston turvallisiksi käyttää. Aikaisemmasta WordPress ja Elementor -projektista olen oppinut sen, että mikäli SSL-suojaus ei ole kunnossa, niin tietyt Elementorin elementit, kuten vektorigrafiikkaa käyttävät .svg tiedostot eivät toimi tarkoitetulla tavalla selaimissa.

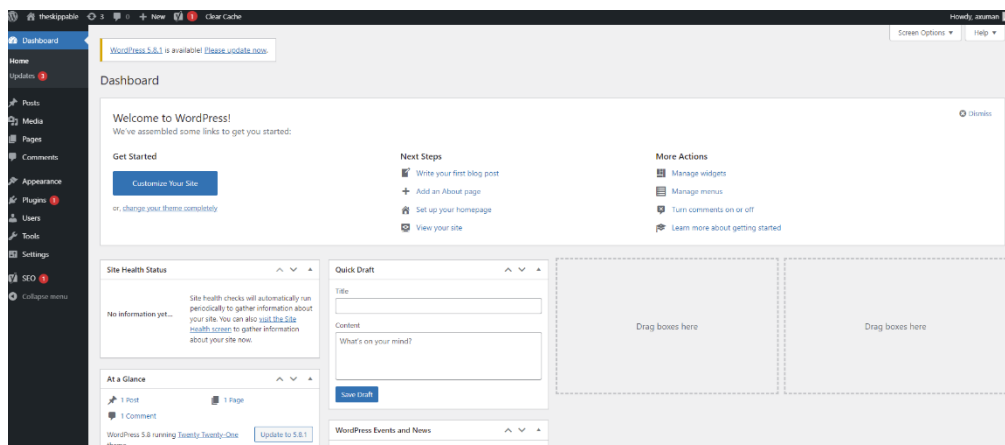
Kun olin saanut GitHubin vahvistuksen siitä, että olin oikeutettu näihin etuihin, pääsin luomaan verkkosivustoni (tunnetaan myös nimellä domain) Name Cheap -palvelussa. Loin palveluun tilin ja domain-nimen theskippable.me. Nimen keksin englanninkielisestä termistä ”skippable”, joka tarkoittaa asiaa, jonka voi ohittaa.

Kun olin saanut verkkotunnuksen ja tilin luotua, pääsin tarkastelemaan palvelun ohjauspaneelia. Huomasin, että opiskelijaetu koskikin ainoastaan verkkotunnusta eikä sivuston verkkoylläpitoa tai aiemmin mainittua WordPress-palvelun asentamista. Päätin silti tilata myös nämä palvelut ja pysyä Name Cheap -palvelussa sen lupaaman tallennustilan ja tuetun kävijämäärän sekä nopeuden takia. Ostin palvelusta halvimman paketin ja sain mukana yhden kuukauden kokeilujakson, jota hyödynnän tämän projektin teossa. Name Cheapin tarjoaman EasyWP:n asennus onnistui helposti. Kun olin valinnut tilauksen ja

täyttänyt maksutiedot, sivusto pyysi minua tekemään muutamia esivalintoja esimerkiksi WordPressiin asennettavien lisäosien suhteen. Näiden valintojen pohjalta pääsin käsiksi ohjauspaneeliin, josta näin tilaukseni tiedot, WP-version sekä painikkeen, josta pääsen WordPressin ohjauspaneeliin ja voin alkaa muokkaamaan sivustoa (kuva 2 ja kuva 3).



Kuva 2. EasyWP:n hallintapaneeli, jossa näkyy perustietoja verkkosivusta.



Kuva 3. WordPressin hallintapaneeli

4.3 WordPress-sivuston kehittäminen vaiheittain

Käyttäen EasyWP-palvelua yhdistettynä Name Cheap -palveluun pystyin helposti käynnistämään WordPressin hallintapaneelin. Ensitöikseni asensin tarpeelliset lisäosat, jotka esittelen seuraavaksi.

Kuten aikaisemmin mainitsin, **Elementor** on lisäosa, joka mahdollistaa niin sanotun vedä ja pudota -tyylisen editoinnin. Tämän ansiosta työskentely on nopeaa ja melko vaivatonta. Tärkein asia Elementorissa on, että sillä voi toteuttaa monimutkaisempia ja dynaamisempia sivustoja verrattuna WordPressin perusversioon ilman yhtään riviä kirjoitettua koodia suunnittelijalta. Elementorin kaltaisia site builder -ohjelmia on muitakin, mutta itse olen tottunut käyttämään Elementoria.

Essential Addons for Elementor tarjoaa lisää ominaisuuksia ja toimintoja Elementoriin.

Elementor Header & Footer Builder mahdollistaa ”Headerin” ja ”Footerin” rakentamisen Elementoria käyttäen. Header eli sivun yläpalkki, joka yleensä sisältää navigaation ja logon sekä Footer eli sivuston alaosio, joka yleensä sisältää sivuston yhteystiedot, omistusoikeustiedon, tietosuojalausannon sekä joissain tapauksissa sivustokartan voidaan koostaa myös WordPressin omalla teemanrakennustyökalulla, mutta tällä lisäosalla niitä voi muokata kuten sivuja Elementorilla. Lisäosan käyttö vapauttaa suunnittelijan tiettyjen muotojen ja järjestyksien rajoituksista.

Envato Elements – Photos & Elementor templates tuo Elementoriin valmiiksi tehtyjä vimpain yhdistelmiä ja -pohjia.

Under Construcion Page lisää sivustolle etusivun, joka ilmoittaa, että sivustoa vielä valmistellaan ja sinne eksynyt ei pääse tarkastelemaan keskeneräistä työtä.

Limit Login Attempts rajoittaa kirjautumisyritysten määrää ja lopulta lukitsee kirjautumisen.

Nämä lisäosat asennettuna aloin työstää sivuston runkoa. Myöhemmät tarvittavat lisäosat lataan, kun sivusto on lähempänä lopullista versiotaan. Tällöin lisään tietosuojaselosteen, evästeiden hallinnan ilmoituksen sekä tarkistan sivuston nopeuden ja suo-
jauksen.

Lisäosien lisäyksen lisäksi valitsin sivustolle Astra-teeman. Se hallitsee pohjaa, johon Elementor rakentuu sekä niitä toiminnallisuuksia tai sivuja, joita en rakenna Elementorilla. Näitä ovat suunnitelmieni mukaan sarjakohtaiset oppaat, jotka teknisesti toteutuvat WordPressissä Artikkeleina. Teemaa voin vaihtaa kesken projektin, mikäli löydän toisen teeman, jonka avulla oppaat ja oppaiden sivu on visuaalisesti miellyttävämpi. Vaikka Elementor korvaa teeman pohjan lähes kokonaan, se ei automaattisesti muokkaa Artikkelien pohjaa. Teema määrittää alustavan Artikkelin pohjan. Astra-teeman valitsin koska se oli minulle tuttu ja sen alustava Artikkelien hallinta oli sivuston suunnitelmaan sopiva.

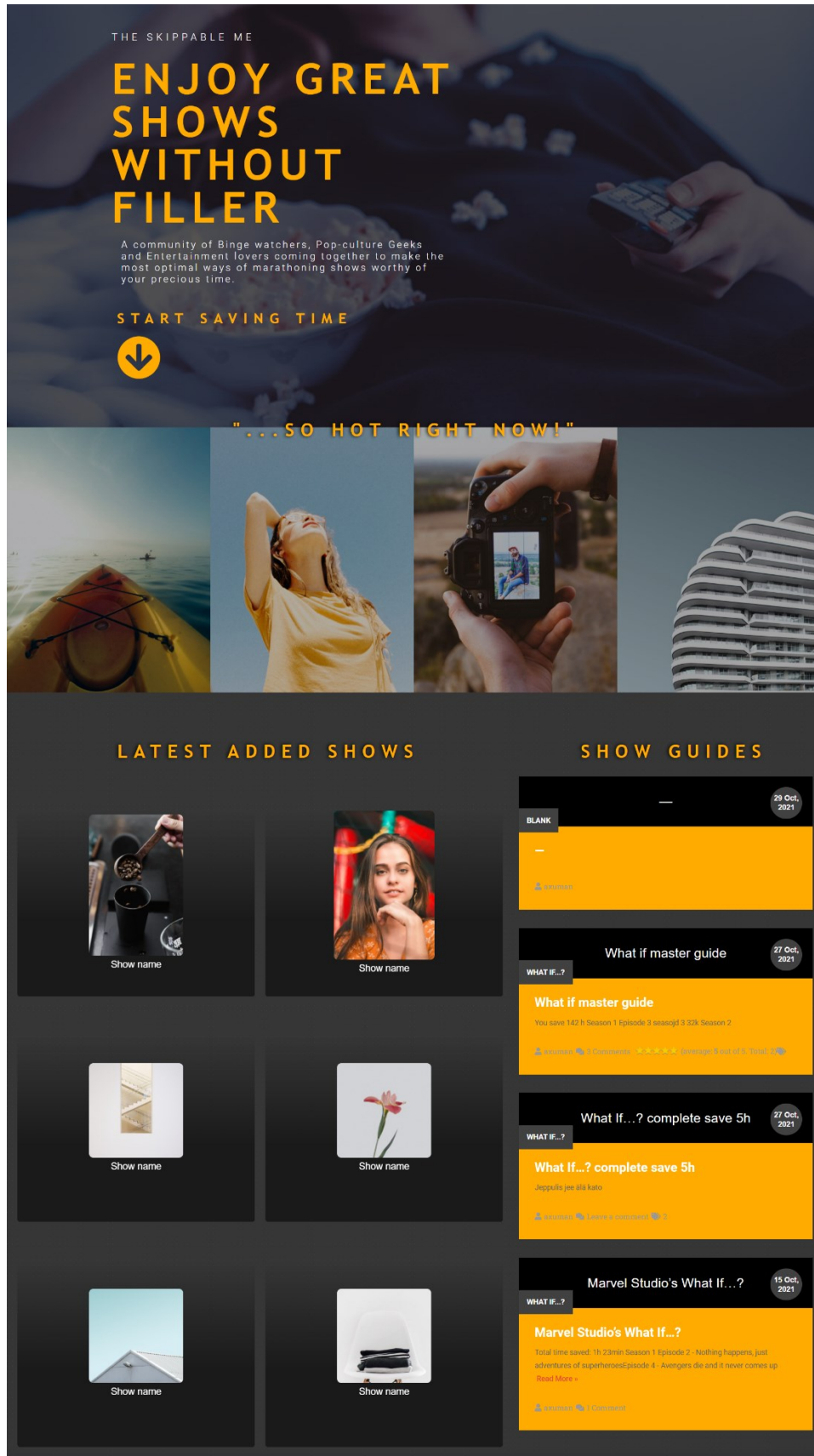
4.4 Sivujen kehittämisen aloitus ja etusivun kehittäminen

Aloitin sivuston työstämisen etusivulta ja törmäsin pian ensimmäiseen ongelmaan: miten esitän sarjojen jaksot ja listat selkeästi etusivulla? Alustava suunnitelmani oli vastaanottaa käyttäjien tekemiä listoja, joita muut käyttäjät voisivat arvioida. Jotta voin esittää sarjaa koskevan oppaan sen tulisi näkyä jaksoa käsittelevällä sivulla. Täten aloin miettimään: Entä, jos kaikki sarjat hakisivat jaksonsa muilta sivuilta ja täten ihmiset voisivat arvostella jaksokohtaisesti ilman, että heidän tarvitsisi luoda kokonaista opasta. Idea siinänsä ei ollut mielestäni huono, joten lähdin sitä toteuttamaan. Latasin WP scraper -lisäosan. Tämän lisäosan avulla pystyn ”repimään” miltä tahansa nettisivulta sisältöä ja tuomaan sen WordPress-projektiin. Vasta lisäosaa testattuani huomasin kaksi ongelmaa tässä metodissa: Ensimmäinen ongelma on, että kyseinen lisäosa lisää sisällön sivustolle blogiartikkelina. Elementorilla ei voi muokata näitä sisältöjä, sillä yksittäisen artikkelin pohjan muokkaus ei kuulu ilmaiseen Elementoriin (Elementor 2021). Kuten aikaisemmin mainittu, sivuston teemat hallitsevat niitä osia, joihin Elementorin ilmaisversio ei pääse käsiksi. Toisekseen tämä metodi vaatisi minulta sivuston kehittäjänä enemmän manuaalista työtä yhtä sivua kohden verrattuna listauksen lisäämiseen yksittäiselle sivulle.

Sarjojen oppaiden päivittämisen sarjakohtaiselle sivulle sain ratkaistua, kun lisäsin WP Show Posts -lisäosan. Tämän lisäksi latasin lisäosan, jonka avulla pystyn lisäämään sivustolle listan artikkeleita ”kortteina”. WP Show Posts -lisäosan ansiosta pystyn määrittelemään, kuinka monta opasta näytetään yhdistetyllä sivulla. Pystyn myös määrittämään, millä perusteilla lisäosa hakee artikkelin, joka näytetään sivulla. Valitsin kriteeriksi artikkelin, jolla on eniten kommentteja tietyistä kategoriasta. Jos käyttäjä on esimerkiksi luonut listan sarjalle Game of Thrones, sivusto hakee eniten kommentoidun artikkelin

näytettäväksi. Kommentointiin lisäsin arvosteluasteikon 1—5 tähteä, jonka avulla käyttäjät voivat jättää mielipiteensä kyseisestä listasta ja arvioida sen. Tieto päivittyy opasta koskevan sarjan sivulla.

Tämän jälkeen aloin muokata etusivua. Kuten Fessenden (2018) tutkimuksessaan toteaa, ihmiset kiinnittävät paljon huomiota sivustolla sivuston ensimmäiseen näkymään. Kehitän etusivusta sekä selkeän että mukaansatempaavan. Alkuperäisestä suunnitelmasta poiketen sijoitin oppaiden ja lisättyjen sarjojen listaukset etusivulle vierekkäin. Tällä pyrin siihen, että etusivulla olisi tärkeät tiedot helposti ja nopeasti nähtävissä (Kuva 4).



Kuva 4. Etusivun uusi tyylittely Elementorilla.

Etusivulle lisäsin väliaikaisia osioita, joissa on tarkoitus näyttää sen hetkiset suositut sarjat, uusimmat sivulle lisätyt sarjat sekä tällä hetkellä puhutuimmat tai aktiivisimmat listat. Koska listat on tehty artikkeleina eli päivityksinä, joita teema hallitsee, niitä ei voi juuri muokata Elementorilla. Elementorilla pystyin väkisin muokkaamaan haettuja oppaita sivuilla, jotka oli luotu Elementorilla. Ongelmana ilmeni kuitenkin se, että Elementor hidastui ja sivun muokkauksien päivittäminen epäonnistui. Näitä muokkauksia väkisin tehdesäni sain virheilmoituksen 502. Yrittäessäni ratkaista virheilmoitusta sain selville, että sivustoni alla oleva Apache-palvelin rajoittaa datamäärän kulkua (Elementor 2019). Apache on verkkopalvelin, jolle WordPress on asennettu. Se määrittelee tiettyjä rajoituksia WP:n ja sen lisäosien toiminnalle. Asiaa tutkiessani löysin ohjeet (Elementor 2019), joissa kerrottiin, että voin mahdollisesti korjata ongelman Elementorin asetuksista. Jos asetusten muokkaaminen ei tuota tulosta toinen keino olisi koodin pätkän syöttäminen sivustoni tiedostoihin. Päädyin kuitenkin poistamaan koko sivuston ja aloittamaan alusta, sillä sivustolla ei ollut vielä paljoa sisältöä ja täten minun ei tarvinnut kirjoittaa koodia sivuston toteutuksessa. Nyt tiesin, miten saan toivotut ominaisuudet toimimaan ja pystyin aloittamaan sivuston alusta asentaen ainoastaan tarvittavat lisäosat ja sotkematta tietokantaa. Aloittaessani sivuston rakentamisen uudelleen päätin myös ladata All-in-One WP Migration -lisäosan, jonka avulla voin helposti tallentaa sivuston varmuuskopioita, mikäli se täytyy palauttaa uudestaan aikaisempaan versioon.

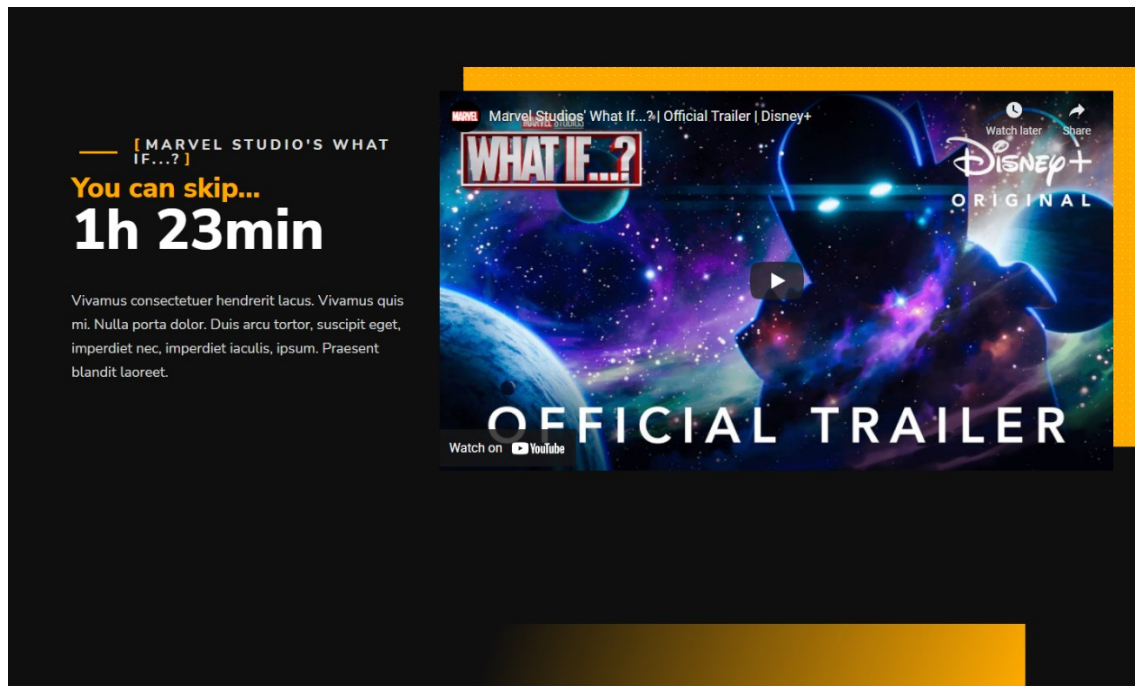
Nyt minulla oli selkeämpi suunnitelma, miten lähden rakentamaan sivustoa, sillä olin saanut sen jo kerran toimivaksi. Tiesin, mitä elementtejä lisään ja miten käytän niitä yhdessä artikkelien noston kanssa. Työskentelyä nopeuttaakseni apunani oli Envaton valmiit elementit. Näiden elementtien avulla sivusto saa nopeasti visuaalisesti näyttävän rungon, joka vaatii vain tekstien ja värien muokkausta.

4.5 Sarjakohtaisen sivuston kehittäminen

Kunkin sarjan oman sivun tarkoituksena on siis näyttää suosituin jaksojen ohitusopas sekä tuoda esille myös muita oppaita, jotka ovat saattaneet jäädä suosituimman oppaan varjoon. Näin myös uudet oppaat, jotka saattavat olla aikaisempia tarkempia, tulevat esille. Seuraavaksi esittelen yksittäisen sarjan sivun rakenteen ja sisällön.

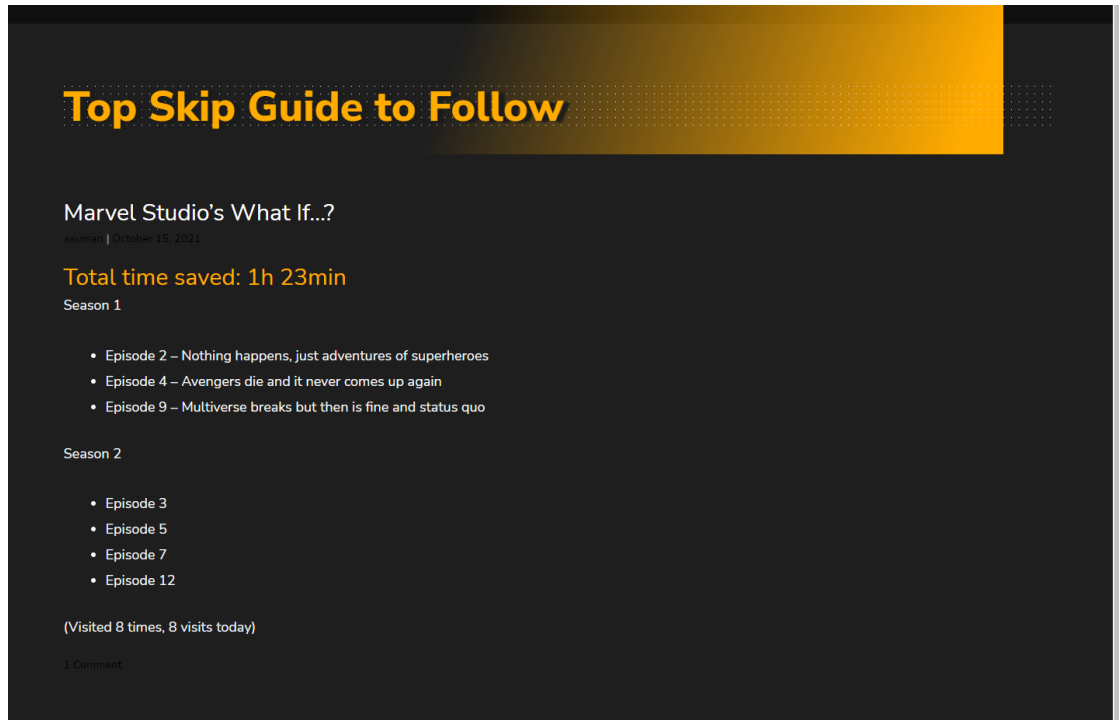
Sarjan sivulla keskeisenä tietona on suosituimman oppaan avulla säästämä aika, sarjan lyhyt kuvaus ja mainos. Näiden perusteella käyttäjä voi päättää, haluaako hän aloittaa

uuden sarjan katsomisen. Nämä tiedot on esitetty sivun yläosassa (kuva 5). Kuvassa 5 näkyvät visuaaliset elementit ovat toteutettu Envato Elements -lisäosalla.



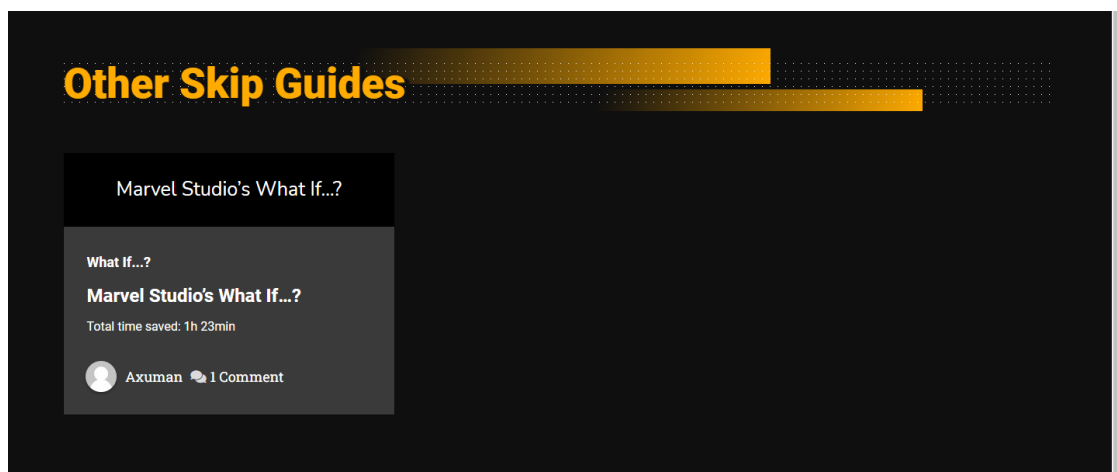
Kuva 5. Sarjan perustiedot ja mainos.

Heti tämän jälkeen sivulla näkyy otsikko "Top Skip Guide to Follow", jonka jälkeen kyseisen sarjan suosituin opas päivittyy sivulle helppolukuisuuden avuksi. Kyseisen oppaan voi myös lukea WordPressin omassa "artikkelit/blogi" -osiossa, mutta helpottaakseni sivuston vierailijoita suosituin opas on sarjan pääsivulla (kuva 6).



Kuva 6. Sarjan suosituin opas haettuna kirjoitetuista artikkeleista.

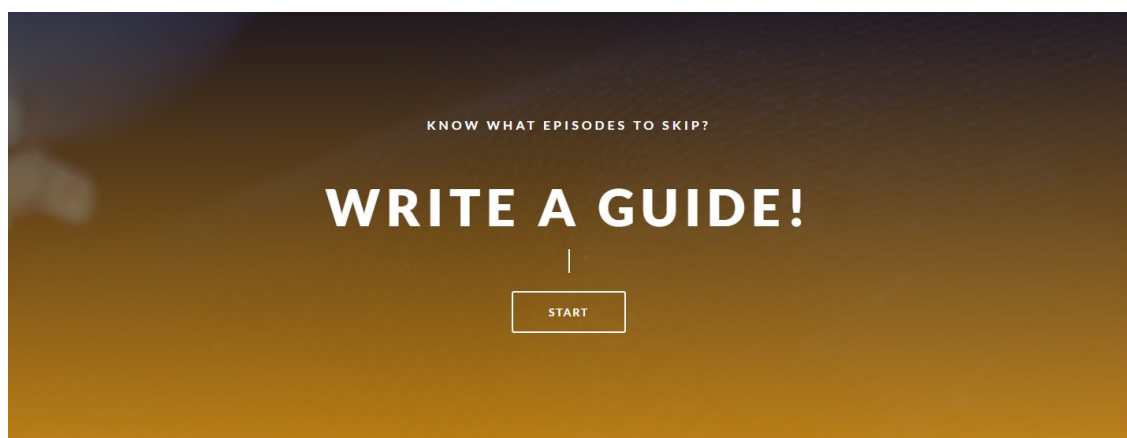
Muut oppaat voi löytää joko sivulle päivittyvästä opaslistasta (kuva 7) tai koko sivuston Guides-osiosta, jonne päivittyvät kaikki sivuston uudet ja vanhat oppaat. Lisäsin sarjalle tehdyille oppaille satunnaisesti päivittyvän osion, jonka avulla suosituimman oppaan varjoon jääneet listat on helpompi löytää.



Kuva 7. Muut oppaat, joita käyttäjät ovat tehneet sarjasta.

Sivun loppuun lisäsin osion, jolla kannustetaan käyttäjiä tekemään lisää ohitusoppaita (kuva 8). Lisään tämän bannerielementin myös etusivulle, jossa vierailijat näkevät kyseisen ilmoituksen ensimmäistä kertaa. Tarkoituksena on pitää tätä elementtiä jokaisella sivulla muistuttaen, että käyttäjät voivat kirjoittaa oppaan, kun ovat hieman enemmän

perehtyneet sivustoon. Kuvan elementti on toteutettu Envato Elements -lisäosaa käyttäen. Se on linkitetty ohjaamaan käyttäjät oppaan kirjoitus sivulle.



Kuva 8. Elementti, joka kutsuu vierailijoita luomaan lisää sisältöä sivustolle.

Sarjaa käsittelevän sivun valmistuminen oli minulle melko iso saavutus tässä projektissa. Sivua tehdessäni ratkaisin monta ongelmaa, joita epäröin teknisessä toteutuksessa. Suurin huolenaiheeni oli yksittäisen oppaan lisäys sivulle ja oppaiden arviointi. Testieni pohjalta oppaiden hallinnointi liittyen sarjan sivua vaikuttaa toimivan ja oppaat päivittyvät, kun lisään niitä kyseiseen kategoriaan. Nämä toiminnallisuudet tukevat sitä, että kun myöhemmin lisään vierailijoille mahdollisuuden kirjautua ja luoda omia oppaita, myös ne päivittyvät oikeille sivuille. Sarjakohtaisen sivun voi kopioida, jolloin sivustolle voi lisätä uusia sarjoja.

4.6 Yläpalkki ja alapalkki

Sarjasivun valmistuttua aloin kehittämään sivustolle header ja footer -osioita eli ylä- ja alapalkkeja. Tässä käytän suomenkielisten termien sijaan englanninkielisiä vastineita, koska ne ovat käytössä myös WordPress-alustalla. Header pysyy tavallisesti samassa paikassa jokaisella sivulla. Header sijoitetaan sivustolla yleensä selaimen ikkunan yläosaan. Header voi olla melkein minkäläinen tahansa, mutta se tulee olla löydettävissä. Nykyaikainen Header on yleensä kiinteä neliskulmio sivuston yläosassa. Se voi olla joissain tapauksissa tyylitelty eri muotoon tai jopa läpinäkyvä, mutta sen sisäiset elementit ja tiedot näkyvät linjassa toisiinsa. Header voi myös olla kiinnitetty näyttöön sivuston sijaan. Tämä tarkoittaa sitä, että header-osio pysyy näkyvillä samassa kohdassa näytöllä, vaikka sivustoa sen alla selaisi. Kuten Kate Shokurova artikkelissaan näyttää ja kertoo

(Shokurova 2020), headereita voi olla monenlaisia, ja vaikka niiden sisältö voi muuttua, tulee headerin silti olla yhdenmukainen sivustolla sekä käyttäjälle löydettävä.

WordPressin oma teemanmuokkaustyökalu tukee headerien luomista, mutta kuten muuallakin WordPressissä, muokkausmahdollisuudet ovat rajalliset. Tästä syystä käytän sivustolla Elementor Header and Footer Builderia. Tällä lisäosalla voi luoda täysin vapaasti muokattavan headerin. Käytännössä tämä luo uuden sivun, yhdistää sen koko sivustolle ja esittää sen päälle liimattuna sillä sivulla, jolla käyttäjä vierailee. Käyttäjä voisi siis nähdä pelkästään headerin menemällä osoitteeseen: <http://theskipable.me/elementor-hf/header/> . Tein headerista läpinäkyvän, kun sivua ei ole selattu (kuva 10), ja kiinteän värisen kun sivua selataan alaspäin (kuva 11).



Kuva 10. Läpinäkyvä header.



Kuva 11. Kiinteä taustaväri.

Halusin lisätä sivustolle hakuosion hakupalkkina, mikä vie paljon tilaa. Sivuston selkeyden säilyttämiseksi en voinut lisätä kokonaista navigaatiota, joten asensin AMP Sidebar Hamburger Menu -lisäosan. AMP Sidebar Hamburger Menu mahdollisti niin sanotun hampurilaisvalikon lisäämisen. Hampurilaisvalikko on vakiintunut nimike ikonille, joka piilottaa navigaatiovalinnat ja näyttää erittäin minimalistiselta hampurilaiselta. Jouduin määrittelemään WordPressin Valikko-asetuksista tämän lisäosan valikon toiminnallisuuden. Valikko siis toimii omana AMP sidebar menu- valikkona. Tämän lisäosan kohdalla jouduin muokkaamaan PHP-koodia, sillä lisäosa ei ollut suoraan Elementorin kanssa yhteensopiva ja muokattava. Muokkasin lisäosan nappia, jotta sitä painamalla aukeava valikko tulisi esiin oikealta vasemman sijaan:

```
<amp-sidebar class="af_sidebar" id="sidenav" layout="nodisplay" side="right">
```

Menu Settings

- Auto add pages Automatically add new top-level pages to this menu
- Display location AMP sidebar menu
 Primary Menu
 Secondary Menu
 Off-Canvas Menu
 Logged In Account Menu
 Footer Menu

Kuva 9. WordPressin valikkoasetukset

Sivuston footer-osioon lisään ainoastaan tekijänoikeusilmoituksen, yksityisyyskäytösivun linkin, sivuston ylläpidon yhteysähköpostiosoitteen sekä sosiaalisen median painikkeet.

Andy Crestodinan ja Anna Fitzgerald mainitsevat artikkeleissaan monia asioita, joita footeriin voi sijoittaa (Crestodian 2015, Fitzgerald 2020), mutta pyrin pitämään oman sivustoni footerin mahdollisimman minimalistisena. Teen näin siksi, että ihmisen huomio keskittyy lähtökohtaisesti sivuston yläosaan (Fessenden 2018). Footerilla pyrin indikoimaan sivuston loppumista ja ilmoittamaan välttämättömät asiat, jotka eivät ilmene muualla sivustolla.

4.7 Käyttäjien tuottaman sisällön mahdollistaminen

Seuraava tärkeä sivustolle lisäämäni toiminnallisuus oli oppaiden kirjoittamisen mahdollistaminen. Käytin apuna User Submitted Post -lisäosaa. Lisäosa mahdollisti lomakepohjan asettamisen sivustolle. Lisään sivustolle myös lomakepohjan täytön ohjeet, jotta listoista saa yhdenmukaisen näköisiä toistensa kanssa. Lomakkeen ja sen täytön ohjeet lisäsin omalle sivulleen, jolle muiden sivujen bannerit ohjaavat.

Vaikka toiminto toimii kuten tarkoitettu, pitäisi siihen saada vielä lisättyä ominaisuus, että ainoastaan kirjautuneet käyttäjät voivat luoda listoja. Tämä on ominaisuus, jota lähdin työstämään seuraavaksi. Vaihdoin käyttäjäasetuksia WordPressissä niin, että kuka tahansa voi tehdä tilin. Tämän ongelmana on kuitenkin se, että mikäli käytän WordPressin sisäistä keinoa, käyttäjät pääsevät käsiksi WP:n ohjauspaneeliin. Tästä syystä päätin

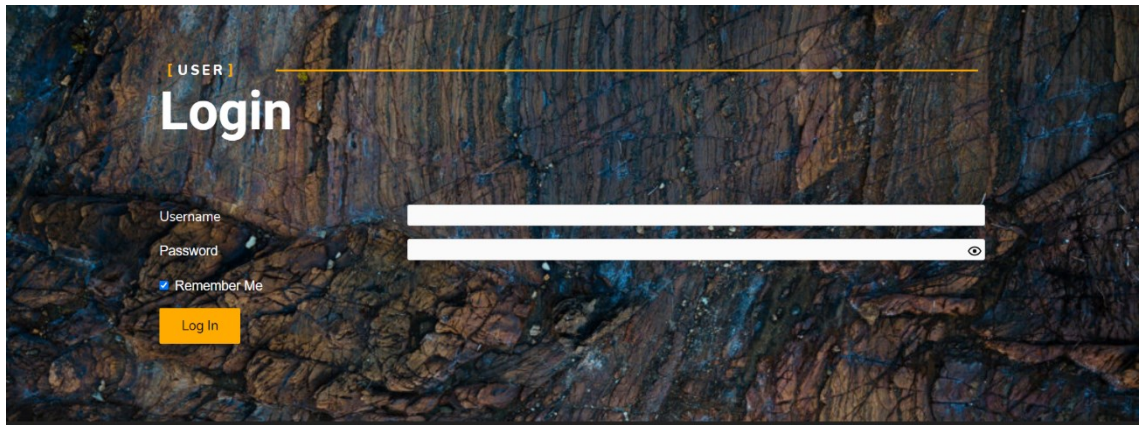
tehdä erillisen kirjautumissivuston, jonka avulla käyttäjä voi rekisteröityä ja kirjautua ilman pääsyä WP:n ohjauspaneeliin.

Muokattavia kirjautumissivu-lisäosia löysin lisäosakaupasta hakusanoilla ”Login” ja ”User registration”. Näistä päätin testilla muutamia ja vertailla niiden toimivuutta ja soveltuvuutta projektiini. Osa kirjautumislisäosista joko muokkaa WordPressin kirjautumissivua, mutta halusin että se olisi täysin itselläni muokattavissa. Koska sivusto on tehty WordPressillä ja uudet käyttäjät tallentuvat WP-käyttäjiksi, on käyttäjän sähköpostiosoitteen tallentaminen lähes välttämätöntä. Yritin kiertää tämän asentamalla Code Snippets -lisäosan. Lisäosa mahdollistaa koodin pätkien lisäämisen WordPressin tietokantaan. Lisäsin koodia, jonka löysin verkosta etsiessäni tietoa siitä, miten WordPressiin voi lisätä käyttäjän ilman sähköpostia. Ohjeistuksen löysin sivustolta wppluginsforyou.com (wppluginsforyou.com 2019). Löytämäni koodin avulla pystyisin poistamaan pakollisen tarkistuksen sähköpostikentän kohdalla. Code Snippets -lisäosan asetuksista pystyin määrittelemään mihin alueisiin sivustolla tämä lisätty koodin pätkä vaikuttaisi. Valitsin, että koodi vaikuttaisi kaikkialla sivustolla, kun luodaan uusi käyttäjä. Näin välttyisin rikkomasta GDPR:ää eli Euroopan yleistä tietosuojasetusta edes vahingossa, sillä en säilyttäisi käyttäjien henkilökohtaisia tietoja kuten sähköpostiosoitteita.

Kirjautumislomaketta työstäessäni huomasin kuitenkin, että missään testaamassani lomakkeessa sähköpostivaatimusta ei voinut poistaa. Tämä ei ollut mahdollista edes WordPressin omalla rekisteröitymissivustolla. Käytin kokonaisen päivän tämän asian tutkimiseen ja testaamiseen. Lopputulos oli kuitenkin se, että sähköposti on välttämätön osa WordPressin käyttäjän todennusta ja hallinnointia. Tämä tulee ilmi myös monilta keskustelupalstoilta (Stack Overflow 201, Stack Exchange 2021, WPQuestions 2012). Syötetty koodi siis toimi ainoastaan WP:n hallintapaneelissa, jossa lisätään uusia käyttäjiä manuaalisesti. Koska tarkoitukseni ei ole lisätä käyttäjiä manuaalisesti, poistin lisäosan ja lähdin tutkimaan toista lähestymistapaa.

Koska joudun tallentamaan käyttäjän sähköpostin, voin käyttää valmiita profiilinhallintalisäosia. Tämän lisäksi minun täytyy tarkistaa ja päivittää sivuston tietosuojalausunto ja ilmoittaa tietojen säilytyksestä. Tässä tapauksessa käytin Profile Builderia. Profile Builder luo Elementor yhteensopivia lomakkeita, jotka hoitavat käyttäjän rekisteröinnin, kirjautumisen ja tilin hallinnan sivuston puolella. Vaikka muita vastaavanlaisia kirjautumislomakkeita löytyy lisäosakaupasta, huomasin tämän olevan Elementorin kanssa yhteensopiva. Tämä mahdollistaa kirjautumis-, rekisteröinti- ja profiilisivujen visuaalisen muokkauksen sopivammaksi muun sivuston kanssa.

WordPress muisti Profile Builderille aiemmin asettamani asetukset, vaikka olin poistanut sen jo kertaalleen. Tämä nopeutti toiminnallisuuksien luontia huomattavasti. Kyseisellä lisäosalla loin kolme uutta sivua, jotka ovat kirjautumissivu (kuva 12), rekisteröitymissivu (kuva 13) sekä profiilin hallintasivu (kuva 14). Sivujen lisäksi loin uuden headerin nimellä ”Header Unlogged”. Tämä oli kopio jo tehdystä Headerista, sillä muutin sen ainoastaan näkymään käyttäjille, jotka eivät ole kirjautuneet. Kirjautuneille käyttäjille uudelleen näkyviin jäi aikaisempi header jonka nimesin ”Header logged in”. Käyttäjille ilmenevä muutos näissä on se, että headerin henkilöä kuvaavaa ikonia klikkaamalla pääsee joko kirjautumissivulle tai profiilin hallintaan riippuen siitä, onko käyttäjä kirjautunut vai ei.



Kuva 12. Elementorilla valmistettu kirjautumissivu.

[NEW USER]

Register

Username *

E-mail *

Password *

Minimum length of 8 characters.
The password must have a minimum strength of Weak

Repeat Password *

Send these credentials via email.

Add User

Already have an account?

Log in

Kuva 13. Elementorilla valmistettu rekisteröitymissivu.

[EDIT]

User Details

User to edit: axuman

Username *

E-mail *

Password *

Minimum length of 8 characters.
The password must have a minimum strength of Weak

Repeat Password *

Update

Log Out

Kuva 14. Elementorilla valmistettu profiilin hallintasivu.

Kun olin saanut kirjautumisen ja profiilin hallinnoinnin toimimaan oli edessä käyttäjän tekemien oppaiden hallinnointi. Tässä vaiheessa oppaiden luominen ja lähettäminen oli mahdollista, vaikka sivun vierailijalla ei olisi tunnuksia. Tämä voisi aiheuttaa tietoturva-ongelmia, roskapostia tai ainakin eripuraa yhteisössä. Kirjautumattomat käyttäjät voisivat

esittää toisia yhteisön jäseniä, syöttää haitallista koodia sivustolle tai lisätä sivustolle mainoksia. Jotta ylläpitäjä voisi valvoa tilien toimintaa, tulisi kaikkien käyttäjien olla sisään kirjautuneita. Jotta käyttäjät voivat tuottaa sisältöä sivustolle, asetin WordPressin asetuksista uusien käyttäjien roolin ”avustajaksi”. Tämä mahdollistaa sen, että käyttäjät voivat kirjoittaa ja hallinnoida oppaitaan, mutta eivät julkaista niitä. Oppaiden julkaisuun vaaditaan ”päätoimittajan” tai korkeamman auktoriteetin rooli. Käyttäjäroolien valinnassa tukeuduin wordpress.orgin omaan ohjeistukseen (WordPress.org 2021).

User Submitted Posts -lisäosan asetuksista pystyin muokkaamaan melko kattavasti sen toiminnallisuuksia. Asetuksissa määrittelin aikaisemmin, mitkä ovat tarvittavat kentät opaslomakkeen täydennykseen. Jätin ainoastaan olennaiset asiat esille kuten oppaan otsikon, oppaan kategorian eli sarjan, jota opas koskee, alkeellisen varmistuskysymyksen, joka hieman vähentää automatisoitujen listauksien syöttöä sekä tekstikentän johon oppaan voi kirjoittaa. Näiden lisäksi lisäsin valintaruudun, jossa pyydetään käyttäjän suostumus kirjoitetun listan arviointiin ja julkaisuun.

Oppaan muotoilun säännöt ilmoitan lomakkeen alapuolella selkeästi esimerkin avulla. Oppaan kirjoittajan käyttäjänimi tulee automaattisesti oppaaseen, kun kyseinen henkilö on kirjautuneena. Oppaiden tekeminen ei ole myöskään mahdollista käyttäjiltä, jotka eivät ole kirjautuneena sivustolle. Kuvassa 15 näkyy oppaiden kirjoittamista käsittelevä sivun näkymä, kun käyttäjä on kirjautuneena sivustolle.

[CREATE]

Episode Skip Guide

If you are new here, please check out the formatting Rules.

[Click here to check the rules](#)

Write a Skip Guide

Post Title

Is up same as down?

Antispam Question

Post Category
 Please select a category.. ▾

Post Content

By checking this box, you agree that this Guide can be published on a publicly visible, theskippable.me -website. The Guide will be reviewed and published if it meets the formatting stated in the Rules section.

[Submit Post](#)

Rules for filling in a Skip Guide

If you want your guide to be published on the site you need to follow the following formatting rules.

1. Title must contain the name of the Show or Series
2. If no suitable Show or Series can be found in the Category selection, leave it as "Uncategorized", new categories will be created according to the title
3. First thing written in the "Post content" should be the total time saved by following your Skip Guide
4. Seasons should be numbered, then press Enter to change line. When inserting Episodes, press Space bar 3 times
5. Add an empty line between last listed Episode of a Season and new Season text
6. Episode description should be brief but give small amount of detail about what happens and why it is Skippable
7. For general formatting follow the example shown on the right

Follow this example

Post Content

```
Total Time Saved: 1 h 23 min

Season 1:
Episode 1 - Pilot episode, nothing happens. Duration 23 min
Episode 4 - People talk, side character development. Duration 30 min

Season 2:
Episode 3 - Character dies, no relation to story. Duration 30 min
```

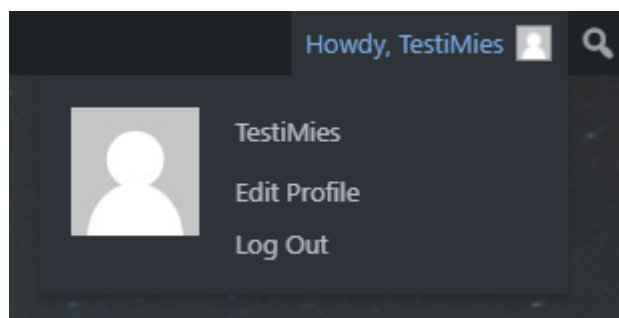
Kuva 15. Kirjautuneelle käyttäjälle näkyvä oppaanluomissivu.

5 TESTAUS JA TEKNISTEN ONGELMIEN KORJAAMINEN

Tässä vaiheessa kaikki tarvittavat ominaisuudet ja toiminnallisuudet olivat sivustolla. Sivusto oli täynnä "Place holder" eli väliaikaiskuvia, -tekstejä sekä -oppaita. Tässä vaiheessa sivusto oli vielä lähes täysin salaamaton.

Jätin sivuston selaimille löydettäväksi yön yli ja otin Under Construcion -asetuksen pois päältä, jotta sivustolla voi vierailia. Tämän avulla pystyin alustavasti testaamaan kirjautumista toisilta laitteilta. Lähetin sivuston linkin myös kolmelle kaverilleni ja sain paljon tietoa sivuston toimivuudesta jo yhden illan aikana. Seuraavana päivänä myös itse tutkin sivustoa ja löydetyistä ongelmista tein listan:

- Sivustolle kirjaututtua WordPressin ylläpitopalkki tulee näkyviin (kuva 16).
- Vierailija voi avata hallintapaneelin ylläpitopalkista.
- Oppaan kirjoituslomakkeessa on kirjatutumattomille "log in" teksti ja linkki. Se ohjaa WP:n omalle kirjautumissivulle.
- Sivuston responsiivisuus ja skaalaus vaihtelee. Sivuston elementit saattavat mennä päällekkäin pienemmillä näytöillä.
- Mobiiliversion sivun elementit eivät skaalaudu lähes ollenkaan.
- Selain ilmoittaa sivun olevan "Ei turvallinen".
- Googlen PageSpeed Insight -palvelun mukaan sivusto on melko hidas (Google 2021).
- Header tulee joidenkin elementtien päälle.
- Footerissa olevat pisteet eivät ole täysin yhdenmukaisia.
- Opas-listassa oppaiden listausten tausta on valkoinen mobiilinäkymässä.



Kuva 16. Kirjautunut käyttäjä pääsee hallinnoimaan tiliään, josta hän ohjautuu WP-hallintapaneliin.

Näitä löytämiäni ongelmakohtia lähdin tutkimaan kohta kohdalta Google Chrome -selaimen Incognito-tilassa. Tämä mahdollisti sivuston testaamisen ilman, että selain kirjaa minut automaattisesti sisään ylläpitäjänä.

Ensimmäisenä haasteena oli saada WordPressin ylläpitopalkki piiloon kaikilta peruskäyttäjiltä. Kirjautunut käyttäjä pääsee tätä kautta WP:n hallintapaneliin ja vaikkei heillä ei ole siellä oikeuksia tehdä juurikaan mitään, voivat he silti kopioida lomakkeita yhdellä napin painalluksella ja mahdollisesti täyttää tietokannan ja palvelimen näillä kopioituilla oppailla. Lähtiessäni selvittämään asiaa löysin nopeasti ohjeita, miten ylläpitopalkin saa pois päältä. Tähän sain avun WPbeginner.com-sivuston oppaasta (WPBeginners 2021). Latasin Hide Admin Bar Based on User Roles -lisäosan ja sen avulla pystyin hallinnoida, millä käyttäjärooleilla tämä palkki näkyy.

Saatuani WordPressin ylläpitopalkin piiloon peruskäyttäjiltä ei heidän pitäisi olla mahdollista päätyä WP:n hallintapaneliin sitä kautta. Huomasin kuitenkin, että jos käyttäjä osaa hakea selaimessa `.../wp-admin`, päätyy hän hallintapaneeliin. Tämä pätee myös kohdassa, jossa sivustolla oleva opas ehdottaa kirjautumattomia kirjautumaan sisään. Tätä kautta vierailijat päätyvät WP:n omalle kirjautumissivulle. Koska lisäämäni käyttäjien toimintaa ohjaavat lisäosat eivät ole tarkoitettu suoranaisesti tähän tarkoitukseen, on niissä ohjelmoituna sisään WP:n hallintapaneeliin ohjaus. Hain apua ongelmanratkaisuun jälleen WPbeginner.com-sivustolta (WPBeginners 2016) ohjeistetaan, miten hallintapaneeliin pääsyä voi säädellä. Asensin uuden lisäosan, Remove Dashboard Accessin. Tämän lisäosan ansiosta ainoastaan ylläpitäjät ja päätoimittajat voivat päästä hallintapaneeliin. Kaikki muut käyttäjäroolit ohjataan suoraan sivuston etusivulle.

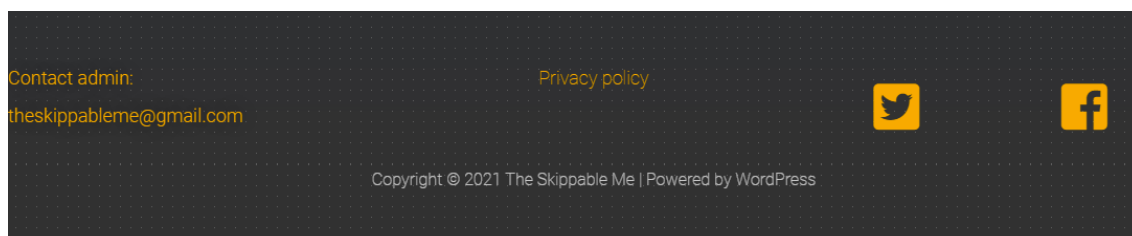
Vaikka sain estettyä käyttäjien pääsyn hallintapaneeliin, voivat he silti päästä WordPressin kirjautumis- ja rekisteröitymissivulle. Tämä ei sinänsä ole ongelma, mutta se tarkoittaa sitä, että aikaisemmin luomieni kirjautumis- ja rekisteröintisivujen tekeminen oli turhaa, jos haluan kirjautumis- ja rekisteröintisivujen olevan yhdenmukaisia käyttäjälle.

Testaillessani kirjautumisvaihtoehtoja törmäsin lisäosaan, joka muokkaa WordPressin omaa kirjautumissivua niin, että siitä voi tehdä haluamansa näköisen. Tällä lisäosalla kirjautumissivua muokataan WordPressin omalla muokkaimella. WordPressin oman kirjautumissivun muokkain ei ole yhtä monipuolinen kuin Elementor, mutta kirjautumissivu oli onneksi erittäin yksinkertainen.

Seuraavaksi muokkasin sivuston responsiivisuutta ja elementtien skaalausta. Eniten ongelmia aiheuttanut ja epäesteettisin sivu skaalauksen suhteen oli etusivu. Etusivulla oli paljon tyylliteltyjä ja muokattuja elementtejä, jotka eivät itsestään skaalaudu tai ne skaalautuvat väärin. Tabletti- ja mobiilinäkymässä huomasin, että fonttien koko kasvoi maksimikokoon. Näitä pystyin muokkaamaan Elementorin responsiivisessa tilassa, joka simuloi tabletti- tai mobiilinäkymää. Näiden lisäksi niiden resoluutioita voi kehittäjä itse muokata. Avasin tablettinäkymän ja korjasin tekstien koon sopivammaksi. Lisäksi muokasin tekstit ja elementit hieman eri muotoon. Työpöytäversiossa minulla oli elementtejä vierekkäin, mutta mobiiliversiolla laitoin elementit allekkain, jotta sivua voi selata pyyhkäisemällä sormella näyttöä ja ainoa suunta, jossa on sisältöä, on pystysuunta.

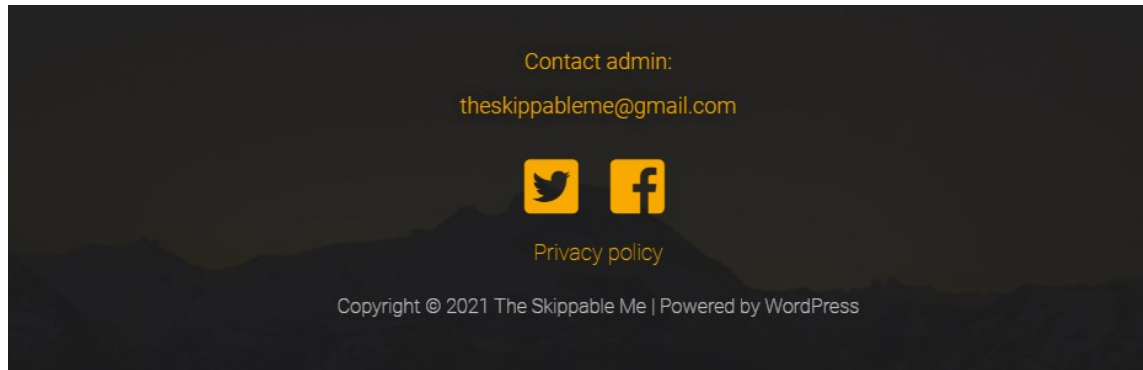
Mobiilinäkymää minun ei tablettinäkymän muokkaamisen jälkeen juurikaan tarvinnut muokata. Vaihdoin joidenkin fonttien koon vielä hieman pienemmäksi ja hieman säädin grafiikan paikkaa sivulla. Tarkastellessani muita sivuja huomasin, että header on mobiilinäkymässä liian iso ja että se peittää osan sivun sisällöstä. Muokkasin molempia sivustolla olevia headereita responsiivisessa mobiilinäkymässä poistaen logon ja siirtämällä hampurilaisvalikon ja käyttäjäikonin toistensa viereen Elementorin mahdollistamalla absoluuttisella asettelulla.

Footerin korjasin muokkaamalla sen taustakuvaa. Koska footer oli alun perin muodostunut kahdesta osiosta aiheuttaen niiden taustakuvatekstuurin päällekkäisyyden (Kuva 17)



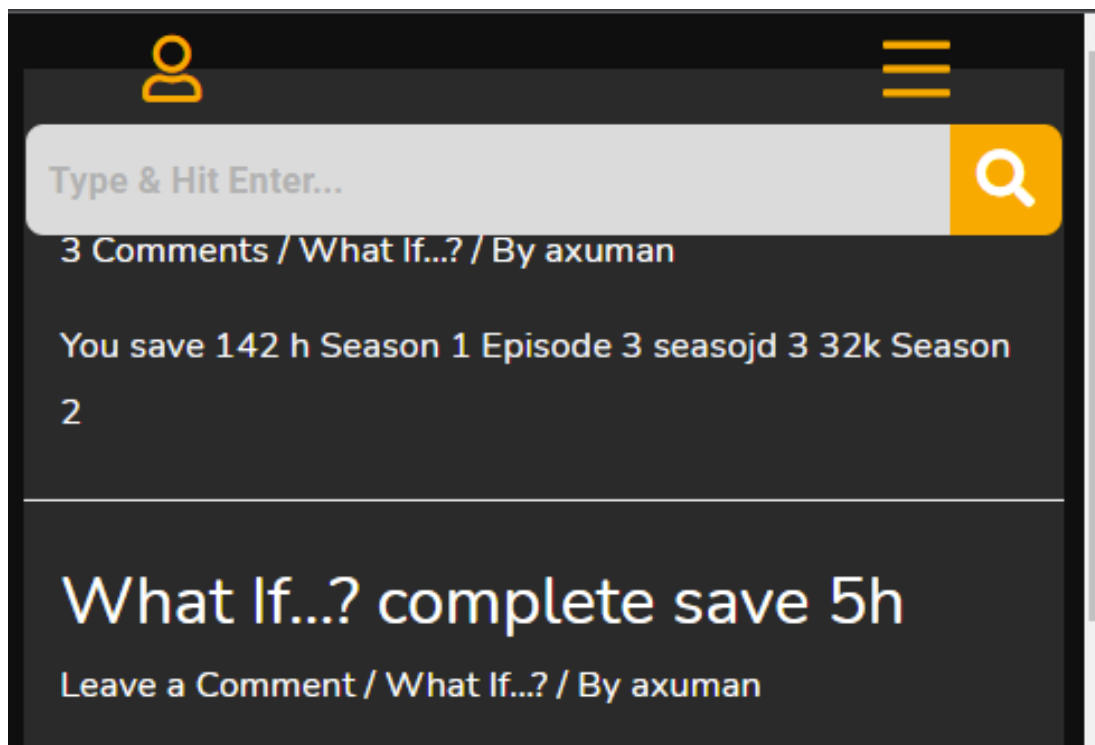
Kuva 17. Alkuperäinen footer. Tekijänoikeustekstin yläpuolella oleva pisterivi on muista huomattavasti poikkeava.

Footerin muokkasin tekemällä uuden osion ja asettamalla kaikki tiedot sen sisään. vaihdoin myös taustakuvan, sillä olin käyttänyt kyseistä pistetaustakuvaa muualla sivustolla ja oli mahdollista, että sivun ja footerin taustassa olevat pisteet menisivät toistensa päälle kuten alkuperäisessä footerissa. Uuteen footeriin tiivistin myös tiedot ja sain ne hieman lähemmäksi toisiaan (kuva 18).



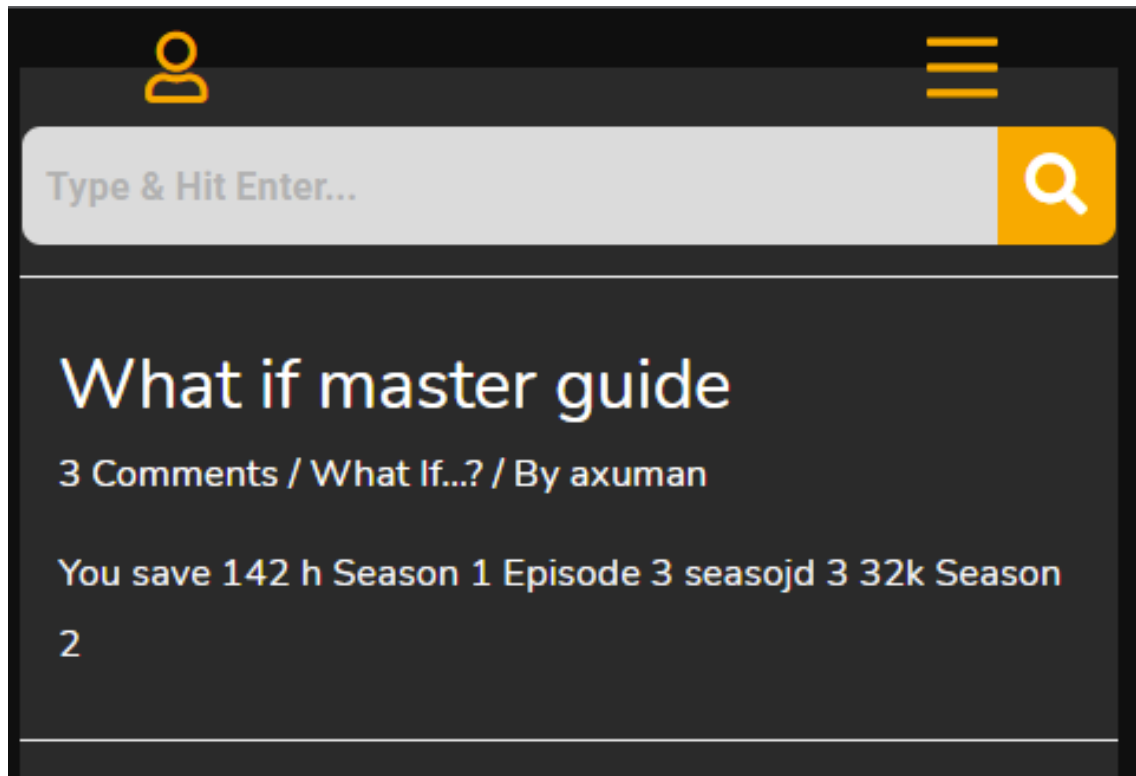
Kuva 18. Päivitetty footer.

Sivuston oppaiden selaamisen korjaaminen mobiililaitteilla olikin melko helppoa. Se vaati vain, että menin vastaavaan mobiililaitetilaan Astra-teeman omista asetuksista ja vaihdoin sisällön taustan värin. Näin sain oppaat seuraamaan sivuston visuaalista ilmettä. Ainoa jäljellejäänyt ongelma oli teeman ylläpitämä artikkelisivu (Guides-sivu), jota en voi muokata Elementorilla. Tämä vaikuttaa siihen, että header tulee joidenkin sen sisältöjen päälle. Ongelma tulee ilmi ainoastaan mobiiliversiossa, ei työpöytäversiossa (kuva 19). Koska header on elementti, joka ilmenee jokaisella sivulla, se myös estää mobiiliversiossa Guides-sivun ensimmäisen tuloksen avaamisen.



Kuva 19. Header peittää ensimmäisen oppaan otsikon, jota klikkaamalla oppaan saa auki.

Tämän ratkaisin lisäämällä sivustolle tyhjän listauksen otsikolla ”—” ja asetin muotoilun status-valintaan, joka poisti kyseisen artikkelin otsikon. Tämän jälkeen poistin kommentit käytöstä ja asetin artikkelin kategoriaan ”Blank”. Lopuksi valitsin valinnan ”Stick to the top of the blog”, jolloin artikkelisivua avatessa ensimmäinen artikkeli on tyhjä ja jää piiloon hakukentän alle (kuva 20). Mikäli jotain tiettyä haetaan sivulta, niin hakusana tulostuu tämän paikalle ja siirtää tulokset hieman alaspäin, joten myöskään jotain haettaessa tämä ei ole ongelma.



Kuva 20. Hakukenttä peittää ensimmäisen julkaisun, joka on kiinnitetty tyhjä julkaisu.

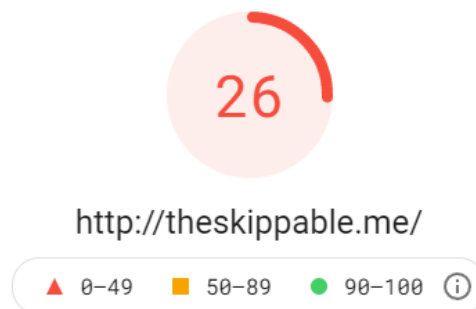
Viimeisenä vaiheena sivuston testaamisessa ja korjaamisessa oli SSL-varmenteen lisääminen ja sivuston latausajan optimoiminen. Käytössäni ollut 100mbps verkkoyhteys lataa sivuston nopeasti, vertasin verkon nopeutta muihin Euroopan maihin ja Yhdysvaltoihin ja huomasin, että ainakin internetin puolesta latausnopeus on suunnilleen samoissa nopeuksissa (Speedtest 2021). Sivuston latausaikaa voi silti hieman optimoida, vaikka suomalaisille käyttäjille sivusto latautuisikin nopeasti.

Googella on palvelu, PageSpeed Insights, jolla voi testata sivuston nopeutta. Tämä nopeustesti on melko pintaraapaisun omainen, mutta hyvin suuntaa antava. Googlen palvelun avulla voin verrata sivuston latausnopeutta ennen ja jälkeen sivuston optimoinnin. PageSpeed Insights arvioi sivuston nopeutta seuraavasti:

- 0–49 (punainen): Huono
- 50–89 (oranssi): Parantamisen varaa
- 90–100 (vihreä): Hyvä

Goolgen ohjeiden mukaan 100 pistettä on täydellinen tulos ja sitä on erittäin vaikea saavuttaa (web.dev 2019). Sivustoilta ei yleisesti odoteta täydellistä tulosta. Omalla sivustollani tähtään mobiili- ja työpöytäversiolla korkeaan kesiverto tulokseen. Projektissa olen käyttänyt tähän mennessä paljon lisäosia ja ne myös hidastavat sivustoa. Koska WordPressiä ei ole tarkoitettu tämänkaltaisen sivuston luontiin varsinkaan ilman koodia, on minun pitänyt ladata raskaita lisäosia.

Googlen laskurin lisäohjeissa on linkkejä useisiin eri sivuston optimointiin vaikuttaviin lisäosiin, joita aion testata sivuston nopeuttamiseksi. Ensimmäinen asennettava lisäosa tähän tarkoitukseen on Autooptimize. Tämä lisäosa siistii sivustolla latautuvia koodeja ja poistaa turhia taustasuorituksia. Ennen optimoinnin aloittamista otin ylös tiedot sivuston pisteistä. Näistä sain tuloksen mobiili- ja työpöytäversiolle (kuva 21 ja kuva 22). Mobiiliversio sai 26 pistettä ja työpöytäversio 61 pistettä, mikä tarkoittaa, että molemmat versiot toimivat melko hitaasti.



Kuva 21. Mobiiliversion tulos.

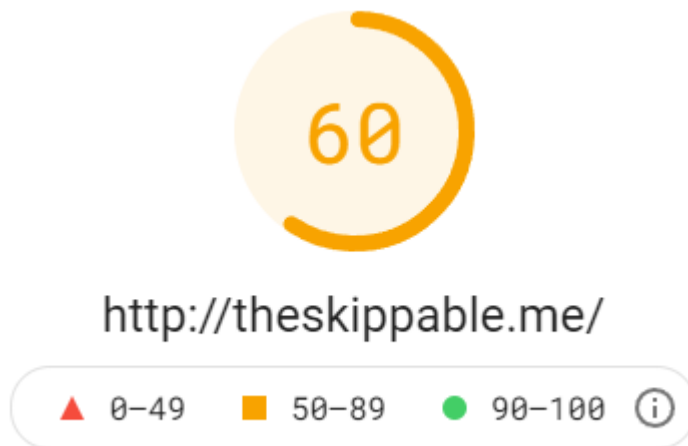


Kuva 22. Työpöytäversion tulos.

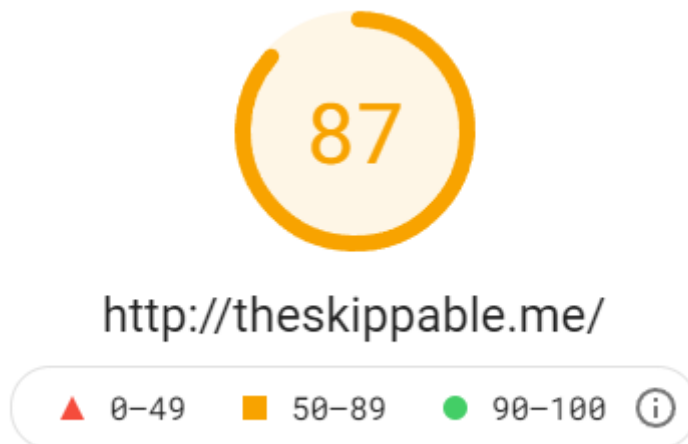
Heti kun käynnistin Autooptimizerin, sivuston nopeus pomppasi mobiiliversiossa tulokseen 44 ja työpöytäversiossa tulokseen 84 jo yksi optimointilisäosa teki siis huomattavan parannuksen sivuston nopeuteen. Tähän päälle lisäsin Cache Enabler -lisäosan. Kyseinen lisäosa luo staattisia HTML-tiedostoja sivuston etupuolen, frontendin, sivuista ja tallentaa ne palvelimelle, jotta sivut latautuvat nopeammin. Tämä teoriassa nopeuttaa sivuston toimintaa, sillä jo ladatun sivun hakeminen on nopeampaa kuin sen uudelleen lataaminen reaaliajassa. Asetin lisäosan asetuksen siten, että aina kun uusi opas luodaan, sivuston välimuisti päivittyy. Näin ollen sivusto päivittyy ainoastaan silloin, kun siinä on tapahtunut muutoksia. Tämä lisäosa nopeutti sivustoani hieman lisää ja antoi nyt PageSpeed Insightin tulokseksi mobiiliversiossa 55 ja työpöytäversiossa 85. Mitä lähemmäksi 100 pistettä mennään, sitä vaikeammaksi sen saavuttaminen tulee (web.dev, 2019). Optimointilisäosat ovat siis vaikuttaneet mobiiliversion nopeuteen työpöytäversiota enemmän, koska mobiiliversion alkuperäinen tulos oli paljon työpöytäversiota matalampi.

Viimeiseksi lisäsin Smush-lisäosan. Tämä lisäosa puristaa mediatiedostot hieman pienemmäksi. Tiedostojen laatu kärsii tästä prosessista, mutta sivustollani olevien taustakuvien ja kuvituskuvien ei tarvitsekaan olla korkealaatuisia. Smushin asetuksessa määrittelin kuville maksimikoon 1080 x 1080 pikseliä. Lisäosan mukaan säästin kuvia muokkaamalla sivustollani tilaa 6,2 megatavua, joka on huomattava määrä sivuston nopeutta optimoitaessa. Kävin vielä jokaisella sivulla lataamassa kuvat uudelleen varmistaakseni, että muuttunut kuvakoko varmasti päivittyy sivustolle.

Näiden optimointien lisäksi poistin mobiiliversiosta turhat suuret kuvat, jotka eivät kuitenkaan näkyisi kunnolla mobiiliversiossa. Lisäksi vähensin muutaman raskaan ulkopuoliselta sivulta ladattavan fontin käyttöä. Sain tämän kaiken jälkeen parhaimmillaan lopputulokseksi mobiililla 60 ja työpöydällä 87 (Kuva 23 ja Kuva 24). Olen tyytyväinen näihin tuloksiin, koska mobiiliversio toimii nyt omiin kriteereihini nähden riittävän nopeasti.



Kuva 23. Mobiiliversion päivitetty tulos.



Kuva 24. työpöytäversion päivitetty tulos.

SSL-suojauksen ja evästeiden käyttöilmoituksen asentamisen jätin viimeiseksi tehtäväksi. Tätä varten lurasin Really Simple SSL -lisäosan, joka hoitaa sivustolle yhdistetyn SSL-todistuksen asettamisen WordPressin eri osiin ja lisäosiin. Kun käynnistin lisäosan

automaattisen asennusavustajan, sain ilmoituksen, että sivustolla ei ole SSL-todistusta. Vaikka olin saanut SSL:n GitHubin opiskelijaetupaketin mukana, se ei ollut automaattisesti käynnistynyt Namecheap-palvelun kautta. Seuraten Namecheapin SSL-asennusohjeita (Namecheap 2021) sekä SSL-todisteen luontiin liittyvää videoa (YouTube 2021) onnistuin aktivoinnissa osittain, mutta SSL-aktivointi jäi tilaan ”pending” eli avoin. Pyysin Namecheapin tuesta apua ja saimme heidän kanssaan hoidettua asian kuntoon kahdessa tunnissa. Koska ilmainen paketti ei ollut niin sanottu kokonaisvaltainen verkkopalvelu, piti SSL-tunnukset luoda ja tuoda palveluun itse. Tämän lisäksi ne piti yhdistää verkko-osoitteeni DNS-asetuksista, jonka jälkeen sain tarvittavat tunnustiedot sähköpostiini. Tunnustiedot syötin EasyWP palvelun puolella sivustolle ja heti kun aktivoin ne, sivustolleni ilmestyi HTTPS-merkintä ja Google Chrome hyväksyi ainakin etusivun turvalliseksi. Lopuksi suoritin Really Simple SSL:n asennuksen loppuun, joka varmisti sen, että koko sivusto ja sen kaikki osat olivat SSL- salauksen alla

Evästeiden keräämisen ilmoitusta varten lisäsin Cookie Notice & Compliance for GDPR/CCPA -lisäosan. Tämä on yksinkertainen lisäosa, joka pyytää suostumusta evästeiden käyttöön. Evästeistä kerrotaan sivuston tietoturvaselosteessa eli Privacy Policy -sivulla. Sivulla myös kerrotaan, miten käyttäjätunnuksia käsitellään ja miten käyttäjien luoma sisältö käyttäytyy sivulla ja kuka tätä sisältöä voi nähdä.

Korjauksien ja muokkauksien jälkeen sivusto oli teknisesti valmis käyttöön. Nyt käyttäjät voivat kirjautua sisään ja laatia sivustolle sarjakohtaisia oppaita. Käyttäjät voivat myös arvioida ja kommentoida toistensa oppaita. Onnistuin siis yhteisöllisen verkkopalvelun prototyypin luomisessa lähes täysin ilman koodaamista, mikä oli opinnäytetyöni tavoite.

6 LOPPUPÄÄTELMÄT

Opinnäytetyössä valmistettiin TV-sarjojen jaksoarviointia tukeva verkkosivusto. Tämä sivusto toteutettiin WordPress-alustalla, joka mahdollisti sivuston kehittämistä ilman ohjelmointia tai koodaamista. WordPressiin ladatut lisäosat mahdollistivat visuaalisten sivumuokkaimien käytön sekä muiden lisätoiminnallisuuksien käyttöönoton. Käytetyin lisäosa sivustolla on visuaalinen sivustonmuokkain, Elementor. Verkkosivusto toteutettiin ajankohtaisesta tarpeesta nopeuttaa TV-sarjojen kulutusta, sillä suoratoistopalvelut ovat vaikuttaneet tapaan, jolla sarjoja nykyään kulutetaan. TV-sarjojen kulutusnopeutta tukee kirjautuneiden käyttäjien mahdollisuus luoda sarjakohtaisia jakso-oppaita. Oppaiden avulla käyttäjät voivat ohjeistaa muita sarjan katsojia ohittamaan turhat jaksot optimoiden heidän ajankäyttönsä. Muut kirjautuneet käyttäjät voivat arvioida ja kommentoida jakso-oppaita ja näin käyttäjät voivat vaikuttaa siihen, mitkä oppaat ovat tarkkoja ja yleisesti oikeaksi todettuja.

Vaikka sivusto on teknisesti toiminnassa osoitteessa theskippable.me, puuttuu sieltä sisältö. Aitoa sisältöä en tähän prototyypiprojektiin itse lisää. Kuten tarkoituksena oli, on sivusto toteutettu toimivasti WordPress-alustalla lähes kokonaan koskematta riviinkään ohjelmointikieltä tai koodia.

Etusivu muuttui melko paljon alkuperäisestä suunnitelmasta. Osittain siksi, että Envato-elementsin avulla sain tehtyä hienoja elementtejä nopeammin ja osittain siksi, että halusin etusivun seuraavan ensin tekemäni sarjakohtaisen sivun tyyllittelyä.

Mobiiliversion etusivu, niin kuin kaikki muutkin sivut, seuraavat työpöytäversion tyyllittelyä melko tarkasti ja mielestäni tämä luo hyvän yhtenäisen kokemuksen. Koen tärkeäksi sen, että käyttäjä tunnistaa olevansa samalla sivustolla laitteesta huolimatta.

Projektin toteutus oli erittäin mielenkiintoista. Koska palvelu käyttää paljon erilaisia WordPressin osia, koen että tunnen WP:n nyt erittäin hyvin, vaikka en sen koodia juuri-kaan tutkinut. Projektissa oli vaiheita, jolloin yritin etsiä lisäosien ja teemojen koodista kohtia, joilla voisin muokata tarvittavia asioita. Onnistuin kuitenkin toteuttamaan suurimman osan tarvittavista muokkauksista lisäosilla. Isoin asia, joka jäin hampaankoloon oli sivustolla latautuva CSS- koodi, joka kutsuu fontawesome.com-sivuston resursseja kuten joitain ikoneja ja fontteja. En saanut selville, mikä lisäosa tai teema näitä haki sivulle.

Jos olisin onnistunut saamaan sen selville, olisin voinut muokata sitä ja olisin ehkä saanut nostettua sivuston latausnopeuden pistearvoa.

Sivustoa rakentaessani eniten aikaa kului käyttäjien rekisteröitymisen mahdollistamiseen. Koska en suostunut uskomaan, että WP vaatii uusien käyttäjien rekisteröitymisen sähköpostilla, käytin paljon aikaa tämän ominaisuuden kiertämiseen. Lopulta päädyin vaihtoehtoon, jota oli testannut sen vaiheen alussa ja se toimi käyttötarkoitukseeni nähdessä tarpeeksi hyvin.

6.1 Asioita, joita olisin voinut tehdä toisin

Elementor toimi projektin toteuttamisessa melko hyvin. Mikäli sivusto olisi ollut staattisempi ja vaatinut visuaalista niin Elementor olisi tähän erittäin loistava valinta. Koska sivuston ensisijainen tarkoitus ja tärkein sisältö ovat oppaat, olisin voinut toteuttaa sivuston pelkällä WordPressin omalla sisältömuokkaimella tai tarvittaessa Gutenbergillä, joka on hieman kehittyneempi versio tästä muokkaimesta. Sivuston sisällön päivittyvyyden takia, sivuston välimuistilla ei voinut juurikaan nopeuttaa sivun toimintaa. Koska sivustolla on dynaamisesti muuttuvia elementtejä, välimuisti ei voi niin useasti tallentaa ja tuoda niitä esiin.

Mikäli sivustoni ei olisi tarvinnut artikkelisivua eli Guides-sivua, olisin voinut valita jonkun Elementorin kanssa paremmin yhteensopivan ja kevyemmän teeman kuten elementorin oman Hello-teeman. Tämä teema ei kuitenkaan olisi mahdollistanut artikkelisivun muokkausta Astra-teeman tavoin. Vaihtoehtoisesti olisin voinut tehdä sivuston WordPressin omalla sivustomuokkaimella, mutta tämä taas olisi rajoittanut muokkausmahdollisuuksia. Sujay Pawar toteaa artikkelissaan Astra vs Elementor Hello Theme: Which One is Best for You?, että Hello on Astraa kevyempi teema (Pawar 2021). Elementor sivustonrakenustyökalu hidastaa sivua (Pawar 2021), joten mikäli haluaisin tehdä nopeat mobiilisivut helpommalla, kannattaisi minun käyttää pelkkää teemaa sekä WP:n omaa sivustomuokkainta. Tämä rajoittaisi sivuston visuaalista ilmettä mobiilissa, mutta toisaalta mobiilisivuston pääsisältö on tärkein (Clearbridge Mobile 2020).

Yksittäisen sivun muokkaukseen olisi pitänyt käyttää jo aluksi enemmän aikaa. En ollut testannut yksittäisen sivujen responsiivisuutta kunnolla ennen kuin siirryin eteenpäin. Kun testausvaiheessa aloin käymään sivuja yksitellen läpi eri resoluutio ja laite konfiguraatioissa huomasin, että sivut eivät olleetkaan responsiivisia. Toimin projektin edetessä

niin, että loin ensin sivujen sisällön ja päätin palata toiminnallisuuksien hiomiseen myöhemmin. Vaikka tämä edistikin motivaatiota projektin suhteen, teki sen responsiivisuuden korjaamisesta hitaampaa, sillä olin lähes pakottanut elementit toimimaan ja näyttämään hyvältä työpöytäversiossa.

Testausta olisi tullut suorittaa yhtäaikaaisesti sivuston kehittämisen kanssa. Projektiin käytetty aika kului miltei kokonaan sen toteuttamiseen. Mikäli olisin testannut palvelua kokonaisvaltaisesti yhden toiminnallisuuden testaamisen sijaan, olisin mahdollisesti huomannut ongelmakohtat ennen kuin ne vaikuttivat useampaan kohtaan tai ennen kun tein samat virheet uudestaan.

Minun olisi lisäksi pitänyt seurata fontteja ja kuvien kokoa jo alusta asti. Jotta saisin sivustoa nopeammaksi, olisi minun pitänyt jatkuvasti seurata, missä vaiheessa tietyt sivustoa hidastavat elementit ja resurssit alkavat vaikuttaa. Näin olisin voinut minimoida niiden aiheuttamat ongelmat.

Koska sivusto on rakennettu WP-alustalle, se ei välttämättä ole paras dynaamisen sisällön ylläpitämiseen. Vaikka WordPress onkin tarkoitettu alun perin blogien ja artikkelien kirjoittamiseen, ei se täysin optimaalinen suuren sivumäärän ylläpitämiseen ja niiden dynaamisesti esille nostamiseen. Tämän pitäisi olla mahdollista kuten WPBeginnerin artikkelissa mainitaan (WPBeginner 2014) esimerkiksi lisäämällä ”lapsi”-sivuja, mutta tällainen vaatisi sivustolla ohjelmoimista, joka taas on asia, jonka pyrin pitämään sivustolla erittäin minimissä. Tähän myös löytyi lisäosia, mutta koska ne ei ollut Elementorille yhteensopivia olivat ne tyyllittelyltään erittäin simppeleitä tai seurasivat artikkelien asetuksia. Näiden muokkaus olisi vaatinut CSS-koodia.

6.2 Muuta pohdintaa ja jatkokehitysideoita

Koska opinnäytetyössä toteutettava projekti oli ennemminkin prototyyppin kaltainen ja niin sanottu proof of concept, ei sinne ole vielä luotu aitoa sisältöä. Mahdollinen jatkokehityksen kohde sivulla olisi sivuston markkinointi verkossa. Tämä markkinointiosuus ei kuitenkaan sisälly tähän opinnäytetyöhön, vaan se olisi seuraava askel. Sivusto toimii tällä hetkellä ennemminkin näyttönä siitä, että tämän kaltainen projekti on mahdollista toteuttaa WordPress-alustalla, mikäli ylläpitäjät voivat toimia aktiivisena osana sivustoa ja mikäli sivuston nopeus ei ole ensisijainen prioriteetti.

Opinnäytetyönä luomani sivuston toinen tarkoitus on myös toimia niin sanotusti konseptin esittäjänä. Mikäli sivustoa markkinoisi ja jatkokehittäisi, tulisi se tarvitsemaan paljon enemmän dynaamista sisältöä ja automatisoituja sivunluomisprosesseja. Näin käyttäjät voisivat luoda oppaita sarjoille, joita ei ole vielä lisätty sivustolle. Järjestelmä voisi luoda sivupohjan avulla uuden sivun hakemalla oikean sarjan, jolle käyttäjä luo listaa. Kokeukseni ja oppimani perusteella en usko, että WordPress pystyisi tällaiseen. Se tarvitsisi vähintään tietokantaan vaikuttavaa koodia, joka päivittyisi ja osaisi itse etsiä internetistä tarvittavat resurssit sivuston rakentamiseen. Tämä sivusto tällaisenaan on kuitenkin hyvä alustavan käyttäjä- ja vierailijatiedon keräämiseen, mikäli sivuston ylläpitäjä on aktiivinen yhteisön rakentamisessa ja sivuston päivittämisessä. Näillä käyttäjätiedoilla sivustoa pystyy esimerkiksi esittelemään mahdollisille sijoittajille ja keräämään rahoitusta, jolla pystyisi palkkaamaan esimerkiksi automatisoitujen toimintojen kehittäjiä tai kustantamaan lisätoimintojen opettelua ja kehittämistä.

LÄHTEET

Chamary, JV. (2021). Dark Mode Saves Battery Life, But You Might Not Notice. <https://www.forbes.com/sites/jvchamary/2021/07/31/phone-battery-life-dark-mode/>. Haettu 13.10.2021.

Cheded, F. (2017). How Memes Help Form Today's Culture Around Movies and TV. <https://film-schoolrejects.com/memes-movies-tv/>. Haettu 15.8.2021.

Clearbridge Mobile (2020). Mobile App Design Fundamentals: 10 Tips for an Effective Content Strategy. <https://clearbridgemobile.com/mobile-app-design-fundamentals-10-tips-for-an-effective-content-strategy/>. Haettu 13.11.2021.

Cohen, F. (2017). How 'Twin Peaks' Got Lost, and Found Its Way Back. <https://www.nytimes.com/2017/05/17/arts/television/twin-peaks-showtime-david-lynch-interview.html>. New York Times. The New York Times Company. Haettu 2.11.2021.

Crestodian, A. (2015). Website Footer Design Best Practices: 27 Things to Put at the Bottom. <https://www.orbitmedia.com/blog/website-footer-design-best-practices/>. Haettu 13.11.2021.

digicert (2021). What is SSL, TLS and HTTPS? <https://www.websecurity.digicert.com/security-topics/what-is-ssl-tls-https>. Haettu 8.12.2021.

Elementor (2021). 502 Error When I Try to Edit with Elementor. <https://elementor.com/help/502-error-when-i-try-to-edit-with-elementor/>. Haettu 12.11.2021

Elementor (2021). How to Create a WordPress Single-Post Template in Elementor. <https://elementor.com/blog/design-your-single-post-template/>. Haettu 12.11.2021.

Elementor (2021). <https://elementor.com/>. Haettu 12.11.2021.

Episodi (2021). <https://www.episodi.fi/>. 8.12.2021.

Fessende, T. (2018). Scrolling and Attention. <https://www.nngroup.com/articles/scrolling-and-attention/>. Haettu 5.11.2021.

Fitton, C. (2019). Here's Why EasyWP Is the Fastest WordPress Hosting Around <https://www.easywp.com/blog/heres-why-easywp-is-the-fastest-wordpress-hosting-around/>. Haettu 13.10.2021.

Fitzgerald, A. (2020). Website Footers: Best Design Practices & Examples. <https://blog.hubspot.com/website/website-footer>. Haettu 13.11.2021.

GitHub (2021). GitHub Student Developer Pack. <https://education.github.com/pack>. Haettu 13.10.2021.

Google (2021). <https://developers.google.com/speed/pagespeed/insights/>. Haettu 05.11.2021

IMDb (2021). <https://www.imdb.com/>. Haettu 8.12.2021.

Ivanova, A.A., Srikant, S., Sueoka, Y., Kean, H.H., Dhamala, R., O'reilly, U.M., Bers, M.U. and Fedorenko, E. (2020). Comprehension of computer code relies primarily on domain-general executive brain regions. *Elife*, 9, p.e58906.

Johnson, J. (2020). GitHub launches Android app in beta. <https://venturebeat.com/2020/01/14/github-launches-android-app-in-beta/>. Haettu 13.10.2021.

Jurgensen, J. (2012). Binge Viewing: TV's Lost Weekends. <https://www.wsj.com/articles/SB10001424052702303740704577521300806686174>. Haettu 31.8.2021

- JustWatch (2021). https://www.justwatch.com/us/tv-shows?rating_imdb=9 . Haettu 12.11.2021
- Lindgaard, G., Fernandes, G., Dudek, C. and Brown, J. (2006). Attention web designers: You have 50 milliseconds to make a good first impression!. *Behaviour & information technology*, 25(2), pp.115-126.
- Marinescu, V., Branea, S. and Mitu, B. eds. (2014). *Contemporary Television Series: Narrative Structures and Audience Perception*. Cambridge Scholars Publishing.
- Merikivi, J., Bragge, J., Scornavacca, E. and Verhagen, T. (2020). Binge-watching serialized video content: A transdisciplinary review. *Television & New Media*, 21(7), pp.697-711.
- Milyavskaya, M., Saffran, M., Hope, N. and Koestner, R. (2018). Fear of missing out: prevalence, dynamics, and consequences of experiencing FOMO. *Motivation and Emotion*, 42(5), pp.725-737.
- Mobiteam (2020). What is Dark Mode and Why is it a Web Design Trend in 2020. <https://mobiteam.medium.com/what-is-dark-mode-and-why-is-it-a-web-design-trend-in-2020-c638b6f22e17>. Haettu 13.10.2021.
- MySQL (2021). What is MySQL? <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>. Haettu 12.11.2021.
- Namecheap (2021). How do I activate an SSL certificate. <https://www.namecheap.com/support/knowledgebase/article.aspx/794/67/how-do-i-activate-an-ssl-certificate/>. Haettu 06.11.2021.
- OLED-info (2021). An introduction to OLED displays. <https://www.oled-info.com/oled-introduction>. Haettu 8.12.2021.
- Pawar, S (2021). Astra vs Elementor Hello Theme: Which One is Best for You? <https://wpastra.com/elementor-hello-theme-vs-astra-review/>. Haettu 12.11.2021.
- Robert Ebert.com (2021). <https://www.rogerebert.com/>. 8.12.2021.
- Rotten Tomatoes (2021). Game of Thrones: Season 5. https://www.rottentomatoes.com/tv/game_of_thrones/s05 . Haettu 2.9.2021.
- Rotten Tomatoes (2021). <https://www.rottentomatoes.com/>. 8.12.2021.
- Sandler, A. (2011). What time is really primetime. <https://www.nielsen.com/us/en/insights/article/2011/what-time-is-really-primetime/> . Haettu 30.7.2021
- Shokurova, K. (2020). Website Header Design in 2020: Best Practices and Examples. <https://uxplanet.org/website-header-design-in-2020-best-practices-and-examples-1992f80ddd69>. Haettu 27.10.2021.
- Speedtest (2021). Speedtest Global Index. <https://www.speedtest.net/global-index>. Haettu 13.11.2021.
- Stack Exchange (2021). how to remove email field from default user registration form on wordpress. <https://wordpress.stackexchange.com/questions/383672/how-to-remove-email-field-from-default-user-registration-form-on-wordpress>. Haettu 13.11.2021.
- Stack Overflow (2014). wordpress remove email when user register.<https://stackoverflow.com/questions/25947954/wordpress-remove-email-when-user-register>. Haettu 13.11.2021
- W3C (2016). HTML & CSS. <https://www.w3.org/standards/webdesign/htmlcss.html>. Haettu 12.11.2021.

Walters, T. (2021). Why Arrested Development Was Canceled (& Why It Came Back). <https://screenrant.com/arrested-development-canceled-why-return-netflix/>. Haettu 2.11.2021.

web.dev (2019). Lighthouse performance scoring. <https://web.dev/performance-scoring/>. Haettu 05.11.2021.

WordPress.org (2021). Democratize Publishing. <https://wordpress.org/about/> . Haettu 12.11.2021.

WordPress.org (2021). Roles and Capabilities. <https://wordpress.org/support/article/roles-and-capabilities/>. Haettu 13.11.2012.

WPBeginner (2014). How to Display a List of Child Pages For a Parent Page in WordPress. <https://www.wpbeginner.com/wp-tutorials/how-to-display-a-list-of-child-pages-for-a-parent-page-in-wordpress/>. Haettu 13.11.2021.

WPBeginner (2016). How to Limit Dashboard Access in WordPress. <https://www.wpbeginner.com/plugins/how-to-limit-dashboard-access-in-wordpress/>. Haettu 05.11.2021.

WPbeginner (2021). How to Disable WordPress Admin Bar for All Users Except Administrators. <https://www.wpbeginner.com/wp-tutorials/how-to-disable-wordpress-admin-bar-for-all-users-except-administrators/>. Haettu 05.11.2021.

wppluginsforyou.com (2019). How to create users without Email in WordPress? <https://www.wppluginsforyou.com/how-to-create-users-without-email-in-wordpress/>. Haettu 13.11.2021.

WPQuestions (2012). Remove email address on WP registration? https://wpquestions.com/Remove_email_address_on_WP_registration/4239. Haettu 13.11.2021.

YouTube (2021). How to generate CSR for an EasyWP website. <https://www.youtube.com/watch?v=dJTMDro5xlo>. Haettu 13.11.2021