

Opinnäytetyö (AMK)

Tietotekniikka

2020

Iiro Järvenpää

DRUPAL 8 -SIVUSTON DYNAAMINEN ULKOASU

Iiro Järvenpää

DRUPAL 8 -SIVUSTON DYNAAMINEN ULKOASU

Sosiaalisen median palvelut ja suoratoistopalvelut ovat nostaneet verkkosivujen käytettävyyden uudelle tasolle. Saumattomat siirtymät sivuston osien välillä saavat nämä käytettävyydeltään erinomaiset verkkosivut tuntumaan sekä yhdeltä suurelta kokonaisuudelta että dynaamisilta. Tämä asettaa uusia haasteita perinteisten verkkosivujen kehityksessä. Jotta näihin sivuston käyttäjien kasvaviin odotuksiin voidaan vastata, on verkkosivujen kehityksessä hyvä huomioida dynaamisen ulkoasun mahdollistavat viitekehukset.

Tässä opinnäytetyössä tutkittiin tapoja toteuttaa Citrus Solutions Oy:lle Drupal 8 -sivuston ulkoasu käyttämällä ulkoisia viitekehysia ja luodaan esimerkkisivusto hyväksikäyttäen Drupalia sisällönhallinnassa. Opinnäytetyön tavoitteena oli löytää tähän tarkoitukseen sopiva viitekehys. Viitekehysten sopivuutta tarkasteltiin sekä yrityksen että opiskelijan näkökulmasta.

Työ toteutettiin testaamalla ja arvioimalla Drupalin tukemia JavaScript-viitekehysia. Opinnäytetyössä valikoitiin testattavaksi kolme suosittua sivustojen ulkoasuun sopivaa JavaScript-viitekehystä: Vue.js, React.js ja Angular.

Esimerkkisivusto toteutettiin lopulta hyödyntämällä Vue.js:ää. Viitekehysten valintaan vaikutti kyseisen viitekehysten yksinkertainen syntaksi kehittäjän näkökulmasta ja mahdollisuus sisällyttää se eri projekteihin useammalla eri tavalla. Työ selvitti onnistuneesti yritykselle mahdollisia eri ratkaisuja Drupal 8 -sivuston ulkoasun luontiin.

ASIASANAT:

Julkaisujärjestelmät, Sisällönhallinta, Internet

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information technology

2020 | 34 pages

Iiro Järvenpää

DYNAMIC FRONT-END OF DRUPAL 8 PAGE

Social media and streaming services have raised the user experience of website to a new level. Seamless transitions between pages of these services make them dynamic and feel like they are one page. This presents new challenges for developing traditional web pages. It's important to take new framework into account in order to answer raising expectations of website users.

This thesis researched existing front-end frameworks which can be used to create Drupal 8 page and creates example webpage while using Drupal 8 as content management system. The goal of this thesis was to find suitable framework for this purpose. The suitability of the the framework was evaluated both from the perspective of the company and student.

Thesis was carried out by testing and evaluating JavaScript frameworks that were supported by Drupal. Three popular JavaScript front-end frameworks were chosen to be tested: Vue.js, React.js and Angular.

Example webpage was created with Vue.js. The Choice was based on the simple syntax of the framework and the possibility of progressively implementing framework to projects. Thesis successfully researched different way to utilize JavaScript-framework while creating front-end of Drupal 8 site.

KEYWORDS:

Publishing systems, Content management, Internet

SISÄLTÖ

| | |
|--|-----------|
| KÄYTETYT LYHENTEET JA SANASTO | 6 |
| 1 JOHDANTO | 7 |
| 2 SISÄLLÖNHALLINTAJÄRJESTELMÄ | 9 |
| 2.1 Sisällönhallintajärjestelmän ominaisuudet | 9 |
| 2.1.1 Yleiset järjestelmäominaisuudet | 10 |
| 2.1.2 Vakio-ominaisuudet | 10 |
| 2.1.3 Lisäominaisuudet | 11 |
| 2.2 Avoin vai suljettu lähdekoodi | 11 |
| 2.3 Markkinatilanne | 13 |
| 2.3.1 WordPress | 13 |
| 2.3.2 Shopify | 14 |
| 2.3.3 Joomla! | 14 |
| 2.3.4 Wix | 15 |
| 2.3.5 Drupal | 16 |
| 3 VIITEKEHYKSET | 17 |
| 3.1 Viitekehukset ja Decoupled CMS | 18 |
| 3.1.1 React.js | 20 |
| 3.1.2 Angular | 21 |
| 3.1.3 Vue.js | 22 |
| 4 TOTEUTUS | 23 |
| 4.1 Tehtävänasettelu | 23 |
| 4.2 Haasteet ja riskit | 23 |
| 5 DYNAAMISEN DRUPAL 8 -ULKOASUN KÄYTTÖÖNOTTO ESIMERKKISIVUSSA | 24 |
| 5.1 Teema | 24 |
| 5.2 Sisällön luonti ja sen hallinta | 26 |
| 5.3 Sisällön hakeminen | 26 |
| 5.4 Vue.js:n tiedostorakenne | 27 |
| 5.5 Lopputulos | 30 |

| | |
|---------------------|-----------|
| 6 YHTEENVETO | 31 |
| 7 LÄHTEET | 33 |

KUVAT

| | |
|--|----|
| Kuva 1. Sisällönhallintajärjestelmien osuus www-sivuista (w3techs, 2020). | 13 |
| Kuva 2. Esimerkki samasta koodista JavaScriptillä ja JQueryllä. | 17 |
| Kuva 3. Bootstrap grid systemin käyttöesimerkki (getbootstrap.com). | 18 |
| Kuva 4. Monoliittinen ja irrotettu sisällönhallintajärjestelmä (Dries Buytaert, Acquia, 2018). | 19 |
| Kuva 5. Yhdensuuntainen tiedonsiirto komponenttien välillä. | 21 |
| Kuva 6. Kuva teeman rakenteesta. | 25 |
| Kuva 7. Vue_thesis.libraries.yml tiedostosta. | 26 |
| Kuva 8. Kuva näkymästä ja sen asetuksista. | 27 |
| Kuva 9. Kuva teeman js-kansion rakenteesta. | 28 |
| Kuva 10. Listaus ensimmäisestä näkymästä. | 29 |

KÄYTETYT LYHENTEET JA SANASTO

| | |
|------------|---|
| CMS | Content Management System, sisällönhallintajärjestelmä, julkaisujärjestelmä. |
| CSS | Cascading style sheets, verkkosivujen ulkoasun määrittelykieli. |
| DOM | Document Object Model, dokumenttioliomalli. Tapa kuvata dokumentin hierarkia puumallina. |
| Drupal 8 | Drupal-sivustoissa käytettävä sisällönhallintajärjestelmä. |
| Framework | Viitekehys. Sisältää valmiita luokkia ja/tai kirjaston, jota hyödynnetään ohjelmoinnissa. |
| Front-end | Sivuston tai järjestelmän osa, joka ajetaan verkkoselaimessa sivua käytettäessä. |
| HTML | Hypertext Markup Language, kuvauskieli, jolla verkkosivut on kirjoitettu. |
| JavaScript | Web-kehityksessä käytetty skriptikieli. |
| MySQL | Relaatiotietokantaohjelmisto |
| Templaatti | Mallipohja, mallinne. |
| XML | Extensible Markup Language. Merkintäkielien standardi. |

1 JOHDANTO

Verkkosivujen ulkoasujen kehityksessä on tapahtunut vuosien saatossa paljon muutoksia. Käyttöjärjestelmän tekstieditori ei ole enää ainoa mahdollinen työkalu sivujen luontiin HTML-kieltä käyttäen. Tekstieditorit ovat oppineet tunnistamaan käytetyn ohjelmointikielen ja kääntämään koodin kieleksi, jota selain ymmärtää. Verkkosivujen kehitystä nopeuttamaan on myös luotu suuri määrä erilaisia viitekehyskielisiä.

Vuosi 2020 on ollut tapahtumarikas myös verkkosivujen kehityksessä. Maailmanlaajuisen koronaviruspandemia ja sen vaikutukset tavallisten ihmisten elämään ovat vauhdittaneet digiloikan ottamista. Siinä missä yrityksillä on ollut tarve siirtää toimintaansa yhä enemmän internetiin toimintansa jatkamiseksi, ovat ihmiset koronarajoitusten seurauksena viettäneet enemmän aikaa suoratoistopalveluiden ja sosiaalisen median parissa.

Suoratoistopalvelut, kuten Netflix ja Instagramin kaltaiset sosiaalisen median palvelut, pyrkivät pitämään käyttäjänsä mahdollisimman tyytyväisenä, ja saamaan heidät viettämään sivustolla mahdollisimman paljon aikaa. Nämä palvelut tuntuvat yhdeltä suurelta kokonaisuudelta. Sivustot reagoivat käyttäjänsä dynaamisesti, ja esimerkiksi siirtyminen sivulta toiselle on jouhevaa ilman erillistä sivulatausta. Nämä sivustot hyödyntävät ulkoasunsa toteutuksessa ulkoista viitekehystä, jonka avulla toteutetaan niiden dynaaminen ulkoasu.

Tämän opinnäytetyön tavoitteena on tutkia tapoja toteuttaa Drupal 8 -sivuston ulkoasu käyttämällä ulkoisia viitekehyskielisiä, ja luoda esimerkisivusto hyväksikäyttäen Drupalia sisällönhallinnassa. Aihe valikoitui yhteistyössä työnantajani Citrus Solutionsin kanssa pohdittaessa mahdollisia aiheita opinnäytetyölleni. Valintaan vaikuttivat sekä kiinnostukseni sivustojen ulkoasujen toteutusta kohtaan että pääasiallinen työtehtäväni niiden tekijänä Citrus Solutionsilla. Opinnäytetyössä keskitytään niihin viitekehyskielisiin, joita Drupal 8 -sisällönhallintajärjestelmä tukee.

Opinnäytetyön teoriaosassa käsitellään aluksi yleisesti sisällönhallintajärjestelmien ominaisuuksia, tarkastellaan avoimen ja suljetun lähdekoodin vaikutusta kehittämiseen ja tutkitaan sisällönhallintajärjestelmien markkinaosuuksia. Lopuksi luvussa tutustutaan Drupaliin ja sen kilpailijoihin.

Teoriaosa etenee esitellen viitekehysten ominaisuuksia. Ominaisuuksia käsitellään yleisesti, ja verkkosivukehityksen näkökulmasta. Luvussa tutustaan sivuston rakenteeseen,

jossa sisällönhallintajärjestelmä toimii taustalla ja viitekehyksellä toteutetaan sivuston ulkoasu. Teoriaosa loppuu kolmen suositun viitekehysten esittelyyn.

Opinnäytetyön toteutusosassa tehdään ja perustellaan viitekehysten valinta. Esimerkksivuston toteutuksessa käsitellään sisällönhallinnan eri vaiheet ja sisältöjen luominen Drupalissa. Luodut sisällöt jaetaan ulkoisen viitekehysten käytettäväksi ja ne esitetään sen avulla luodussa yksisivuisessa sivussa. Lopuksi toteutustyössä arvioidaan valitun viitekehysten sopivuus käyttötarkoitukseensa.

2 SISÄLLÖNHALLINTAJÄRJESTELMÄ

Sisällönhallintajärjestelmä, englanniksi content management system, on nimitys yksittäiselle tai useammalle toisiinsa liittyvälle työkalulle, jonka avulla hallitaan ja luodaan järjestelmässä olevaa sisältöä. Verkkosivujen sisällönhallintajärjestelmien tapauksessa järjestelmä mahdollistaa sisällön tuottamisen, muokkaamisen ja poistamisen verkkosivuilta ilman, että käyttäjältä vaaditaan ohjelmointitaitoa. (Rouse, 2019) Tämän opinnäytetyön puitteissa sisällönhallintajärjestelmällä viitataan järjestelmiin, joilla hallitaan internetsivujen sisältöä.

Suomen kielessä sisällönhallintajärjestelmä-termin rinnalla käytetään usein myös termiä julkaisujärjestelmä. Näiden termien välillä ei ole tarkkaa jaottelua, mutta julkaisujärjestelmien tärkeimpänä ominaisuutena voidaan pitää mahdollisuutta julkaista jotain jossain. Julkaisujärjestelmällä voidaan myös kuvata suuremman sisällönhallintajärjestelmän osaa, joka vastaa sisällön julkaisusta ja siihen liittyvistä ominaisuuksista. Yleistäen voidaan todeta, että mitä monimutkaisempi järjestelmä, sen helpompi sitä on kuvailla sisällönhallintajärjestelmäksi julkaisujärjestelmän sijaan. (Tolvanen, 2009)

Internet on täynnä erilaisia sivustoja, jotka tarjoavat käyttäjilleen erilaisia ominaisuuksia. Näiden sivustojen taustalla toimivia järjestelmiä on miltei yhtä paljon, ja uusia järjestelmiä kehitetään koko ajan. Eri järjestelmiä on saatavilla eri kokoisille sivustoille aina kattavista verkkokauppojen hallintaan suunnitelluista järjestelmistä blogikirjoitusten julkaisemiseen (Tolvanen, 2009).

2.1 Sisällönhallintajärjestelmän ominaisuudet

Järjestelmän keskeisiä ominaisuuksia ovat sisällön hallinta, asiasanoittaminen ja hakeminen, sekä versiohallinta ja julkaisu. Lisäksi suosittuja ominaisuuksia ovat esimerkiksi tiedostojenhallintaominaisuudet, ylläpitäjäpaneelin tuki useille eri kielille ja hakukoneoptimoidut Web-osoitteet. (Rouse, 2019) Pro gradu -tutkielmassaan, joka käsittelee web-sisällönhallintajärjestelmien ominaisuuksia ja käyttöönottoa, Perttu Tolvanen (2007) jaottelee sisällönhallintajärjestelmän ominaisuudet kolmeen kategoriaan: yleisiin järjestelmäominaisuuksiin, vakio-ominaisuuksiin ja lisäominaisuuksiin.

2.1.1 Yleiset järjestelmäominaisuudet

Sisällönhallintajärjestelmällä voidaan olettaa olevan lista yleisiä ominaisuuksia, jotka mahdollistavat mielekkään sisällönhallinnan. Näihin perusominaisuuksiin voidaan lukea muun muassa tietoturvallisuus, mahdollisuus erilaisille käyttäjärooleille ja -oikeuksille ja sivujen kieliversiointi (Tolvanen, 2007).

Drupalin tapauksessa voidaan todeta, että se ei poikkea yleisestä sisällönhallintajärjestelmien linjasta. Varsinkin Drupalin tärkeimpään osaan, ytimeen (englanniksi core), julkaistaan aktiivisesti tietoturvapäivityksiä. Ytimeen sisältyvät kaikki sisällönhallintajärjestelmän perusominaisuudet.

Kieliversiointi on olennainen ominaisuus, varsinkin Suomen kaltaisessa kaksikielisessä maassa. Useimmiten sivujen halutaan tukevan vähintään suomea, ruotsia ja englantia. Drupalissa on mahdollista vaihtaa ylläpitäjien kieltä tai asettaa niistä useampikin mahdolliseksi. Lisäksi järjestelmässä voidaan luoda sisällöstä toiselle kielelle käännettyjä versioita. Järjestelmään on helppo lisätä käyttäjiä eri rooleilla, joilla on erilaiset oikeudet esimerkiksi lisätä tai muokata sisältöä.

2.1.2 Vakio-ominaisuudet

Olennainen osa sisällönhallintajärjestelmää on mahdollisuus hallita sisältöä. Tällä voidaan tarkoittaa esimerkiksi tekstin kirjoittamista, erilaisten sivujen luontia tai olemassa olevan tekstin muokkausta. Sisällönhallintajärjestelmät pääasiallisesti tallentavat informaation tietokantaan tai hakevat informaatiota sieltä. Näitä tietokantakyselyitä ei kuitenkaan näytetä käyttäjälle, vaan ne näytetään visuaalisessa muodossa järjestelmän käyttöliittymän välityksellä. Järjestelmä muokkaa tietokantaa joko käyttämällä tietokantakäskyjä, tai omaa ohjelmointikieltään, kuten esimerkiksi PHP:ta Drupalin tapauksessa.

Sisällön muokkaamisen lisäksi on olennaista pystyä luomaan toisistaan rakenteellisesti eroavaa sisältöä samalle sivustolle, kuten esimerkiksi blogitekstejä sekä uutisia. Näihin erilaisiin sivupohjiin, Drupalin tapauksessa sisältötyyppeihin, on mahdollista sisällyttää toisistaan huomattavastikin poikkeavaa informaatiota.

Ylläpitäjien on kyettävä julkaisemaan sisältö joko suoraan tai ajastetusti sivuston muiden käyttäjien nähtäviksi. Loppukäyttäjän kannalta on mielekästä, että sisältöihin on mahdollista liittää asiasanoja, joiden avulla sisältöä voidaan jäsenellä. (Tolvanen, 2007)

2.1.3 Lisäominaisuudet

Yleisten ja vakio-ominaisuuksien lisäksi tiedostonhallintajärjestelmiin voi usein hallitusti lisätä ominaisuuksia toivotun lopputuloksen tuottamiseksi. Tolvanen (2007) luettelee lisäominaisuuksiksi esimerkiksi toiminnallisuuksien, personoinnin ja mittauksen hallinnan.

Drupalissa tämän kaltaisia lisäominaisuuksia koostetaan omiksi pienemmiksi koodikonaisuuksiksi, eli moduuleiksi, joita voidaan ladata, kytkeä päälle tai pois päältä tarpeen mukaan. Näin sisällönhallintajärjestelmän ylläpitäjä pystyy asentamaan esimerkiksi Webform-moduulin, jos hän tarvitsee sivustolle lomakkeita.

2.2 Avoin vai suljettu lähdekoodi

Erikoistumisalojensa lisäksi sisällönhallintajärjestelmät voidaan jakaa avoimen ja suljetun lähdekoodin mukaan. Avoin lähdekoodi on käyttäjänsä tarkasteltavissa ja muokattavissa. Puolestaan suljetun lähdekoodin käyttäjällä ei ole samanlaista mahdollisuutta vaikuttaa koodin toimintaan.

Maailmanlaajuinen avoimen lähdekoodin käyttöä tukeva Open Source Initiative (The Open Source Initiative, 2007) määrittelee sivuillaan avoimen lähdekoodin seuraavasti:

1. Ohjelman lisenssi ei rajoita sen myymistä tai levittämistä.
2. Ohjelman lähdekoodin täytyy tulla sen mukana tai olla helposti saatavissa.
3. Ohjelman lisenssi ei rajoita ohjelman muokkaamista tai muokatun ohjelman jakamista.
4. Ohjelman lähdekoodin levitystä voidaan rajoittaa vain, jos lisenssi mahdollistaa sellaisten korjaustiedostojen jakamisen, joihin sisältyy lähdekoodi.
5. Ohjelman lisenssi ei aseta ihmisiä tai ryhmiä eri asemaan.
6. Ohjelman lisenssi ei rajoita ohjelmiston käyttötarkoitusta.
7. Kaikilla, joille ohjelma on jaettu, on samat oikeudet.

8. Ohjelman lisenssi ei saa rajoittua suurempaan ohjelmistokokonaisuuteen, vaan koskee myös itsenäistä ohjelmaa.
9. Ohjelman lisenssi ei rajoita sen kanssa levitettäviä muita ohjelmia.
10. Ohjelman lisenssiä ei rajoiteta mihinkään teknologiaan tai käyttöliittymään.

Näin ollen kuka tahansa voi ladata esimerkiksi avoimen lähdekoodin sisällönhallintajärjestelmän, ja muokata sitä. Avoin lähdekoodi mahdollistaa sen, että nettisivuja tilaava asiakas voi valita kenet tahansa kehittämään nettisivujaan, tai vaikkapa vaihtaa nykyisestä toimittajastaan toiseen.

Suurimpia avoimen lähdekoodin etuja verrattuna suljettuun lähdekoodiin on se, että se on ilmainen. Asiakas ei joudu maksamaan ohjelmistosta lisenssimaksuja, ja hän omistaa tehdyn koodin kokonaisuudessaan.

Avoimen lähdekoodin ohjelmistojen ympärille on myös usein kehittynyt aktiivisia yhteisöjä, jotka yhdessä kehittävät ohjelmistoja eteenpäin, ja luovat niille uusia toiminnallisuuksia. Nämä uudet toiminnallisuudet ovat kehittyneet täyttämään ohjelmistojen käyttäjien tarpeita. Parhaassa tapauksessa ohjelmiston kehittäjä ei joudu itse kehittämään jo ker-
ran yhteisön toteuttamaa ominaisuutta uudelleen, vaan voi ladata sen valmiina pakettina.

Avoimuudella ja yhteisöllisyydellä on kuitenkin kääntöpuolensa. Koska ohjelmiston lähdekoodi on kaikkien nähtävissä, ovat myös niissä olevat tietoturva-aukotkin kaikkien nähtävissä. Nämä voivat pahimmassa tapauksessa aiheuttaa suuriakin tietoturvariskejä, kuten esimerkiksi vuoden 2018 maaliskuussa korjattu Drupal-sisällönhallintajärjestelmän ytimen Drupalgeddoniksi kutsuttu tietoturva-aukko, joka mahdollisti ulkopuolisen koodin suorittamisen sivustolla (Drupal, 2018). Avoimen lähdekoodin ohjelmisto onkin riippuvainen sen aktiivisesta yhteisöstä.

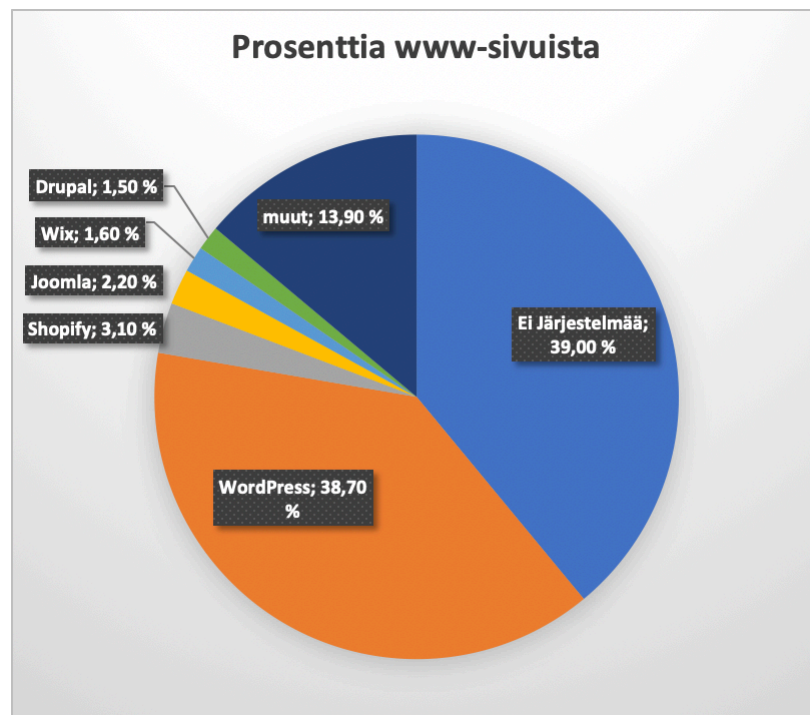
Suljetun lähdekoodin ohjelmistoa käyttävä asiakas maksaa lisenssimaksuja, mutta ei omista ohjelmiston koodia. Lisenssimaksut voivat olla kuukausittaisia tai kertaluontoisia. Rajattu pääsy koodiin tarkoittaa kuitenkin sitä, että tietoturvaongelmat ovat paremmin hallittavissa, koska ulkopuoliset tahot eivät pääse helposti etsimään koodista tietoturva-aukkoja.

Suurimpia ongelmia suljetussa lähdekoodissa on sitoutuminen järjestelmän toimittajaan. Koska asiakas ei omista koodia, hän joutuu useimmiten aloittamaan alusta vaihtaessaan toimittajasta toiseen. Samanlainen ongelma voi tulla vastaan myös järjestelmän toimittajan ajautuessa konkurssiin tai lopettaessa palvelun.

Parhaimmassa tapauksessa koodin omistajan ja asiakkaan välinen suhde on tiivistä yhteistyötä, jolla on selkeät tavoitteet, ja joka on paljon arvokkaampaa kuin lisenssimaksuista aiheutuneet kulut. Toisaalta lisenssimaksut voivat kattaa myös ohjelmiston ylläpidon.

2.3 Markkinatilanne

Eri sisällönhallintajärjestelmien välinen kilpailu on kovaa, ja se näkyy alati vaihtuvissa markkinaosuuksissa. Vielä vuosi sitten puhuttiin niin kutsutusta suuresta kolmikosta, jolla viitattiin kolmeen suurimpaan sisällönhallintajärjestelmään: WordPressiin, Joomlaan ja Drupaliin. Näiden rinnalle on viimeisen vuoden aikana noussut muun muassa verkkokauppoihin erikoistunut Shopify ja varsinkin pieniin sivustoihin erikoistunut Wix. (Kuva 1.)



Kuva 1. Sisällönhallintajärjestelmien osuus www-sivuista (w3techs, 2020).

2.3.1 WordPress

WordPress on Matt Mullenwegin ja Mike Littlen vuonna 2003 kehittämä vapaan lähdekoodin sisällönhallintajärjestelmä (Wordpress), jota käyttää 26.10.2020 38.7% kaikista

Internetin sivustoista, kuten kuvasta 1 nähdään. Sisällönhallintajärjestelmän kehitystä on ohjannut sarja kehittäjien periaatteita. Ohjelmiston tulee olla käyttövalmis mahdollisimman vähäisillä säädöillä. Se on suunniteltu keskivertokäyttäjälle, joka haluaa kirjoittaa ilman ongelmia. Käyttäjän ei tule joutua tekemään turhaan päätöksiä. Sisällönhallintajärjestelmän tärkeimmät ominaisuudet ovat ne, joita 80% tai enemmän käyttäjistä käyttää. (Wordpress)

WordPress käyttää ohjelmointikielenään PHP:ta, joka tallentaa ja noutaa dataa sisällönhallintajärjestelmän MySQL-tietokannasta. Tällä hetkellä järjestelmän uusin versio on 26.10.2020 julkaistu versio 5.5.1 (Wordpress, 2020).

2.3.2 Shopify

Vuonna 2004 perustettu Shopify on sähköisen kaupankäynnin ratkaisuihin erikoistunut suljetun lähdekoodin sisällönhallintajärjestelmä. Se perustettiin alun perin kehittäjiensä verkkokaupparatkaisuksi, koska sen aikaiset ratkaisut eivät täyttäneet heidän vaatimuksiaan. (Shopify, 2020)

Palvelu tarjoaa kolme eri tason kuukausimaksullista lisenssiä. Kaikki näistä sisältävät nettikaupan rajoittamattomalla määrällä tuotteita, ympärivuorokautisen tukipalvelun ja muun muassa tuen tarjouskupongeille. Kalliimmat lisenssit sisältävät enemmän käyttäjätilejä verkkokaupan henkilöstölle, paremmat raportit, useampien kielten tuen ja muun muassa maakohtaisia verkkotunnuksia.

2.3.3 Joomla!

Vuonna 2000 yritys nimeltä Miro kehitti sisällönhallintajärjestelmän nimeltä Mambo. Joomla on avoimen lähdekoodin järjestelmä, joka haarautui vuonna 2005 Mambo-sisällönhallintajärjestelmästä. Nimensä se on saanut swahilin kielen sanasta jumla, joka tarkoittaa ”kaikki yhdessä.” Tämä kuvastaa vapaaehtoisista kehittäjistä koostuvan yhteisön ylläpitämää sisällönhallintajärjestelmää (JavaTpoint).

Joomlan tärkeimpiä ominaisuuksia ovat muun muassa tuki yli seitsemälle kymmenelle kielelle, hakukoneoptimointi ja -ystävällisyys, muokattavuus, ilmaisuus ja kattava, yli 8000 laajennuksen kirjasto (Joomla). Sisällönhallintajärjestelmän laajennuksilla

sivustoille voidaan lisätä vakio-ominaisuuksien lisäksi haluttuja ominaisuuksia. Kattava laajennuskirjasto onkin yksi monista syistä, jotka tekemvät Joomlaista niin muokattavan.

Kuten Wordpress, myös Joomla käyttää ohjelmointikielensä PHP:ta, ja tietokantanaan MySQL:aa. Sivuston responsiivisuuden takaamiseksi se käyttää Bootstrap-viitekehystä, ja ulkoasun muokkaamiseen CSS:n LESS-esikäntäjää.

Ensimmäinen suurempi julkaisu 1.5 julkaistiin tammikuussa 2008 (JavaTpoint). Tällä hetkellä siitä on julkaistuna 3.9 haaran lokakuussa 2020 julkaistu versio 3.9.22 ja 4.0 haaran 4.0-beta 2 versio. (Joomla)

2.3.4 Wix

Idea Wixistä sai alkunsa kolmen kehittäjän, Avishai Abrahamin, Nadav Abrahamin ja Giora Kampin, luodessa www-sivustoa toisen startup-yrityksen pohjaksi. Oman verkkosivuston kehittäminen oli heille vaikea, turhauttava ja kallis kokemus. Kokemuksen pohjalta kuitenkin perustettiin Wix, joka on alusta, jonka tavoitteena on mahdollistaa sivujen luonti kenelle tahansa taidoista riippumatta. (Wix, 2020)

Wix on suljetun lisenssin sisällönhallintajärjestelmä, joka tarjoaa käyttäjälleen useita eri tason lisenssejä eri tarkoituksiin. Ilmainen lisenssi tarjoaa käyttäjilleen valmiita ilmaisia sivustopohjia, 500 Mb palvelintilaa, 500 Mb kaistaa palvelimelta ja asiakastuen. Ilmaisilisenssin huonoina puolina ovat esimerkiksi valmiiksi asetettu verkko-osoite muodossa käyttäjätunnus.wixsite.com/sivustonosoite, jokaisella sivulla esitetyt mainokset ja se, ettei käyttäjällä ole mahdollisuutta lisätä sivustolleen Googlen analytiikkaa. Maksulliset sivustolisenssit antavat mahdollisuuden esimerkiksi omaan verkkotunnukseen, käyttäjäanalyysiin ja rajoittamattomaan palvelinkaistaan. Lisäksi Wix myy erillisiä lisenssejä verkkokaupoille. Nämä lisenssit mahdollistavat verkkomaksujen suorittamisen, myyntianalytiikan ja raporttien koostamisen. (Wix)

Sivustojen rakentaminen Wixillä perustuu visuaaliseen käyttöliittymän raahaa ja pudota-ominaisuuteen, jolla palvelun käyttäjä voi raahata haluamaansa sivustopohjaan haluamiaan elementtejä. Näitä elementtejä voivat olla esimerkiksi kuvagalleria tai tekstilohkot. Sivustot ovat myös automaattisesti optimoituja hakukoneita ja mobiililaitteita varten. (Wix)

2.3.5 Drupal

Vuonna 2000 Dries Buyert kehitti opiskelukavereilleen keskustelupalstan, joka myöhemmin siirrettiin internettiin hänen valmistuttuaan. Kyseisestä sivustosta tuli sivuston taustalla pyörivän sisällönhallintajärjestelmän testausympäristö ja ideariihi. Vuonna 2001 sivuston taustalla toimiva sisällönhallintajärjestelmä julkaistiin muiden käytettäväksi. Näin syntyi avoimen lähdekoodin sisällönhallintajärjestelmä Drupal. (Drupal, 2020)

Drupalin toiminta perustuu sen Drupal Coren, eli ytimen, ympärille. Ydin koostuu itse sisällönhallintajärjestelmän välttämättömien ominaisuuksien lisäksi niin kutsutuista ydinmoduuleista ja ydinteemoista. Drupalin moduulit ovat Joomlaan laajennusten kaltaisia pienempiä koodipaketteja. Niitä asentamalla voidaan laajentaa sivuston toiminnallisuuksia. Drupalin ydinmoduulit ovat yhteisön yhdessä valitsemia ominaisuuksia. Ydinmoduulien lisäksi sivuston kehittäjä voi ladata Drupalin projektisivulta yhteisön julkaisemia moduuleita, joita on tällä hetkellä (28.10.2020) valittavana yli 46 000 kappaletta (Drupal.org, 2020). Drupal perustuu avoimen lähdekoodin teknologioihin. Koodikielenään se käyttää PHP:ta ja tietokantanaan MySQL:aa.

Kehittäjän näkökulmasta Drupalin heikkoudet ja vahvuudet kulkevat käsi kädessä. Se on hyvien sisällönhallintaominaisuuksiensa, monikielisyytensä ja joustavuutensa takia hyvä valinta suurten ja monimutkaisten sivujen tuottamiseen. Toisaalta Drupalin joustavuudesta johtuen sen käyttäminen vaatii käyttäjältään asiantuntemusta ja aikaa, joten esimerkiksi pienten sivustojen luominen Drupalilla ei ole välttämättä kustannustehokkain ratkaisu.

Tällä hetkellä Drupalista on olemassa kolme tuettua versiota. Drupal 7 sisällönhallintajärjestelmän uusin versio 7.73 julkaistiin vuoden 2020 syyskuussa, Drupal 8 versio 8.9.7 lokakuussa 2020 ja Drupal 9 versio 9.0.7 lokakuussa 2020 (Drupal, 2020). Drupalin versio 9 on kehitetty pitkälti 8:n pohjalta ja siirtymä versiosta toiseen pitäisi olla ”helpoin vuosikymmeniin”. Versio 10 on suunniteltu julkaistavaksi vuonna 2022. (Drupal, 2020)

3 VIITEKEHYKSET

Tietotekniikassa viitekehukset ovat valmiita koodikokoelmia ja toiminallisuuksia, jotka toimivat ennalta määritettyjen sääntöjen mukaan. Näitä sääntöjä kutsutaan ohjelmointirajapinnaksi, tai sen englanninkielisen termin application programming interface lyhenteellä API. Viitekehysten tarkoituksena on yksinkertaistaa kehitystyötä tarjoamalla valmiita rakenteita vähentäen kehittäjän tarvetta määritellä kaikkia asioita. Näitä rakenteita voivat olla esimerkiksi valmiit luokat ja funktiot, joita hyödyntäen kehittäjä voi muun muassa manipuloida dataa. (Techterms)

```
1
2
3 <div id="welcome-text">
4   <p>Hello world!</p>
5 </div>
6
7 // Example of hiding div element with JavaScript
8 var variable;
9 variable = document.getElementById("welcome-text");
10 variable.style.display="none";
11
12 // Example of hiding div element with JQuery
13
14 $("#welcome-text").hide();
15
16
```

Kuva 2. Esimerkki samasta koodista JavaScriptillä ja JQueryllä.

Nettikehityksessä käytettävät viitekehukset voidaan karkeasti jakaa palvelimen puolella ja käyttäjän puolella käytettäviin viitekehyyksiin. Palvelinpuolen viitekehyyksiä ovat esimerkiksi Ruby on Rails ja Laverte. Vastaavasti käyttäjäpuolen viitekehyyksiä ovat esimerkiksi AngularJS, Ember.js, React.js ja Vue.js.

Palvelinpuolen viitekehukset helpottavat ja yksinkertaistavat nettikehitystä kehityksen back-end -puolella. Viitekehukset voivat esimerkiksi mahdollistaa palvelinten ja selainten välisen HTTP-protokollaan perustuvaan kommunikoinnin käsittelyn yksikertaisemmassa muodossa. Muita olennaisia ominaisuuksia ovat sivustojen osoitteiden hallinta ja muodostus, yksinkertaistetumpi tietokantojen käyttö, ja tallennetun datan esittäminen templateattien avulla. (Mozilla, 2020)

Käyttäjäpuolen, eli front-endin, viitekehyyksiä voidaan palvelinpuolen viitekehysten tapaan käyttää vähentämään sivustoa tehdessä tarvittavan kirjoitettavan koodin määrää

käyttämällä viitekehysten valmiita toiminnallisuuksia sen sijaan, että sivuston tekijä joutuisi kirjoittamaan kaiken siihen vaadittavan koodin itse. Esimerkiksi suosituimpien frontend -viitekehysten joukkoon kuuluva Bootstrap sisältää sivuston palstapohjaisen elementtien leveyden hallintaan tarkoitetun Grid Systemin, jossa elementtien leveyttä voidaan hallita esimerkiksi lisäämällä siihen viitekirjastossa määritelty luokka, joka sisältää halutun leveyden määrittelyt (Kuva 3.). Sama Grid System tukee myös mobiililähtöisen suunnittelunsa ansiosta eri kokoisia selaimia.

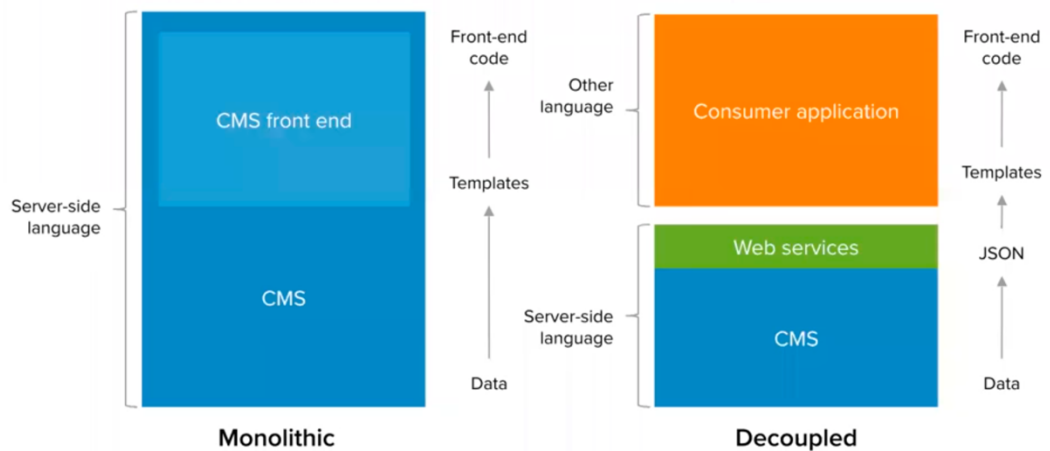


Kuva 3. Bootstrap grid systemin käyttöesimerkki (getbootstrap.com).

3.1 Viitekehykset ja Decoupled CMS

Decoupled Content management systemillä (CMS), jota joskus kutsutaan headless Content management systemiksi, tarkoitetaan tilannetta, jossa sisällönhallintajärjestelmää käytetään sivustokokonaisuuden toteutuksessa sisällönhallintatehtävissä tuottamaan sisältöä toisten ohjelmistojen käytettäväksi esimerkiksi sisällön esittämiseen (Dries Buytaert, Acquia, 2018). Näissä tapauksissa sisällön esittämiseen voidaan käyttää esimerkiksi JavaScript-viitekehysä, kuten React.js, Angular tai Vue.js.

Monolithic vs. decoupled CMS



Kuva 4. Monoliittinen ja irrotettu sisällönhallintajärjestelmä (Dries Buytaert, Acquia, 2018).

Buyert (2018) toteaa sisällönhallintajärjestelmän irrottamisen pelkästään sisällönhallintatehtäviin tuovan mukanaan useita etuja, joita ei ole perinteisessä kytketyssä sisällönhallintajärjestelmällä toteutetussa järjestelmässä. Perinteisissä sisällönhallintajärjestelmissä itse järjestelmä myös hoitaa hallitsemansa sisällön esittämisen. Koska irrotettu sisällönhallintajärjestelmä voi olla täysin erillinen kokonaisuus ulkoasusta, se mahdollistaa saman informaation esittämisen useassa toisistaan erillisessä paikassa. Sisällönhallintajärjestelmä julkaisee hallitsemansa informaation esimerkiksi Drupalin tapauksessa JSON-muodossa, jonka ulkoasusta huolehtiva ohjelma esittää halutussa muodossa (Kuva 4.). Näin sisällöntuottajat voivat käyttää hallintajärjestelmää sisällön luomiseen kerran usean erillisen paikan sijasta, ja samalla sisällöntuottajat voivat myös hallita kaikkia sisältöä yhdestä paikasta. Tämä tarkoittaa myös sitä, että sivuston back-end ja front-end ovat hyvin selkeästi omat kokonaisuutensa, jolloin niiden kehittäminen on myös toisistaan erillään CMS-arkkitehtuurissa. Näin sivuston kehityksessä back-end -kehittäjät ja front-end -kehittäjät voivat keskittyä itsenäisesti omien osa-alueidensa kehittämiseen. Back-end -kehittäjät kehittävät rajapintaa ja samanaikaisesti front-end -kehittäjät rakentavat sivuston käyttäjille näkyvää osaa.

Kun sivusto toteutetaan headless-menetelmällä, siihen sisältyy myös omat ongelmansa (Dries Buytaert, Acquia, 2018). Monissa nykyisissä sisällönhallintajärjestelmissä on mahdollista muokata sisältöä sivulla, jossa se esiintyy. Tätä mahdollisuutta

sisällönhallintaan ei ole, kun sisältö esitetään irrotettuna sisällönhallintajärjestelmästä, ellei muokkaustoimintoja erikseen mahdollisteta.

Toinen heikkous on siinä, että missä perinteisillä sisällönhallintajärjestelmillä on mahdollista hallita missä ja miten sisältö esitetään, sivuston ulkoasusta irrotettu sisällönhallintajärjestelmä ei ota kantaa siihen, missä tai miten hallittu informaatio esitetään (Dries Buytaert, Acquia, 2018). Näin ollen esitettävän datan ulkoasun muokkaaminen täytyy tehdä front-end -kehityksen puolella.

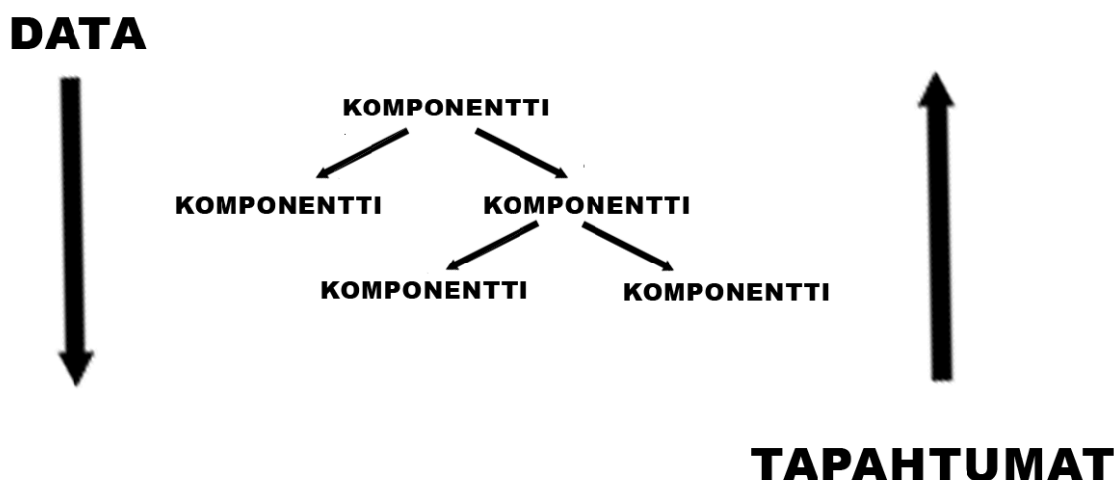
Kolmantena heikkoutena voidaan mainita se, että koska sisällönhallinta ei suoraan vaikuta siihen, miltä asiat näyttävät, on niistä vaikea esittää esikatselua. Kehittäjien on nähtävä paljon vaivaa esikatselun näyttämiseen sisällönsyöttäjille. Käytännössä tämä tarkoittaa sitä, että ulkoasu täytyy toteuttaa erikseen myös sisällönhallinnan puolelle (Dries Buytaert, Acquia, 2018).

3.1.1 React.js

React.js on Facebookille työskentelevän Jordan Valken kehittämä avoimen lähdekoodin JavaScript kirjasto, jota käytetään varsinkin yksisivuisten ohjelmistojen luomiseen. Se otettiin käyttöön Facebookin uutissyötteessä vuonna 2011, ja Instagramissa vuonna 2012 (Hunt, 2015).

React käyttää templaattien rakentamiseen JavaScriptin sijasta sitä laajentavaa JavaScript XML-kieltä, joka mahdollistaa HTML-kielen käyttämisen Reactin JavaScript-komponenttien renderöinnissä. HTML-syntaksi muutetaan JavaScript kutsuiksi Reactin viitekehyksessä (React).

Reactille tärkeitä ominaisuuksia ovat esimerkiksi yhdensuuntainen tiedonsiirto ja virtual document object model. Yhdensuuntaisessa tiedonsiirrosta komponenttien muuttujat, joita kutsutaan reactissa tiloiksi (englanniksi state), ovat muuttumattomia sen jälkeen, kun ne välitetään HTML-tagien mukana renderöitäviksi. Komponentti voi kuitenkin välittää tiedon lapsikomponenteilleen, jolloin komponentin tilaan tehtyjä muutoksia voidaan hyödyntää niissä (React). Virtual Document Object Model (Virtual DOM) tarkoittaa Reactin luomaa ja muistissa olevaa datarakennetta, joka prosessoi muutokset ja päivittää selaimen (Rascia, 2018). Tämä mahdollistaa sen, että Reactin kirjasto renderöi ainoastaan komponentteja, jotka muuttuvat oikeasti.



Kuva 5. Yhdensuuntainen tiedonsiirto komponenttien välillä.

3.1.2 Angular

Ensimmäisen kerran vuonna 2010 julkaistu Angular (Github, 2010), joka tunnettiin ennen nimellä AngularJS, on TypeScriptillä kirjoitettava avoimen lähdekoodin JavaScript-viitekehys. Vuonna 2016 Google julkaisi täysin uudelleen rakennetun version tästä viitekehystä, lisäten samalla komentoliittymän ja täysin komponenttipohjaisen arkkitehtuurin (AngularJS, 2016). Uusin julkaistu versio on tällä hetkellä versio 10, joka julkaistiin kesäkuussa 2020 (Fluin, 2020).

Se, että Angular on hyvin dokumentoitu, ja on Googlen tukema, ovat arvokkaita valtteja kehittäjän näkökulmasta. Suosiosta johtuen Angularille löytyy myös paljon kolmannen osapuolen työkaluja ja komponentteja, joita voidaan hyödyntää Angular-projekteissa parantamaan ja lisäämään toiminnallisuuksia. Muita huomionarvoisia ominaisuuksia ovat esimerkiksi Angularin etuaikaan kääntäjä (Ahead-of-Time Compiler, AOT) ja Angular Ivy. AOT-kääntäjä nopeuttaa sivuston latautumista muuttaen TypeScriptin ja HTML:n JavaScriptiksi jo kääntämisen aikana. (Angular) Angular Ivy on nimitys Angularin uuden sukupolven kääntö- ja renderöintikanavalle, joka kääntää ohjelmiston komponentit ja templaatin selaimessa näytettäväksi JavaScriptiksi (Angular.io). Angular Ivy poistaa renderöinnin aikana käyttämättömän koodin ja näin nopeuttaa sivustolatauksia. (AngularConnect, 2019)

3.1.3 Vue.js

Vue.js on vuonna 2014 julkaistu, entisen Google työntekijän, Evan Youn, kehittämä avoimen MIT-lisenssin alainen JavaScript-viitekehys (Cromwell, 2016). Viitekehysten version 1.0.0 julkaisu tapahtui lokakuussa 2015, ja se sisälsi parannetun templaattisyntaksin, parannellun rajapinnan ja nopeamman renderöinnin verrattuna aikaisempiin kehitysjulkaisuihin (Vuejs.org, 2015). Syyskuussa 2016 julkaistu Vue.js 2.0 nopeutti viitekehysten suorituskykyä entisestään, ja sen renderöintitaso korvattiin kevyellä virtuaalisella DOM:illa (You, 2016). Vue.js 3.0 julkaistiin syyskuussa 2020 (You, 2020).

Vue.js on asteittainen viitekehys, jota käytetään sivustojen ulkoasujen luontiin. Asteittaisella viitekehyksellä tarkoitetaan viitekehystä, joka voidaan sisällyttää projekteihin erilaisissa määrin, tai vaihtoehtoisesti projektin toiminnallisuutta voidaan korvata Vuella toteutetulla ulkoasulla. Viitekehys onkin suunniteltu alusta alkaen sen sisällyttämisen mahdollistamiseksi projekteihin erilaisissa määrin. (Cromwell, 2016)

Vuen ydin on keskittynyt ulkoasun renderöintiin. Lisäksi Vuella on erinäinen määrä kirjastoja, jotka mahdollistavat Vuen skaalaamisen kokonaiseksi projektiksi. Viitekehys onkin suunniteltu helposti lähestyttäväksi, monipuoliseksi ja tehokkaaksi. Se on mahdollista sisällyttää projektiin script-tagilla, eikä vaadi Webpackin kaltaista erillistä työkalua. Koska viitekehysten toiminta perustuu HTML:n, CSS:n ja JavaScriptin käyttämiseen, riittää näiden tunteminen projektin aloittamiseen.

4 TOTEUTUS

4.1 Tehtävänasettelu

Opinnäytetyön tutkimuskysymykseksi asetettiin: ”Mikä olisi hyvä JavaScript-viitekehys yrityksellemme dynaamisten Drupal 8 -sivustojen toteuttamiseen?” Työn toimeksiantajana toiminut Citrus Solutions Oy voi hyödyntää ratkaisua yrityksen tulevissa, että nykyisissä projekteissa, jos viitekehys mahdollistaa sen hyödyntämisen helposti vanhoissa projekteissa. Opinnäytetyön tekijän tulee ottaa tämä huomioon projektin yhteydessä. Viitekehysten mukautuvuus voidaan siis laskea toivotuksi ominaisuudeksi. Lisäksi opiskelijan tulee ottaa huomioon viitekehysten sopivuus yrityksen työntekijöiden taitojen näkökulmasta. Täyttääkseen esimerkkisivuston ehdot tulee sivuston sisältää vähintään kaksi sisältötyyppiä, joista ainakin toisen tulee olla haettavissa hakutoiminnolla. Lisäksi esimerkkisivuston tulee sisältää mahdollisuus siirtyä linkin avulla tiettyyn osaan sivustoa.

4.2 Haasteet ja riskit

Opinnäytetyön tekijä oli tutustunut ennen sen tekemistä Drupal-sivuston toteutukseen vain ilman ulkoisia viitekehyskieliä. Näitä tehdessä opiskelija oli keskittynyt pääasiallisesti sivustojen front-end -ratkaisujen toteuttamiseen, joten projektin toteuttaminen vaatisi uuden koodikielen opiskelua toteutusta varten. Näitä kieliä opiskelija ei ole myöskään opinut opiskeluidensa aikana. Lisäksi, koska yhdessä Drupalin kanssa käytettävien viitekehysten määrä on suuri, joten aikaa tulisi kulumaan suurehko määrä sopivan viitekehysten valintaa tehtäessä.

Riskinä onkin, ettei opiskelija saa toteutettua Drupalin dynaamista ulkoasua valintojen laajan kirjon ja vaaditun uuden ohjelmointikielen opettelun takia. Opiskelija on kuitenkin motivoitunut, ja haluaa sekä kehittää jo nykyisin osaamiaan taitoja, että lisätä uusia tietoja, jotka vievät häntä eteenpäin front-end -kehittäjänä. Pienempi riski on se, etteivät ulkoiset viitekehyskirjastot sopisi Drupal 8 -sivuston ulkoasun luontiin joko yleisellä tasolla, tai yrityksen näkökulmasta. Riski tähän on kuitenkin pienehkö, sillä teknologiat eivät ole uusia, ja ne on otettu hyvin huomioon Drupalin kahdeksannessa versiossa esimerkiksi tietojen esittämisessä.

5 DYNAAMISEN DRUPAL 8 -ULKOASUN KÄYTTÖÖNOTTO ESIMERKKISIVUSSA

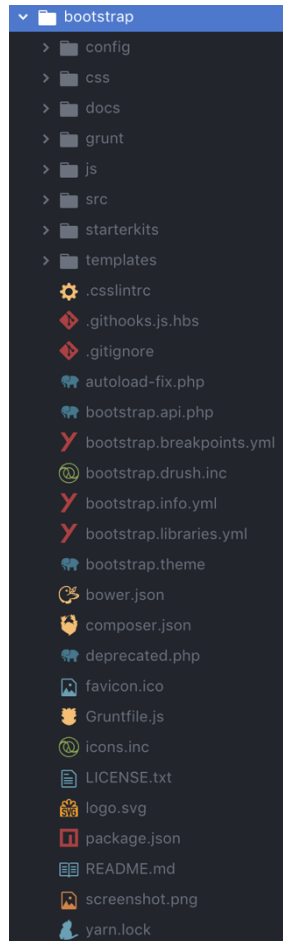
Drupal-sivuston dynaamisen ulkoasun toteutuksen viitekehyykseksi valikoitui Vue.js versionumeroltaan 2.0+. Viitekehyyksen valintaan vaikutti kyseisen viitekehyyksen yksinkertainen syntaksi front-end -kehittäjän näkökulmasta, sillä sen kehityksessä käytetään suurimmaksi osaksi kieliä, joita kehittäjä osaa entuudestaan. Toinen merkittävä valintaan vaikuttanut ominaisuus oli mahdollisuus sisällyttää se sivustoprojektiin useammalla eri tavalla ja eri määrissä.

Vue.js komponenttipohjainen rakenne yhdistettynä Drupalin sivustojen rakenteen muodostavien Twig-templattien kanssa mahdollistaa sen sisällyttämisen sivuston yksittäisiin sivuihin tai elementteihin. Vaihtoehtoisesti sillä voi tuottaa yksisivuinen nettisivun yksittäisellä HTML-tiedostolla, jossa esitetään halutut tiedot.

5.1 Teema

Simuloitaessa yrityksessä normaalisti käytettyjä käytäntöjä sivulle asetettiin Bootstrap-teema Drupal-ytimen mukana tulevien viiden vakioiteeman sijaan. Teeman asennus toteutettiin pakettihallintaa hyödyntämällä käyttäen composer.json tiedostoa. Asennus asettaa teeman Drupal-juuressa olevaan themes/contrib/-kansioon (Kuva 6.).

Teema sisältää yleisiä CSS-asetuksia, valmiit tyylittelut tärkeimmille HTML-elementeille, joihin on lisätty valmiita laajennettavia luokkia, ja elementtien leveyksien määrittämistä helpoittava Bootstrap Grid system. Lisäksi teeman mukana tulee valmiiksi määriteltyjä



Kuva 6. Kuva teeman rakenteesta.

komponentteja esimerkiksi pudotusvalikoille, ilmoituksille ja sivuston navigaatiolle. Mukana tulee myös edellämänittuja komponentteja elävöittäviä JQuery-liitännäisiä.

Lisäksi tästä teemasta luotiin alateema, joka mahdollistaa teeman räätälöinnin ja muokkaamisen ilman alkuperäisen teeman muokkausta. Alateema kopioidaan /themes/-kansioon alle omaksi kansiokeeseen, ja nimetään uudelleen vue_thesis:ksi. Jotta Drupal tunnistaa teeman, kansion nimi kirjoitetaan konemuodossa pienillä kirjaimilla, ja alaviivaa lukuunottamatta ilman erikoismerkkejä ja välilyöntejä. Vue_thesis-kansiossa olevat THEMENAME.info.yml- ja THEMENAME.libraries.yml-tiedostot nimetään uudelleen alateeman koneluettavan nimen mukaisesti muotoon vue_thesis.info.yml ja vue_thesis.libraries.yml. vue_thesis.info.yml sisältää alateeman tiedot ja viittauksen bootstrap-isäntäteemaan, jolloin Drupal osaa käyttää isäntäteemassa määriteltyjä asioita, jos niitä ei ole määritelty alateemassa. Kuten kuvasta 7 nähdään, Vue_thesis.libraries.yml sisältää teeman versionumeron, viittauksen käytettävään main.css-teemoitustiedostoon ja käytetyt JavaScript-tiedostot.

```

1  vue:
2    version: 1.x
3    css:
4      theme:
5        css/main.css: {}
6    js:
7      https://cdn.jsdelivr.net/npm/vue/dist/vue.js: { external: true }
8      js/axios.min.js: {}
9      js/vue-router.js: {}
10     js/index.js: {}
11

```

Kuva 7. Vue_thesis.libraries.yml tiedostosta.

5.2 Sisällön luonti ja sen hallinta

Tehtävänasettelun mukaisesti Drupalin käyttöliittymää tulee käyttää uuden sisällön luontiin ja sen hallintaan. Sivustolle luodaan uusi sisältötyyppi Teams, jolle asetetaan erityyppisiä kenttiä. Kenttien monimuotoisuudella havainnollistetaan viitekehysten käyttökelpoisuutta niiden esittämiseen. Sisältötyypin hallintaan, muokkaamiseen, poistamiseen ja uuden luomiseen asetetaan oikeus sekä sisällöntuottajille, että pääkäyttäjälle.

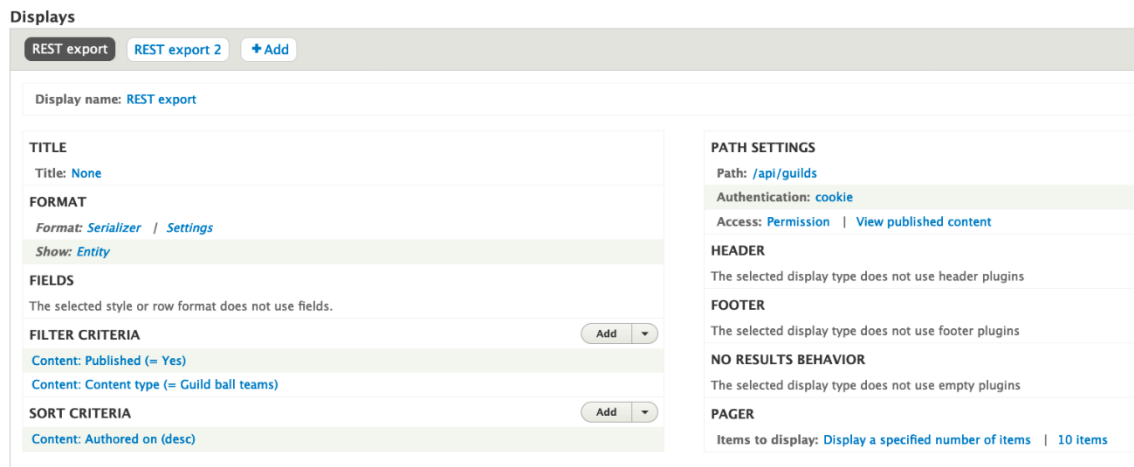
5.3 Sisällön hakeminen

Käyttääkseen hyväksi Drupalin hallittua sisältöä on viitekehysten saatava sisällön tieto jossain muodossa. Asettamalla Drupalin ytimen mukana tulevat RESTful web services- ja Serialization-moduulit päälle, on Drupalin esittävien elementtien, eli näkymien, mahdollista esittää haluttu sisältö JSON-muodossa.

Tälle datalle voidaan asettaa osoite, josta sen voi hakea esimerkiksi käytettäväksi Vue.js varten. Tätä dataa voidaan myös suodata käyttämällä näkymien suodattimia. Suodattimiksi voidaan asettaa erilaisia arvoja, kuten "Content: (Published = yes)" ja Content: content type (= Guild ball teams). Näin JSON-muodossa esitetty data rajataan vain järjestelmästä julkaistuun sisältöön, joka on sisältötyyppiä "Guild ball teams".

Kuvassa 8 esitetty näkymä on jaoteltu neljään erilliseen näyttömuotoon eri tapauksia varten. Niistä ensimmäinen, eli REST export, esittää kaikki ensimmäisen sisältötyypin sisällöt JSON-syötteenä Drupal-sivuston sivulla /api/guilds. Toinen, eli REST export 2, esittää yksittäisen sisällön tiedot käyttäen hyväksi Drupalin asettamaa sisällölle uniikkia tunnustetta. Esimerkiksi osoite /api/guilds/1 esittää Drupalissa ensimmäiseksi luodun

sisällön. Kolmas esittää kaikki julkaistut sisällöt toisesta sisältötyypistä, ja neljäs esittää yksittäisen sisällön toisesta sisältötyypistä.

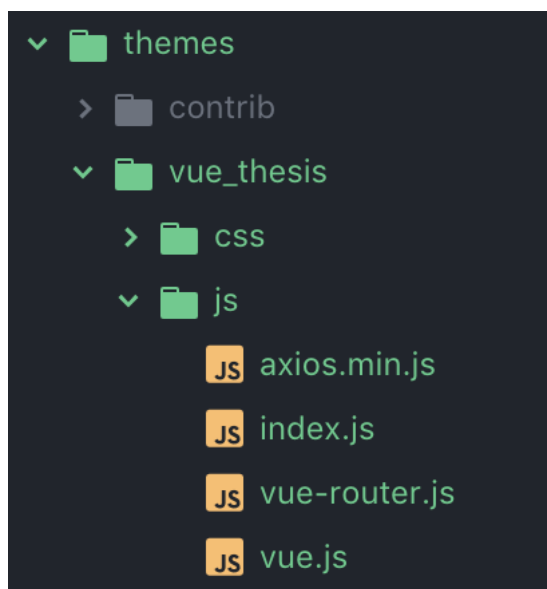


Kuva 8. Kuva näkymästä ja sen asetuksista.

5.4 Vue.js:n tiedostorakenne

Teeman js-kansio sisältää projektille olennaiset JavaScript-tiedostot (Kuva 9.). Tiedostot ladataan käytettävästä riippuen templaatissa, tai toisessa tiedostossa. Kansiossa olevat axios.min.js, vue.js ja vue-router.js ovat projektissa tarvittavia kirjastoja, ja index.js on itse projektin tiedosto.

Axios.min.js sisältää JavaScript-kirjaston, joka mahdollistaa näkymien tuottamien json-syötteiden noutamisen näkymälle asetetusta sivustopolusta. Kirjastoa hyväksi käyttäen tiedot otetaan käyttöön index.js:ssä niiden esittämistä varten. Vastaavasti Vue.js-tiedosto sisältää itse Vue.js viitekehysten ja vue-router.js dynaamisten webosoitteiden luonnin sivulle.

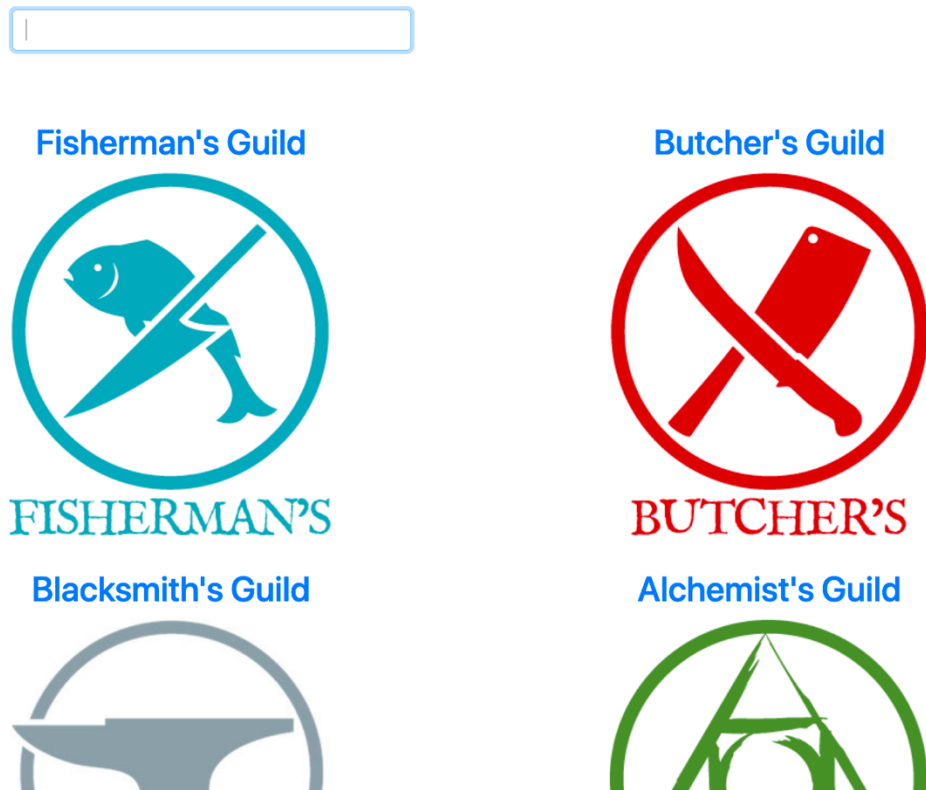


Kuva 9. Kuva teeman js-kansion rakenteesta.

Index.js-tiedostossa määritellään kaikki tarvittava toiminnallisuus, mitä projekti käyttää. Esimerkkiprojektissa tiedosto sisältää Axiosin käyttämät kaksi osoitetta, joita käytetään sisältöjen hakemiseen, ja metodit, joissa niitä käytetään tietojen hakemiseen, sekä osoitteenhallintajärjestelmän, jolla voidaan palata sivuston tiettyyn tilaan. Tämän lisäksi tiedosto sisältää määrittelyt sisältöjen listauksesta, niiden suodattamisesta hakukentän mukaan, ja yksittäisten sisältötyyppien määrittelyyn. Määrittelyissä asetetaan muuttujien tiiloille arvot, ja määritellään templaattien tunnisteet, jossa muuttujia käytetään.

JavaScript-tiedostojen lisäksi tarvitaan ulkoasun luontia varten niiden vastapariksi tiedosto, joka sisältää HTML-pohjaisen templaatin. Nimetty templaatti, jonka nimi vastaa JavaScript-tiedostossa muuttujalle asettua tunnistetta, sisältää Vue.js:lle ominaista syntaksia, joka mahdollistaa JavaScriptissä tietojen sitomisen dynaamisesti ulkoasun luontia varten.

Drupal 8 -projektissa Vue.js:n käyttöönotto voidaan toteuttaa käyttämällä hyväksi sisälönhallintajärjestelmän Twig-templaatteja, itsenäistä moduulia, tai itsenäistä HTML-tiedostoa käyttämällä. Esimerkkiprojektissa toteutus tehtiin käyttämällä yksittäistä HTML-tiedostoa.



Kuva 10. Listaus ensimmäisestä näkymästä.

Toteutuksessa listataan sivulla ensimmäisen sisältötyypin nimet ja niihin asetettu kuva, joista käyttäjä voi siirtyä yksittäisen sisällön sivulle. Etusivulta löytyy hakukenttä, joka suodattaa dynaamisesti reaaliajassa näytetyt sisällöt (Kuva 10.). Hakukriteereinä toimivat ensimmäisen sisältötyypin otsikko, sisältötyypissä oleva tekstikenttä, ja entiteettiviitaukset toiseen sisältötyyppiin. Sisällön otsikko toimii linkkinä yksittäiseen sisältöön. Yksittäinen sisältö voidaan ladata ilman erillistä sivulatausta, jo ladattua dataa hyödyntäen.

Ensimmäisen sisältötyypin sivulla listataan sisällön otsikko, kuva, kappale tekstiä, ja sisällönsyöttäjän määrittämä listaus toisesta sisältötyypistä. Listaus toimii myös linkkinä toisen sisältötyypin näkymään.

Siirryttäessä sivulta toiselle, sivuston web-osoite päivittyy, ja mahdollistaa sen eri osiin navigoinnin suoraan. Yksittäisillä sivuilla on myös paluulinkki edelliseen näkymään, ja suora linkki etusivulle.

5.5 Lopputulos

Opinnäytetyössä toteutettiin single page app, yksisivuinen sivu, jossa vaihdettaessa sivuston eri osien välillä ei tehdä erillistä sivulatausta. Sisällönhallinta tehtiin Drupalin käyttöliittymää käyttäen, ja ulkoasu toteutettiin Vue.js:llä. Etusivulla esitettiin ensimmäisen sisältötyypin sisältöjen otsikot ja kuva. Sekä kuva että otsikko toimivat linkkeinä sisällön näkymään. Etusivun hakukentällä voidaan hakea sisältötyyppien eri sisältöjen mukaan, ja näkymä päivittyy dynaamisesti ladaten hakua vastaavat sisällöt.

Opiskelijan näkökulmasta Vue.js on hyvä sivuston toteutukseen, koska sen syntaksi on kolmesta vaihtoehdoista yksinkertaisin ja perustuu opiskelijan jo osaamiin kieliin. Yksinkertainen syntaksi mahdollistaa keskittymisen viitekehysten logiikan ymmärtämiseen. Logiikan ymmärtäminen auttaa opiskelijaa vastaisuudessa myös muiden viitekehysten käyttämisessä.

Helppo omaksuttavuus on myös tärkeä ominaisuus yrityksen kannalta, koska yrityksessä on paljon eritaustaisia kehittäjiä. Vue.js on hyvä valinta myös sen mukautuvuuden takia. Koska Vue.js on asteittainen viitekehys, se soveltuu hyvin erilaisiin projekteihin erilaisissa määrissä. Se voidaan upottaa projektiin Drupalin templaattien mukana, tai vaihtoehtoisesti sillä voidaan luoda koko sivuston ulkoasu.

Kokonaisuudessaan esimerkkisivusto on onnistunut esimerkki Vue.js:n potentiaalista sekä opiskelijan että yrityksen näkökulmasta. Drupal 8 -sivuston ulkoasu toteutettiin onnistuneesti, ja sille laaditut kriteerit täytettiin. Opiskelija lisäsi tietotaitoaan JavaScript-viitekehysten kehittäjänä, ja pystyy hyödyntämään oppimiaan asioita työskennellessään front-end -kehittäjänä.

6 YHTEENVETO

Opinnäytetyön tavoitteeksi asetettiin kehittäjälle ja yritykselle sopivan JavaScript-viitekehysten valinta Drupal 8 -sisällönhallintajärjestelmän kanssa käytettäväksi. Valitulla viitekehyksellä tuli luoda ulkoasu Drupal 8 -sivustolle, jossa sisällönhallinta tehdään Drupalilla ja ulkoasu esitetään viitekehysten avulla. Sivustolla tuli olla vähintään kaksi erilaista sisältötyyppiä, joilla voidaan havainnollistaa sisältöjen suhteita toisiinsa. Opinnäytetyön toinen tavoite oli lisätä opiskelijan taitoja front-end -kehittäjänä.

Drupal on sisällönhallintajärjestelmänä erittäin tehokas ja monipuolinen alusta sivustojen ylläpitoa varten. Kuitenkin se on parhaimmillaan suurien sisältömäärien hallinnassa, ja sivuston ulkoasun toteuttaminen jää herkästi joko valmiiden teemojen varaan tai pitkälti kehittäjien suurilta osin räätälöitäväksi. Tämä aiheuttaa herkästi sen, että kokeneen kehittäjän silmiin Drupal-sivustot näyttävät juuri Drupalilla toteutetuilta sivuilta. Sivustojen muokkaamiseen visuaaliselta ilmeeltään monimuotoisemmaksi halutaan löytää uusia tapoja toteuttaa sivustojen peruselementtejä.

Tämä opinnäytetyö on tutkinut erilaisia JavaScript-viitekehysjä, ja arvioinut niiden sopivuutta yrityksen toimintatapoihin, henkilöstön osaamiseen ja jo olemassa oleviin projekteihin. Työn aikana tutkittiin erilaisia menetelmiä sisällyttää valittu viitekehys projekteihin. Viitekehysesä valikoitui Vue.js.

Opinnäytetyön tavoitteessa onnistuttiin, ja opitut asiat on todettu myös hyödylliseksi projekteissa, joissa on käytetty muita JavaScript-viitekehysjä. Opiskelija on siis onnistuneesti sekä lisännyt ymmärrystään viitekehysten toiminnasta sivustoilla että nostanut yleistä taitotasoaan JavaScriptin käyttäjänä.

Opinnäytetyötä voidaan kehittää eteenpäin usealla eri tavalla. Viitekehysten käyttämistä projekteissa voidaan parantaa luomalla erilaisia yritykselle standardoituja tapoja sisällyttää ne projekteihin. Näitä tapoja voivat olla esimerkiksi valmiiksi toteutettu moduuli, joka sisältää valmiiksi valmiin pohjan viitekehysten sisällyttämistä varten sivustolle. Toinen vaihtoehto voisi olla valmiin sivustopohjan luonti Drupal 8 -sivuston TWIG-mallipohjilla, joilla voidaan luoda esimerkiksi yksittäisiä lohkoja sivustolle. Kolmas mahdollinen kehityssuunta on lähteä viemään projektia pidemmälle päivittämällä Vue.js versioon 3.0 ja testata sen yhteensopivuutta Drupal 9:n kanssa.

Jos opinnäytetyö aloitettaisiin kokonaan alusta, työn toteutus voitaisiin tehdä eri tavalla. Koska olen oppinut huomattavan määrän JavaScriptillä kehittämistä, toteutuksessa voitaisiin luoda kolme erillistä esimerkkisivustoa eri viitekehysten avulla, ja arvioida niiden sopivuutta eri näkökulmista. Eri viitekehysten suorituskykyä voitaisiin myös tutkia.

Tulevaisuudessa viitekehysten voidaan olettaa olevan tärkeässä asemassa verkkosivujen kehityksessä, koska niiden taustajoukkoina toimii Facebookin ja Googlen kaltaisia suuria yrityksiä. Näiden yritysten omissa intresseissä on kehittää aktiivisesti tapoja toteuttaa verkkopalveluita, varsinkin koska verkkopalveluiden käyttäjien määrä tulee nousemaan huomattavasti diginatiivin sukupolven vanhetessa.

7 LÄHTEET

- Angular.io. N. d. Angular Ivy. <https://angular.io/guide/ivy> Viitattu 1.11.2020.
- AngularConnect, 2019. https://www.youtube.com/watch?v=anphffaCZrQ&ab_channel=Angular-Connect Viitattu 1.11.2020.
- AngularJS, 2016. <http://blog.angularjs.org/2016/09/angular2-final.html> Viitattu 1.11.2020.
- Angular. N. d. <https://angular.io/guide/aot-compiler> Viitattu 1.11.2020.
- Cromwell, V., 2016. Betweenthewires.org. <https://betweenthewires.org/2016/11/03/evan-you/> Viitattu 12.8.2019.
- Dries Buyert, Acquia, 2018. Dries Buytaert Shares His View on Decoupled Drupal: When, Why, and How. <https://www.youtube.com/watch?v=Em-Z-oHNMSo> Viitattu 1.11.2020.
- Drupal.org, 2020. https://www.drupal.org/project/project_module Viitattu 28.10.2020.
- Drupal.org, 2018. <https://www.drupal.org/sa-core-2018-002> Viitattu 25.10.2020.
- Drupal.org, 2020. Drupal.org Viitattu 27.10.2020.
- Drupal.org, 2020. <https://www.drupal.org/about/9/faq> Viitattu 28.10.2020.
- Drupal.org, 2020. <https://www.drupal.org/project/drupal/> Viitattu 28.10.2020.
- Fluin, S., 2020. <https://blog.angular.io/version-10-of-angular-now-available-78960babb41?gi=22a8d727a265> Viitattu 1.11.2020.
- getbootstrap.com. N. d. <https://getbootstrap.com/docs/4.0/layout/grid/> Viitattu 28.10.2020.
- Github, 2010. <https://github.com/angular/angular.js/releases?after=v0.9.4> Viitattu 1.11.2020.
- Hunt, P., 2015. Pete hunt | TXJS 2015. <https://www.youtube.com/watch?v=A0Kj49z6WdM> Viitattu 1.11.2020.
- JavaTpoint, N. d. <https://www.javatpoint.com/history-of-joomla> Viitattu 27.10.2020.
- Joomla.org, N. d. https://docs.joomla.org/Joomla!_CMS_versions Viitattu 27.10.2020.
- Joomla.org, N. d. <https://www.joomla.org/core-features.html> Viitattu 27.10.2020.
- Mozilla, 2020. https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Web_frameworks Viitattu 28.10.2020.
- Rascia, T., 2018. <https://www.taniarascia.com/getting-started-with-react/> Viitattu 1.11.2020.
- Reactjs.org, N. d. <https://reactjs.org/docs/introducing-jsx.html> Viitattu 1.11.2020.
- Reactjs.org, N. d. <https://reactjs.org/docs/state-and-lifecycle.html> Viitattu 1.11.2020.
- Rouse, M., 2019. TechTarget. <https://searchcontentmanagement.techtarget.com/definition/content-management-system-CMS> Viitattu 26.10.2020.
- Shopify.com, 2020. <https://www.shopify.com/about> Viitattu 26.10.2020.
- Techterms.com, N. d. <https://techterms.com/definition/framework> Viitattu 28.10.2020.

The Open Source Initiative, 2007. <https://opensource.org/docs/osd> Viitattu 25.10.2020.

Tolvanen, P., 2007. Web-sisällönhallintajärjestelmä - ominaisuudet ja käyttöönotto. Teoksessa: s. 32. Viitattu 25.10.2020

Tolvanen, P., 2007. Web-sisällönhallintajärjestelmä - ominaisuudet ja käyttöönotto. Teoksessa: s. 37-38. Viitattu 25.10.2020

Tolvanen, P., 2009. Vierityspalkki.fi. <https://vierityspalkki.fi/2009/11/03/kasitesekamelskaa-julkaisujarjestelma-cms-portaali-sisallonhallintajarjestelma/> Viitattu 20.10.2020.

w3techs, 2020. https://w3techs.com/technologies/overview/content_management Viitattu 26.10.2020.

Wix, N. d. ,Wix, the Leader in Website Creation, <https://www.wix.com/about/us> Viitattu 27.10.2020.

Wix, N. d. Free Vs. Premium Site. <https://support.wix.com/en/article/free-vs-premium-site> Viitattu 27.10.2020.

Wix.com, N. d. Wix Editor. <https://www.wix.com/features/main> Viitattu 27.10.2020.

Wordpress.org, 2020. <https://fi.wordpress.org/download/releases/> Viitattu 26.10.2020.

Wordpress.org, N. d. <https://wordpress.org/about/philosophy/> Viitattu 26.10.2020.

Wordpress.org, N. d. <https://wordpress.org/support/article/history/> Viitattu 26.10.2020.

Vuejs.org, 2015. <https://vuejs.org/2015/10/26/1.0.0-release/>. Viitattu 1.11.2020.

You, E., 2016. Medium.com. <https://medium.com/the-vue-point/vue-2-0-is-here-ef1f26acf4b8#vh9f94s2m> Viitattu 2.11.2020.

You, E., 2020. vuejs.org. <https://news.vuejs.org/issues/186> Viitattu 2.11.2020.