



Sijaintikohtaisen ponnahdus- ikkunan toteutus Shopify- verkkokauppaympäristössä

Hilda Lindfors

OPINNÄYTETYÖ
Joulukuu 2022

Tieto- ja viestintäteknikan tutkinto-ohjelma
Ohjelmistotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tieto- ja viestintätekniikan tutkinto-ohjelma
Ohjelmistotekniikka

LINDFORS, HILDA
Sijaintikohtaisen ponnahdusikkunan toteutus Shopify-
verkkokauppaympäristössä

Opinnäytetyö 41 sivua
Joulukuu 2022

Opinnäytetyön tarkoituksena oli kehittää Shopify-pohjaisiin verkkokauppoihin soveltuva asiakkaan maakoodia hyödyntävä ponnahdusikkuna, jota voitaisiin hyödyntää laajalti erilaisissa Shopify-verkkokauppaympäristöissä. Työ oli toiminnallinen ja sen tilaajana toimi Woolman Oy, joka on Euroopan suurin Shopify Plus -yhteistyökumppani. Opinnäytetyön tehtävinä oli tutkia Shopify-alustan tarjoamia mahdollisuuksia sekä tutustua Shopify-teemakehitykseen ja sen ominaispiirteisiin. Työn tavoitteena oli ennaltaehkäistä verkkokaupan asiakkaiden ostoprosessin katkeaminen tekemällä verkkokauppojen välillä siirtyminen mahdollisimman vaivattomaksi.

Opinnäytetyö aloitettiin tutkimalla Shopify-pohjaisten verkkokauppojen rakennetta ja Shopify:n joustavuutta kehitettävänä alustana. Työn tutkiva osuus rajattiin käsittelemään Shopify-verkkokauppoja yleisellä tasolla ja kehitystyössä keskityttiin Shopify:n frontend-kehitykseen. Työn ulkopuolelle rajattiin esimerkiksi kehitystyössä käytettävät versionhallintatyökalut sekä Shopify-alustan backend-kehitys. Tutkimustyön perusteella todettiin, että Shopify-pohjaiset teemat ovat monipuolisia ja joustavia kehitysympäristöjä, jotka taipuvat monenlaiseen jatkokehitystyöhön.

Opinnäytetyön pohjalta toteutettiin asiakkaan maakoodia hyödyntävä ponnahdusikkuna, joka toimii selaimesta tai laitteesta riippumatta moitteettomasti. Toteutuksessa hyödynnettiin Shopify:n tarjoamia teemakehitystyökaluja mahdollisimman kattavasti. Toteutus tehtiin useamman verkkokaupan kokonaisuudet huomioiden, ja se on mahdollista monistaa erilaisiin Shopify-verkkokauppaympäristöihin verkkokaupan teemasta riippumatta.

Työn tuloksena kehitetty ponnahdusikkuna vastaa työelämässä havaittuja tarpeita. Shopify-verkkokauppiilla on toistuvasti tarvetta vastaavanlaisille ratkaisuille, sillä monilla heistä on useamman verkkokaupan kokonaisuus, jolla mahdollistetaan yksilökohtainen myynti eri maissa. Ponnahdusikkunan avulla asiakkaan on vaivatonta siirtyä oikeaan verkkokauppaan, eikä ostoprosessi katkea.

Asiasanat: shopify, teemakehitys, frontend, web-ohjelmointi, Liquid

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in ICT Engineering
Software Engineering

LINDFORS, HILDA:
Location Based Popup implementation on the Shopify E-Commerce Platform

Bachelor's thesis 41 pages
December 2022

The purpose of the thesis was to develop a pop-up window for Shopify-based online stores that utilizes the customer's country code. This could be widely used in various Shopify online store environments. The thesis was ordered by Woolman Ltd which is Europe's largest Shopify Plus partner. The objectives of the thesis were to investigate the possibilities offered by the Shopify platform and get to examine Shopify theme development and its features. The goal of the work was to prevent interruptions in the sales process of online store customers by making switching between online stores as effortless as possible.

The thesis began by an investigation of the structures of Shopify-based online stores and the flexibility of Shopify as a development platform. The investigative part of the work was limited to Shopify's online stores at a general level, and the development work focused on Shopify's front-end development. Based on the research, it was concluded that Shopify-based themes are versatile and flexible development environments that lend themselves well to a wide range of further development work.

Based on the thesis, a pop-up window using the customer's country code was implemented, which works regardless of the browser or device. The theme development tools offered by Shopify were used in the implementation as comprehensively as possible. The implementation was made considering the entities of several online stores, and it is possible to duplicate it in various Shopify online store environments, regardless of the theme of the online store.

The pop-up window was developed as a result of the thesis, and it meets the needs found in working life. Shopify online retailers are in need of similar solutions, as many of them have a set of several online stores that enable individual sales in different countries. With the help of a pop-up window, it is easy for the customer to switch to the right online store, and the purchase process will not be interrupted.

Key words: shopify, theme development, frontend, web-development, liquid

SISÄLLYS

1	JOHDANTO	6
2	SHOPIFY	7
	2.1 Shopifyn perusta	7
	2.2 Shopify tilastoissa	7
3	SHOPIFY-TEEMOJEN RAKENNE	10
	3.1 Liquid	11
	3.2 JavaScript Shopifyssa	12
	3.3 JSON-tiedostot Shopifyssa	12
	3.3.1 Asetustiedostot	12
	3.3.2 Kielitiedostot	13
4	TEEMAKEHITYS	14
	4.1 Kehitysympäristö	14
	4.1.1 Koodieditori	14
	4.1.2 Teemaeditori	16
	4.1.3 Kielieditori	18
	4.2 Theme Kit	18
	4.2.1 Theme Access	19
	4.2.2 Teeman lataus Theme Kitilla	19
5	TOTEUTUS	21
	5.1 Suunnitelma	21
	5.2 Esivalmistelut	21
	5.3 Rakenne	21
	5.4 Asetusten toteuttaminen teemaeditoriin	22
	5.5 Liquid-toteutus	24
	5.5.1 Ponnahdusikkunan otsikon toteutus kielieditoriin	27
	5.6 Tyyllittelyt	28
	5.7 Toiminnallisuudet	31
	5.7.1 Sulkevan painikkeen poisto maakoodin perusteella	34
	5.8 Lopputulos	35
6	TESTAUS	37
	6.1 Testiteema ja Theme Kit	37
	6.2 Testaus VPN-yhteydellä	37
	6.3 Testausvaiheessa havaitut ongelmat	38
7	POHDINTA	39
	LÄHTEET	40

LYHENTEET JA TERMIT

AJAX	Asynchronous JavaScript and XML
API	Application programming interface
CSS	Cascading Style Sheets
JSON	JavaScript Object Notation
POS	Point of sale
SaaS	Software as a Service
VPN	Virtual Private Network

1 JOHDANTO

Tässä opinnäytetyössä perehdytään Shopify-pohjaisten verkkokaupparatkaisujen rakenteeseen Shopify-teemakehityksen näkökulmasta. Työn tilaajana toimii Euroopan suurin Shopify Plus -yhteistyökumppani Woolman Oy. Opinnäytetyön aiheen lähtökohtana on tilaajan tarve helpottaa verkkokaupan asiakkaiden ostoprosessin jatkumista uudelleenohjaavalla ponnahdusikkunalla. Työn tavoitteena on ennaltaehkäistä asiakkaiden ostoprosessin katkeaminen tekemällä verkkokauppojen välillä siirtyminen mahdollisimman vaivattomaksi.

Opinnäytetyön tarkoituksena on kehittää Shopify-pohjaisiin verkkokauppoihin soveltuva, asiakkaan maakoodia hyödyntävä ponnahdusikkuna, joka on toteutukseltaan selkeä ja jota voidaan soveltaa laajalti erilaisissa Shopify-verkkokauppa-kokonaisuuksissa. Toteutus tehdään alusta alkaen Shopify-ympäristössä ja kehitystyössä hyödynnetään Shopify-teemakehityksessä tarvittavia työkaluja.

2 SHOPIFY

Shopify on käyttäjäystävällinen verkkokauppa-alusta, jonka avulla yritykset voivat rakentaa verkkokauppoja ja myydä verkossa yhden selkeän hallintapaneelin kautta. Shopify:n avulla kauppiat voivat rakentaa modernin verkkokaupan ja myydä tuotteitaan sosiaalisen median alustoilla, erilaisilla kauppapaikoilla, blogeilla ja verkkosivustoilla sekä sähköpostien, tekstiviestien ja chatin välityksellä. Shopifylla on mahdollista myydä tuotteita myös vähittäismyyntipisteissä sen sisäänrakennetun sovelluksen, Shopify POS:n kautta. Shopify POS on luotu muun muassa vähittäiskauppoja, pop-up-myymlöitä ja torimyyntiä varten. (Fabregas & Main 2022.)

2.1 Shopify:n perusta

Shopify on SaaS-mallinen pilvipalvelu, jonka ylläpito ja hallinta tapahtuu selaimen kautta internetin välityksellä. Tämän ansiosta kauppiat voivat käyttää sitä miltä tahansa yhteensopivalta laitteelta ilman asennettavia ohjelmia. Shopify myös hoitaa ohjelmisto- ja palvelintason päivitykset sekä muun ylläpidon, eikä kauppiaan täten itse tarvitse huolehtia niistä. (Shopify Staff n.d.)

Shopify on tilausmallinen palvelu ja se tarjoaa neljä perustason maksusuunnitelmaa. Maksusuunnitelmien kuukausitason kustannukset ovat 29:stä dollarista ylöspäin. Shopify tarjoaa näiden lisäksi myös Lite-tilaustason ratkaisun, joka ei sisällä verkkokauppaa, mutta tukee POS-myyntiä. Tämä tilaustaso mahdollistaa myös Shopify:n itse kehittämän ostopainikkeen (Buy Button) käytön muilla verkkosivuilla, blogisivustoilla ja sähköpostin välityksellä. (Fabregas & Main 2022.)

2.2 Shopify tilastoissa

Shopify on perustettu vuonna 2006 ja se on siitä asti kasvanut huimaa vauhtia maailmanlaajuisesti. Alun perin Shopify:n oli tarkoitus palvella vain sen perustajia,

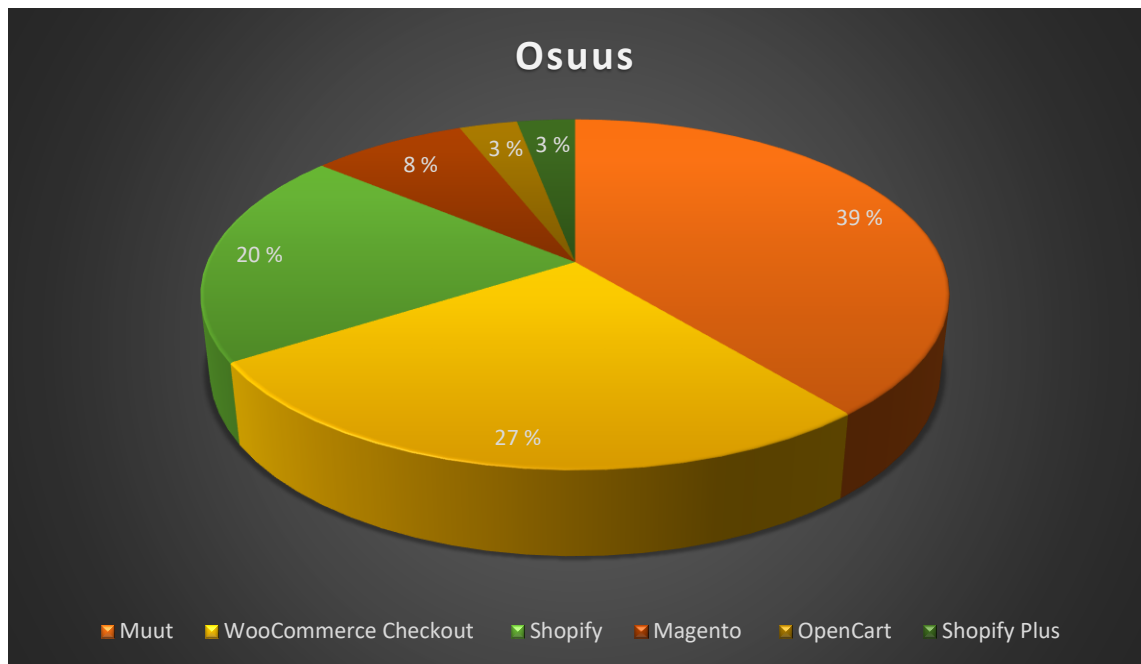
Tobias Lütkea bisneskumppaneineen. Vuonna 2004 he olivat perustamassa verkkokauppaa lumilautailuyritykselleen, mutta huomasivat pian, että silloiset verkkokauppatyökalut aiheuttivat lähinnä hämmennystä ja niiden kanssa oli turhauttavaa työskennellä. Tämän vuoksi Lütke päätti hyödyntää taitojaan ohjelmistokehittäjänä ja alkaa rakentaa omaa verkkokauppa-alustaansa Shopifyä. (Sorensen 2022.)

Vaikka Shopify on suunnattu alun perin pienyrityksille, käyttävät alustaa nykyään myös suuret, kansainväliset yritykset (TAULUKKO 1). Maailmanlaajuisesti tunnettuja Shopify-pohjaisia verkkokauppiaita ovat esimerkiksi Kylie Cosmetics, Gymshark ja FashionNova. Shopifyä käytetään nykyään jo 175:ssä eri maassa ja aktiivisia Shopify-verkkokauppoja on tänä päivänä yli miljoona. Shopify myös työllistää yli 7000 ihmistä, jonka ohella Shopifyyn yhteistyökumppanina työskentelee ainakin 700 000 työntekijää. (Özşahan n.d.)

TAULUKKO 1. Suurimpia Shopifyä käyttäviä brändejä (Site Builder Report 2022)

Yritys	Markkina-arvo (mrd.)	Liikevaihto (mrd.)	Valuutta
Heinz To Home	49.7	-	USD
Raycon	25.1	-	USD
Staples (CAN)	-	14	USD
JB Hi-Fi	8+		AUD
Crate and Barrel (Singapore)	1.6	-	USD

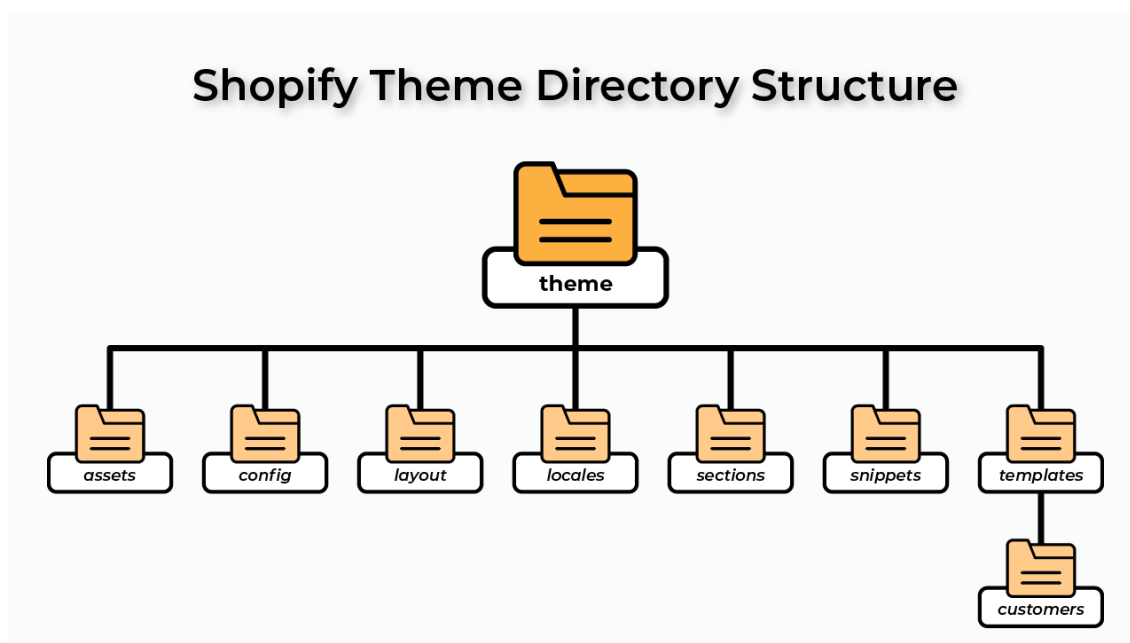
Kesäkuusta 2022 lähtien erilaisten verkkokauppojen tai verkkokauppasivustojen osuus maailman miljoonan käytetyimmän verkkosivun joukossa on 25.5 % (KUVIO 1). Näistä verkkokauppasivustoista peräti 20 % oli Shopifyyn tarjoamia. (Kinsta 2022.)



KUVIO 1. Verkkokauppojen käyttöaste miljoonan käytetyimmän verkkosivuston joukossa (mukaan Kinsta 2022)

3 SHOPIFY-TEEMOJEN RAKENNE

Shopify:n teemalla tarkoitetaan tietynlaista tiedostokokonaisuutta, joka määrittää verkkokaupan ulkoasun (Tovey 2021). Shopify:n teemoissa käytetään Liquid, HTML, CSS, JSON ja JavaScript -tyyppisiä tiedostomuotoja. Nämä tiedostot järjestävät ja tyyllittelevät verkkokaupan sekä määrittelevät verkkokaupan toiminnallisuudet. Teemakoodi on organisoitu Shopify:n nimeämän hakemistorakenteen mukaan, joka sisältää teeman määrittelevät tiedostot sekä siihen liittyvät lisäresurssit (KUVA 1). Shopify:ssa on saatavilla useita ilmaisia ja maksullisia teemoja, mutta teeman voi halutessaan kehittää myös itse. (Endertech Team 2021.)



KUVA 1. Shopify-teemojen hakemistorakenne (Endertech Team 2021)

Teeman tiedostot on kategorisoitu kolmeen pääkategoriaan. Pääkategoriat ovat Markup and features, Supporting assets ja Config files (TAULUKKO 2). Jokainen Shopify-verkkokauppa rakentuu edellä mainittuja hakemistoja ja pääkategorioita noudattamalla. Mikäli kehittäjä rakentaa itse oman teemansa Shopify:n teemakauppaan, hänen tulee noudattaa Shopify:n antamia vaatimuksia teeman toteutukseen liittyen. (Theme architecture n.d.)

TAULUKKO 2. Shopify-teemojen tiedostorakenteen pääkategoriat (mukaillen Theme architecture n.d.)

Kategoria	Tarkoitus
Markup and features	Nämä tiedostot hallitsevat teeman ulkoasua ja toiminnallisuuksia. Tiedostot käyttävät Liquid-syntaksia generoidakseen HTML-koodia, jonka pohjalta verkkokaupan sivut toteutetaan.
Supporting assets	Nämä tiedostot ovat asetteja, skriptejä tai lokaaleja tiedostoja, joita joko kutsutaan tai hyödynnetään toisissa teeman tiedostoissa.
Config files	Nämä tiedostot varastoivat dataa ja ovat JSON-muotoisia. Varastoitua dataa voidaan hyödyntää ja kustomoida Shopify:n teemaeditorissa.

3.1 Liquid

Liquid on Shopify:n luoma avoimen lähdekoodin templaattikieli (template language), joka on kirjoitettu Ruby-ohjelmointikielillä. Se on Shopify:n teemojen perusta ja sitä käytetään dynaamisen sisällön lataamiseen verkkokaupassa. Liquid on ollut Shopifylla käytössä vuodesta 2006 lähtien ja sitä käyttävät nykyään myös monet muut verkkosovellukset. (Liquid n.d.)

Liquid-sivupohja eli templaatti on syntaksiltaan yksinkertainen ja helposti opetettavissa. Liquid käyttää objekteja, tageja sekä filttäreitä renderöimään eli visualisoimaan templaatilla näytettävää dynaamista sisältöä. Esimerkiksi tuotesivutemplaatilla on käytettävissä tuoteobjekti, josta voidaan dynaamisesti renderöidä muun muassa nimi, hinta ja tuotokuva.

Liquid tarjoaa kahdenlaisia, erityisiä merkintätapoja, jotka ovat tuloste- ja tunnistelohkot (KUVA 2). Tulostelohkot ympäröidään kaksilla aaltosulkeilla ja ne palauttavat muuttujia tai vakioita HTML-merkinnöissä. Tunnistelohkot taas ympäröidään aaltosulkeilla ja prosenttimerkeillä ja ne suorittavat sivupohjan logiikan. (TwentyThree n.d.)

```
{% assign variable = "Joe Pichardo" %}  
<h1>{{ variable }}</h1>
```

KUVA 2. Liquidin tunniste- ja tulostelohkot (Pichardo 2020)

3.2 JavaScript Shopifyssa

JavaScript on dynaaminen ohjelmointikieli, jonka avulla voidaan toteuttaa monimutkaisia toimintoja verkkosivuilla (MDN Web Docs 2022). JavaScriptia voidaan käyttää Shopify-verkkokaupoissa muun muassa räätälöityjen toiminnallisuuksien toteuttamiseen sekä kaupan ulkoasun muuttamiseen. JavaScriptia ja sillä luotuja toiminnallisuuksia voidaan käyttää Shopifyssa esimerkiksi pudotusvalikoissa, lomakkeiden validoinnissa ja erilaisissa ponnahdusikkunoissa. JavaScriptin käytön edellytyksenä Shopify-verkkokaupoissa on se, että kehittäjä tuntee HTML-merkintäkielen ja CSS-tyylikielen. (McFarland 2022.)

3.3 JSON-tiedostot Shopifyssa

Shopify-verkkokauppojen teemat sisältävät Config- ja Locales-nimiset hakemistot. Nämä hakemistot koostuvat JSON-päätteisistä konfiguraatitiedostoista, joita kutsutaan teeman asetus- ja kielitiedostoiksi. Asetus- ja kielitiedostojen avulla kauppiaat voivat muokata verkkokauppansa käännöksiä ja ulkoasua ilman kooditason muutoksia Shopifyyn omassa teema- ja kielieditorissa. (Theme architecture n.d.)

3.3.1 Asetustiedostot

Shopify-verkkokaupat sisältävät kahdenlaisia asetustiedostoja (TAULUKKO 3), joita kutsutaan teeman asetuksiksi. Nämä asetustiedostot ovat `settings_schema.json` ja `settings_data.json` ja ne sijaitsevat teemakoodin

Config-hakemistossa. Teeman asetukset on luotu helpottamaan kaupan ulkoasun kustomointia. (Tse 2018.)

TAULUKKO 3. Teemojen asetustiedostot (mukaillen Theme architecture n.d.)

Tyyppi	Kuvaus
<code>settings_schema.json</code>	Hallitsee teeman asetusten järjestystä ja sisältöä teemaeditorissa.
<code>settings_data.json</code>	Varastoi tallennetut arvot <code>settings_schema.json</code> -tiedoston asetuksista.

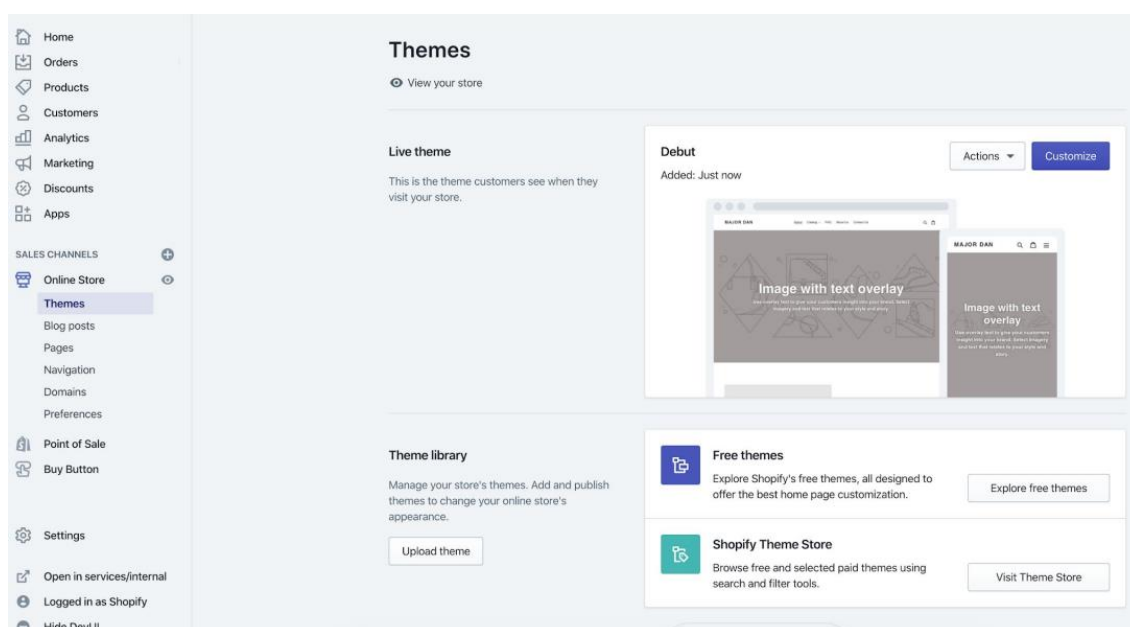
3.3.2 Kielitiedostot

Asetustiedostojen lisäksi Shopify-verkkokaupat sisältävät kielitiedostoja, joilla voidaan hallita verkkokaupan käännöksiä. Näitä käännöksiä käytetään teksti-muuttujissa eri puolilla teemaa ja teemaeditoria. Jokainen kielitiedosto sisältää tietyn kielen käännökset ja tiedosto nimetään käännettävän kielen mukaisesti. Jokaisessa teemassa on myös oletuskielitiedosto, joka on useimmiten `en.default.json`. (Theme architecture n.d.)

4 TEEMAKEHITYS

4.1 Kehitysympäristö

Shopify-verkkokauppoja hallinnoidaan Shopify:n admin-paneelin (KUVA 4) avulla. Kuka tahansa voi perustaa Shopify-pohjaisen verkkokaupan, sillä Shopify on mahdollista testata ilmaiseksi muutaman päivän ajan. Kun kauppa on perustettu, valitaan kauppaan soveltuva teema admin-paneelin Online Store -valikon alta. Shopify tarjoaa kahdeksan ilmaista teemaa, mutta teeman voi ostaa myös Shopify:n teemakaupasta. (Offline to Online with Shopify n.d.)

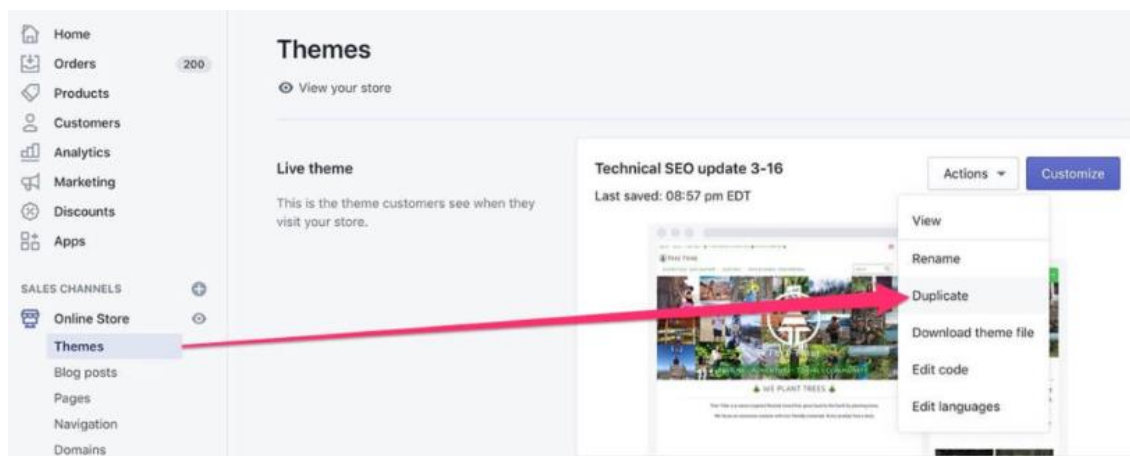


KUVA 4. Shopify:n admin-paneeli (Offline to Online with Shopify n.d.)

4.1.1 Koodieditori

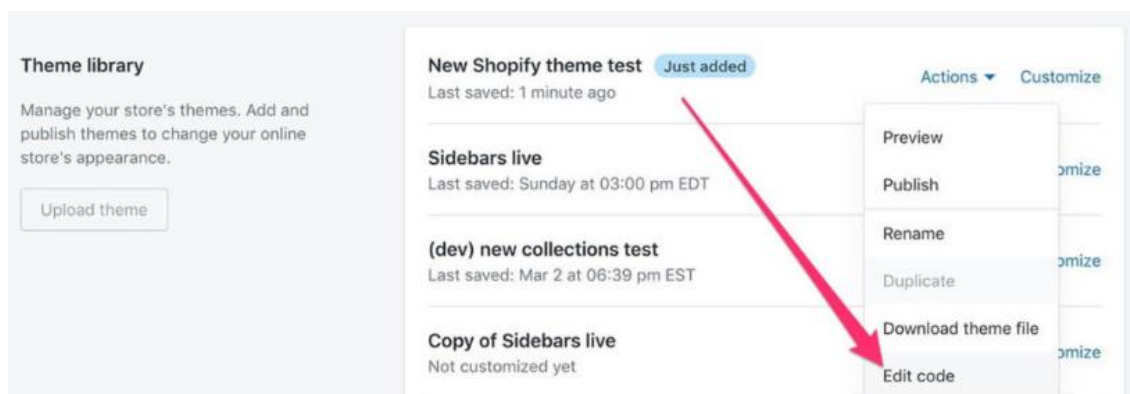
Mikäli Shopify-verkkokaupan teemaa tai ulkoasua halutaan editoida yksityiskohteisesti, voidaan se tehdä Shopify:n oman koodieditorin kautta (How to Edit Shopify Theme Code 2022). Turvallisuuden kannalta on äärimmäisen tärkeää ottaa tuotantoteemasta varmuuskopio ennen muutosten tekoa (KUVA 5). Kopion ottaminen varmistaa myös sen, ettei tuotannossa oleva teema voi kehitystyön myötä rikkoutua. Teemakopiot ovat tärkeitä myös pieniä muutoksia tehdessä,

sillä niiden avulla voidaan jäljittää teeman muutoksia ja havaita mahdollisia bugeja. (Lannen 2020.)



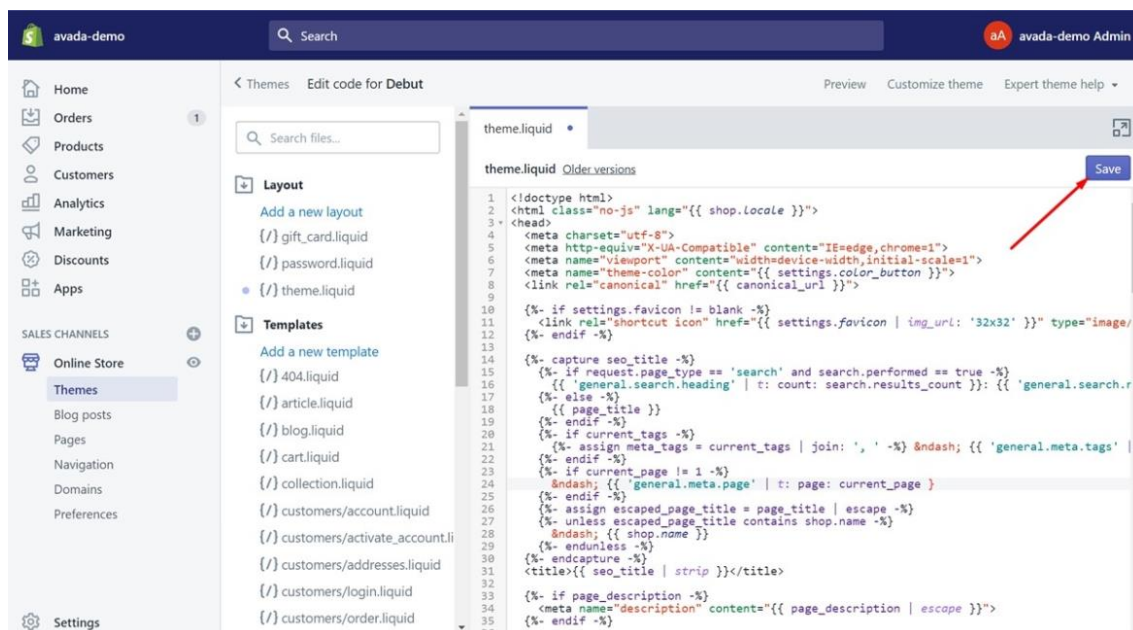
KUVA 5. Teemakopion ottaminen Shopify Adminissa (Lannen 2020)

Kun tuotannossa oleva teema on kopioitu, voidaan kehitystyötä jatkaa uudessa teemakopiossa eli testiteemassa. Testiteema löytyy tuotantoteeman alapuolelta Theme library -valikosta. Viimeiseksi kopioitu teema sijoittuu teemaluettelossa ylimmäiseksi ja sitä päästään muokkaamaan Actions-painikkeesta avautuvan pudotusvalikon Edit code -painikkeesta (KUVA 6). (Lannen 2020.)



KUVA 6. Koodieditorin käyttö Shopify Adminissa (Lannen 2020)

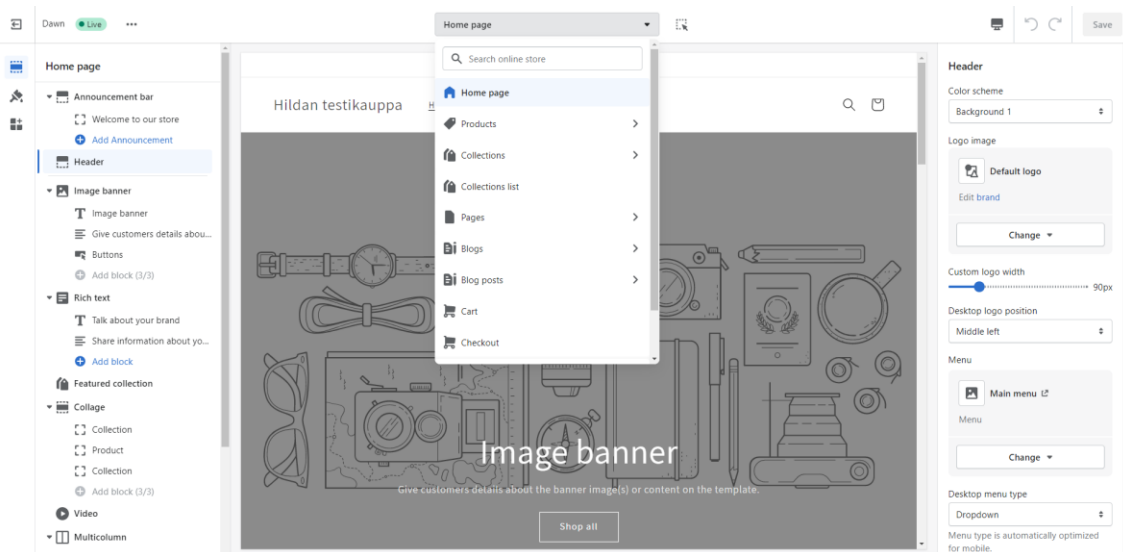
Shopify-teemat sisältävät useita HTML- ja CSS-tiedostoja. Nämä tiedostot on ryhmitelty kansioihin ja ne löytyvät koodieditorin (KUVA 7) vasemmassa laidassa sijaitsevasta sivupalkista. Sivupalkista voidaan myös lisätä jokaisen kansion alle uusia tiedostoja. Kun halutut tiedostot on lisätty tai valittuja tiedostoja on muokattu, voidaan muutokset tallentaa koodieditorin Save-painikkeesta. (How to edit HTML Code on Shopify 2022.)



KUVA 7. Shopifyn koodieditori (How to edit HTML Code on Shopify 2022)

4.1.2 Teemaeditori

Shopify-teemojen sisältöä ja ulkoasua voidaan muokata monella eri tavalla. Yksi vaihtoehtoista on käyttää Shopifyn visuaalista kustomointityökalua eli teemaeditoria (KUVA 8). Teemaeditorin avulla voidaan työstää jo olemassa olevia ominaisuuksia, joita verkkokaupan teema sisältää. Kustomointimahdollisuudet ovat kuitenkin rajallisia, sillä yksilöllisemmät kustomoinnit toteutetaan teemakoodia kehittämällä. (Le 2021.)



KUVA 8. Shopifyn teemaeditori

Teemaeditoriin voidaan luoda uusia asetuksia kehittämällä `settings_schema.json` -tiedostoa. Tiedoston data on ympäröity hakasulkein, joiden avulla ryhmitellään tietyn asetuksen tai asetuskokouksen vaihtoehdot yhteen. Jokaisella ryhmällä on nimi ja joukko asetuksia, jotka määritellään taulukko-muodossa. Jokainen asetusvaihtoehto on objekti asetustaulukon sisällä (KUVA 9). (Code Shopify 2018.)

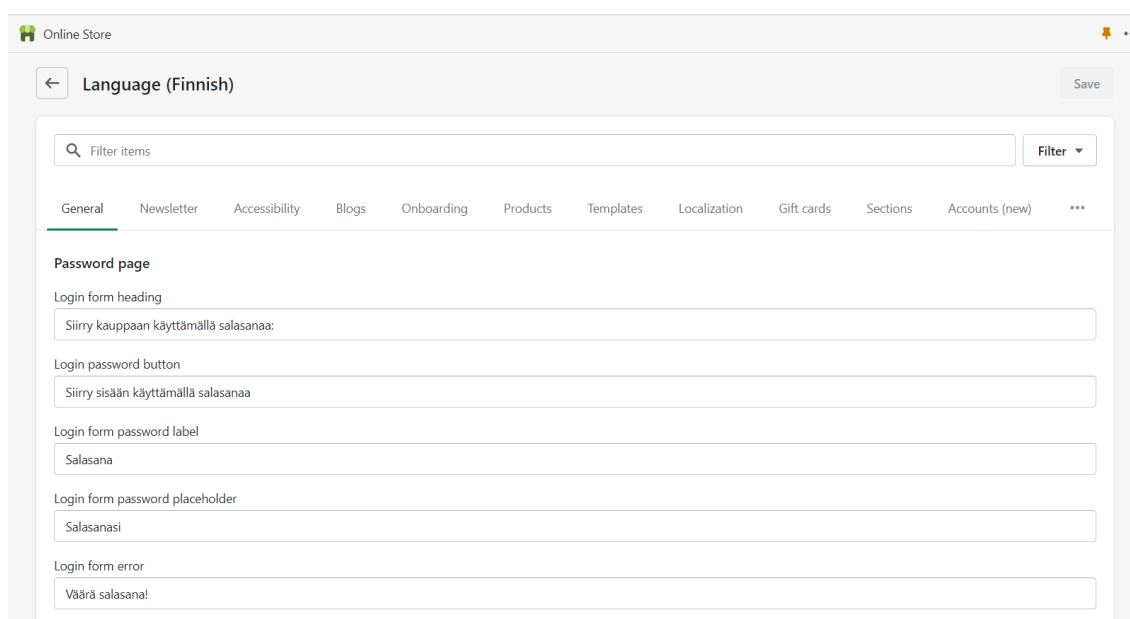
```
[
  {
    "name" : "Colors",
    "settings" : [
      {
        "type": "color",
        "id": "color_borders",
        "label": "Border Colors",
        "default": "#e5e5e5",
        "info" : "this will be the colors for borders"
      },
      {
        "type": "color",
        "id": "color_body_text",
        "label": "Body Text",
        "default": "#2980b9"
      }
    ]
  }
]
```

KUVA 9. `settings_schema.json` -tiedoston rakenne (Code Shopify 2018)

Koodiesimerkissä (KUVA 9) luodaan Shopify:n teemaeditoriin asetus, jonka nimi on Colors. Kun asetusta klikataan teemaeditorissa, avautuu joukko Colors-otsikon alle määritellyjä vaihtoehtoja. Nämä vaihtoehdot ovat Border Colors ja Body Text, joille on määritetty asetustiedostossa oletusvärit. Näiden vaihtoehtojen väriä voidaan muuttaa helposti teemaeditorissa color picker -väriyökalulla. (Code Shopify 2018.)

4.1.3 Kielieditori

Shopifyn teemat sisältävät useimmiten käännökset yhdelle tai useammalle kielelle. Shopify tarjoaa kuitenkin kielieditorin (KUVA 10), missä voidaan muokata teemakohtaisesti jo olemassa olevia tai kielitiedostoihin luotuja käännöksiä. Kielieditoriin päästään teeman Actions-pudotusvalikon Edit languages -painikkeesta. Käännökset on järjestelty kielieditorissa teeman osa-alueiden mukaan, joita ovat esimerkiksi blogit, kokoelmat ja ostoskori. (Pixel Union n.d.)



KUVA 10. Shopify'n kielieditori

4.2 Theme Kit

Theme Kit on Shopify'n kehittämä, erilaisten alustojen välillä toimiva komentorivin työkalu, jota voidaan käyttää teemakehityksessä. Theme Kit toimii Linux-, macOS- ja Windows-käyttöjärjestelmillä ja sen avulla kehittäjät saavat synkronoitua Shopify-alustan ja oman koodieditoriympäristönsä. Theme Kit saadaan asennettua tietokoneelle noudattamalla käyttöjärjestelmäkohtaista ohjeistusta. (Theme Kit n.d.)

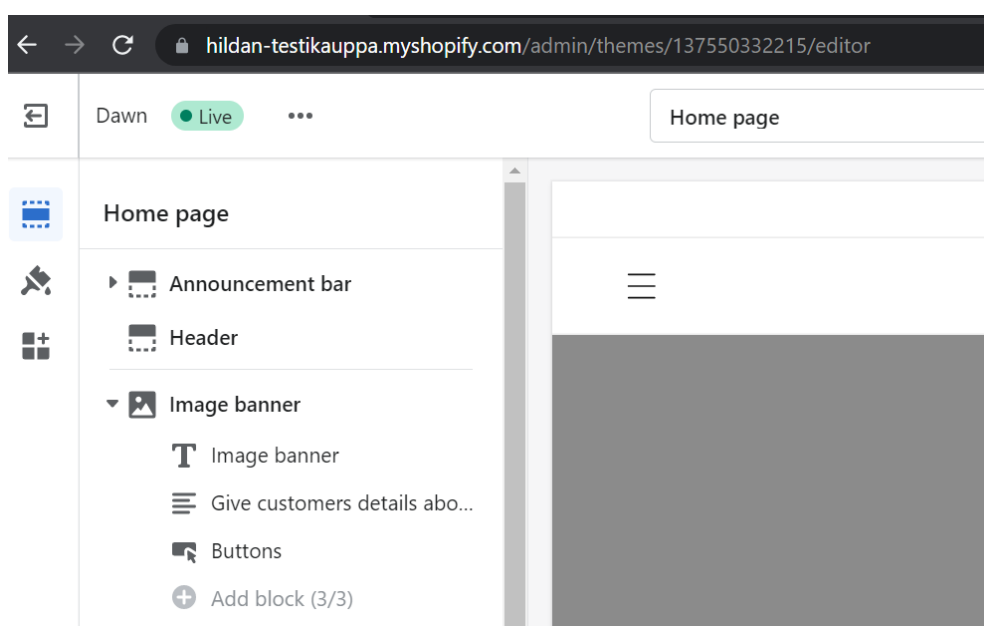
4.2.1 Theme Access

Theme Access on Shopify:n luoma sovellus, jota käytetään Theme Kit -työkalun salasanojen hallintaan. Theme Kitin käyttöön tarvitaan salasana, jotta saadaan yhdistettyä koodieditorisovellus ja verkkokauppa. Yksityinen salasana luodaan Theme Access -sovelluksella, joka lähettää salasanalinkin annettuun sähköpostiin. (Kundale 2022.)

4.2.2 Teeman lataus Theme Kitilla

Teeman asentaminen tietokoneelle vaatii Theme Access -sovelluksesta saadun henkilökohtaisen salasanan sekä verkkokaupan ja teeman tunnisteiden. Verkkokaupan tunniste saadaan kaupan URL-osoitteesta ja se on muodoltaan `mystore.myshopify.com`. (TecForFun 2021.)

Verkkokaupan teeman tunniste on numerosarja, joka on saavutettavissa muutamalla eri tavalla. Yleisin käytetty tapa on avata valitun teeman teemaeditori Shopify:n Online store -välilehdeltä ja kopioida teeman tunniste URL-osoitteesta (KUVA 11). Tunnisteen saa myös esimerkiksi lähdekoodista avaamalla teeman esikatselussa kehittäjän työkalut ja kirjoittamalla hakukenttään "id". Haun voi myös toteuttaa `ctrl+f`-komennolla. (Sacac 2021.)



KUVA 11. Teeman tunniste teemaeditorin URL-osoitteessa

Kun henkilökohtainen salasana sekä verkkokaupan ja teeman tunniste on tiedossa, voidaan teema ladata tietokoneelle. Ensin luodaan kansio sopivaan tiedostosijaintiin ja avataan se koodieditorissa. Koodieditorissa avataan terminaali, minne syötetään teeman noutamiseksi komento (KUVA 12). (TecForFun 2021.)

```
theme get -password=[your-password] -store="[your-store.myshopify.com]" -themeid=[your-theme-id]
```

KUVA 12. Teeman lataus Theme Kitin avulla

Komennon syöttämisen jälkeen kaikki teeman tiedostot latautuvat valittuun kansioon. Komento luo myös ylimääräisen `config.yml` -tiedoston, joka sisältää konfiguraatiodatan. Koska teeman `settings_data.json`-asetustiedosto sisältää teeman ulkoasuun liittyviä asetuksia, tulee sitä varten lisätä `config`-tiedostoon `ignore_files`-parametri (KUVA 13), jolla poissuljetaan tiedostoja synkronoimasta. Parametriin asetetaan edellä mainitun asetustiedoston polku, jolla varmistetaan, etteivät teeman tyylit ylikirjoitu vietäessä muutoksia teemaan. (TecForFun 2021.)

```
! config.yml
1  development:
2    password: shptka_2aad1582d907450c57dc31051443d6fc
3    theme_id: 138413932855
4    store: hildan-testikauppa.myshopify.com
5    ignore_files:
6      - config/settings_data.json
7
```

KUVA 13. `settings_data.json`-tiedoston poissulkeminen tiedostojen synkronoinnissa

5 TOTEUTUS

5.1 Suunnitelma

Opinnäytetyön pohjalta toteutetaan käyttäjän maakoodin mukaan käyttäytyvä ponnahdusikkuna Shopify-pohjaiseen verkkokauppaan. Ponnahdusikkuna avautuu sellaisille käyttäjälle, jotka eivät selaile kauppaa sen kohdemaasta ja käyttäytyy sen mukaan, mistä maasta sivustoa selaillaan. Ponnahdusikkuna sisältää tekstiä, tyyllittelyä sekä painikkeet, joiden avulla päästään siirtymään oikeaan verkkokauppaan ja voidaan sulkea ponnahdusikkuna. Toteutustyössä ja testauksessa hyödynnetään Shopify:n kehittämää Theme Kit -työkalua.

5.2 Esivalmistelut

Toteutusta varten luotiin Shopify-verkkokauppa, johon valittiin teemaksi Shopify:n kehittämä ilmainen Dawn-teema. Teeman asentamisen jälkeen verkkokauppaan asennettiin Theme Access -sovellus, jotta teema saatiin ladattua omalle tietokoneelle Visual Studio Code -koodieditoriin. Lopuksi tuotantoteemasta otettiin kopia kehitystyötä varten mahdollisten virheiden ennaltaehkäisemiseksi.

5.3 Rakenne

Jotta opinnäytetyöhön kerättyjä tietoja päästiin hyödyntämään mahdollisimman laajasti, päätettiin toteutuksessa hyödyntää Shopify:n tarjoamia mahdollisuuksia, kuten teemaeditoria ja kielieditoria. Näiden avulla ponnahdusikkunan kustomointi ja asetusten muuttaminen on mahdollista toteutuksen jälkeen myös ilman koodiosaamista.

Kehitystyö aloitettiin luomalla teemakoodin sections-kansioon geo-popup.liquid-niminen tiedosto. Tähän tiedostoon sisällytettiin ponnahdusikkunan ehdot ja Liquidilla kirjoitettu logiikka. Ponnahdusikkunan toiminnallisuuksia varten luo-

tiin snippets-kansion alle `geo-popup-scripts.liquid`-tiedosto, joka sisältää toteutukseen tarvittavan JavaScript-koodin. Ponnahdusikkunaan liittyvät, teemaeditorissa näytettävät asetukset luotiin `settings_schema.json`-asetustiedostoon. Ponnahdusikkunan ulkoasuun liittyviä CSS-muotoiluja varten luotiin erillinen `geo-popup.css`-tiedosto teeman `assets`-kansion alle. Tämän lisäksi luotiin kaupan kielieditorissa editoitavissa oleva käännös valmiiseen `fi.json`-käännöstiedostoon.

5.4 Asetusten toteuttaminen teemaeditoriin

Ponnahdusikkunan toteutuksessa hyödynnettiin teeman asetustiedostoa, josta teematiedosto hakee käytettävät parametrit. Tätä varten luotiin erillinen `GEO Popup`-niminen asetusvalikko, jonka alta ponnahdusikkunaan liittyvät asetukset löytyvät. Aetusvalikkoon toteutettiin ensimmäisenä valintaruutu (checkbox) ja tekstisyötekenttä (KUVA 14). Valintaruutu (`"id": geo_popup`) määrittää, käytetäänkö ponnahdusikkunaa kaupassa vai ei. Tekstisyötekenttään (`"id": store_country_code`) syötetään kaupan maakoodi, jota käytetään vertaamaan kaupan käyttäjän maakoodeja. Kaupan maakoodia tarvitaan ponnahdusikkunan logiikassa, joten se voidaan kätevästi hakea koodiin teemaeditorin kautta.

```
{
  "name": "GEO Popup",
  "settings": [
    {
      "type": "header",
      "content": "Popup settings"
    },
    {
      "type": "checkbox",
      "id": "geo_popup",
      "default": false,
      "label": "Enable",
      "info": "Uncheck to disable"
    },
    {
      "type": "text",
      "id": "store_country_code",
      "label": "Store country code"
    }
  ]
}
```

KUVA 14. Ponnahdusikkunan valintaruutu ja tekstikenttä maakoodia varten `settings_schema.json`-tiedostossa

Ponnahdusikkunan toteutusta suunnitellessa mietittiin erilaisia ratkaisuvaihtoehtoja, jotka hyödyntäisivät mahdollisimman laajasti Shopify-kokonaisuutta. Jotta teeman asetustiedostot ja tätä kautta myös teemaeditori saatiin käyttöön mahdollisimman kattavasti, päätettiin ponnahdusikkunan logiikassa hyödyntää teemaeditoriin määriteltäviä modaaleja.

Ponnahdusikkunan asetusvalikkoon luotiin GEO Popup Definitions -niminen asetus, joka sijoitettiin aikaisemmin luotujen asetusten alle (KUVA 15). GEO Popup Definitions on tekstikenttä, jossa eri modaalit on eroteltu putkiviivoin. Modaalissa määritellään selaajan maakoodi ja linkki, johon selaaja ohjataan modaalin ensimmäisen osan maakoodin perusteella. Tämän lisäksi modaalit sisältävät ponnahdusikkunan tekstin ja ponnahdusikkunan painikkeen tekstin. Modaaleja voidaan hyödyntää laajalti ponnahdusikkunan toteutuksessa, sillä niiden avulla määritellään maakohtaisesti ponnahdusikkunan käyttäytyminen.

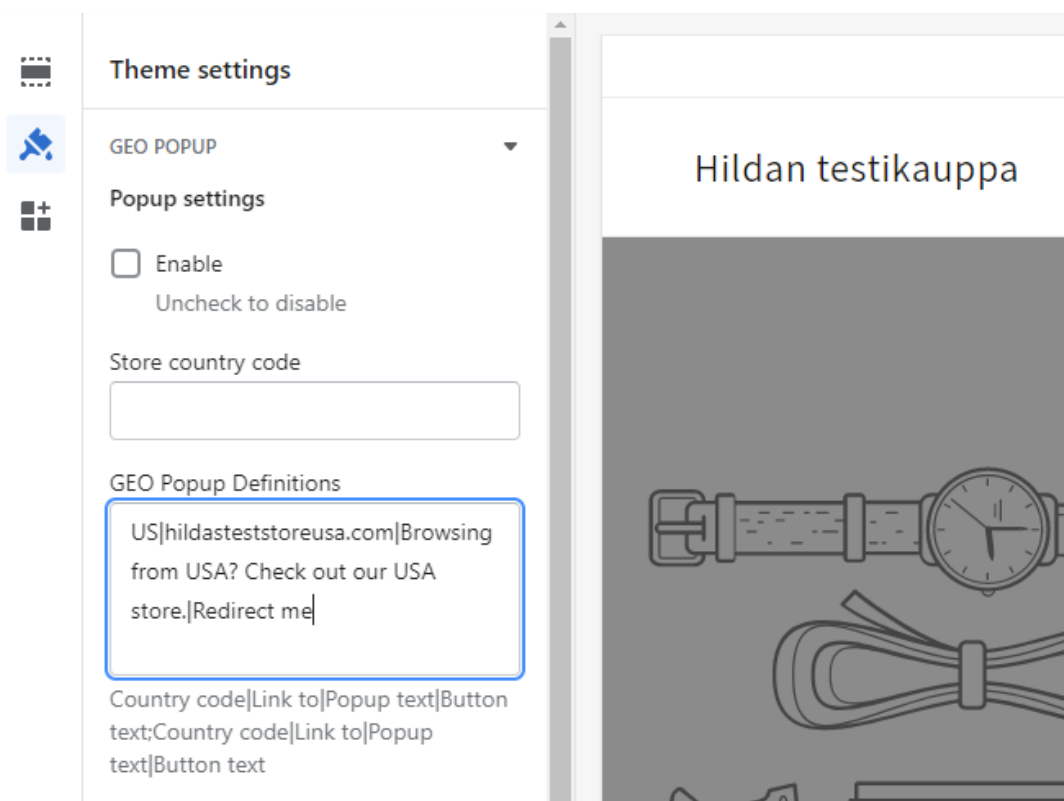
```

26     "label": "Store country code",
27     "label": "Store country code"
28   },
29   {
30     "type": "textarea",
31     "id": "geo_popup_definition",
32     "label": "GEO Popup Definitions",
33     "info": "Country code|Link to|Popup text|Button text;Country code|Link to|Popup text|Button text",
34     "default": "US|hildasteststoreusa.com|Browsing from USA? Check out our USA store.|Redirect me"
35   }
36 ]
37 },

```

KUVA 15. GEO Popup Definitions -asetus teeman asetustiedostossa

Teemaan luodut asetukset löytyvät muokatun teeman teemaeditorista, teeman asetukset-välilehdeltä (KUVA 16). Shopify:n teemoissa on jo ennalta joukko asetuksia, joiden sisältö riippuu teemasta tai siihen tehdyistä kustomoinneista. Asetukset ovat siis teemakohtaisia, eivätkä yhteen teemaan tehdyt asetukset siirry toisen teemaversioon teemaeditoriin automaattisesti.



KUVA 16. GEO Popup -asetusvalikko teemaeditorin asetuksissa

5.5 Liquid-toteutus

Ponnahdusikkunan käyttöliittymä toteutetaan `geo-popup.liquid`-tiedostossa. Ponnahdusikkunan asetukseen toteutettiin alussa valintaruutu, jonka avulla teeman asetuksista voidaan valita, näytetäänkö ponnahdusikkunaa kaupassa. Asetuksen valinta voidaan tarkistaa toteutuksen Liquid-logiikan puolella yksinkertaisella `if`-ehtolauseella. Ehtolauseessa (KUVA 17) tarkastetaan, onko ponnahdusikkuna käytössä.

```

2
3   {% if settings.geo_popup %}
4
5   {% endif %}
6

```

KUVA 17. Ehtolause ponnahdusikkunan näyttämiseksi

Mikäli teeman asetuksissa valintaruutu on valittuna, siirrytään koodissa ehtolauseeseen sisälle. Ensimmäiseksi luodaan ehtolauseeseen sisälle muuttuja, johon säilötään ponnahtusikkunan modaalien sisältö. Muuttuja alustetaan tallentamalla asetuksista löytyvä arvo, jotta modaaleja voidaan selkeämmin käsitellä muualla koodissa. Liquid-ohjelmointikielessä muuttujat luodaan ja nimetään assign-tagilla (KUVA 18).

```
7 | {% assign geopopup_modals = settings.geo_popup_definition %}
```

KUVA 18. geopopup_modals-muuttujan luonti

Kuten ponnahtusikkunan asetuksissa määriteltiin, modaalikokonaisuudet erotellaan toisistaan puolipisteellä (KUVA 19). Modaalikokonaisuuksia voi olla siis samassa kaupassa useampi, joten puolipisteellä erottelu tulee ottaa huomioon muuttujan luonnin yhteydessä. Liquidissa on split-niminen filteri, joka jakaa merkkijonon taulukoksi käyttämällä argumenttia erottimena (Liquid n.d).

```
7 | {% assign geopopup_modals = settings.geo_popup_definition | split: ';' %}
```

KUVA 19. Modaalikokonaisuuksien erotus puolipisteellä

Kun modaalien sisältö on säilötty muuttujaan ja modaalikokonaisuudet eroteltu puolipisteellä, voidaan modaalit käydä läpi for-silmukassa eli toistolauseessa, joka suorittaa toistuvasti koodilohkon sisällön. Toistolauseeseen sisällä luodaan vielä erillinen modal_parts-muuttuja modaalin osille (KUVA 20), sillä modaalit on eritelty modaalikokonaisuuksissa putkiviivoin. Modaalin osat erotellaan Liquidin split-filterillä.

```
7 | {% assign geopopup_modals = settings.geo_popup_definition | split: ';' %}
8 |   {% for modal in geopopup_modals %}
9 |     <div class= "geo-js--modal">
10 |       {% assign modal_parts = modal | split: '|' %}
```

KUVA 20. modal_parts-muuttujan luonti ja erottelu for-toistolauseeseen sisällä

Ponnahtusikkuna sisältää tekstin ja painikkeen, joiden sisältö määräytyy käyttäjän maakoodin perusteella. Koska Liquidin split-filteri jakaa merkkijonon taulu-

koksi, voidaan modaalien osia käsitellä taulukkomuodossa (KUVA 21). Taulukoiden indeksointi alkaa nolasta ja tiettyyn elementtiin päästään käsiksi viittaamalla indeksinumeroa hakasulkeissa (Liquid n.d).

```

7   {% assign geopopup_modals = settings.geo_popup_definition | split: ';' %}
8   {% for modal in geopopup_modals %}
9     <div class= "geo-js--modal">
10      {% assign modal_parts = modal | split: '|' %}
11      <div class="reveal geo-popup--modal" id="geoPopupModal {{ modal_parts[0] | strip }}"
12          aria-labelledby="geoPopupModal {{ modal_parts[0] | strip }}" data-reveal>
13        <div class="modal--contents">
14          <span>{{ modal_parts[2] | strip }}</span>
15          <button class="geopopup-redirect">
16            <a href="{{ modal_parts[1] }}">{{ modal_parts[3] | strip }}</a>
17          </button>
18          <button class="geopopup-close">x</button>
19        </div>
20      </div>
21    </div>
22  {% endfor %}

```

KUVA 21. Modaalien osien käyttö taulukkomuodossa for-toistolauseen sisällä

Ponnahdusikkunan sisältö käydään läpi for-toistolauseen sisällä hyödyntämällä HTML-tyyppisiä elementtejä, tageja, tyyliluokkia ja tunnisteita. Modaalien sisällöt tulostetaan luodun modal_parts-muuttujan avulla Liquidin tulostelohkoissa, jotka muodostuvat kahdesta ympäröivästä aaltosulkeesta. Ponnahdusikkunan modaalien tulostuksessa käytetään lisäksi reveal-modaalilaajennusta, jonka avulla modaalit saadaan luotua ja käynnistettyä niiden JavaScript-toiminnallisuudet huomioiden. Ponnahdusikkunaan luotiin myös x-painike, jolla saadaan suljettua ponnahdusikkuna.

Koska ponnahdusikkunan modaalien luonnissa ja käynnistämisessä käytetään reveal-modaalilaajennusta, tulee se huomioida Liquid-koodin ympäröivän ehtolauseen sisällä. Jottei ponnahdusikkunalle luoda modaaleja siinä tapauksessa, kun ehto näyttämislle ei täyty, lisätään alkuperäiseen ehtolauseeseen else-ehto (KUVA 22). Ehdon avulla luokan geo-popup-modal data jätetään huomioida.

```

21   </div>
22   {% endfor %}
23 </div>
24 {% else %}
25   <div data-geo-popup--modal="false"></div>
26 {% endif %}

```

KUVA 22. Uusi ehto modaalien luonnin ehkäisemiseksi

Ponnahdusikkuna sisältää tyylliluokkia ja sen toiminnallisuudet on toteutettu JavaScript-koodilla. Ponnahdusikkunaa varten luodut CSS-tyylitiedosto ja skriptit sisältävä Liquid-tiedosto tulee sisällyttää ponnahdusikkunan Liquid-toteutukseen, jotta ponnahdusikkuna toimii halutulla tavalla. Koska tyylitiedosto on luotu teemakoodin assets-kansion alle, saadaan se huomioitua Liquid-koodissa tulostelohkoa hyödyntäen. Ponnahdusikkunan JavaScript-koodin sisältävä tiedosto on luotu teemakoodin snippets-kansion alle, joten se voidaan lisätä Liquid-koodiin `include`-tagin avulla. Tarvittavat tyylitiedostot lisätään useimmiten Liquid-koodin alkupäähän ja skriptit loppupäähän (KUVA 23).

```

sections > geo-popup.liquid > ...
1  {{ 'geo-popup.css' | asset_url | stylesheet_tag }}
2
3  {% if settings.geo_popup %}
4
5  <div id="geoPopupOverlay" class="geo-js--overlay">
6
7  {% assign geopopup_modals = settings.geo_popup_definition | split: ';' %}
8
9  {% for modal in geopopup_modals %}
10 <div class="geo-js--modal">
11   {% assign modal_parts = modal | split: '|' %}
12   <div class="reveal geo-popup--modal" id="geoPopupModal" data-reveal="{{ modal_parts[0] | strip }}"
13     aria-labelledby="geoPopupModal" data-reveal="{{ modal_parts[0] | strip }}" data-reveal>
14     <div class="modal--contents">
15       <span>{{ modal_parts[2] | strip }}</span>
16       <button class="geopopup-redirect">
17         <a href="{{ modal_parts[1] }}">{{ modal_parts[3] | strip }}</a>
18       </button>
19       <button class="geopopup-close"></button>
20     </div>
21   </div>
22 </div>
23 {% endfor %}
24 </div>
25 {% else %}
26 <div data-geo-popup--modal="false"></div>
27 {% endif %}
28
29 {% include 'geo-popup-script' %}
30

```

KUVA 23. Ponnahdusikkunan JavaScript- ja CSS-tiedoston sisällytys Liquid-toteutukseen

5.5.1 Ponnahdusikkunan otsikon toteutus kielieditoriin

Ponnahdusikkuna sisältää entuudestaan teemaeditorissa määriteltävää tekstisisältöä. Shopifyssa lokalisoinnit luodaan kuitenkin useimmiten käännöstiedostojen (locale) avulla, jotta niitä voidaan myöhemmin hallinnoida kielieditorista käsin. Jotta käännöstiedostot saatiin käyttöön, luotiin ponnahdusikkunalle otsikko niitä

hyödyntäen. Koska ponnahdusikkuna ilmestyy vain ulkomailta kauppaa selaileville käyttäjille, päätettiin otsikosta tehdä englanninkielinen. Kaupan pääkieli on suomi, joten muutokset tehtiin `fi.json`-tiedostoon (KUVA 24). Ponnahdusikkunan otsikko sisällytettiin `general`-nimiseen käännösryhmään.

```
13     "geo_popup": {
14         "heading": "Welcome to our store!"
15     },
```

KUVA 24. Ponnahdusikkunan otsikko `fi.json`-käännöstiedostossa

Ponnahdusikkunan otsikko sijoitettiin kielitiedostossa `general`-käännösten sisälle, joten otsikkoa päästään muokkaamaan teeman kielieditorissa `General`-välilehdellä. Käännös tulostetaan `Liquid`-tiedostossa viittaamalla käännöstiedoston ryhmän, alaryhmän ja käännöksen nimeä pistenotaatiolla (KUVA 25).

```
14     <div class="modal--contents">
15         <span class="h1" style="font-size: 28px; margin: 0px 0px 18px ">{{ 'general.geo_popup.heading' | t }}
16     </span>
```

KUVA 25. Otsikon käännöksen tulostus `Liquid`-tiedostossa

5.6 Tyylittelyt

Ponnahdusikkuna sisältää useita tyyliluokkia, jotka on määritelty tyylitiedostossa. Tyylit määräävät muun muassa ponnahdusikkunan ja sen sisällön muotoilun, ponnahdusikkunan sijainnin sivustolla sekä ponnahdusikkunan tekstien värit ja fonttikoot. Tyyleillä saadaan myös esimerkiksi tuotua ponnahdusikkuna sivuston päällimmäiseksi elementiksi. Tyylitiedoston alkupäässä (KUVA 26) on määritelty muun muassa ponnahdusikkunan näkyvyyteen, sijaintiin ja kokoon liittyviä tyylejä.

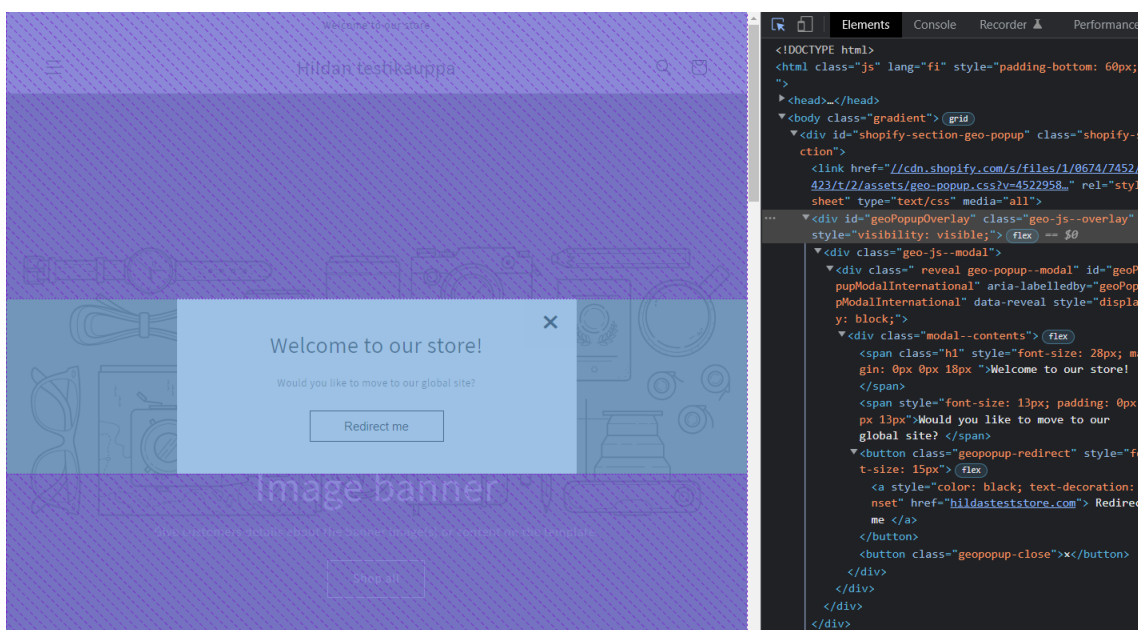
```

assets > # geo-popup.css > ...
1
2   .geo-popup--modal {
3     display: none;
4   }
5
6   .geo-js--overlay {
7     visibility: hidden;
8     position: fixed;
9     left: 0;
10    top: 0;
11    width: 100%;
12    height: 100%;
13    z-index: 9999;
14    display: flex;
15    flex-direction: column;
16    align-items: center;
17    justify-content: center;
18    background-color: rgba(27, 27, 27, 0.2);
19  }
20
21  .geo-js--modal {
22    width: 90%;
23    background-color: rgb(242, 242, 242);
24    max-width: 500px !important;
25  }

```

KUVA 26. Ponnahdusikkunan yleisen tason tyylit tyylitiedostossa

Koska `geo-popup--modal`-luokkaa tarvitaan vain, jos ponnahdusikkunan modaalit ovat käytössä, on se asetettu oletuksena näkymättömäksi. Ponnahdusikkunalle luodulla `geo-js--overlay`-luokalla ladataan koko sivuston kokoinen overlay eli päällyste (kuva 27), jonka avulla sivustoa ei päästä klikkaamaan ennen ponnahdusikkunaan reagoimalla.



KUVA 27. Ponnahdusikkunan ympärillä oleva `geo-js--overlay`-luokka Google Chrome-selaimen inspector-työkalussa

Ponnahdusikkunalle luotiin `geo-js--modal`-luokka, joka ympäröi koko ponnahdusikkunan. Ponnahdusikkunan sisällön tyylit määritellään `geo-js--modal`-luokan sisällä olevassa `modal--contents`-luokassa. Koska ponnahdusikkunan painikkeet ovat `modal--contents`-luokan sisällä, asetetaan luokat tyylitiedostossa peräkkäin välilyönneillä eroteltuna (KUVA 28). Tällä asettelulla saadaan kohdistettua tyylittelyt oikeaan luokkaan, kun luokkia on useampi sisäkkäin.

```
27 .geo-js--modal .modal--contents {
28   padding: 40px;
29   position: relative;
30   display: flex;
31   flex-direction: column;
32   text-align: center;
33 }
34
35 .geo-js--modal .modal--contents .geopopup-redirect {
36   padding: 10px;
37   cursor: pointer;
38   border: solid 1px;
39   border-color: black;
40   position: relative;
41   display: flex;
42   flex-direction: column;
43   align-items: center;
44   align-self: center;
45   width: 40%;
46   margin-top: 10px;
47 }
48
49 .geo-js--modal .modal--contents .geopopup-close {
50   font-size: 42px;
51   position: absolute;
52   top: 0;
53   right: 0;
54   line-height: 0.5;
55   padding: 18px;
56   cursor: pointer;
57   border-color: transparent;
58 }
```

KUVA 28. `modal--contents`-luokan ja sen sisältämien luokkien tyylit tyylitiedostossa

Ponnahdusikkunaa varten luotiin oma tyylitiedosto, mutta teemakoodi sisältää jo entuudestaan useita CSS-muotoisia tyylitiedostoja. Muista tyylitiedostoista johdun osa tyyleistä jouduttiin asettamaan ponnahdusikkunan Liquid-toteutuksessa inline-muotoiluna, eli asettamalla muotoilutyylin HTML-elementin `style`-attribuuttiin (KUVA 29). Näin teeman muut tyylitiedostot eivät päässeet ylikirjoittamaan ponnahdusikkunaan haluttuja tyylittelyjä.

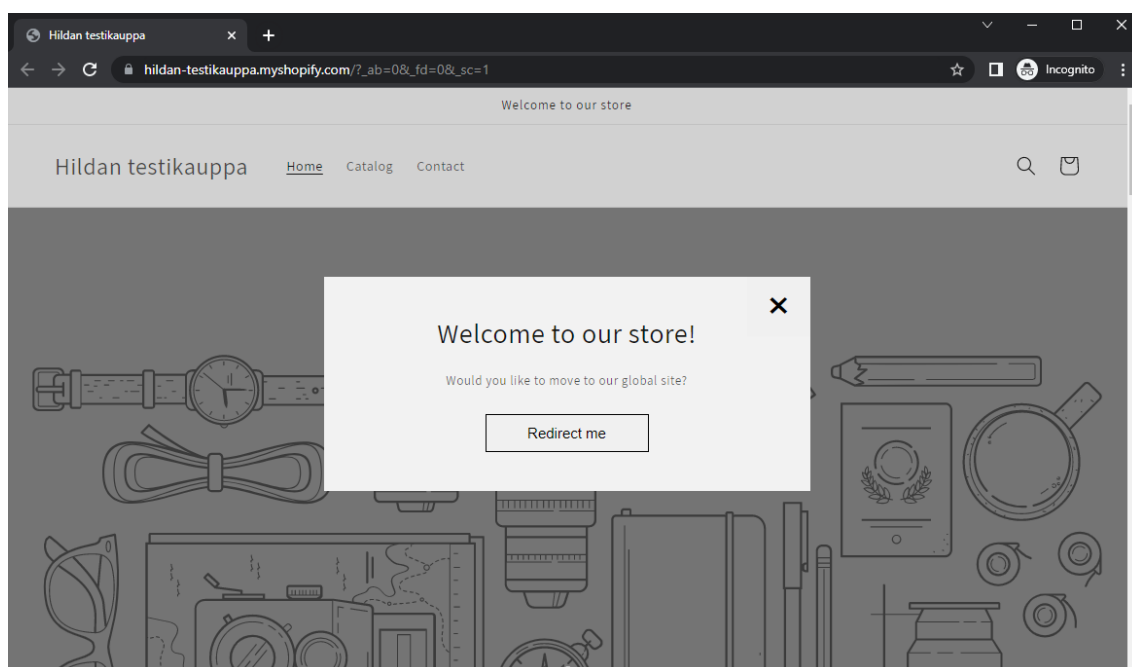
```

15 <div class="modal--contents">
16 <span class="h1" style="font-size: 28px; margin: 0px 0px 18px ">{{ 'general_geo_popup.heading' | t }}
17 </span>
18 <span style="font-size: 13px; padding: 0px 0px 13px">{{ modal_parts[2] | strip }}
19 </span>
20 <button class="geopopup-redirect" style="font-size: 15px">
21 <a style="color: black; text-decoration: unset" href="{{ modal_parts[1] }}">
22 <span style="font-size: 13px; padding: 0px 0px 13px">{{ modal_parts[3] | strip }}
23 </a>
24 </button>

```

KUVA 29. Ponnahdusikkunan Liquid-toteutuksen inline-muotoilut

Ponnahdusikkuna sisältää useita luokkia, jotka sisältävät melko paljon erilaisia tyyllittelyjä. Tyyli tiedostossa olevien luokkien lisäksi ponnahdusikkunan ulkoasuun (KUVA 30) vaikuttavat sen Liquid-toteutukseen inline-muotoiluna asetetut tyylit.





KUVA 30. Ponnahdusikkunan ulkoasu

5.7 Toiminnallisuudet

Toteutuksen JavaScript-osuus suorittaa ponnahdusikkunan toiminnallisuudet. Koska skriptejä varten on luotu Liquid-muotoinen snippet-tiedosto, tulee sen sisällä oleva JavaScript-koodi ympäröidä `script`-tageilla. Ponnahdusikkunan toiminta perustuu maakodiin, joten selaajan maakoodi tarvitsee olla saavutettavissa.

Mikäli käyttäjän selaimen sisäiseen tallennustilaan päästään käsiksi, voidaan käyttäjän maakoodi hakea joko sessionStoragesta eli istuntokohtaisesta muistista tai käyttäjän IP-osoitetta hyödyntävästä sijainnihakupalvelusta. Heti skriptin alussa varmistetaan, että selaimen tallennustila on käsiteltävissä. Mikäli edellä mainittu ehto täyttyy, voidaan haettava arvo eli maakoodi myös varastoida muuttajaan (KUVA 31).

```
snippets >  geo-popup-script.liquid >  script
1 <script>
2   if (typeof sessionStorage == 'object') {
3     const countryCode = sessionStorage.getItem('countryCode');
4
```

KUVA 31. Selaimen tallennustilan saavutettavuuden tarkistus ja haettavan maakoodin tallennus country_code-muuttajaan

Toteutuksen sijainnihakuosuudessa päätettiin hyödyntää GeoJS-rajapintaa JavaScriptin oman Geolocation API:n sijaan, sillä GeoJS palauttaa maakoodin lisäksi myös muita sijaintitietoja. GeoJS-rajapinta palauttaa muun muassa kaupungin, kunnan, osavaltion ja mantereiden nimien tai koodin. Se ei myöskään vaadi erillistä hyväksyntää selaajan sijaintitietojen käyttöön.

Jos sessionStoragesta eli istuntokohtaisesta muistista ei löydy käyttäjän maakoodia, määritellään se sijainnihakupalvelun avulla kutsumalla GeoJS-rajapintaa fetchCountryCode-funktiossa. Tämä mahdollistaa sen, että haettava maakoodi päivitetään selaajan istuntokohtaiseen muistiin lataamatta koko sivua uudelleen (KUVA 32).

```
5   if (countryCode == null) {
6     fetchCountryCode()
7   } else {
8     handleCountryCode(countryCode);
9   }
10
11  async function fetchCountryCode() {
12    const response = await fetch('https://get.geojs.io/v1/ip/geo.json');
13    if (response.status == 200) {
14      const data = await response.json();
15      sessionStorage.countryCode = data.country_code;
16      handleCountryCode(data.country_code);
17    }
18  }
```

KUVA 32. Maakoodin haku GeoJS-rajapinnasta

Maakoodi käsitellään seuraavaksi `handleCountryCode`-funktiossa (KUVA 33). Mikäli teemaeditoriin määritelty kaupan maakoodi on eri kuin GeoJS-rajapinnasta haettu maakoodi eli `countryCode`, avataan GeoJS-rajapinnasta saadulle maakoodille määritetty ponnahtusikkuna. Tämä tehdään kutsumalla `openModal`-funktioita. Mikäli taas teemaeditoriin määritelty kaupan maakoodi on sama kuin GeoJS-rajapinnasta saatu maakoodi, poistetaan ponnahtusikkuna.

```

19
20     function handleCountryCode(countryCode) {
21         if ('{{settings.store_country_code}}' != countryCode) {
22             openModal(countryCode);
23         } else {
24             document.querySelector('#shopify-section-geo-popup').remove();
25         }
26     }
27

```

KUVA 33. Selaimen ja kaupan maakoodin vertailu `handleCountryCode`-funktiossa

Maakoodi käsitellään `openModal`-funktiossa (KUVA 34). Ensin haetaan maakoodille asetettu ponnahtusikkuna. Jos maakoodille ei ole määritelty teeman asetuksissa yksilöityä ponnahtusikkunaa, käytetään siinä tapauksessa International-modaalia. Eri maista tulevat selaajat voidaan siis ohjata esimerkiksi kansainväliseen kauppaan luettelematta kymmeniä maakoodeja teemaeditorin modaaliin erikseen. Ponnahtusikkuna avataan, mikäli selaimen maakoodi on eri kuin kaupan maakoodi.

```

27
28     function openModal(countryCode) {
29         const modalSelector = document.getElementById('geoPopupModal' + countryCode) ||
30             document.getElementById('geoPopupModalInternational');
31         const closeButtons = document.querySelectorAll('.geopopup-close');
32

```

KUVA 34. Maakoodin käsittely `openModal`-funktiossa

Koska ponnahtusikkunan on tarkoitus näkyä selaajalle vain kerran, tulee määrittää, missä tapauksissa ponnahtusikkuna on jo näytetty (KUVA 35). Tämä tarkistetaan selaimen istuntokohtaisesta muistista hakemalla arvo `geoPopupModalShown`-avaimella avain-arvoparista. Mikäli arvoa ei löydy, asetetaan ensiksi ponnahtusikkunan päällimmäinen `.geo-js--overlay`-luokka näkyviin. Tämän

jälkeen istuntokohtaiseen muistiin asetetaan arvo merkiksi siitä, että ponnahtusikkuna on näytetty selaajalle. Lopuksi käyttäjälle näytetään maakoodikohtainen ponnahtusikkuna asettamalla modalSelector-elementti näkyväksi.

```

33     if (sessionStorage.geoPopupModalShown != 'shown') {
34         document.querySelector('.geo-js--overlay').style.visibility = 'visible';
35         sessionStorage.geoPopupModalShown = 'shown';
36         modalSelector.style.display = 'block';
37     }

```

KUVA 35. Ponnahtusikkunan aktivoiva ehtolause

Koska ponnahtusikkunaan toteutettiin erillinen ponnahtusikkunan sulkeva painike, tulee sen toiminnallisuudet määrittellä JavaScript-koodilla. Toiminnallisuudet määrittellään luodun openModal-funktion sisällä, jossa closeButton-muuttujaan on tallennettu jokaisen modaalin sulkeva painike-elementti. Jokaiselle elementille kiinnitettiin tapahtumankäsittelijä kutsumalla forEach-toistolauseessa button-muuttujaa addEventListener-metodilla. Metodi tunnistaa, kun käyttäjä painaa nappia. (KUVA 36).

```

39     closeButton.forEach(button => {
40         button.addEventListener('click', event => {
41             document.getElementById('shopify-section-geo-popup').style.display = 'none';
42             document.getElementById('geoPopupOverlay').style.display = 'none';
43         });
44     });
45 }
46 }
47 </script>

```

KUVA 36. Tapahtumankäsittelijän lisäys ponnahtusikkunan sulkupainikkeisiin

5.7.1 Sulkevan painikkeen poisto maakoodin perusteella

Kun samalla verkkokauppialla on useampi kauppa, saatetaan tietystä sijainnista saapuvat asiakkaat haluta ohjata tiettyyn kauppaan. Ohjaus onnistuu luodun ponnahtusikkunan välityksellä, mutta joissain tapauksissa esimerkiksi eri maiden kauppojen hinnastot ovat keskenään poikkeavia. Tästä syystä käyttäjälle ei välttämättä haluta antaa vaihtoehtoa selailta toisen maan kauppaan. Mikäli esimerkiksi tietystä maakoodista tulevilta käyttäjiltä halutaan estää pääsy verkkokaup-

paan, voidaan se toteuttaa yksinkertaisesti poistamalla ponnahtusikkunan sulkeva painike (KUVA 37). Tämäkin toiminnallisuus lisättiin ehtolauseella `openModal`-funktioon.

```

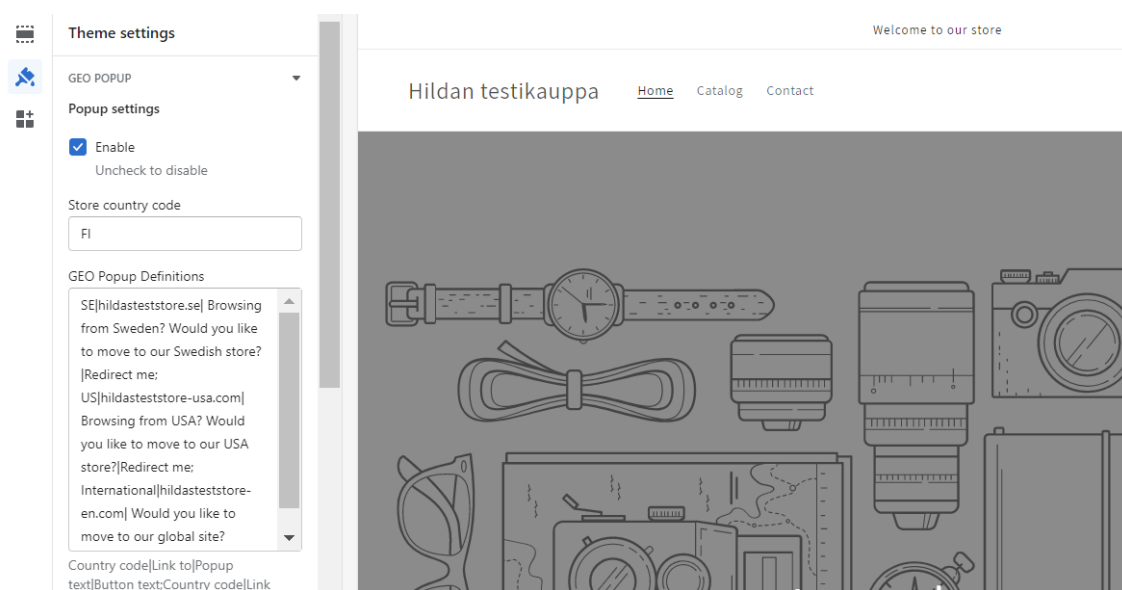
32
33     if (countryCode == 'US') {
34         closeButton.forEach(button => {
35             button.remove();
36         });
37     }

```

KUVA 37. Sulkemispainikkeen poisto Yhdysvalloista kauppaa selailevilta

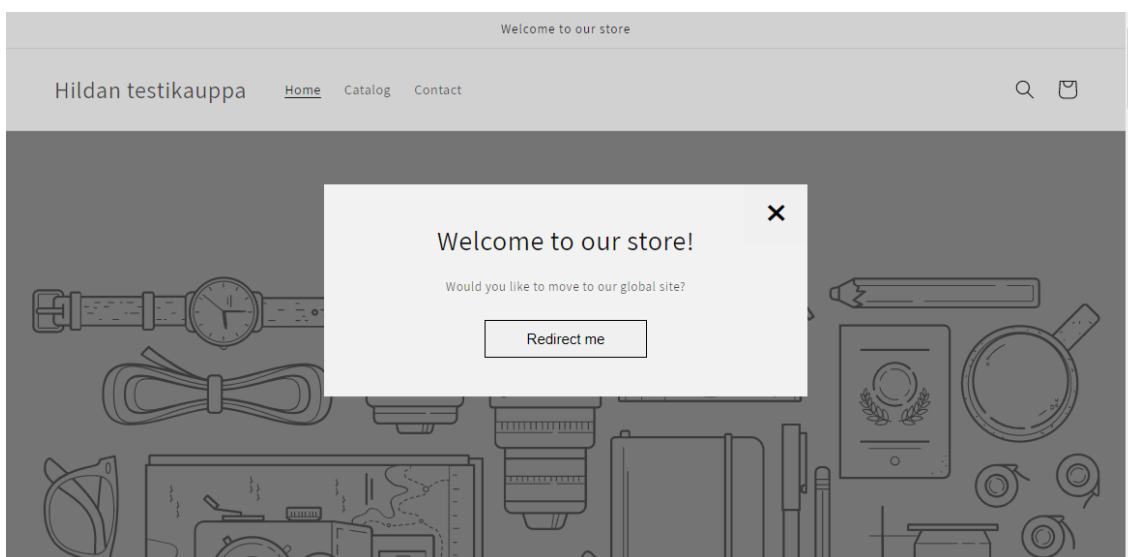
5.8 Lopputulos

Toteutuksen tuloksena syntyi toimiva ponnahtusikkuna, jonka sääntöjä ja määritelmiä voidaan jatkossa muuttaa teemaeditorin välityksellä. Teemaeditoriin asetettiin oletusarvoiset asetukset (KUVA 38), joilla määritellään ponnahtusikkunan käyttäytyminen perustuen käyttäjän sijaintiin. GEO Popup Definitions -kenttään asetettiin alustavasti kaksi maata, joista ohjataan selaaja oman maansa kauppaan. Muista maista tulevat asiakkaat ohjataan nykyisten asetusten mukaan kansainväliseen kauppaan. Tämä tarkoittaa sitä, että ponnahtusikkuna ilmestyy kaikille Suomen ulkopuolelta kaupassa vieraileville käyttäjille.



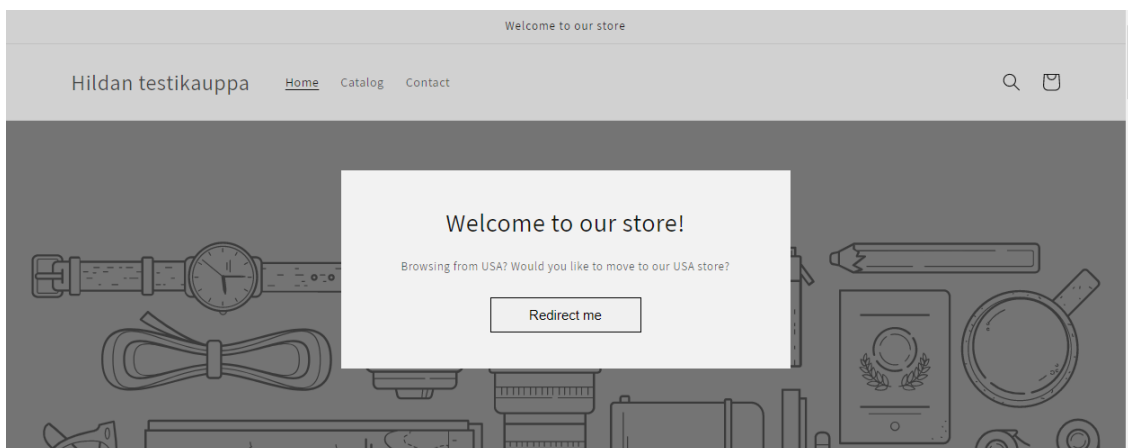
KUVA 38. Teeman asetusten määrittely teemaeditorissa

Kun käyttäjä tulee kauppaan Suomen ulkopuolelta jostain muusta maasta kuin Ruotsista tai Yhdysvalloista, hänet ohjataan ponnahtusikkunan avulla kansainväliseen kauppaan (KUVA 39). Kun käyttäjä tulee kauppaan Ruotsista, hänet ohjataan vastaavasti ruotsinkieliseen kauppaan. Teemaeditoria käyttäen verkkokauppias pystyy yksinkertaisesti muokkaamaan sekä ponnahtusikkunan että sen painikkeen tekstiä.



KUVA 39. Suomen verkkokauppaan ilmestytävä ponnahtusikkuna kansainväliseen kauppaan ohjattaville asiakkaille

Toteutuksen JavaScript-koodissa luotiin toiminnallisuus, jonka johdosta ponnahtusikkunan sulkeva painike piilotetaan kauppaan Yhdysvalloista selailevilta asiakkailta. Kun verkkokauppaan tullaan Yhdysvalloista, käyttäjän ainoa mahdollisuus on siirtyä painikkeesta oman maansa kauppaan (KUVA 40).



KUVA 40. Ponnahtusikkuna ilman sulkumahdollisuutta

6 TESTAUS

Toteutusta testattiin usealla eri selaimella sekä työpöytä- että mobiilinäkymä huomioiden. Erilaisia näytön kokoja testattiin selainten inspector-työkalun (esim. Google Chromen kehittäjän työkalu) avulla. Ponnahdusikkunan tyylien todettiin istuvan hyvin erikokoisilla sisältöalueilla sekä mobiili- että työpöytänäkymään.

6.1 Testiteema ja Theme Kit

Shopifyssa testiteemaa voidaan esikatsella selaimessa teeman Preview-esikatselutilalla. Esikatselutoiminto löytyy teemakohtaisesti Shopify:n admin-paneelista sekä teema- ja koodieditorista. Koska toteutuksessa tarkasteltiin käyttäjän maakoodia ja maakoodi voidaan hakea selaimen istuntokohtaisesta muistista, avattiin esikatselu aina yksityisessä selaintilassa eli Incognito-ikkunassa. Yksityinen selaintila mahdollistaa sen, ettei selain tallenna istuntokohtaisia tietoja selaimen sisäiseen muistiin.

Teemakehityksessä on tyypillistä, että jokainen uusi ominaisuus tai toteutus tehdään osa kerrallaan. Tästä syystä jokaisen uuden ominaisuuden tai korjauksen jälkeen muutokset päivitettiin tehtyyn testiteemaan Theme Kitin theme watch -komennon avulla. Komento ajetaan koodieditorityökalun terminaalissa (KUVA 41) ja se ajaa muutokset valittuun teemaan aina tiedoston tallennuksen yhteydessä (Theme Kit n.d).

```
PS C:\Users\Hilda Lindfors\Testikauppa> theme watch --env development
19:13:08 [development] Hildan testikauppa: Watching for file changes to theme 138413932855
19:13:12 [development] processing config/settings_schema.json
```

KUVA 41. Muutosten päivittäminen testiteemaan theme watch -komennolla

6.2 Testaus VPN-yhteydellä

Koska toteutusta tuli testata useasta eri maasta, hyödynnettiin testaamisessa VPN-yhteyttä. VPN mahdollistaa virtuaalisen erillisverkon käytön, jonka avulla

selaimen IP-perusteinen sijainti saadaan muutettua valitun näennäisverkon sijainnin mukaan. VPN-sovellukseksi valittiin toiminnallisuuksiltaan parhaalta vaikuttava, turvallinen sekä ilmainen ProtonVPN-sovellus, jonka avulla eri testitapaukset saatiin toistettua.

6.3 Testausvaiheessa havaitut ongelmat

Testausvaiheessa havaittiin, että joidenkin maiden ilmaiset VPN-yhteydet toimivat puutteellisesti ja testitapaukset jouduttiin toistamaan useamman kerran testitulosten varmistamiseksi. Lisäksi huomattiin, ettei ponnahdusikkuna toimi oikein tehokkaimmilla, ammattilaistason mainoksenestolaajennuksilla. Nämä laajennukset siis estävät ponnahdusikkunan näyttämisen, mutta eivät kuitenkaan poista ponnahdusikkunan `geo-js--overlay` -luokkaa. Tämä tarkoittaa sitä, että sivustoa ei pääse klikkaamaan ollenkaan. Sivusto siis olettaa, että ponnahdusikkunaan ei ole reagoitu, eikä tällöin päästä selaajaa selailemaan sivustoa. Ponnahdusikkunan todettiin kuitenkin toimivan yleisimmin käytössä olevilla mainoksenestolaajennuksilla.

7 POHDINTA

Shopify-pohjaisten teemojen kustomointimahdollisuudet ovat laajat, joten ominaisuuksiltaan vastaavanlainen ponnahtusikkuna on mahdollista toteuttaa monella eri tavalla. Ponnahtusikkunan toteutuksessa hyödynnettiin opinnäytetyöprosessissa opittujen asioiden lisäksi jo työelämästä saatua kokemusta Shopify-teemakehityksestä. Ponnahtusikkunasta pyrittiin saamaan verkkokaupan teemasta riippumatta toimiva kokonaisuus.

Opinnäytetyön pohjalta toteutettua ponnahtusikkunaa voidaan hyödyntää laajalti erilaisissa Shopify-verkkokauppaympäristöissä ja pienellä jatkokehityksellä toteutuksesta saataisiin helposti kaupasta kauppaan siirrettävä monistettava ratkaisu. Opinnäytetyön toteutuksen kaltaiset ominaisuudet ovat verkkokauppojen kansainvälisen myynnin kannalta elintärkeitä, sillä niiden avulla pidetään huolta asiakkaan ostoprosessin jatkumisesta.

Ponnahtusikkunan kustomointimahdollisuudet ovat loputtomat, mutta joitain ominaisuuksia voisi olla hyvä jatkokehittää. Esimerkiksi teemaeditoriin luotuja asetuksia voitaisiin selkiyttää erottelemalla modaalien osat omiin laatikkoihinsa. Tämä tosin saattaisi aiheuttaa sekaannusta modaalikokonaisuuksien erottelussa ja modaalien käsittelyssä koodin puolella. Teemaeditoriin olisi kuitenkin helppo lisätä esimerkiksi color-pickereita eli värityökaluja, joiden avulla ponnahtusikkunan värimaailmaa voisi entistä paremmin muuttaa muokkaamatta koodia. Myös tehokkaiden mainostenestolaajennusten kierto on olisi hyvä löytää jokin ratkaisu.

Mikäli ponnahtusikkunaa ja sen toiminnallisuuksia jatkokehittäisi ja testaisi ajan kanssa, siitä saataisiin muodostettua entistä selkeämpi kokonaisuus. Tästä kokonaisuudesta voitaisiin lopulta kehittää esimerkiksi sovellus, joka olisi ladattavissa Shopify:n omasta sovelluskaupasta.

LÄHTEET

Code Shopify. 2018. Theme Customization - The Settings File. Code Shopify - blogi 12.4.2018. Viitattu 17.11.2022.

https://www.codeshopify.com/blog_posts/theme-customization-the-settings-file

Endertech Team. 2021. The Shopify Theme and Online Store Editor Explained. E-commerce-blogi 14.12.2021. Viitattu 6.11.2022. <https://endertech.com/blog/the-shopify-theme-and-online-store-editor-explained>

Fabregas, K. & Main, K. 2022. What Is Shopify & How Does It Work (2022 Guide). Forbes. Luettu 5.11.2022. <https://www.forbes.com/advisor/business/software/what-is-shopify/>

How to edit HTML Code on Shopify? 2022. Avada-blogi 8.1.2022. Viitattu 17.11.2022. <https://blog.avada.io/shopify/docs/edit-html.html>

How to Edit Shopify Theme Code. 2022. Avada-blogi 3.8.2022. Viitattu 17.11.2022. <https://blog.avada.io/shopify/docs/how-edit-theme-code-shopify.html>

Kinsta. Shopify Market Share and Usage Statistics in 2022. Luettu 6.11.2022. <https://kinsta.com/shopify-market-share/>

Lannen, J. 2020. How to Safely Edit Your Shopify Theme. Speed Boostr. Luettu 17.11.2022. <https://speedboostr.com/how-to-safely-edit-your-shopify-theme/>

Le, T. 2021. How to edit Shopify theme effectively – 10+ Guides and Examples. HappyPoints-blogi 17.8.2021. Viitattu 17.11.2022. <https://happy-points.io/blog/tips-to-edit-shopify-theme-what-and-how-to-do-c2-stt-141/>

Liquid. n.d. Shopify. Luettu 11.11.2022. <https://shopify.github.io/liquid/>

McFarland, K. 2022. Can I Use JavaScript in Shopify? WebsiteBuilderInsider. Luettu 12.11.2022. <https://www.websitebuilderinsider.com/can-i-use-javascript-in-shopify/>

MDN Web Docs. 2022. What is JavaScript? Luettu 12.11.2022. https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript

Mendpara, J. 2022. Biggest Companies Using Shopify. Site Builder Report 13.5.2022. Viitattu 6.11.2022. <https://www.sitebuilderreport.com/biggest-companies-using-shopify>

Offline to Online with Shopify. n.d. Shopify. Pdf-dokumentti. Viitattu 17.11.2022. https://uploads-ssl.webflow.com/5ebd9d65d819123e03b274fd/5ed571a609d7762aa36fd2bb_Offline%20to%20Online%20with%20Shopify.pdf

Özşahan, H. n.d. Impressive Shopify Statistics You Need to Know. Popupsmart. Digital Marketing -blogi n.d. Viitattu 6.11.2022. <https://popupsmart.com/blog/shopify-statistics>

Pichardo, J. 2020. Liquid syntax and whitespace control. Shopify Tutorials -blogi 5.10.2020. Viitattu 12.11.2022. <https://joepichardo.com/blogs/shopify/liquid-syntax-and-whitespace-control>

Pixel Union. n.d. Adjusting your theme's language. Luettu 18.11.2022. <https://support.pixelunion.net/hc/en-us/articles/360022053294-Adjusting-your-theme-s-language>

Sacas, C. 2021. How to find the shopify theme id? HowCommerce-blogi 1.9.2021. Viitattu 19.11.2022. <https://www.howcommerce.com/shopify-theme-id/>

Shopify Staff. 2022. What is Shopify and How Does it Work? (2022). Shopify-blogi n.d. Viitattu 5.11.2022. <https://www.shopify.com/blog/what-is-shopify>

Sorensen, K. 2020. The ultimate list: 50+ surprising Shopify statistics. Construct Digital. Shopify-blogi 13.5.2020. Viitattu 6.11.2022 <https://www.konstruktdigital.com/shopify/shopify-statistics/>

TechForFun. 2021. How to set up your local environment for Shopify theme customization. Luettu 19.11.2022. <https://www.tecforfun.com/misc/how-to-set-up-your-local-environment-for-shopify-theme-customization/>

Theme architecture. n.d. Shopify.dev. Luettu 6.11.2022. <https://shopify.dev/themes/architecture>

Theme Kit. n.d. Shopify.dev. Luettu 18.11.2022. <https://shopify.dev/themes/tools/theme-kit>

Tovey, H. 2021. A Beginner's Guide to the Shopify Theme Structure. Shopify-blogi n.d. Viitattu 6.11.2022. <https://www.heathertovey.com/blog/guide-to-shopify-theme-structure/>

Tse, T. 2018. Understanding Shopify Theme Styles and Presets With settings_data.json. Web Design and Development -blogi 30.5.2018. Viitattu 17.11.2022. <https://www.shopify.com/partners/blog/shopify-theme-styles-and-presets>

TwentyThree. n.d. Introduction to Liquid. Luettu 12.11.2022. <https://www.twentythree.com/help/introduction-to-liquid>