

WordPress-pohjainen nimipäiväsi- vusto

Jesse Huttunen

Opinnäytetyö
Tammikuu 2021
Tietojenkäsittely ja tietoliikenne
Insinööri (AMK), tieto- ja viestintätekniikka

Tekijä(t) Huttunen, Jesse	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Tammikuu 2021
	Sivumäärä 49	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty: x
Työn nimi WordPress-pohjainen nimipäiväsivusto		
Tutkinto-ohjelma Tieto- ja viestintätekniikka		
Työn ohjaaja(t) Pasi Manninen, Kari Niemi		
Toimeksiantaja(t) Preoni Oy		
<p>Tiivistelmä</p> <p>Nimipäivät ovat monille suomalaisille tärkeitä. Nimipäivät on mahdollista katsoa kalenterista, mutta nimien hakeminen tällä menetelmällä on hidasta. Opinnäytetyön aiheena oli toteuttaa toimeksiantajana toimineelle Preonille uudet verkkosivut nimipäivien näyttämiseen ja nopeaan hakemiseen.</p> <p>Opinnäytetyön tarkoituksena oli oppien syventäminen REST-rajapintojen käytöstä WordPressin ja PHP:n kanssa. Lähtökohta nimipäiväsivustolle oli suhteellisen yksinkertainen, mutta se oli kuitenkin myös hyvin laajennettavissa erilaisilla nimiin liittyvillä tiedoilla.</p> <p>REST-rajapinnat ovat nykypäivänä todella yleinen tapa jakaa dataa internetin välityksellä. Netistä löytyykin melkein loputon määrä erilaisia tietoja sisältäviä REST-rajapintoja. Suurin osa nimipäiväsivuston nimipäivätiedoista haetaan ulkoisen Abalin REST-rajapinnan kautta. Tästä rajapinnasta puuttuu kuitenkin joitakin suomalaisille tärkeitä nimipäiviä sekä muita nimiin liittyviä tietoja. Loput tiedoista haetaan omasta MySQL-tietokannasta, johon tietoja on lisätty useasta eri lähteistä löytyneitä avoimia dataa.</p> <p>Aiempi kokemus REST API:n peruskäytöstä Reactilla sekä PHP:n käytöstä tietokantojen kanssa auttoivat opinnäytetyössä, mutta etenkin WordPress toi toteutukseen odottamatomia haasteita ja ongelmia. Ongelmia ratkoessa oppi paljon uutta siitä, miten näitä toimintoja tulee käyttää etenkin WordPressin päällä toimivalla sivustolla.</p> <p>Työ onnistui ja lopputuloksena tulikin näppärä pieni nimipäiviä näyttävä sivusto, josta on käyttäjän mahdollista nopeasti hakea useanlaisia nimipäiviin ja nimiin liittyviä tietoja, mukaan lukien usean eri valtion nimipäiviä.</p>		
Avainsanat (asiasanat) JavaScript, nimipäivät, PHP, REST API, WordPress		
Muut tiedot (Salassa pidettävät liitteet)		

Author(s) Huttunen, Jesse	Type of publication Bachelor's thesis	Date January 2021
		Language of publication: Finnish
	Number of pages 49	Permission for web publication: x
Title of publication WordPress-based name day website		
Degree programme Information and Communication Technology		
Supervisor(s) Manninen, Pasi & Niemi, Kari		
Assigned by Preoni Oy		
<p>English description</p> <p>Name days are important to many Finns. A common way to view name days is from a calendar but searching for specific names with this method is slow and tedious. The topic of the thesis was to create a new website for Preoni that focuses on displaying and searching for name days and other name-related information.</p> <p>The purpose of the thesis was to deepen the knowledge of using REST APIs on a WordPress-based website using PHP with a relatively simple starting point. After the name day information was added to the site it was also very extendable with various other name-related information.</p> <p>REST APIs are a common way to share data over the internet. There are almost an endless number of REST APIs with widely varying information in them. Most of the name day data is retrieved from Abalin's International name day API. Although Abalin's API has a wide variety of name days, some important name days for Finns are missing. The missing name day and name-related information is retrieved from our own database, where the wanted information has been added from various sources.</p> <p>Previous experience on the basic use of REST APIs with React and the use of PHP with databases helped in the thesis, but especially WordPress brought its own challenges and problems to the project. Solving these problems brought in a lot of experience on how these features should be implemented on a WordPress site.</p> <p>The work was successful and the outcome was a handy little website for showing name days, as well as searching for various types of information related to name days and names, including name days from several different countries.</p>		
Keywords/tags (subjects) JavaScript, Name days, PHP, REST API, WordPress		
Miscellaneous (Confidential information)		

Sisältö

1	Johdanto	5
1.1	Nimipäivät	5
1.2	Tilaaja.....	5
1.3	Tavoite	5
2	Teoreettista pohjaa	6
2.1	Nimipäivät Suomessa	6
2.2	WordPress	7
3	Kehitysympäristö.....	8
3.1	Ampps-virtuaalipalvelimen käyttöönotto	8
3.2	WordPress lapsiteeman käyttöönotto	9
3.3	Nimipäiväsivustolla käytössä oleva data	10
3.3.1	Omat tietokannat	10
3.3.2	Nimipäivä API.....	11
3.3.3	MediaWiki API	12
4	Toteutus.....	12
4.1	Nimipäiväsivusto	12
4.2	Etusivu	13
4.2.1	Etusivun nimipäivät osio.....	13
4.2.2	Nimipäivien tulostus.....	16
4.2.3	Suosituimmat etunimet Suomessa.....	21
4.3	Nimihakusivu	25
4.3.1	Nimihakusivun pääosio.....	25
4.3.2	Väestötietojärjestelmän tiedot	33
4.3.3	Nimen tietoja Wikipedia APIlla.....	36
4.4	Nimitilastosivu	40
5	Pohdinta.....	45
	Lähteet	47

Kuviot

Kuvio 1. Tyylytiedoston lapsiteema kommentti	9
Kuvio 2. Tyylytiedoston käyttöönotto WordPress-funktioitiedostossa	9
Kuvio 3. PHP-kooditiedoston aloitus	10
Kuvio 4. Etunimitilastot omassa tietokannassa	11
Kuvio 5. Nimipäiväsivuston rakenne	13
Kuvio 6. Etusivun nimipäivät osio	14
Kuvio 7. Vaihtoehtoinen kuva valtion lipulle	14
Kuvio 8. Ohjelmointi käyttäjän hakuarvojen vastaanottamiselle	15
Kuvio 9. Kokonaisten sanojen luonti maatunnuksista ja eläinten nimipäivätaulun valinta	15
Kuvio 10. Eläinten nimipäivien SQL-hakukomento	16
Kuvio 11. Ohjelmointi päiväyksen tarkistamiselle	16
Kuvio 12. Eläinten nimipäivien tulostus	18
Kuvio 13. CURL-yhteys Abalinin nimipäivä APIin	18
Kuvio 14. Nimipäivien tulostus	19
Kuvio 15. Suomenruotsalaisten nimipäivien tulostus	20
Kuvio 16. Painike edelliseen päivään siirtymiselle	20
Kuvio 17. Form-elementti päiväyksellä hakemiseen	21
Kuvio 18. Suosituimmat nimet	21
Kuvio 19. Suosituimpien naisten etunimien tulostus	22
Kuvio 20. Suosituimmat etunimet Google Chart -taulukkona.....	22
Kuvio 21 Tietokantatietojen lähetys Google Chartille	23
Kuvio 22. WordPress-funktio PHP-tietojen siirtämiseen JavaScript-koodille	24
Kuvio 23. Google Chartia käyttävä JavaScript-ohjelmointi suosituimmille miesten nimille	25
Kuvio 24. Nimihakusivun pääosio	26
Kuvio 25. Hakukomento eläinten tietokannasta	26
Kuvio 26 Eläinten nimipäivien tulostus	27
Kuvio 27. Nimihaku Abalinin nimipäivä rajapintaan	28
Kuvio 28. Nimipäivätietojen haku Abalinin nimipäivä rajapintaan	29
Kuvio 29. Nimen nimipäivien tulostus	30

Kuvio 30. Nimen ruotsinkielisen nimipäivän tulostus	31
Kuvio 31. Nimipäivien lähetys Wordpress-funktio tiedoston puolelle.....	32
Kuvio 32. WordPress-funktio nimipäivätietojen siirtämiseen JavaScript- ohjelmoinnin puolelle	32
Kuvio 33. JavaScript-ohjelmointi juoksevalle ajastimelle seuraavaan nimipäivään	33
Kuvio 34. Väestötiedot käyttäjälle	34
Kuvio 35. Nimen määrän tulostus suomalaisten etunimenä	35
Kuvio 36. Etunimien ranking-arvojen ohjelmointi	36
Kuvio 37. Wikipedia-nimitiedot käyttäjälle	36
Kuvio 38 Nimiartikkelitiedot sisältävä JSON-tiedoston rakenne	38
Kuvio 39. Nimiartikkelitalukon järjestysnumeron haku	38
Kuvio 40. Sivun id:n haku nimiartikkelin taulukosta.....	39
Kuvio 41. Wikipedia-tietojen haku MediaWiki-rajapinnasta.....	40
Kuvio 42. Nimitilastosivu	41
Kuvio 43. Ohjelmointi nimitilastojen rajaamiseen erisivuille	42
Kuvio 44. Infotekstien tulostus	43
Kuvio 45. SQL-hakukomento nimitilastoille.....	44
Kuvio 46. Nimitilastojen tulostus.....	44
Kuvio 47. Sivunvaihto painikkeiden ohjelmointi	45

Termit ja lyhenteet

cURL	cURL:n avulla voidaan PHP-koodissa hakea tietoja ulkoiselta palvelimelta.
JavaScript	JavaScript on PHP:n tapaan web-ohjelmointikieli. JavaScript-koodi on mahdollista päivittää ja ajaa uudestaan sivujen ollessa auki. JavaScriptiä käytetään etenkin web-selaimissa eli sivujen visuaalisella puolella erilaisten toimintojen luomiseen.
JSON	JavaScript Object Notation on etenkin JavaScript-tietojen tallentamiseen tarkoitettu tiedostomuoto.
MySQL	MySQL on yksi yleisimmin käytössä oleva avoin tietokantamuoto web-palveluille.
PHP	PHP on web-ohjelmointikieli, joka on yleisesti käytössä etenkin palvelinpuolella eli backendissä. PHP toteuttaa koodin vain kerran sivuja luodessa.
REST API	RESTful API:t ovat infrastruktuuri, jolla dataa voidaan siirtää paikasta toiseen.
SQL	Structured Query Language on standardoitu kieli tietokantakyselyjä varten.
SwaggerHub	Swaggerhubin avulla voi luoda, testata ja jakaa Swagger REST APIa kätevästi suoraan selaimen kautta.

1 Johdanto

1.1 Nimipäivät

Nimipäivät sekä nimiin liittyvät tiedot ovat monille suomalaisille mielenkiintoista informaatiota. Vanha tapa nimipäivien hakemiseen on selata nimipäiviä kalenterista, mikä toimii tulevien nimipäivien osalta, mutta jonkun tietyn nimen nimipäivän hakeminen on hidasta. Nykypäivänä nimipäivien hakeminen on mahdollista toteuttaa huomattavasti nopeammilla menetelmillä.

1.2 Tilaaja

Opinnäytetyön toimeksiantajana toimi Jyväskyläläinen Preoni. Preoni omistaa useita erilaisia sivustoja ja opinnäytetyöhön liittyen oleellisin sivusto on Preonin paivyri.fi-kalenterisivusto, jossa käyttäjät voivat tarkistella kalenterin avulla liputuspäiviä sekä nimipäiviä.

1.3 Tavoite

Työn aiheena oli toteuttaa Preonille uudet verkkosivut, jotka keskittyvät nimenomaan nimipäivien näyttämiseen ja hakemiseen. Nimipäiviin keskittyvien sivujen hyödyt ovat etenkin erilaiset hakutoiminnot, joiden avulla voidaan hakea nimipäivät tietyllä arvolla esim. nimellä. Nimipäiväsivuilla voidaan näyttää käyttäjälle kätevästi suomalaisten nimipäivien lisäksi myös muidenkin maiden nimipäiviä. Nimipäivätietojen lisäksi voidaan näyttää muitakin tietoja kuten esim. kuinka monella suomalaisella on tietty nimi käytössä ja mikä on tämän nimen alkuperä ja historia.

Nimipäiväsivustolla nimipäivät ja nimiin liittyvät tiedot saadaan useasta avoimesta lähteestä. Tietoja haetaan suoraan useasta REST-rajapinnasta ja myös omiin tietokantoihin lisätyistä arvoista. Pääosa nimipäivätiedoista on peräisin Abalinin nimipäivä APIsta, josta löytyy suomalaisten nimipäivien lisäksi monen muun maan nimipäivä tietoja (International NameDay API. N.d). Nimipäivä APIsta puuttuu

kuitenkin joitakin etenkin suomalaisille tärkeitä nimipäivä tietoja kuten esim. ruotsinkieliset ja eläinten nimipäivät. Puuttuvat nimipäivätiedot on haettu muista lähteistä ja lisätty omaan MySQL-tietokantaan. Tietokannasta tiedot haetaan PHP:n PDO:lla, jolla ajetaan SQL-komentoja omiin tietokantoihin. Preoni käyttää sivustojensa luomiseen WordPress-ohjelmistoa, opinnäytetyön päärajoitus olikin käyttää nimipäivä sivujen luontiin WordPressia, jotta uusi sivusto olisi myöhemmin siirrettävissä heidän palvelimellensa.

2 Teoreettista pohjaa

2.1 Nimipäivät Suomessa

Nimipäiviin liittyvillä perinteillä on pitkät juuret suomalaisessa historiassa. Nimipäivät ovat monille suomalaisille tärkeitä ja monilla onkin tapana tarkistella kalentereista, milloin kenenkin nimipäivä on. Nimipäivien vieton alkuperä on katollisen kirkon pyhimyskalenterissa. Keskiajalla kristillisissä kulttuureissa pyhimykset olivat tärkeässä asemassa ja heidän muistopäivien vietto olikin tärkeää. 1500 luvun Saksassa pyhimyksien muistopäivien vietto alkoi muuttumaan henkilön sijasta, myös kyseisen nimen juhlistamiseen. Suomen nimipäivä perinne juontaa juurensa 1700-luvulle, jolloin nimipäivä perinne levisi tänne Ruotsista. (Nimipäivien vietto. N.d)

Helsingin yliopiston almanakkatoimisto ylläpitää suomalaista, suomenruotsalaista sekä suomalaisia eläinten nimipäiväluetteloita. Almanakkatoimistolla on tekijänoikeus suomalaisiin nimipäiväluetteloihin, joka vahvistettiin korkeimmassa oikeudessa vuonna 2000. Nimipäivä tietojen näyttäminen verkkosivuilla on kuitenkin sallittua ilman erillistä lupaa, jos sivuilla ei julkaista kerralla enempää kuin kahden viikon nimipäivä tietoja tai viidentoista nimen listaa. Almanakkatoimistolta on mahdollista myös ostaa täydet nimipäivien käyttöoikeudet sekä nimipäivälistat kohtuullisilla hinnoilla. Yhden nimipäiväluettelon hinta on 57 + 24% alv (Nimipäivä hinnasto. N.d). (Nimipäivien tekijänoikeus. N.d)

Almanakkatoimisto lisää uusia nimiä kalenteriin viiden vuoden välein. Suomenkieliseen nimipäiväkalenteriin pääseminen vaati, että nimi on käytössä etunimenä vähintään viidelläsadalla suomenkielisellä kansalaisella. Suomenruotsalaisen nimipäiväkalenteriin pääsy vaatii puolestaan viisikymmentä etunimeä ruotsinkielisellä kansalaisella. (Suomalaiset nimipäivät 2020. 2018)

2.2 WordPress

WordPress on sisällönhallintajärjestelmä verkkosivujen luontia varten. WordPress on kirjoitettu PHP-ohjelmointikielellä ja tietokantanaan se käyttää MySQL:ää. WordPress luotiin alun perin helpompaa blogisivustojen valmistusta ja ylläpitoa varten.

WordPress on lähdekoodiltaan avoin, eikä sen käyttö vaadi erillisiä maksuja tai lupia. (WordPress kansantajuisesti ja helposti, 2016)

Alkunsa jälkeen WordPress on kasvanut suosituimmaksi WWW-sisällönhallintajärjestelmäksi ja nykyään sitä käyttääkin 39 prosenttia miljoonasta suosituimmasta internet-sivustosta. Seuraavaksi suosituin WWW-sisällönhallintajärjestelmä on Drupal, joka on käytössä vain 3.25 prosentilla verkkosivulla. (CMS Usage Distribution in the Top 1 Million Sites. N.d.)

Yleisiin perustarpeisiin WordPressin käyttö ei välttämättä vaadi käyttäjältä ohjelmointi osaamista lainkaan, vaan sivuja on mahdollista luoda myös pelkästään visuaalisilla valikoilla. WordPressin vakio-ominaisuuksien lisäksi WordPress-sivustoon voidaan lisätä haluttuja ominaisuuksia suurella määrällä ilmaisia ja maksullisia lisäosia (Lisäosat. N.d.). Koska WordPress on täysin avoimen lähdekoodin, sen koodeja on kenen tahansa mahdollista muokata ja jatkaa ilman esteitä. WordPressin suosion johdosta siihen liittyviä ohjeita löytyy todella paljon verkosta.

Visuaalisten toimintojen lisäksi WordPress-sivuja sekä toimintoja on mahdollista kirjoittaa verkkopohjaisilla kielillä kuten HTML, CSS, PHP ja JavaScript. WordPress tukee vakiona myös suosittua JavaScript-pohjaista jQuery-kirjastoa. WordPressin päällä ajettavassa jQuery-koodissa on kuitenkin vaihdettava aina dollarimerkin tilalle

jQuery-sana (Pataki & Crawshaw. 2019). WordPress itsessään on kirjoitettu PHP:llä ja siitä löytyykin laajakirjo sen omia PHP-funktiota (Function Reference. N.d.).

WordPressillä sivuja koodatessa kannattaa huomioida, että siinä on useita etenkin turvallisuuden kannalta tärkeitä rajoituksia. WordPress estää vakiona esim. JavaScriptien ajon. JavaScriptien ajoa varten on tehty useita ilmaisia WordPress-lisäosia, mutta JavaScriptit saa toimimaan myös WordPressin wp_enqueue_script-funktiolla (Wp_enqueue_script. N.d.).

3 Kehitysympäristö

3.1 Ampps-virtuaalipalvelimen käyttöönotto

Nimipäiväsivuston prototyyppi tehtiin Ampps-virtuaalipalvelimella. Amppsin asentaminen on suoraviivaista, eikä asennuksen aikana ole tarvetta muuttaa sen vakioasetuksia. Asennuksen jälkeen sen lisäosaksi lisättiin Wordpress. Ampps käynnistyy piilossa, mutta Ampps-valikon saa auki tuplaklikkaamalla Amppsin ikonia Windowsin oikeasta alareunasta löytyvästä ylänuoli-valikosta. Ampps-valikon talokuvakkeella päästään Amppsin ohjaussivustolle. (Ampps. N.d.)

Opinnäytetyössä Ampps tarvitsi lisäosakseen vain WordPressin, sillä Ampps asentaa automaattisesti MySQL-tietokannan, jota voi ohjata esim Amppsisista löytyvällä phpMyadminilla tai laajemmilla toiminnoilla varustetulla MySQL Workbench -ohjelmistolla. WordPress-lisäosa löytyy Amppsin ohjaussivuston vasemmasta valikosta blogien kohdasta. Lisäosan asentamisen on yksinkertaista ja testikäytössä WordPress-sivusto toimii hyvin asennuksen vakioasetuksillakin.

Vakioasetuksilla WordPress-sivusto löytyy asennuksen jälkeen osoitteesta <http://127.0.0.1/wp/>. WordPress admin kirjautuminen löytyy lisäämällä osoitteeseen wp-admin, esim. <http://127.0.0.1/wp/wp-admin>. Kirjautumisen jälkeen pääsee WordPressin admin asetussivuille.

3.2 WordPress lapsiteeman käyttöönotto

Lapsiteemaa vaatii oman teemakansion WordPressin asennuskansioon. Windows pohjaisissa käyttöjärjestelmissä Amppsin vakioasetuksilla sijainti on C:\Program Files\Ampms\www\wp\wp-content\themes. Lapsiteemaa varten tulee lisätä uusi kansio esim. twentytwenty-child. Lapsiteeman kansio sisältää vähintään kolme eri tiedostoa: PHP-kooditiedosto, johon sivun sisältö kirjoitetaan, CSS-tyylitiedoston sekä WordPress-funktiot sisältävän PHP-tiedoston. Tyylitiedoston alkuun kommentoidaan lapsiteeman nimi sekä mitä teemaa se käyttää pohjanaan (Ks. Kuvio 1).

```
/*
Theme Name:  twentytwenty Child
Template:    twentytwenty
Version:     1.0.0
*/
```

Kuvio 1. Tyylitiedoston lapsiteema kommentti

WordPressin funktiotiedostossa CSS-tyylitiedosto otetaan käyttöön wp_enqueue- ja add_action-toimintojen avulla. (Ks. Kuvio 2)

```
add_action( 'wp_enqueue_scripts', 'my_theme_enqueue_styles' );
function my_theme_enqueue_styles() {

    $parent_style = 'twentytwenty-style';

    wp_enqueue_style( $parent_style, get_template_directory_uri() . '/style.css' );
    wp_enqueue_style( 'child-style',
        get_stylesheet_directory_uri() . '/style.css',
        array( $parent_style ),
        wp_get_theme()->get('Version')
    );
}
```

Kuvio 2. Tyylitiedoston käyttöönotto WordPress-funktiotiedostossa

PHP-kooditiedoston alkuun kommentoidaan sivupohjan nimi. WordPress-sivua lisätessä sille valitaan listasta oikea sivupohja kyseisen nimen avulla. Kommentin jälkeen haetaan pääteeman yläpalkki ja lisätään sivun päärunkona toimivat div-elementit (Ks. Kuvio 3). Div-elementtien jälkeen sivustoa voi kirjoittaa normaalisti HTML:llä ja PHP:llä. (Child themes. N.d.)

```
<?php
/*
Template Name: Nimipaivat
*/
get_header();
?>

<div id="primary" class="content-area">
  <main id="main" class="site-main">
```

Kuvio 3. PHP-kooditiedoston aloitus

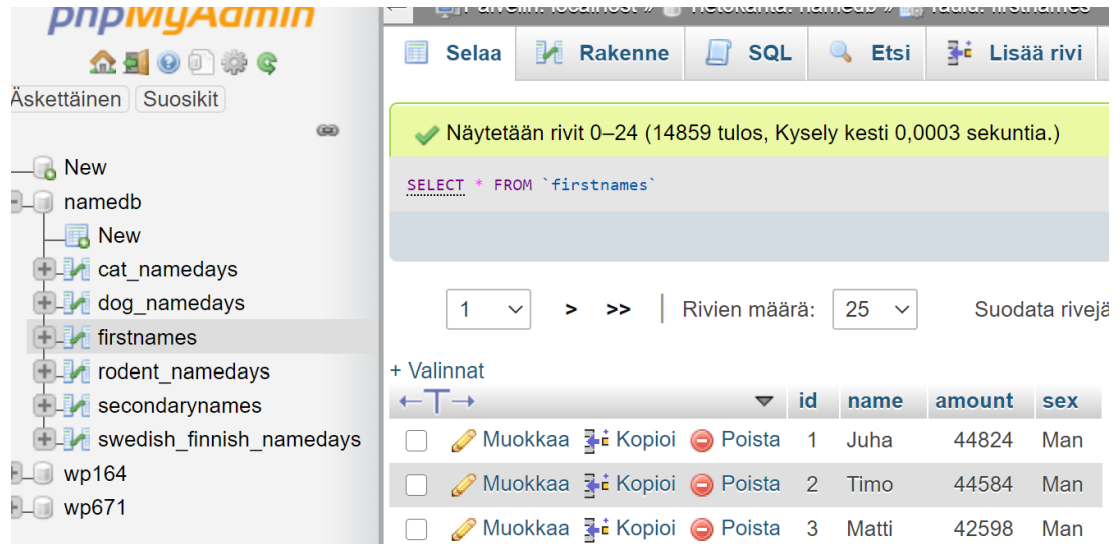
WordPress-sivustolla lapsiteema ja sivupohja on otettava kuitenkin vielä käyttöön. WordPressin admin-sivuilta löytyy teemavalikko. Teemavalikossa pitäisi näkyä uusi lapsiteema muiden vakioiteemojen joukossa. WordPress löytää uuden lapsiteeman automaattisesti CSS-tyylitiedostossa olevan kommentin avulla. Lapsiteema otetaan käyttöön ja lisätään sivu, joka käyttää aiemmin luotua sivupohjaa. WordPressin sivun lisäys- tai muokkausvaiheessa sivupohjan saa valittua oikealta löytyvästä valikosta. Kun sivupohja on käytössä, sivu ajaa sivupohjan sisältämät koodit automaattisesti. Kaikki sivupohjatiedostoon tallennetut ohjelmointimuutokset tulevat voimaan heti, kun sivu päivitetään selaimella.

3.3 Nimipäiväsivustolla käytössä oleva data

3.3.1 Omat tietokannat

Etunimi- ja välinimitilastot on saatu Väestötietojärjestelmästä (Etunimitilasto. 2020). Ruotsinkieliset nimipäivä tiedot ovat peräisin Vårdsvenska-sivustolta (Swedish name-days in Finland. N.d.). Eläinten nimipäivät on haettu puolestaan Webcal-sivustolta (Eläinten nimipäivät. N.d.). Suomen nimitilastot jaetaan Excel-tiedostona, jossa data

on jaettu useaan eri taulukkoon. Tietokantaa varten nimitilaston tiedot yhdistettiin vain kahdeksi tauluksi: etunimi- ja välinimitauluksi. Taulukot tallennettiin kahdeksi CSV-muotoiseksi tiedostoksi. Kaikkien omien tietokantojen tiedot on viety tietokantaan CSV-tiedostoista MySQL Workbenchin Table Data Import Wizard -toiminnolla. Toiminnon jälkeen tietokantatauluihin lisättiin myös ID-osio. (Ks. Kuvio 4)



Kuvio 4. Etunimitilastot omassa tietokannassa

3.3.2 Nimipäivä API

Suurin osa nimipäivätiedoista haetaan Abalinin avoimesta nimipäivä APIsta. Abalinin sivustolla ei ilmoiteta kuka tai ketkä API:n ovat luoneet, mutta API:n liittyviin yhteydenotoista vastaa Vojtěch Nekvapil. Abalinin APIsta löytyy useita tiedonhakutoimintoja, kuten tämän päivän, huomisen sekä eilisen nimipäivän haku tietylle valtiolle. Nimipäiväsivustolla oli viisaampi kuitenkin käyttää nimipäivien hakuun aina päiväystä, tällöin päivien välillä siirtyminen painikkeiden avulla oli helpompaa toteuttaa muuttamalla vain päiväyksen numeroarvoja. Nimipäiväsivustolle otettiin käyttöön myös APIsta löytyvä toiminto hakea nimipäiviä nimen perusteella. Nimipäivä API sisältää Suomen lisäksi useiden muiden valtioiden nimipäivä tiedot. (International NameDay API. N.d.)

Abalinin nimipäivä API löytyy Swaggerhub-sivustolta. Swaggerhubissa nimipäivä API:n eri toimintoja voi testata sekä tarkistella saadun JSON-tiedoston rakennetta (Swaggerhub International NameDay API. N.d.)

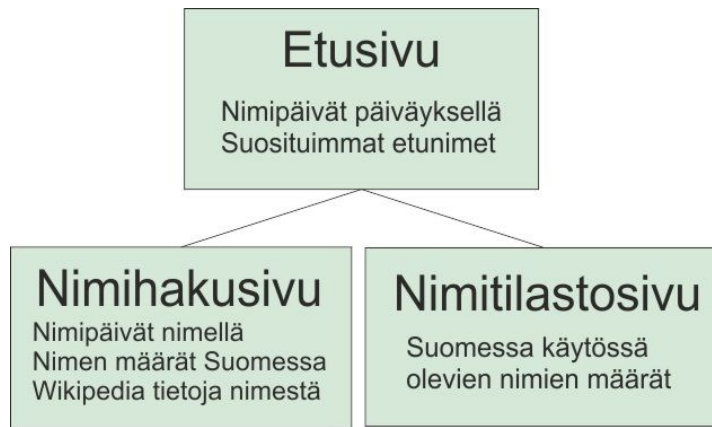
3.3.3 MediaWiki API

MediaWikin APIlla on mahdollista hakea Wikipedia sivujen eli artikkeleiden sisältö (MediaWiki. N.d). MediaWikin APIlla voidaan hakea myös kaikki artikkelit tietyn kategorian sisältä, mutta rajoitteena on se, että sillä pystyy hakemaan kerralla vain 500 artikkelia kategorian sisältä. Opinnäytetyössä oli tarvetta hakea kaikki nimiartikkelit Wikipedian etunimi kategoriasta, joka sisältää kuitenkin yli 1700 nimiartikkelia. Petscan-sivuston avulla kaikkien nimiartikkeleiden tiedot saatiin lisättyä paikalliseen JSON-tiedostoon. JSON-tiedostoa käyttäen oli sitten mahdollista hakea haluttuun nimeen liittyvän Wikipedia artikkelin sivu id (Petscan. N.d). Sivun id mahdollistaa tietyn nimeen liittyvän artikkelin sisällön tulostamisen MediaWikin APIlla.

4 Toteutus

4.1 Nimipäiväsivusto

Nimipäiväsivusto koostuu kolmesta verkkosivusta: etusivusta, nimihakusivusta sekä nimitilastosivusta. Etusivun päätarkoitus on näyttää nimipäivät päiväyksen perusteella sekä mahdollistaa helppo siirtyminen toisille sivuille. Nimihakusivun päätarkoitus on näyttää nimipäivä nimellä haettaessa sekä näyttää muita tähän nimeen liittyviä tietoja. Nimitilastosivu on tarkoitettu Väestötietojärjestelmältä saatujen suomen nimitilastojen selaamiseen. (Ks. Kuvio 5)



Kuvio 5. NimiPäiväSivuston rakenne

WordPress on toteutettu PHP:llä ja osittain tämän takia pääosa nimiPäiväSivuston koodipuolestakin on toteutettu PHP:llä. WordPressin päällä toimivaa sivustoa olisi ollut mahdollista koodata muullakin kielellä, mutta PHP oli muutenkin mieluinen valinta. PHP:n syntaksi on helppo lukuista ja suoraviivaista. PHP:n edut ovat etenkin sen hyvässä yhteensopivuudessa omien tietokantojen kanssa. PHP:n päällimmäisenä puutteena on kuitenkin se, että koodi ajetaan vain kerran sivuja luodessa. Puutteen takia sivustolla on pieniä pätkiä JavaScriptiä etusivun nimitilastojen sekä nimihakusivun ajastimen muodossa.

4.2 Etusivu

4.2.1 Etusivun nimipäivät osio

Jotta käyttäjä näkisi nopeasti minkä valtion nimipäivä tietoja tällä hetkellä näytetään, etusivun yläosasta löytyy haetun valtion lippu. Lippukuvakkeet haetaan Countryflags-sivustolta (Countryflags. N.d). Sivustolla on myös eläinten nimipäivätietoja, tällöin näytetään Icon8-sivustolta haettuja eläinkuvakkeita (Animals. N.d.).

Kuvakkeen jälkeen käyttäjälle näytetään etusivun tärkein sisältö eli haetun päivän nimipäivät. Nimipäivien alta löytyy näppäimet päivien välillä siirtymistä varten. Päiväystä voi vaihtaa myös näppäinten alta löytyvällä hakupalkilla. Nimipäivien haku on

mahdollista myös nimellä, joka vie käyttäjän nimihakusivulle. Molemmista hakupalkeista löytyy valtion valintaominaisuus. (Ks. Kuvio 6)





Nimipäivät tänään

Tänään suomenkielisiä nimipäivä sankareita ovat [Tapani](#), [Teppo](#).

Tänään ruotsinkielistä nimipäivä sankareita ovat [Staffan](#), [Stefan](#).

EILEN	TÄNÄÄN	HUOMENNA
Hae päivällä ja maalla:		
26/12/2020	📅	Suomi ▾ HAE
Hae nimellä ja maalla:		
Kirjoita haettava nimi tähän		Suomi ▾ HAE

Kuvio 6. Etusivun nimipäivät osio

Etusivun koodin alussa lisätään kommentilla sivupohjan nimi WordPressia varten. Otetaan käyttöön WordPress-päätteeman yläpalkki `get_header`-komennolla. Luodaan sivu runkona toimivat `div`-elementit. Sivun yläosassa olevalle kuvakkeelle luodaan linkki vaihtoehtoiseen kuvaan, jota käytetään silloin, jos kuvaketta ei onnistuta hakemaan Countryflags- tai Icons 8 -sivustolta (Ks. Kuvio 7). PHP:n `include`-toiminnolla lisätään koodiin tietokantayhteyden muodostavan tiedoston sisältö.

```
$alt_img='onerror="this.onerror=null;this.src=\'flag_not_found.jpg\';" ';
```

Kuvio 7. Vaihtoehtoinen kuva valtion lipulle

Käyttäjän valitsevat hakuarvot otetaan vastaan HTTP:n GET-metodilla. Jos käyttäjältä ei saada hakuarvoa, esim. silloin, kun sivulle tullaan ensimmäisen kerran, hakuarvoille luodaan myös vakioarvot. Vakioarvoina toimii tämänhetkinen päivä ja valtiona Suomi, koska käyttäjä todennäköisimmin haluaa tietää juuri nämä tiedot. Kokopitkä päiväys jaetaan kolmeen erilliseen muuttujaan: päivä, kuukausi ja vuosi, jotta arvoihin olisi yksinkertaisempi verrata koodissa. Vertailuissa käytetään aina vain

kuukautta ja päivää, koska käyttäjälle ilmoitettavaksi vuodeksi laitetaan joko kuluva vuosi tai siihen lisätään yksi, jos nimipäivä tulee vasta ensi vuonna. (Ks. Kuvio 8)

```
//Gets the values given from the forms, and if some value wasnt
if(isset($_GET["country"]) && !empty($_GET["country"]))
{
    $language_code = $_GET["country"];
}
else{$language_code = "fi";}

if(isset($_GET["dateselect"]) && !empty($_GET["dateselect"]))
{
    $selected_date = $_GET["dateselect"];
}
//date function sets the value as current day in a format like
else{$selected_date = date('Y-m-d');}

//Divides selected_date into 3 different variables: year, month
list($year,$month,$day)=explode("-", $selected_date);

//Year is not needed in SQL commands for animal and swedish finn
$month_day = $month."-".$day;
```

Kuvio 8. Ohjelmointi käyttäjän hakuarvojen vastaanottamiselle

Valittu valtio saadaan maatunnuksena, jota käytetään tässä muodossa URL-osoitteissa, mutta käyttäjää informoivia tekstejä varten luodaan maatunnuksista myös käyttäjää paremmin informoivia sanoja. Eläinten kohdalla otetaan myös animaldb muuttujaan paikallisen tietokantataulun nimi, jota käytetään myöhemmin SQL-komennon sisällä.(Ks. Kuvio 9)

```
if ($language_code=="gr") {$language="kreikkalaisa";}
if ($language_code=="ru") {$language="venäläisiä";}
if ($language_code=="cat") {$language="kissojen"; $animaldb="cat_namedays";}
if ($language_code=="dog") {$language="koirien"; $animaldb="dog_namedays";}
```

Kuvio 9. Kokonaisten sanojen luonti maatunnuksista ja eläinten nimipäivätäulun valinta

4.2.2 Nimipäivien tulostus

Ennen nimipäivää viettäviä nimien tulostusta tarkistetaan, jos käyttäjä on hakenut eläinten nimipäiviä. Tällöin nimipäivä tiedot haetaan omasta MySQL-tietokannasta. Jotta käyttäjä näkisi helposti, että tällä hetkellä näytetään eläinten nimipäiviä, sivun yläosaan lisätään eläin kuvake. Tällä hetkellä eläinten nimipäiviä löytyy kissoille, koirille sekä jyr sijöille. Eläinten nimipäivien hakemista varten luodaan SQL-komento, jolla saadaan halutut tiedot omasta tietokannasta, oikean eläimen taulusta (Ks. Kuvio 10).

```
$animals = $db->query("SELECT * FROM $animaldb where nameday like '%$month_day'");
```

Kuvio 10. Eläinten nimipäivien SQL-hakukomento

Seuraavana eläinten nimipäiväkoodissa selvitetään onko haettu nimipäivä tänään. Jos haettu nimipäivä on tänään, käyttäjälle päiväys ilmoitetaan "tänään"-teksteillä. Jos haettu päiväys ei ole tänään, päiväys ilmoitetaan numeroilla (Ks. Kuvio 11).

```
//Used to count if a name was found or not.
$i=0;
//Selects which text to show the use, the date or "today"
if ($selected_date!=date('Y-m-d')) {
    echo "<h5 class='cent_text' >Nimipäivät ".$day." ".$month." ".$year." "</h5>";
    $is_it_today= "<b class='lightly_bold'>".$day." ".$month." ".$year."</b> ".$language.
}
else{
    echo '<h5 class="cent_text">Nimipäivät tänään</h5>';
    $is_it_today= "Tänään ".$language." nimipäivä sankareita ovat ";
}
echo "<p class='cent_text'>";
echo $is_it_today;
```

Kuvio 11. Ohjelmointi päiväyksen tarkistamiselle

PHP:ssä tämän hetkinen päivä on mahdollista hakea date-funktiolla. Date-funktio antaa tämän hetkisen päiväysarvon antamalla funktiolle halutun arvon lyhenteen, esim. vuosi saadaan yksinkertaisesti Y-kirjaimella. Kirjaimen koko on tärkeä, sillä funktio antaa arvon eri muotoisena sen perustaalla onko kirjain iso vai pieni (Date. N.d).

Eläinten nimipäivät haetaan PDO:n fetch-funktiolla ajamalla SQL-komento while-silmukan ehtona kyseisen eläimen tietokanta tauluun. Silmukka tulostaa saadut tiedot rivi kerrallaan. Silmukan sisällä tietokannasta saadusta päiväyksestä otetaan vain päivä ja kuukausi vertailua varten ja nimipäivän nimet jaetaan yksittäisiksi nimiksi. Eläinten nimipäivät on tietokannassa siten, että jokaisella päivällä on oma nimijono, joka sisältää pilkulla erotettuna kaikki sen päivän nimet. Jotta jokaiselle nimelle saadaan oma linkki sen nimen sivulle, PHP:n explode-funktiolla nimet jaetaan pilkun kohdalta taulukkoon yksittäisiksi nimiksi. Nimi taulukon rivit tulostetaan sitten linkkeinä foreach-silmukalla. Linkkien jälkeen tarkistellaan onko päiväys ollut jo tänä vuonna. Vertailu toteutetaan yhdistämällä kuukauden ja päivän luvut, esim marraskuun 22 päivä olisi 1122, mikä on lukuna suurempi kuin esim lokakuun 13 päivä, joka olisi 1013. Vertailun avulla tiedetään laitetaanko vuodeksi nykyinen vuosi vai nykyinen vuosi +1, jos nimipäivä on seuraavan kerran vasta ensi vuonna. (Ks. Kuvio 12)

```

while($row = $animals->fetch(PDO::FETCH_ASSOC)) {
    $name = $row['name'];
    $nameday = $row['nameday'];

    //Dividing full date into year month and day variables
    list($year,$month,$day)=explode("-", $nameday);

    //Dividing string of that days nameday heroes into individual names
    $names = explode(" ", $name);

    //Loop through the names
    foreach ($names as $one_name){
        echo '<a href="wp/nimihaku/?fname='.$one_name.'&country='.$language_code.'">'.$one_name."</a>";
        //Gives the right value for year based on if the day is already in the past this year.
        $comparison_date=$month.$day;
        $now= date('md');
        if($comparison_date < $now) {
            $year=date('Y')+1;
        }
        else {
            $year=date('Y');
        }
    }
}

```

Kuvio 12. Eläinten nimipäivien tulostus

Jos käyttäjä hakee eläinten nimipäivien sijasta valtion nimipäivää. Haetaan cURLilla Abalinin nimipäivä APIsta halutun valtion tiedot kyseiselle päivälle (Ks. Kuvio 13).

```

$curl = curl_init();
curl_setopt_array($curl, array(
    //For example finnish language code is fi.
    CURLOPT_URL => "https://api.abalin.net/namedays?country=".$language_code."&month=".$month."&day=".$day,
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_FOLLOWLOCATION => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    //CURLOPT_TIMEOUT => 30,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => "GET",
));

$response = curl_exec($curl);
$err = curl_error($curl);
curl_close($curl);

//Checks curl response for errors
if ($err) {
    //Curl had an error
    if ($debug) echo "cURL Error #:" . $err;
    else echo $err;
} else {
    //Data successfully received from api without errors.

    //Decoding received json array
    $data = json_decode($response);
}

```

Kuvio 13. CURL-yhteys Abalinin nimipäivä APIin

Sivun yläosaan tulostetaan eläinten tapaan kuvake, mutta tässä tapauksessa se on haetun valtion lippu. Seuraavana APIlta saatua JSON-dataa aletaan käsittelemään. Ensinnäkin tehdään sama kuin eläinten kohdalla eli tarkistetaan mikä päiväys käyttäjälle näytetään, ”tänään”-teksti vai päiväys numeroina.

APIlta saadusta JSON-tiedosta otetaan nimitiedot muuttujaan. Muuttuja sisältää eläinten tapaan useita nimiä yhdessä merkkijonossa. Nimet jaetaan explode-toiminnolla yksittäisiksi nimiksi taulukkoon. Taulukon sisällä olevat nimet tulostetaan sitten foreach-silmukan avulla nimi linkkeinä (Ks. Kuvio 14).

```
$nameheroes = $data->data->namedays->{$language_code};

//Divides multiple name results into an array
$singular_names=explode(", ", $nameheroes);

//Creating links for each name
foreach($singular_names as $one_name)
{
    echo '<a href="/wp/nimihaku/?fname='.$one_name.'&country='.$language_code.'">'.$one_name.'</a>';
}
```

Kuvio 14. Nimipäivien tulostus

Jos valittu valtio on Suomi, haetaan myös ruotsinkieliset nimipäivät. Suomen ruotsinkielisiä nimipäiviä ei ole Abalinin nimipäivä API:ssa, joten data on haettu eri lähteestä ja lisätty omaan MySQL-tietokantaan. Suomen ruotsinkieliset nimipäivä data on Vardsvenska sivustolta (Swedish namedays in Finland. N.d).

Ennen ruotsinkielisten nimipäivien tulostusta tarkistetaan muiden nimipäivien tapaan miten päiväys näytetään, numeroilla vai ”tänään”-tekstillä. Itse tiedot haetaan omasta tietokannasta SQL-komennolla, jonka hakutuloksena kaikki haetun päivän nimet tulostetaan yksitellen while-silmukalla käyttäen PDO:lla SQL-komentoa silmukan ehtona. (Ks. Kuvio 15)

```
$swedish_finnish = $db->query("SELECT * FROM swedish_finnish_namedays where nameday like '%$month_day'");

//Shows name links
while($row = $swedish_finnish->fetch(PDO::FETCH_ASSOC)) {
    $name = $row['name'];
    echo '<a href="wp/nimihaku/?fname='.$name.'&country=fi">'.$name."</a> ";
}
```

Kuvio 15. Suomenruotsalaisten nimipäivien tulostus

Nimipäivää viettävien nimien jälkeen lisätään käyttäjälle napit päivien välillä siirtymiseen. Nimipäivien välillä hyppimiseen hyödynnetään GET-metodina käyttäjältä saatu päiväys-hakuarvo. Hakuarvoon joko lisätään tai siitä poistetaan yksi päivä. Päiväyksestä ei voi vain poistaa yhtä lukua, koska silloin kuukadet eivät vaihdu oikein. Strtotime-funktiolla saadaan PHP-ohjelmointi lukemaan päiväys arvot oikeasti päiväyksenä ja tällöin siitä voidaan poistaa tai lisätä päivä toisella strtotime-funktiolla. Päiväys ja muut käyttäjän aiemmin valitsemat rajausarvot saadaan vietyä URL-osoitteen sisällä kysymysmerkin jälkeisinä arvoina. Jotta tiedot saadaan lähetettyä oikealle WordPress-sivulle, käytetään WordPressin bloginfo('url') funktiota kirjoittamaan palvelimen URL-osoite ja palvelimen osoitteen perään kirjoitetaan vielä halutun sivun nimi. (Ks. Kuvio 16)

```
<a class="button page_change_button" lang="fi"
href="<?php bloginfo('url'); ?>/nimipaivat/?country=<?php echo $language_code ?>&dateselect=
<?php echo date('Y-m-d',(strtotime ( '-1 day' , strtotime ( $selected_date) ) )); ?>">
<?php if ($selected_date!=date('Y-m-d')) { echo "Edellinen päivä"; }else{ echo "Eilen"; } ?></a>
```

Kuvio 16. Painike edelliseen päivään siirtymiselle

Käyttäjälle luodaan hakupalkit, joilla käyttäjä voi hakea tuloksia päiväyksellä tai nimellä. Molemmissa hakupalkeissa on myös valtion valinnan ominaisuus. Hakupalkit on toteutettu HTML form-elementillä. Oikea valtio pysyy valittuna sivun päivittämisen jälkeen lisäämällä selected-sana sille option-elementille, jonka käyttäjä valitsi. (Ks. Kuvio 17).

```

<label for="country">Hae päivällä ja maalla:</label>
<form action="" method="GET" style='display:flex;'>
  <input type="date" id="start" name="dateselect" value="<?php echo $selected_date;?>">
  <select name="country" id="country" value=language_code>
    <option value="fi" <?php if ($language_code == "fi") echo "selected" ?>>Suomi</option>
    <option value="se" <?php if ($language_code == "se") echo "selected" ?>>Ruotsi</option>
    <option value="dk" <?php if ($language_code == "dk") echo "selected" ?>>Tanska</option>
  </select>
</form>

```

Kuvio 17. Form-elementti päiväyksellä hakemiseen

4.2.3 Suosituimmat etunimet Suomessa

Etusivulla näytetään nimipäivien lisäksi myös muutamia suosituimpia suomalaisia miesten ja naisten etunimiä (Ks. Kuvio 18). Suomalaiset nimitilasto tiedot on saatu Väestötietojärjestelmästä, joka on jakanut tilastot Avoindata-sivustolla (Etunimitilasto. 2020).

Suosituimpia suomalaisia naisten etunimiä

- [Tuula](#) on etunimenä 31 037:lla suomalaisella naisella.
- [Anne](#) on etunimenä 30 458:lla suomalaisella naisella.
- [Päivi](#) on etunimenä 30 088:lla suomalaisella naisella.
- [Anna](#) on etunimenä 29 461:lla suomalaisella naisella.
- [Ritva](#) on etunimenä 29 099:lla suomalaisella naisella.

[Näytä kaikki suomalaiset naisten nimet](#)

Suosituimpia suomalaisia miesten etunimiä

- [Juha](#) on etunimenä 44 824:lla suomalaisella miehellä.
- [Timo](#) on etunimenä 44 584:lla suomalaisella miehellä.
- [Matti](#) on etunimenä 42 598:lla suomalaisella miehellä.
- [Kari](#) on etunimenä 40 091:lla suomalaisella miehellä.
- [Mikko](#) on etunimenä 38 521:lla suomalaisella miehellä.

[Näytä kaikki suomalaiset miesten nimet](#)

[Näytä kaikki suomalaiset nimet](#)

Kuvio 18. Suosituimmat nimet

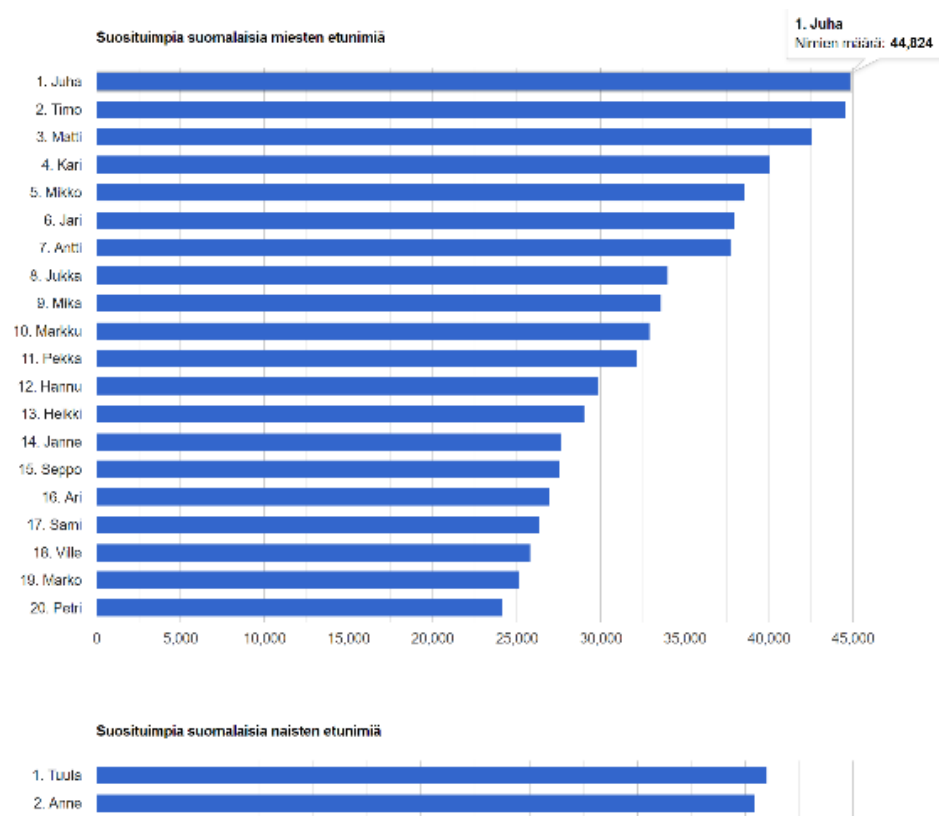
Nimitilastot on lisätty omaan MySQL-tietokantaan. Tietokannasta haetaan viisi suosituinta nimeä sekä niitä käyttävien suomalaisten määrät SQL-komennolla.

Tulokset järjestetään laskevasti sekä rajoitetaan viiteen nimeen jo heti SQL-komennoissa. Koska SQL-komento rajoittaa tulokset, nimet tulostava while-silmukka tulostaa vain viisi riviä nimiä. (Ks. Kuvio 19)

```
$ladies = $db->query("SELECT * FROM firstnames where sex like 'woman' order by amount desc limit 5 ;");
while($row = $ladies->fetch(PDO::FETCH_ASSOC)) {
    $name = $row['name'];
    $amount = $row['amount'];
    echo "<li>".<a href='wp/nimihaku/?fname='.$name.'&country=fi'>".$name."</a> on etunimenä
    <b class='lightly_bold'>".number_format($amount, 0, '.', ' ')."</b>:lla suomalaisella naisella.</li>";
}
```

Kuvio 19. Suosituimpien naisten etunimien tulostus

Suosituimmat etunimet näytetään sivustolla myös visuaalisina tilastoina Google Chartin avulla. Google Chart taulukko mahdollistaa selkeämmän vertaamisen suosituimpien etunimien määrien välillä. Jokainen Google Chartin palkki toimii klikatessa linkkinä sen nimen sivulle. (Ks. Kuvio 20).



Kuvio 20. Suosituimmat etunimet Google Chart -taulukkona

Google Chart tarvitsee sivulle ensimmäisenä tyhjät div-elementit, jonne taulukot tullaan piirtämään JavaScript koodilla. Google Chart itsessään toimii JavaScriptillä, mutta nimitiedot haetaan ensin omasta tietokannasta PHP:llä (Anam. 2018). PHP soveltuu tietokanta kyselyihin paremmin kuin JavaScript. Tietokannasta saadut nimitilasto tiedot lisätään PHP-talukkoon, joka muutetaan JavaScriptiä varten JSON-muotoon. Molempien sukupuolien tiedot lähetään WordPress-funktion kautta JavaScript-koodin puolelle `do_action`-toiminnolla. (Ks. Kuvio 21)

```
//Google chart for women.
$title2 = "Suosituimpia suomalaisia naisten etunimiä";
$div_id2 = "women_chart";
$result2 = $db->query("SELECT * FROM firstnames where sex like 'woman' order by amount desc limit 20 ;")
$rows2 = array();
$table2 = array();
$table2['cols'] = array(
    array('label' => 'Nimi', 'type' => 'string'),
    array('label' => 'Nimien määrä', 'type' => 'number')
);
$ord2=0;
foreach($result2 as $r2) {
    $ord2++;
    $temp2 = array();
    $temp2[] = array('v' => (string) $ord2.". ".$r2['name']);
    $temp2[] = array('v' => (int) $r2['amount']);
    $rows2[] = array('c' => $temp2);
}
$table2['rows'] = $rows2;
$jsonTable2 = json_encode($table2);

//Sending both men and women data through a Wordpress funtion to the google chart javascript.
do_action('run_chart',$title, $jsonTable, $div_id, $title2, $jsonTable2, $div_id2);
```

Kuvio 21 Tietokantatietojen lähetys Google Chartille

Tiedot JavaScriptille vievä WordPress-funktion ottaa etusivulta `do_action`-toiminnolla lähetetyt arvot vastaan `add_action` toiminnolla ja luo näistä arvoista halutun nimiset muuttujat, jotka JavaScript-koodin on sitten mahdollista avata (Ks. Kuvio 22). (Pille J. 2013.)

```
//Nameday Google charts
function chart_run($title, $names, $div_id, $title2, $names2, $div_id2) {
    wp_enqueue_script('my-script', get_stylesheet_directory_uri() . '/skripti.js', array('jquery'))
    wp_localize_script('my-script', 'my_variables', array(
        'title' => ($title),
        'names' => ($names),
        'div_id' => ($div_id),
        'title2' => ($title2),
        'names2' => ($names2),
        'div_id2' => ($div_id2)
    ))
};
}
add_action('run_chart','chart_run',10, 6);
```

Kuvio 22. WordPress-funktio PHP-tietojen siirtämiseen JavaScript-koodille

Google Chart -paketti lisätään WordPressin sivunmuokkaus sivulla etusivun head-osiin Scripts and Styles -lisäosan avulla. Google Chart -paketti otetaan käyttöön JavaScript-koodin alussa. Taulukon tiedoiksi otetaan WordPress-funktiolta saadut nimetilat. Funktiolta tilastot saadaan itse määrittelyillä my_variables-nimillä. Google Chart -taulukon ulkonäköä säädetään options-arvoilla. Taulukolle valitaan div-elementti, jonne se tullaan piirtämään. AddListener-toiminnon avulla listataan ominaisuus, jolla palkkia klikkaamalla pääsee kyseisen nimen sivulle (Tuncay Elvanağaç. 2018.).

Lopuksi suosituimmat suomalaiset nimet näyttävä Google Chart -taulukko piiretään, mutta jotta taulukko skaalautuisi sivukokoa muuttaessa, taulukko on piirrettävä uudestaan jokaisen muutoksen tapahtuessa. JQueryn avulla havaitaan milloin ikkunan koko muuttuu window.resize-toiminnolla. Kun havaitaan sivukoon muuttuminen, Google Chart -taulukko piirretään uudestaan muuttuneelle sivukoolle sopivana. (Ks. Kuvio 23)

```

google.charts.load("current", { packages: ["corechart"] });
google.charts.setOnLoadCallback(Chart_men);
google.charts.setOnLoadCallback(Chart_women);

//Google chart for most popular mens names
function Chart_men() {
  var data = new google.visualization.DataTable(my_variables.names); //Mens data from php
  var options = { //Settings for graph
    title: my_variables.title, //Chart title from php
    hAxis: { minValue: 0,}, //Graph blocks start from value 0, by default it started from abc
    chartArea: { 'width': '70%', 'height': '80%' }, // Chart graph size
    legend: 'none', //Removes unnecessary info block from chart
  };
  var chart = new google.visualization.BarChart(document.getElementById(my_variables.div_id));

  //Click bar to open that names page
  google.visualization.events.addListener(chart, 'select', function (e) {
    var selection = chart.getSelection();
    if (selection.length) {
      var row = selection[0].row;
      let name_with_rank = data.getValue(row, 0);
      let name_without_rank = name_with_rank.substring(name_with_rank.indexOf(' ') + 1);
      location.href = "/wp/nimihaku/?fname="+name_without_rank+"&country=fi";
    }
  });

  //Draw the google chart of men
  chart.draw(data, options);

  //Draw the chart again when ever the window size changes to make the chart responsive.
  jQuery(document).ready(function ($) {
    $(window).resize(function () {
      chart.draw(data, options);
    });
    $(window).trigger('resize');
  });
}

```

Kuvio 23. Google Chartia käyttävä JavaScript-ohjelmointi suosituimmille miesten nimille

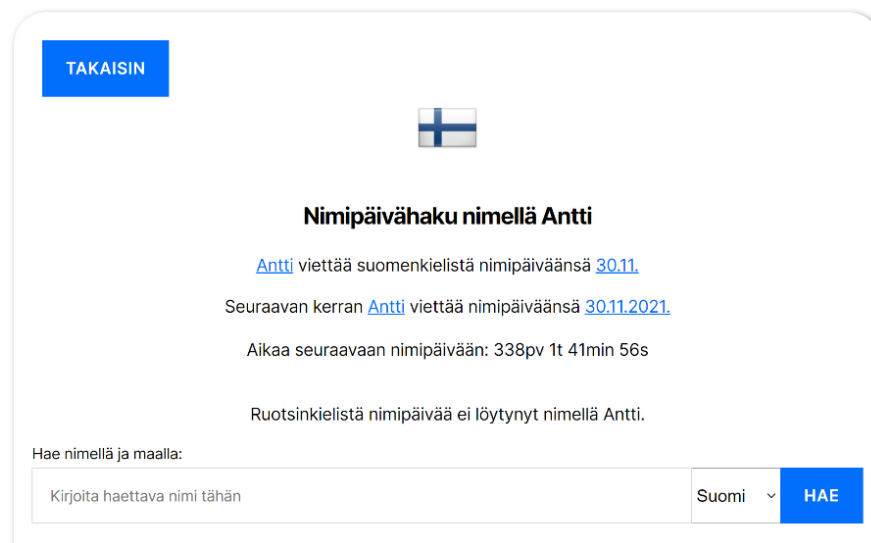
4.3 Nimihakusivu

4.3.1 Nimihakusivun pääosio


Nimihakusivulla käyttäjä pääsee katsomaan laajemmin johonkin tiettyyn nimeen liittyviä nimipäivätietoja. Kuten etusivulla, nimihakusivun yläosiossa käyttäjä saa lipun muodossa tiedon minkä maan tietoja hän on hakenut. Käyttäjälle ilmoitetaan milloin kyseinen nimi viettää nimipäivää valitussa maassa. Päiväys toimii myös linkkinä takaisin etusivulle kyseisen päivän kohdalle. Sivun pääosissa on myös juokseva ajastin seuraavaan nimipäivään.

Pääosion jälkeen käyttäjälle annetaan hakupalkki uudella nimellä hakemista varten. Hakupalkki päiväyksellä hakemista varten on jätetty tarkoituksella pois tältä sivulta,

koska se koettiin testikäytössä vievän tarpeettomasti tilaa. Päiväyksellä hakemista varten käyttäjän voi mennä nopeasti etusivulle takaisinnapilla. (Ks. Kuvio 24)



TAKAISIN



Nimipäivä haku nimellä Antti

[Antti](#) viettää suomenkielistä nimipäiväänsä [30.11.](#)

Seuraavan kerran [Antti](#) viettää nimipäiväänsä [30.11.2021.](#)

Aikaa seuraavaan nimipäivään: 338pv 1t 41min 56s

Ruotsinkielistä nimipäivää ei löytynyt nimellä Antti.

Hae nimellä ja maalla:

Kirjoita haettava nimi tähän

Suomi **HAE**

Kuvio 24. Nimihakusivun pääosio

Nimihakusivun koodin alussa on paljon samaa kuin etusivullakin. Alussa kommentoidaan WordPressille sivupohjan nimi, haetaan pääteeman yläpalkki, lisätään vaihtoehtoinen kuva, jos lippukuvaketta ei löydy netinvälityksellä, otetaan tietokantayhteys ja otetaan vastaan käyttäjän valitsemat hakuarvot. Sivulle lisätään myös sivun päärunkona toimivat div-elementit sekä annetaan käyttäjälle nappi takaisin etusivulle.

Jos käyttäjä on hakenut eläinten nimipäiviä, sivun yläosaan lisätään sen eläimen kuvake valtion lipun sijasta sekä määritetään SQL-komento eläinten nimipäivä tietokannan oikeaan tauluun (Ks. Kuvio 25).

```
$animals = $db->query("SELECT * FROM $animaldb where name like '%$fname%'");
```

Kuvio 25. Hakukomento eläinten tietokannasta

Kaikki tietyn päivän eläinten nimipäivän nimet saadaan yhdessä merkkijonossa. Tämä merkkijono täytyy jakaa useaan eri nimeen explode-funktiolla, jotta jokaiselle nimelle saadaan yksilöity linkki sen nimen sivulle. Jaetut nimet käydään läpi while-silmukalla ja samalla rajataan, että hakutuloksena tulostetaan vain haetun nimen tiedot, ei kaikkia sen päivän nimiä. Tämän jälkeen selvitetään minä vuonna nimipäivä on seuraavan kerran ja tulostetaan tiedot käyttäjän nähtäväksi.

JavaScript-ajastimelle tiedot lähetetään taulukkona, koska joillakin valtioilla on monta nimipäivää samalle nimelle. Taulukosta on sitten helppo JavaScript-ohjelmoinnilla luoda jokaiselle päiväykselle yksilöity ajastin. (Ks. Kuvio 26)

```
while($row = $animals->fetch(PDO::FETCH_ASSOC)) {
    //The wanted data
    $name = $row['name'];
    $nameday = $row['nameday'];
    //Dividing full date from DB into year month and day variables
    list($result_year,$result_month,$result_day)=explode("-", $nameday);
    //Splits name strings as the data is written in a way where each nameday has its names grouped into one string.
    $names = explode(" ", $name);
    //Loop through the names
    foreach ($names as $one_name){
        //We only want data for the exact name we were searching, not all names for that day.
        if($one_name==$fname){
            //Gives the right value for year based on if the day is in the past this year.
            $comparison_date=$result_month.$result_day;
            $now= date('md');
            if($comparison_date < $now) {
                $result_year=date('Y')+1;
            }
            else {
                $result_year=date('Y');
            }
            //Result text for the user, for example "Tupsu viettää kissojen nimipäiväänsä 17.12."
            echo "<a href='/wp/nimihaku/?fname='.$one_name.'&country='.$language_code.' '>'. $one_name."</a> ";
            echo "viettää ".$language." nimipäiväänsä <a href='/wp/?dateselect='.date('Y').'-'.'$result_month.'-'.'$result_day.'&country='.$language_code.' '>'. $result_day."</a>";
            //Second result text row, for example "Seuraavan kerran Tupsu viettää kissojen nimipäiväänsä 17.12.2020."
            echo "<p class='cent_text'>";
            echo "Seuraavan kerran ". "<a href='/wp/nimihaku/?fname='.$fname.'&country='.$language_code.' '>'. $fname."</a>";
            echo "viettää ".$language." nimipäiväänsä <a href='/wp/?dateselect='.date('Y').'-'.'$result_month.'-'.'$result_day.'&country='.$language_code.' '>'. $result_day."</a>";
            //Third result text row, for example "Aikaa seuraavaan kissojen nimipäivään: 1pv 1t 1min 1s".
            echo "<p class='cent_text'>Aikaa seuraavaan ".$language." nimipäivään: ";
            $number++; //Counter for if there are multiple namedays for the name, probably unnecessary for Finnish animals.
            $row_to_add= $result_year."/". $result_month."/". $result_day." 00:00 AM"; //The row of text that will be added to the array.
            array_push($namedays, $row_to_add); //Add new row to the end of the array.
            echo "<span id='countdown'."$number."></span>"; //Element that holds the countdown text.
        }
    }
}
```

Kuvio 26 Eläinten nimipäivien tulostus

Jos käyttäjä ei hakenut eläinten nimipäivää, hän on hakenut tietoja jollain valtiolla. Eri valtioiden nimipäivät haetaan Abalinin nimipäivä APIsta cURL-toiminnon avulla. Ennen cURL-hakua, käyttäjää varten lisätään kuitenkin haetun valtion lippu kuvake. Urlencode-toiminnolla haettu nimi muutetaan URL-osoitteeseen sopivaksi. (Ks. Kuvio 27)

```
$curl = curl_init();

//Establishes connection with the Abalin api where nameday data is retrieved from.
curl_setopt_array($curl, array(
  CURLOPT_URL => "https://api.abalin.net/getdate?name=".urlencode($fname)."&country=".$language_code,
  CURLOPT_RETURNTRANSFER => true,
```

Kuvio 27. Nimihaku Abalinin nimipäivä rajapintaan

Abalinin nimipäivä Apilta tiedot saadaan JSON-muodossa. JSON-tiedosto muutetaan PHP-taulukoksi `Json_decode`-toiminnon avulla määrittämällä sille myös `true`-arvo. Ilman `true`-arvoa dekodaus antaa tiedot objektina (`Json_decode` function. N.d).

Abalinin API antaa kaikkien haetun päivän nimet yhdessä merkkijonossa. Kaikki nimet sisältävä merkkijono katkotaan eri nimiin `preg_split`-toiminnolla. Aiemmin nimet jaettiin `explode`-toiminnolla, mutta jostain syystä `explode` ei osannut jakaa joidenkin eri valtioiden nimiä pilkun kohdalta, vaikka se toimikin suurimmaksi osaksi oikein.

Nimet sisältävä taulukko käydään läpi ja tulostetaan nimenomaan vain haetun nimen nimipäivätiedot. Seuraavana selvitetään vielä minä vuonna nimipäivä on tulossa seuraavan kerran ja annetaan se arvo vuosimuuttujalle. (Ks. Kuvio 28)

```

$data = json_decode($response, true);
$rows = $data["results"];

//Counter for if right name wasnt found and to
$number=0;

//Used to know when the need is to inform
$name_found=false;
echo "<div class='cent_text'";
//If seach string is less than 3 letters abalin
if($rows){
//If user searches with partial name or even a
foreach ($rows as $row) {
//Each day has a string that has multiple name
$names=preg_split('/\s*,\s*/', $row['name']);

//Loop through the names
foreach ($names as $one_name){
//We only want results for the name we sear
if($one_name==$fname){

//Creating date from Aballing api data and
$result_day=$row['day'];
$result_month=$row['month'];
if($result_day <= 9){
$result_day = "0".$result_day;
}
if($result_month <= 9){
$result_month = "0".$result_month;
}
$result_date=$result_month.$result_day;
$now = date ('md');

//Aballin's nameday api doesnt give years
if($result_date < $now) {
$result_year= date('Y')+1;
}
else {
$result_year=date('Y');
}
}
}
}

```

Kuvio 28. Nimipäivätietojen haku Abalinin nimipäivä rajapintaan

Vuoden selvittämisen jälkeen tulostetaan käyttäjälle hänen hakemansa nimipäivä tiedot. Nimipäiväsivustoa testatessa tuli vastaan ongelma, että joillakin valtioilla on monta nimipäivää samalle nimelle. Tämä tuotti ongelmia JavaScript-ajastimen kanssa, joka ratkesi lähettämällä ajastimelle tiedot taulukkona yksittäisten tietojen sijasta. (Ks. Kuvio 29)


```

//Result text for the user, for example "Raakel viettää suomenkielistä nimipäiväänsä
echo "<p>";
echo "<a href='wp/nimihaku/?fname=".$one_name."&country=".$language_code.">"
  . $one_name."</a> viettää ".$language." nimipäiväänsä
<a href='/wp/?dateselect=".date('Y')."-" . $result_month . "-" . $result_day . "&country="
  . $language_code.">" . $result_day . "." . $result_month . "</a>";
echo "</p>";

//Second text row, for example "Seuraavan kerran Raakel viettää nimipäiväänsä 17.12.2017"
echo "<p>";
echo "Seuraavan kerran " . "<a href='wp/nimihaku/?fname=".$one_name."&country="
  . $language_code.">" . $one_name . "</a> viettää nimipäiväänsä
  <a href='/wp/?dateselect=".date('Y')."-" . $result_month . "-" . $result_day . "&country="
  . $language_code.">" . $result_day . "." . $result_month . "." . $result_year . "</a>";
echo "</p>";

//Third text row, for example "Aikaa seuraavaan nimipäivään: 1pv 1t 1min 1s".
echo "<p>Aikaa seuraavaan nimipäivään: ";
$number++; //Counter for if there are multiple namedays in a year for the name, need
$row_to_add= $result_year . "/" . $result_month . "/" . $result_day . " 00:00 AM"; //The row
array_push($namedays, $row_to_add); //Add new date to namedays array.
echo "<span id='countdown' . $number . "></span>";

```

Kuvio 29. Nimen nimipäivien tulostus

Jos käyttäjä on hakenut suomalaisia nimipäiviä, näytetään myös ruotsinkieliset nimipäivät omasta tietokannasta. Tätä koodiosiota voisi parantaa siten, että siinä olisi vain yksi SQL-komento, mutta ei löytynyt toista yksinkertaista tapaa saada tarkistettua onko komennon hakutulokset tyhjä. While-silmukka ei tulosta mitään tietoja, jos sama fetch-komento ajetaan if-ehdossa ja while-silmukassa. Tähän on varmasti joku elegantimpikin ratkaisu, kuin tehdä sama SQL-komento kahdesti, mutta tämäkin tapa kuitenkin toimii. (Ks. Kuvio 30)

```

$swedish_finnish3 = $db->query("SELECT * FROM swedish_finnish_namedays where name like '$fname'");

if($swedish_finnish->fetchAll(PDO::FETCH_ASSOC)) {
    echo "<p>";
    while($row = $swedish_finnish2->fetch(PDO::FETCH_ASSOC)) {
        $name = $row['name'];
        $nameday = $row['nameday'];
        //Dividing full date from DB into year month and day variables
        list($year2,$month2,$day2)=explode("-", $nameday);
        echo '<a href="wp/nimihaku/?fname='.$name.'&country=fi">'.$name."</a> viettää ruotsinkielistä nimipäiväänsä  
<a href='/wp/?dateselect=".date('Y')."-'."$month2."-'."$day2."&country=".$language_code.">".$day2."."$month2."$year2."</a>';
    }
    echo "</p>";

    //Gives the right value for year based on if the day is already in the past this year.
    $comparison_date2=$month2.$day2;
    $now2 = date('md');
    if($comparison_date2 < $now2) {
        $year2=date('Y')+1;
    }
    else {
        $year2=date('Y');
    }

    echo "<p>";
    echo "Seuraavan kerran ".<a href='wp/nimihaku/?fname=".$fname."&country=fi">".$fname."
    </a> viettää ruotsinkielistä nimipäiväänsä <a href='/wp/?dateselect=".date('Y')."-'."$month2."-'."$day2."
    &country=".$language_code.">".$day2."."$month2."."$year2."</a></p>";
    echo "<p>Aikaa seuraavaan ruotsinkieliseen nimipäivään: ";

```

Kuvio 30. Nimen ruotsinkielisen nimipäivän tulostus

Nimihakusivulla on käytössä sama nimihakupalkki kuin etusivullakin. Koska samaa form-elementtiä käytetään usealla sivulla, se on viisainta lisätä erilliseen tiedostoon. Erillisen tiedoston sisältö on sitten mahdollista lisätä jokaiselle sivulle yksinkertaisesti PHP:n include-toiminnolla.

Juoksevalle ajastimelle tiedot lähetetään WordPressin do_action-toiminnolla (Ks. Kuvio 31). Toiminto vie nimipäiväarvot WordPress-funktion kautta JavaScript-ohjelmoinnin puolella olevalle ajastin funktiolle. Ajastimeen käytetään JavaScriptiä sen takia, että PHP:llä on mahdoton tehdä juoksevaa ajastinta. PHP-ohjelmointi ajaa koodin vain kerran sivuja luodessa. JavaScriptillä ajastimen arvoja on mahdollista päivittää joka sekunti.

```
do_action('countdown', $year2, $month2, $day2, $namedays);
```

Kuvio 31. Nimipäivien lähetys Wordpress-funktio tiedoston puolelle

Ruotsinkielisissä nimipäivissä ei löytynyt nimeä, jolla olisi ollut kaksi nimipäivää samana vuonna, joten ruotsinkieliset nimipäivä tiedot lähetään tällä hetkellä ajastimelle yksinkertaisesti kolmella muuttujalla: päivä, kuukausi ja vuosi. Kaikissa muissa nimipäivissä käytetään puolestaan taulukkoa, johon on koodin aikana lisätty kaikki saadut nimipäivät. Yleensä taulukkoon tulee vain yksi päiväys, mutta joissakin maissa, joillakin nimillä on monta nimipäivää samana vuonna. (Ks. Kuvio 32)

```
function countdown_run( $year2, $month2, $day2, $namedays) {
    wp_enqueue_script('scripti', get_stylesheet_directory_uri() . '/countdown_timer.js', array('jquery'))
    wp_localize_script('scripti', 'my_variable', array(
        //Swedish finnish nameday
        'year2' => ($year2),
        'month2' => ($month2),
        'day2' => ($day2),
        //Other namedays
        'namedays' => ($namedays),
    ))
};
}
add_action('countdown', 'countdown_run', 10, 4);
```

Kuvio 32. WordPress-funktio nimipäivätietojen siirtämiseen JavaScript-ohjelmoinnin puolelle

JavaScript-ajastin on toteutettu Javad Shariaty nimisen käyttäjän esimerkin mukaisesti (Javad Shariaty. 2018). Ajastin luo juoksevat kellot kaikille nimipäivätaulukon päiväyksille ja erikseen myös ruotsinkielisille nimipäiville, jos sellainen on saatu (Ks. Kuvio 33).

```

var namedays = my_variable.namedays;
namedays.forEach(myFunction);
function myFunction(item, index) {
    if (index != 0) {
        timedown(item, "countdown" + index);
    }
}

//Timer for Swedish finnish namedays.
var end2 = my_variable.year2 + "/" + my_variable.month2 + "/" + my_variable.day2 + " 00:00 AM";
if (my_variable.year2) {
    timedown(end2, "countdown_swe");
}

//Creates a countdown timer
function timedown(ti,id){
    // Set the date we're counting down to
    var countDownDate = new Date(ti).getTime();

    // Update the count down every 1 second
    var x = setInterval(function() {

        // Get todays date and time
        var now = new Date().getTime();

        // Find the distance between now an the count down date
        var distance = countDownDate - now;

        // Time calculations for days, hours, minutes and seconds
        var days = Math.floor(distance / (1000 * 60 * 60 * 24));
        var hours = Math.floor((distance % (1000 * 60 * 60 * 24)) / (1000 * 60 * 60));
        var minutes = Math.floor((distance % (1000 * 60 * 60)) / (1000 * 60));
        var seconds = Math.floor((distance % (1000 * 60)) / 1000);

        // Output the result in an element with id="demo"
        document.getElementById(id).innerHTML = (days!=0 ? days + "pv " : '') + (hours!= 0 ? hours + "t " : '')
        + (minutes != 0 ? minutes + "min " : '') + seconds + "s ";

        // If the count down is over, write some text
        if (distance < 0) {
            clearInterval(x);
            document.getElementById(id).innerHTML = "Nimipäivä on tänään.";
        }
    }, 1000);
}

```

Kuvio 33. JavaScript-ohjelmointi juoksevalle ajastimelle seuraavaan nimipäivään

4.3.2 Väestötietojärjestelmän tiedot

Nimihakusivulla näytetään myös Väestötietojärjestelmän tietoja haetun nimen määrästä Suomessa (Ks. Kuvio 34). Väestötietojärjestelmä on jakanut nimitilastot avoimesti avoindata sivustolla (Etunimitilasto. 2020).

Väestötietojärjestelmän nimitietoja nimellä Eeri

- [Eeri](#) etunimisiä miehiä on **21** kappaletta.
- [Eeri](#) etunimisiä naisia on **13** kappaletta.
- Yhteensä [Eeri](#) etunimi on käytössä **34** henkilöllä.
- [Eeri](#) nimen ranking miesten etunimenä on **2 939**.
- [Eeri](#) nimen ranking naisten etunimenä on **3 943**.
- [Eeri](#) on muuna nimenä **7**:lla miehellä.
- [Eeri](#) on muuna nimenä **8**:lla naisella.
- Yhteensä [Eeri](#) on muuna nimenä käytössä **15** henkilöllä.
- [Eeri](#) nimen ranking miesten muuna nimenä on **4 482**.
- [Eeri](#) nimen ranking naisten muuna nimenä on **4 154**.

[Näytä kaikki suomalaiset nimet](#)

Kuvio 34. Väestötiedot käyttäjälle

Tällä hetkellä nimipäiväsivustolla on vain Suomen nimitilastoja. Muistakin maista löytyy varmasti samankaltaista dataa, jota nimipäiväsivustolle on mahdollista lisätä sivuja jatkokehittäessä, mutta koska tilastotietoja on tällä hetkellä vain suomalaisista nimistä, tilastot luodaan vain silloin, kun käyttäjä on etsinyt suomalaisia nimipäiviä.

Nimitilastot haetaan omasta tietokannasta SQL-komennolla while-silmukan ehtona PDO:lla (Ks. Kuvio 35). Suomessa nimet on usein jaettu sukupuolen mukaan, mutta Suomessa on käytössä myös sukupuolineutraaleja nimiä. Silmukka tulostaa kerralla molempien sukupuolien tiedot, jos sellaisia löytyy. Jos toisella sukupuolella ei ole yhtäkään nimen käyttäjää, riviä ei tulosteta. Tilastotiedot tulostetaan käyttäjälle lauseiden muodossa, esim. ”Meri etunimisiä naisia on 3 337 kappaletta.”. Jos nimi on käytössä molemmilla sukupuolilla, näytetään myös summatut tulokset, esim. ” Yhteensä Eeri etunimi on käytössä 34 henkilöllä.”.

```

while($row = $sql->fetch(PDO::FETCH_ASSOC)) {
    $name = $row['name'];
    $amount = $row['amount'];
    $gender = $row['sex'];
    if($gender == "Woman"){
        $sex = " naisia ";
    }
    else if($gender == "Man"){
        $sex = " miehiä ";
    }
    else $sex = " ihmisiä ";
    echo "<li><a href='wp/nimihaku/?fname=".$name."&country=fi'>".$name."</a> etunimisiä ".$sex."
on <b class='lightly_bold'>".number_format($amount, 0, '.', ' ')
."</b> kappaletta.</li>";
    $total += $amount;
    $i++;
}

if($i > 1){
    echo "<li>Yhteensä <a href='wp/nimihaku/?fname=".$name."&country=fi'>".$name."</a> etunimi on käytössä <b class='lightly_bold'>".
    number_format($total, 0, '.', ' ')."</b> henkilöllä.</li>";
}

```

Kuvio 35. Nimen määrän tulostus suomalaisten etunimenä

Nimien rankingit eli järjestysnumerot nimen suosiolle on toteutettu Juan nimisen käyttäjän esimerkin mukaisesti (Juan. 2013.). Esimerkiksi Suomen suosituimman naisten etunimen eli Tuulan ranking on 1.

Järjestysnumerot luodaan suoraan SQL-komennossa. PDO:n avulla ajetaan kyseinen SQL-komento ja tulostetaan sen tulokset while-silmukalla. Silmukan sisällä tarkistetaan kumman sukupuolen tulokset ovat ja tulostetaan nimen ranking-tiedot käyttäjälle lauseen muodossa, esim ”Meri nimen ranking naisten etunimenä on 192.”. Pitkälti sama koodi tehdään uudestaan myös muille nimille, jossa tiedot haetaan vain tietokannan eri taulusta kuin etunimien kohdalla. (Ks. Kuvio 36)

```

$sql3 = $db->query("SELECT sex, name, 1+(SELECT count(*) from firstnames
a WHERE a.amount > b.amount AND sex like '$gender') as RNK, amount FROM
firstnames b WHERE Name = '$fname'");

while($row = $sql3->fetch(PDO::FETCH_ASSOC)) {
    $name = $row['name'];
    $amount = $row['amount'];
    $rank= $row['RNK'];
    $gender = $row['sex'];
    if($gender == "Woman"){
        $sex = " naisten ";
    }
    else if($gender == "Man"){
        $sex = " miesten ";
    }
    else $sex = " ihmisten ";

    echo "<li><a href='wp/nimihaku/?fname=".$name."&country=fi'>".$name."</a>
nimen ranking ".$sex." etunimenä on <b class='lightly_bold'>".
number_format($rank, 0, '.', ' ')."</b></li>";

```

Kuvio 36. Etunimien ranking-arvojen ohjelmointi

4.3.3 Nimen tietoja Wikipedia APilla

Nimihakusivulla näytetään myös nimeen liittyvän Wikipedia-artikkelin alkutiivistelmät. Tiivistelmästä löytyy yleensä tiivistetysti nimen alkuperää, historiaa ja muita mielenkiintoisia tietoja. Tiivistelmän alta löytyy myös linkki, jolla käyttäjä pääsee lukemaan koko nimiartikkelin Wikipedian puolella. (Ks. Kuvio 37)

Wikipedia tietoja nimellä Anelma:

Anelma on suomalainen naisen etunimi. Nimi johtuu pyytämistä tarkoittavasta verbistä anoa mutta sen esikuvina ovat olleet myös vanhemmat nimet Nelma, Ihanelma ja Sanelma. Nimen keksijänä pidetään raittius- ja naisasianajana tunnettua Edla Kojosta, joka antoi sen vuonna 1906 syntyneelle tyttärelleen, Anelma Vuoriolle. Anelma nimipäivä on 2. joulukuuta. Suomessa nimi Anelma on 1900-luvun alusta vuoden 2019 loppuun mennessä annettu 4 464 naiselle ja alle viidelle miehelle.

[Lue lisää wikipediassa](#)

Kuvio 37. Wikipedia-nimitiedot käyttäjälle

Wikipedia-tietojen haku tiettyyn nimeen liittyen toteutettiin ensin suoraviivaisesta hakemalla MediaWikin APilla Wikipedia-artikkelia nimen perusteella, joka toimikin esim. Pekka nimeä hakemalla, mutta testatessa tässä tavassa esiintyi useita ongelmia. Hakemalla esim. Simo nimeä, hakutuloksena saatiinkin Simon kunta, eikä

haluttua nimiartikkelia. Joissain tilanteissa, joissa haluttu artikkeli oli epäselvä, API ei näyttänytkään mitään artikkelia, vaan listaa mahdollisista artikkeleista.

Ongelman ratkaiseminen vaati sitä, että nimiartikkeli täytyy avata suoraan sen omalla sivu id:llä. Wikipedian etunimikategoriasta löytyy kaikki nimiin liittyvät Wikipedia-artikkelit. MediaWikin APIlla on mahdollista hakea Wikipedia-kategorian sisällä olevien artikkeleiden sivu id:t, mutta toiminto on rajoitettu maksimissaan vain 500:n artikkelin tietojen hakemiseen kategorian sisältä. MediaWikin API:n haun tuloksia ei pystynyt myöskään rajaamaan suoraan kyseiseen nimeen. API:n maksimi tuloksien määrä ei ollut tässä tilanteessa riittävä, sillä etunimikategoria sisältää yli 1700 nimiartikkelia.

Ongelma kierrettiin lisäämällä etunimikategorian artikkeleiden tiedot Petscan-sivuston avulla Json-tiedostoon (Petscan. N.d). JSON tiedosto tallennetaan paikallisesti WordPress-palvelimelle, josta se saadaan avattua yksinkertaisesti PHP-ohjelmoinnin `file_gets_content`-funktiolla. Avatusta JSON-tiedostosta on sitten mahdollista hakea halutun nimiartikkelin sivu id. Id:llä voidaan sitten lopulta hakea sen nimiartikkelin sisältö MediaWikin APIlla.

Petscan-sivulla luodussa JSON-tiedostossa nimet ovat taulukoissa päätaulukon sisällä. Nimipäiväsivuston käytössä JSON-tiedostosta selvitetään nimitaulukon järjestysnumero päätaulun sisällä, tällöin saadaan avattua suoraan oikean nimen taulukko. Nimen taulukosta löytyvistä tiedosta tarvitaan sitten ottaa vain Wikipedia artikkelin sivu id. (Ks. Kuvio 38)


```

array(1738) {
  [0]=>          Taulukon numero
  array(9) {
    ["id"]=>
    int(5905)          Sivun id
    ["len"]=>
    int(19771)
    ["metadata"]=>
    array(1) {
      ["wikidata"]=>
      string(9) "Q12805716"
    }
    ["n"]=>
    string(4) "page"
    ["namespace"]=>
    int(0)
    ["nstext"]=>
    string(0) ""
    ["q"]=>
    string(9) "Q12805716"
    ["title"]=>
    string(7) "Vilhelm"          Nimi
    ["touched"]=>
    string(14) "20201129073604"
  }
  [1]=>
  array(9) {
    ["id"]=>

```

Kuvio 38 Nimiartikkelitiedot sisältävä JSON-tiedoston rakenne

Koodissa haetaan ensin PHP:n `array_search`-toiminnolla halutun nimen nimitaulukon järjestysnumero päätaulukon sisällä. Luku laitetaan `key`-nimiselle muuttujalle. Yleensä artikkelin nimi on vain nimi, esim Pekka. Joissain nimiartikkeleissa otsikko on kuitenkin nimi + (nimi), esim. Simo (nimi). Ainakin yhdessä artikkelissa otsikko oli myös nimi + (etunimi). `Array_search`-toiminnolla on mahdollista hakea tuloksia vain tarkalla arvolla, joten se ei osannut hakea oikein kaikkia artikkeleita. Ongelma ratkaistiin siten, että jos tuloksia ei löydy pelkällä nimellä, hakusanaan lisätään "(nimi)" ja jos sekään ei toiminut koitetaan lisätä hakuun vielä "(etunimi)". (Ks. Kuvio 39)

```

//Key is the array number with the wanted name. Key is later used to get the right pag
$key = array_search($fname, array_column($json_decoded['*'][0]['a']['*'], 'title') );

//Trying to find the name article again. Needed only because some name articles have
if ($key == false) {
    $key = array_search($fname."_(nimi)" , array_column($json_decoded['*'][0]['a']['*']

```

Kuvio 39. Nimiartikkelitalukon järjestysnumeron haku

Taulukon järjestysnumerolla eli tässä tapauksessa key-muuttujalla päästään suoraan halutun nimen taulukkoon. Nimen taulusta löydetään Wikipedia-artikkelin sivu id, joka annetaan pageid-muuttujalle jota käytetään myöhemmin tiedot hakevassa URL-osoitteessa (Ks. Kuvio 40).

```
if($key != false) {
    //Gets the Wikipedia page id that will be used to open
    $pageid = $json_decoded['*'][0]['a']['*'][$key]['id'];
```

Kuvio 40. Sivun id:n haku nimiartikkelin taulukosta

Jos nimiartikkeli löytyi, haetaan cURLilla sivun id:n avulla Wikipedia-artikkelin alkutiivistelmät. URL-osoite, jota käytetään hakemaan Wikipedia-artikkelin sisältö, pitää sisällään useita rajoituksia siihen mitä tietoja ja missä muodossa ne saadaan rajapinnasta. Pitkä URL-osoite on äkkiseltään hankalalukuinen, mutta selkenee kun huomaa, että &-merkki aloittaa aina uuden rajausarvon.

- /w/api.php? ilmoittaa, että halutaan tietoja API:ltä, ei vain tavallista Wikipedia-sivua.
- Format=JSON muuttaa saadun paketin muodon JSON:iksi.
- Action=query muuttaa miten data saadaan ja miten sitä pystyy rajoittamaan. Parse on yksi toimiva vaihtoehto tähän, mutta query osoittautui paremmaksi tässä tilanteessa.
- Prop=extracts antaa Wikipedia-artikkelin sisällön puhtaana tekstinä tai rajoitettuna HTML:nä.
- Exintro rajoittaa saadut tekstit vain artikkelin alkutiivistelmiin.
- Explaintext antaa artikkelin sisällön vain puhtaana tekstinä.
- Redirects=1 antaa arvon kuinka monesti haku saa ohjata uudestaan.
- Pageids=\$pageid antaa API:lle haettavan nimiartikkelin sivun id:n.

Kaikki MediaWikin Wikipedia-apissa toimivat rajausarvot ja niiden merkitykset löytyvät MediaWiki-sivustolta (MediaWiki. N.d). (Ks. Kuvio 41)

```

$url = "https://fi.wikipedia.org/w/api.php?format=json&action=query
&prop=extracts&exintro&explaintext&redirects=1&pageids=".$pageid;

$ch = curl_init( $url );
curl_setopt( $ch, CURLOPT_RETURNTRANSFER, true );
$output = curl_exec( $ch );
curl_close( $ch );

if ($err) {
    if ($debug) echo "cURL Error #:" . $err;
    else echo $err;
} else {
    //Wikipedia json received
    $result = json_decode( $output, true );

    //Writes the wikipedia name article summary for the user to read.
    echo $result["query"]["pages"][$pageid]["extract"];

```



Kuvio 41. Wikipedia-tietojen haku MediaWiki-rajapinnasta

Minkä tahansa Wikipedia-sivun saa avattua sivu id:n avulla kirjoittamalla URL-osoitteen `/?curid=`, minkä jälkeen sivu id kirjoitetaan lukuina. Esim. Pekka nimen sivu on `/?curid=231089`. Wikipedia-tietojen alle käyttäjälle annetaankin linkki suoraan kyseiseen nimiartikkeliin juuri tällä menetelmällä tai Wikipedian etusivulle, jos nimeä ei löytynyt.

4.4 Nimitilastosivu

Nimitilastosivulla käyttäjälle näytetään nimitilastoja Suomessa käytössä olevista nimistä. Nimitilastoja näytetään 40 kappaletta per sivu. Jos tuloksia ei rajoitettaisi ollenkaan, sivut latautuisivat erittäin hitaasti ja loisivat sivulle erittäin pitkät listat nimistä, joita on työläs selata etenkin puhelimella. Tuloksien rajaaminen nimenoman neljäänkymmeneen nimeen päädyttiin siksi, että määrä on jaettavissa tasan neljään kymmenen nimen riviin desktop-näkymässä, kahteen kymmenen nimen riviin tablet-näkymässä sekä yhteen sopivan lyhyen mittaiseen riviin mobiililla. Riveihin jako on toteutettu CSS:n `columns`-arvolla, jonka avulla rivien määrä on helppo muuttaa sivukoon vaihtuessa.

Hakutulokset on mahdollista rajata joko etunimiin tai muihin nimiin. Käyttäjä voi rajata tuloksia myös sukupuolen mukaan sekä päättää ovatko tulokset nousevassa vai laskevassa järjestyksessä. Sivujen välillä siirtymistä varten sivun alaosasta löytyy navigointipainikkeet. Navigointipainikkeiden välissä on myös sivulista, jonka avulla käyttäjä voi valita haluamansa tietyn sivunumeron. Jos käyttäjä tietää minkä nimen tiedot hän haluaa, nimitilastosivulta löytyy myös hakupalkki, jolla hän pääsee suoraan sen nimen nimihakusivulle. Nimihakusivulta käyttäjä löytää kyseisen nimen tilastotiedot muiden nimipäivä tietojen lisäksi. (Ks. Kuvio 42)

Nimien määrät Suomessa

Rajaa hakutuloksia

Etunimi ▾

Naiset ▾

Laskeva ▾

Hae nimellä ja maalla:

Kirjoita haettava nimi tähän

Suomi ▾

HAE

Sivu 1. Naisten etunimiä löydetty yhteensä **8 488**. Nimien järjestys laskeva.

• 1. Tuula 31 037	• 11. Tiina 25 923	• 21. Satu 21 456	• 31. Ella 19 544
• 2. Anne 30 458	• 12. Riitta 25 495	• 22. Raija 21 350	• 32. Johanna 19 478
• 3. Päivi 30 088	• 13. Tarja 24 822	• 23. Eeva 21 337	• 33. Kirsi 19 402
• 4. Anna 29 461	• 14. Laura 23 479	• 24. Heidi 21 330	• 34. Arja 18 814
• 5. Ritva 29 099	• 15. Pirkko 23 284	• 25. Hanna 21 260	• 35. Maija 18 173
• 6. Leena 28 680	• 16. Selja 21 949	• 26. Sanna 21 059	• 36. Paula 17 797
• 7. Pirjo 28 259	• 17. Aino 21 811	• 27. Anja 20 625	• 37. Ulla 17 387
• 8. Sari 27 761	• 18. Liisa 21 499	• 28. Eija 20 286	• 38. Jenni 16 851
• 9. Minna 27 731	• 19. Jaana 21 481	• 29. Maria 20 242	• 39. Elina 16 275
• 10. Marja 26 794	• 19. Sirpa 21 481	• 30. Merja 19 718	• 40. Sirka 16 251

EDELINEN SIVU

Sivu 1 ▾

SEURAAVA SIVU

Kuvio 42. Nimitilastosivu

Nimitilastosivulla näytetään Väestötietojärjestelmän jakamia avoimia nimitilastoja Suomessa käytössä olevista etunimistä. Väestötietojärjestelmä jakaa tilastoja Avoin-data sivustolla Excel-tiedoston muodossa (Etunimitilasto. 2020). Excel-tiedostossa tiedot on jaettu useisiin eri taulukoihin. Tietokantaa varten taulukkojen tiedot täytyi yhdistää ensin Excelillä kahteen tauluun, etunimiin ja muihin nimiin. Lopuksi molemmat taulut tallennettiin erikseen kahteen CSV-tiedostoon. CSV-tiedostosta etunimi ja

välinimitilastot on sitten mahdollista lisätä omaan tietokantaan MySQL Workbench sovelluksen Table data import wizard -toiminnolla.

Nimitilastosivun alussa on muiden sivujen tapaan kommenttina sivupohjan nimi WordPressiä varten sekä sivun päärunkona toimivat div-elementit. Lippukuvakkeelle linkitetään paikallinen korvike, jos lippu kuvaa ei saada ulkoisesta lähteestä. Vasemman ylänurkkaan lisätään takaisin nappi etusivulle. Koska nimitilastoja on tällä hetkellä vain Suomesta, tälle sivulle lisätään suoraan Suomen lipun kuvake. Käyttäjältä saadut rajausarvot otetaan vastaan ja muodostetaan myös yhteys omaan tietokantaan.

Nimitilastosivun rajausominaisuudet näytettävien nimien suhteen luodaan kolmella eri select-elementillä. Tietojen rajauslomake on pitkälti samankaltainen kuin esim. nimihakupalkissa oleva form-elementti, mutta erona on, että rajausarvoa muuttaessa form-elementti lähettää tiedot automaattisesti ilman tarvetta erilliselle vahvistus napille. HTML-lomake on mahdollista lähettää automaattisesti muutoksen tapahduessa antamalla this.form.submit-käsky onchange-arvolle.

Nimitilastojen määrä per sivu rajoitetaan neljään kymmeneen kappaleeseen. Paginointi eli nimitilastotietojen laittaminen eri sivuihin on toteutettu Saqlain nimisen Stackoverflow käyttäjän esimerkin mukaisesti (2017). Paginointia varten lasketaan kuinka monta sivua tietokannan tiedoilla saadaan. (Ks. Kuvio 43)

```
//Limit for how many names are shown per page;  
$results_per_page = 40;  
//Counts total amount of name results  
$query = "SELECT count(*) FROM $which_name Where sex like '$gender';"  
//Counting number of pages for pagination.  
$sql = $db->query($query);  
$total_results = $sql->fetchColumn();  
$total_pagenumbers = ceil($total_results/$results_per_page);  
$starting_limit = ($pagenumber-1)*$results_per_page;
```

Kuvio 43. Ohjelmointi nimitilastojen rajaamiseen erisivuille

Käyttäjää varten luodaan informaatiotekstejä. Informaatioteksteissä käyttäjälle ilmoitetaan yksinkertaisella lauseella millä tavalla nimitilastojen tuloksia tällä hetkellä rajoitetaan ja kuinka paljon hakutuloksia on saatu. Jos käyttäjä on hakenut esimerkiksi naisten etunimiä, informaatioteksti on ”Naisten etunimiä löydetty 8 488 kpl”. Informaatio teksteissä käytettiin aiemmin myös ilmoitusta ovatko tulokset nousevassa vai laskevassa järjestyksessä, mutta tämä koettiin turhaksi, koska tilastoista näkee helposti kummassa järjestyksessä tulokset ovat ja myös rajauskohdasta on mahdollista tarkistaa tämänhetkiset rajausarvot. (Ks. Kuvio 44)

```
//Counts amount of names to show in the info
$count = "SELECT count(*) FROM $which_name Where sex like '$gender'";
$sql2 = $db->query($count);
$total = $sql2->fetchColumn();

//Whatnames are shown (firstname, secondary or all)
if($which_name == "secondarynames"){
    $translate_name = "muuta nimiä";
}
else $translate_name = "etunimiä";

//Pagenumber for info section
echo "<br><div class='justify_cent flexing wrap'>";
//echo "<p><b class='lightly_bold'>Sivu ".$pagenumber.". </b></p>";
echo "<p class='cent_tex_mobile'>";

//What sex is selected (men, women or all)
if($gender == "woman"){
echo " Naisten ".$translate_name." löydetty <b class='lightly_bold'>".number_format($total, 0, '.', ' ');
}
else if($gender == "man"){
    echo " Miesten ".$translate_name." löydetty <b class='lightly_bold'>".number_format($total, 0, '.', ' ');
}
else if($gender == "%"){
    echo ucfirst($translate_name). " löydetty <b class='lightly_bold'>".number_format($total, 0, '.', ' ');
}
}

echo "</b> kpl.</p>";
```

Kuvio 44. Infotekstien tulostus

Nimitilastot sisältävään tietokantatauluun luodaan SQL-komento. Komento luo samalla nimille myös näiden järjestysnumerot (Juan. 2013). Tuloksiksi rajoitetaan vain halutun paginointisivun tiedot. (Ks. Kuvio 45)

```
$show = "SELECT name, 1+(SELECT count(*) FROM $which_name a Where a.amount > b.amount
AND sex like '$gender') as RNK, amount, sex FROM $which_name b WHERE sex like '$gender'
order by amount $order LIMIT ?,?";

$r = $db->prepare($show);
$r->execute([$starting_limit, $results_per_page]);
```

Kuvio 45. SQL-hakukomento nimitilastoille

Saadut nimet tulostetaan while-silmukalla ajamalla SQL-komento PDO:n avulla silmukan ehtona. Silmukan sisällä jokaiselle nimelle luodaan linkki sen nimen sivulle sekä tulostetaan sen järjestys- ja tilastonumerot. (Ks. Kuvio 46)

```
while($res = $r->fetch(PDO::FETCH_ASSOC)):
    <li>
        <?php echo $res['RNK']. ". ";?>
        <a href="wp/nimihaku/?fname=<?php echo $res['name'];?>&country=fi"><?php echo $res['name'];?></a>
        <?php echo number_format($res['amount'], 0, '.', ' ');?>
    </li>
<?php endwhile; ?>
```

Kuvio 46. Nimitilastojen tulostus

Käyttäjä voi vaihtaa tilastosivuja kahdella menetelmällä, seuraava tai edellinen painikkeilla ja sivunumerolistalla. Sivunumeron valintalistalla on toteutettu form-elementillä, joka lähettää tiedot automaattista muutoksen jälkeen. Jotta tuloksien rajausarvot pysyisivät samana sivuja vaihtaessa, rajausarvot lähetetään uudestaan käyttäjältä piilotettuina. (Ks. Kuvio 47)

```

<a class="button page_change_button"
href="<?php bloginfo('url'); ?>/nimet/?pagenumber=<?php echo $pagenumber-1; ?>&which_name=<?php echo $which_name;
<?php echo $gender; ?>&order=<?php echo $gender; ?>">Edellinen sivu</a>

<!-- Select any page number. For better UX, form is submitted automatically when user changes the value. -->
<form action="" method="GET" class=" flexing justify_cent mid_button_margin">
  <input type="hidden" name="which_name" value="<?php echo $which_name; ?>">
  <input type="hidden" name="gender" value="<?php echo $gender; ?>">
  <input type="hidden" name="order" value="<?php echo $order; ?>">
  <select name="pagenumber" onchange="this.form.submit()" class="page_selector">
    <?php
    for ($page=1; $page <= $total_pagenumbers ; $page++):
      echo '<option value="'. $page. '" ' . (($page==$pagenumber)?'selected="selected"':"").'>Sivu '. $page. '</option>';
    endfor;
  >

```

Kuvio 47. Sivunvaihto painikkeiden ohjelmointi

5 Pohdinta

Opinnäytetyön aiheena oli luoda uudet nimipäivätietoja esittävät verkkosivut toimeksiantajana toimineelle Preonille. Aihe oli juuri sopivan kokoinen.

Yksinkertaiset nimipäiväsivut on mahdollista toteuttaa suhteellisen nopeasti, jos sivuilla ei tarvitsisi nähdä kuin esim. suomenkieliset nimipäivät. Aihe oli kuitenkin myös sellainen, että sen varjolla sivustoa oli helppo lähteä laajentamaan erilaisilla ominaisuuksilla ja tiedoilla. Koska aihe oli suhteellisen yksinkertainen, opinnäytetyön epäonnistumisen riski ei ollut kovin suuri, mutta sivuista tulivat kuitenkin hyvät ja paljon erilaista nimiin liittyvää tietoa sisältävät.

Aiemmat kokemukset REST-rajapintojen peruskäytöstä Reactilla sekä PHP:n käytöstä tietokantojen kanssa auttoivat opinnäytetyössä, mutta yllättäen etenkin WordPress toi toteutukseen odottamattoman paljon haasteita. WordPressissä on useita turvallisuuteen liittyviä rajoitteita, etenkin JavaScriptin suhteen. Näitä ongelmia selvitellessä oppi paljon uutta etenkin WordPressin omista funktioista, jotka alkuun vaikuttivat erittäin vaikeasti ymmärrettäville logiikaltaan.

Lopputuloksena nimipäiväsivustosta tuli näppärät pienet verkkosivut nimipäivien etsimiseen ja selailuun, josta käyttäjän on myös mahdollista löytää muita

nimiaiheista tietoa. Preoni Oy:n päivyrisivuston uudistaminen on vielä kesken, mutta Preoni pystyy hyödyntämään nimipäiväsivustoa tulevaisuudessa heidän uudella päivyrisivustolla. Prototyypin avulla Preoni Oy:n on mahdollista myös jatkokehittää nimipäiväsivustoa lisäämällä sinne myös muista maista löytyviä nimiin liittyviä tietoja opinnäytetyössä esille tuoduilla metodeilla.

Lähteet

Ampps. N.d. Ampps-sovelluksen lataussivu. Viitattu 29.12.2020.
<https://ampps.com/download>

Anam. 2018. Google Chart-ohje Stackoverflow-foorumilla. Viitattu 21.12.2020.
<https://stackoverflow.com/questions/12994282/php-mysql-google-chart-json-complete-example>

Animals. N.d. Eläinkuvakkeet Icons 8-sivustolta. Viitattu 23.12.2020.
<https://icons8.com/icon/set/animals/color>

Child themes. N.d. WordPress aliteeman käyttöönotto ohjeet. Viitattu 29.12.2020.
<https://developer.wordpress.org/themes/advanced-topics/child-themes/>

CMS Usage Distribution in the Top 1 Million Sites. N.d. Tilasto Built With sivustolla. Viitattu 14.12.2020. <https://trends.builtwith.com/cms>.

Countryflags. N.d. Lippukuvakkeet Countryflags-sivustolta. Viitattu 23.12.2020.
<https://www.countryflags.io/>

Date. N.d. PHP date-funktion ohjesivut. Viitattu 30.12.2020
<https://www.php.net/manual/en/function.date.php>

Eläinten nimipäivät. N.d. Suomalaiset eläinten nimipäivät. Viitattu 21.12.2020.
http://www.webcal.fi/fi-FI/muut_tiedostomuodot.php

Etunimitilasto. 2020. Väestötietojärjestelmän nimitilastot Suomessa. Viitattu 29.12.2020. https://www.avoindata.fi/data/fi/dataset/none/resource/08c89936-a230-42e9-a9fc-288632e234f5?view_id=0cfa5fa9-f4e5-43b5-9638-9664d3772401

Function Reference. N.d. WordPressin funktiot ohjesivu. Viitattu 14.12.2020.
https://codex.wordpress.org/Function_Reference

International NameDay API. N.d. Abalinin nimipäivä API:n kotisivut. Viitattu 29.12.2020. <https://API.abalin.net/>

Javad Shariaty. 2018. JavaScript-ajastin esimerkki Stackoverflow-foorumilla. Viitattu 29.12.2020. <https://stackoverflow.com/questions/41464114/multiple-countdown-timer>

JSON_decode function. N.d. PHP-funktion ohjesivu. Viitattu 27.12.2020.
https://www.geeksforgeeks.org/php-json_decode-function/

Juan. 2013. SQL-ranking ohje Stackexchange-foorumilla. Viitattu 20.12.2020.
<https://dba.stackexchange.com/questions/13703/get-the-rank-of-a-user-in-a-score-table>

Lisäosat. N.d. WordPress-lisäosien hakusivu. Viitattu 14.12.2020.
<https://fi.WordPress.org/plugins/>

MediaWiki. N.d. Wikipedia tietojen hakemiseen tarkoitetun API:n ohjesivut. Viitattu 29.12.2020. https://www.mediawiki.org/wiki/API:Main_page

Nimipäivien tekijänoikeus. N.d. Almanakkatoimiston suomalaistan nimipäivien tekijä-oikeudet. Viitattu 31.12.2020. <https://almanakka.helsinki.fi/fi/nimipaivat/nimipaivien-tekijanoikeus.html>

Nimipäivien vietto. N.d. Almanakkatoimiston artikkeli nimipäivien historiasta. Viitattu 31.12.2020. <https://almanakka.helsinki.fi/fi/nimipaivat/nimipaivien-vietto.html>

Nimipäivähinnasto. N.d. Almanakkatoimiston hinnastot nimipäiville. Viitattu 5.1.2021. <https://almanakka.helsinki.fi/fi/kalenterin-tekeminen/nimipaivahinnasto.html>

Pataki.D. & Crawshaw.R. 2019. How To Properly Add jQuery Scripts To WordPress. WordPress ohje sivu. Viitattu 14.12.2020. <https://premium.wpmudev.org/blog/adding-jquery-scripts-wordpress/>

Petscan. N.d. Sivusto Wikipedia tietojen tallentamista varten. Viitattu 26.12.2020.
https://petscan.wmflabs.org/?max_age=&cb_labels_yes_l=1&edits%5Bflagged%5D=both&cb_labels_any_l=1&ns%5B0%5D=1&language=fi&project=wikipedia&cb_labels_no_l=1&edits%5Bbots%5D=both&interface_language=fi&categories=etunimet&edits%5Banons%5D=both&search_max_results=500&doit=

Pille J. 2018. PHP-muuttuja JavaScriptille WordPress-ympäristössä esimerkki Stackexchange-sivustolla. Viitattu 19.12.2020. <https://WordPress.stackexchange.com/questions/96370/pass-php-variable-to-javascript>

Saqlain. 2017. PHP-paginointi ohje Stackoverflow-foorumilla. Viitattu 25.12.2020.
<https://stackoverflow.com/questions/44792931/how-to-create-pagination-with-pdo-php>

Suomalaiset nimipäivät 2020. 2018. Almanakkatoimiston tiedote. Viitattu 30.12.2020. <https://almanakka.helsinki.fi/fi/julkaisut/ajankohtaista/nimipaivatiedotteet.html>

Swaggerhub International Nameday API. N.d. Abalininin nimipäivä API Swaggerhubissa. Viitattu 22.12.2020. <https://app.swaggerhub.com/APIs/nekvAPI/InternationalNamedayAPI/3.0.0>

Swedish namedays in Finland. Ruotsinkieliset nimipäivät Suomessa sisältävä Vårsvenska artikkeli. Viitattu 28.12.2020. <http://www.vardsvenska.fi/en/about-finland/culture-in-finland/namedays/swedish-namedays-in-finland>

Tuncay Elvanağaç. 2018. Google Chart linkitysohje Stackoverflow-foorumilla. Viitattu 29.12.2020. <https://stackoverflow.com/questions/6205621/how-to-add-links-in-google-chart-API/54164499>

WordPress kansantajuisesti ja helposti. Muokattu viimeksi 28.1.2016. Artikkelin WordPressiin liittyen. Viitattu 14.12.2020. <https://sivustonikkari.fi/wordpress-kansantajuisesti-ja-helposti/>.

Wp_enqueue_script. N.d. WordPressin funktion ohjesivu. Viitattu 14.12.2020. https://developer.WordPress.org/reference/functions/wp_enqueue_script/