



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Janina Mattila

JAVASCRIPT-KIRJASTON HYÖDYNTÄMINEN DATAN VISUALISOINNISSA

Liiketalous
2022

TIIVISTELMÄ

Tekijä	Janina Mattila
Opinnäytetyön nimi	JavaScript-kirjaston hyödyntäminen datan visualisoinnissa
Vuosi	2022
Kieli	suomi
Sivumäärä	37
Ohjaaja	Päivi Rajala

Opinnäytetyön aiheena on tutkia datan visualisoinnin hyötyjä työelämässä ja projekteissa. Tavoitteena on selvittää, kuinka JavaScript-kirjastoa käytetään datan visualisointiin.

Tutkimus aloitetaan selvittämällä datan visualisoinnin tarkoitusta ja hyötyä työelämässä. Datan visualisointi toteutetaan JavaScript-kirjastoilla, joita käsitellään opinnäytetyössä.

Opinnäytetyön teoriaosassa tutkitaan erilaisia JavaScript-kirjastoja ja niiden ominaisuuksia. Datan visualisointiin tarkoitettua ApexCharts JavaScript-kirjastoa esitellään tarkemmin, tutkitaan sen ominaisuuksia ja selvitetään, miten kirjasto toimii käytännössä.

Opinnäytetyön lopputuloksena olisi saada kattava käsitys datan visualisoimisesta JavaScript-kirjastoilla. ApexCharts-kirjastolla toteutettu esimerkki kuvaa kirjaston käyttöä käytännössä.

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Tietojenkäsittely

ABSTRACT

Author	Janina Mattila
Title	Using the JavaScript library in data visualization
Year	2022
Language	Finnish
Pages	37
Name of Supervisor	Päivi Rajala

The subject of the thesis was to investigate the benefits of data visualization in business and projects. The goal was to find out how to use JavaScript library for data visualization.

The research begins by finding out the purposes and benefits of data visualization in business. Data visualization is executed with JavaScript libraries, which are discussed in the thesis.

The theory section of the thesis examines various JavaScript libraries and their characteristics. ApexCharts JavaScript library, intended for data visualization, is introduced in more detail, its features are explored and how the library works in practice is examined.

The result of the thesis was to get a comprehensive view of data visualization with JavaScript libraries. An example executed with the ApexCharts library describes the use of the library in practice.

Keywords

Data, Visualization, ApexCharts, JavaScript-library

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO.....	8
2	DATAN VISUALISOINTI JAVASCRIPT-KIRJASTOLLA	9
	2.1 JavaScript-kirjastot datan visualisoinnissa.....	9
	2.2 Datn visualisoinnin hyödyt.....	9
	2.3 JavaScript-kirjastoja datan visualisointiin	11
	2.3.1 D3.js.....	12
	2.3.2 Chart.js	13
	2.3.3 ApexCharts	14
3	APEXCHARTS JAVASCRIPT-KIRJASTO	15
	3.1 Kirjaston historiaa	15
	3.2 ApexCharts kaavio tyypit	15
	3.3 Ominaisuudet.....	16
	3.3.1 Responsiivisuus	16
	3.3.2 Interaktiivisuus.....	17
	3.3.3 Dynaaminen	17
	3.4 Tiedon haku kaavioon AJAX-pyynnöllä ApexCharts:ssa	18
4	KAAVION TEKO JA MUOKKAUS APEXCHARTSILLA.....	20
	4.1 Kirjaston käyttöönotto.....	20
	4.2 Yksinkertaisen kaavion teko ApexCharts:lla	20
	4.3 Kaavion muokkaaminen.....	22
	4.3.1 Erilaiset kaavio tyypit	23
	4.3.2 Ulkonäön muokkaaminen	24
	4.3.3 Muita kaavion muokkaustapoja.....	27
	4.4 Kaavion datan haku JSON-tiedostosta.....	29
5	APEXCHARTS-KIRJASTON KÄYTTÖ DATAN VISUALISOINNISSA	32

5.1 ApexCharts:n hyödyt datan visualisoinnin kirjastona.....	32
5.2 Yhteenveto.....	33
6 JOHTOPÄÄTÖKSET	34
LÄHTEET	36

LYHENTEET JA TERMIT

JavaScript-kirjasto	Valmiiksi koodattu JavaScript-koodin kirjasto, joka sisältää valmiiksi koodattuja ominaisuuksia verkkosivustolle.
Datan visualisointi	Tiedon muuttamista visualiseenmuotoon, kuten kaavioiksi tai diagrammeiksi.
ApexCharts	Datan visualisointiin tarkoitettu JavaScript-kirjasto.
JavaScript	Ohjelmointikieli, jolla pystytään lisäämään erilaisia toimintoja verkkosivulle.
HTML	Hypertext Markup Language. Tunnetuin kieli, millä verkkosivuja kirjoitetaan.
JSON	JavaScript Object Notation. Kevyt tiedostomuoto, jota käytetään tiedonvälitykseen ja tallennukseen.
AJAX	Asynchronous JavaScript And XML. Sen avulla voidaan kommunikoida palvelimien kanssa. Voidaan tehdä pyyntöjä palvelimelle, ja vastaanottaa ja käsitellä tietoja palvelimelta.

KUVALUETTELO

Kuva 1. Kuvakaappaus ApexCharts:n modernista dashboard-kaaviosta. (ApexCharts 2022 C.).....	11
Kuva 2. Kuvakaappaus D3.js Bar chart race -kaaviosta. (D3 Data-Driven Documents 2021 B.)	12
Kuva 3. Kuvakaappaus Chart.js Horizontal Bar Chart:sta. (Chart.js 2022 A.)	13
Kuva 4. Kuvakaappaus ApexCharts kuplakaavion malleista. (ApexCharts 2022 B.)	16
Kuva 5. ApexCharts dokumentin mallin avulla koodattu timeline, eli aikajana, kaavio.	17
Kuva 6. Kuvakaappaus ApexCharts dynaamisen kaavion esimerkistä. (ApexCharts 2022 E.).....	18
Kuva 7. ApexCharts pylväskaavio.....	22
Kuva 8. Kuva näyttää kaavion type: 'line' -koodilla	23
Kuva 9. Kuva näyttää kaavion type: 'radar' -koodilla	23
Kuva 10. Kuva näyttää kaavion type: 'area' -koodilla.....	24
Kuva 11. ApexCharts JavaScript-koodin chart-osion muokkaamisen tulokset. ...	25
Kuva 12. Kaavio, jossa on muokattu dataviivan väriä, ruudukon värejä ja kuvan lisäämistä dataan. Kaavion "kukkia.jpg"-kuva on otettu Unsplash-sivustolta. (Henry Be 2016.)	27
Kuva 13. Kaavion muokkaus if-else-funktiolla ja pylväiden arvojen sijoittaminen ylös.	29
Kuva 14. Kuvakaappaus Visual Studio Code:lla tehdystä JSON-tiedostosta.	31

1 JOHDANTO

Opinnäytetyön aiheena on tutkia datan visualisointiin tarkoitettujen JavaScript-kirjastojen käytön hyödyllisyyttä verkkosivuilla. Opinnäytetyö toteutettiin empiirisenä tutkimuksena, jossa testattiin JavaScript-kirjaston käyttöä ja arvioitiin sen toimivuutta käytännössä.

Aihe valittiin, koska harjoittelujaksolla työhön kuului etsiä datan visualisointiin tarkoitettuja JavaScript-kirjastoja, joista havaittiin, kuinka aiheesta hyödyttiin työelämässä. Tutkimus aloitettiin selvittämällä eri JavaScript-kirjastoja, jotka soveltuvat datan visualisointiin ja valitsemalla näistä yksi esimerkki tapaukseksi opinnäytetyössä.

Datan visualisointi on tärkeää, etenkin yritysten datan tarkkailussa. Datan näyttäminen erilaisilla kaavioilla auttaa näkemään tärkeitä asioita suuresta tietomassasta. Visualisoimalla tietoa se näyttää selkeältä ja sitä on helpompi analysoida. Suuren tietomäärän ohjelmoiminen verkkosivuille on aikaa vievää kehittäjille, missä JavaScript-kirjastot tulevat hyödyksi.

Teoriaosuudessa esitellään yhden JavaScript-kirjaston, ApexCharts:n, ominaisuuksia ja datan visualistamisen mahdollisuuksia. Kirjastosta olisi tarkoitus saada mahdollisimman laaja käyttökokemus. Lopuksi kirjastoa hyödyntämällä tulisi toteuttaa esimerkkitapaus datan visualisoimisesta ja kirjaston sisällöstä.

Tavoitteena on tutkia ja selvittää, mitä JavaScript-kirjastot ovat ja kuinka datan visualisoinnista voidaan hyötyä. Tavoitteena on myös löytää helppokäyttöinen kirjasto datan visualisoimiseen, jota voisivat käyttää muutkin kuin kehittäjät. Tutkimuksen lopputuloksena olisi tarkoitus saada hyvä lähtökohta datan visualisointiin tarkoitettujen JavaScript-kirjastojen käyttöön.

2 DATAN VISUALISOINTI JAVASCRIPT-KIRJASTOLLA

Tässä luvussa käsitellään JavaScript-kirjastojen hyötyjä datan visualisoinnissa ja siihen liittyviä JavaScript-kirjastoja. Tavoitteena on tutkia, mitä JavaScript-kirjastot ovat ja kuinka tällaista kirjastoa voidaan hyödyntää kaavioiden teossa verkkosivulla.

2.1 JavaScript-kirjastot datan visualisoinnissa

JavaScript-kirjasto on valmiiseen käyttöön kirjoitettu JavaScript-koodi, joka sisältää toimintoja, joita voidaan käyttää verkkosivulla tai sovelluksissa. Kehittäjä voisi kirjoittaa koko koodin itse, mutta on hyödyllisempää käyttää jo valmiiksi toisen kehittäjän kirjoittamaa koodia. JavaScript-kirjastojen avulla voidaan säästää paljon aikaa. (Khan Academy 2022.)

JavaScript-kirjastoja löytyy moniin eri tarkoituksiin, mutta kyseisessä opinnäytetyössä tutkitaan juuri datan visualisointiin tarkoitettuja kirjastoja. Datan visualisointi on sitä, että tieto muutetaan visuaaliseksi kaavioksi tai diagrammiksi, jossa tieto näkyy kokonaisuutena, jolloin sitä on helppo analysointia. (XAMK. Data ja sen visualisointi.)

JavaScript-kehittäjälle interaktiivisten verkkosivujen tekeminen on yhtä tärkeää kuin datan visualistaminen. JavaScript-koodikielen kasvaessa myös JavaScript-kirjastoja julkaistaan enemmän. (Majorek J. 2021.) Tärkeintä on valita sopiva kirjasto kyseiseen projektiin, mikä voi olla haastavaa löytää monien eri kirjastojen joukosta.

2.2 Datan visualisoinnin hyödyt

Datan visualisoinnilla tarkoitetaan tiedon muuttamista visuaaliseksi kuvaksi, kuten kaavioksi tai diagrammeiksi, jotta sitä ymmärrettäisiin ja havainnollistettaisiin paremmin. Visualisointi tietyllä esitys tavalla voi antaa datasta uutta tietoa. (XAMK. Data ja sen visualisointi.)

Uuden tiedon esille tuominen antaa paremman kokonaiskuvan projektista. Yritykselle voidaan tehdä sivu, joka voi kuvastaa yrityksen myyntiä tältä vuodelta, tuotteiden myyntiä tai asiakkaiden määrää, kuten nähdään kuvassa 1. Tietoa voidaan visualisoida aikajaksoittain tai kokonaisuudessa, jolloin on helpompi nähdä projektin etenemistä.

Yrityksessä ja projekteissa kasaantuu yleensä paljon tietoa, mikä on vaikeaa tulkita vain katsomalla numeroita ja prosentteja. Vaikka jotkut pitävät enemmän lukujen lukemisesta, kaikille tämä ei sovi. Tämän takia on hyödyllistä koota tieto kuvaksi, jota on kaikkien helppo analysoida.

Etenkin suurissa yrityksissä datan määrä on suurta ja on hankalaa huomata tärkeimpiä ongelmia, joihin yrityksen täytyisi keskittyä. Datan visualistaminen antaa kyvyn nähdä tärkeät tiedot suuren tietomassan sisältä, jolloin on helpompi analysoida projektin heikkouksia ja vahvuuksia. (Pengon 2022.)

On otettava huomioon, että visualistamisen tavoitteena on yleensä saada näyttää tietoa monelle eri ihmiselle, kuten työkumppaneille tai asiakkaille. Tämän takia datan visuaalinen ulkoasu täytyisi olla miellyttävä ja helppo lukea. Asiakas ottaa enemmän huomiota ulkonäöltään miellyttäviin kaavioihin, kuin monimutkaisiin ja värittömiin, mikä toimii myös hyvänä mainontastrategiana.



Kuva 1. Kuvakaappaus ApexCharts:n modernista dashboard-kaaviosta. (ApexCharts 2022 C.)

2.3 JavaScript-kirjastoja datan visualisointiin

Datan visualisointiin löytyy paljon erilaisia ja erinäköisiä JavaScript-kirjastoja. Opinnäytetyöhön täytyi löytää yksi JavaScript-kirjasto, jota tutkittaisiin lähemmin, ja käytettäisiin tutkimuksen teoriaosuuteen.

Tutkimus aloitettiin etsimällä ja käymällä läpi eri visualisointiin tarkoitettuja JavaScript-kirjastoja. Tärkeintä oli, että kirjasto olisi helppokäyttöinen ja avoimen lähdekoodin kirjasto, mikä tarkoittaa, että sitä voi käyttää vapaasti kaupallisissa

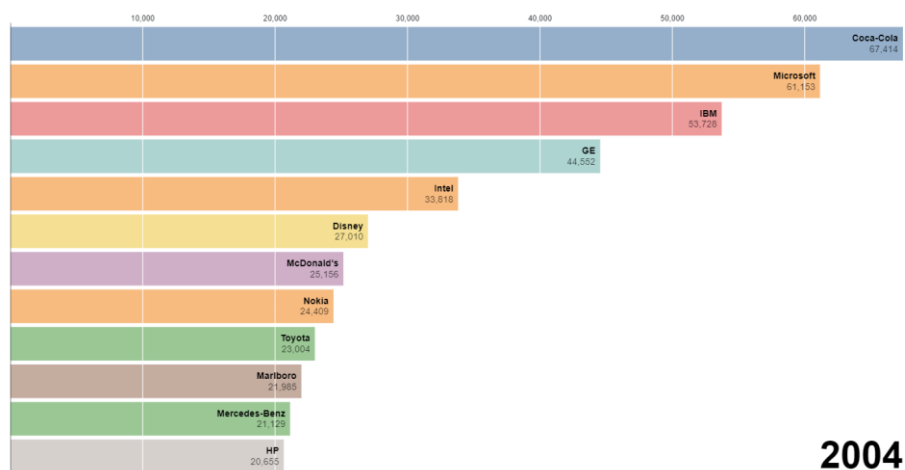
sekä ei-kaupallisissa projekteissa. Tavoitteena oli löytää kirjasto, jota voisivat muutkin käyttää kuin kehittäjät. Kirjastot, jotka valittiin lähempään tulkintaan, olivat D3.js, Chart.js ja ApexCharts.

2.3.1 D3.js

D3.js eli Data-Driven Documents on yksi suosituimmista datan visualisointiin tarkoitetuista kirjastoista, mikä tuli ilmi, koska se löytyi melkein kaikista blogikirjoituksista ensimmäisenä, jotka käsittelivät datan visualisointiin tarkoitettuja kirjastoja. Kirjasto on todella laaja ja sillä voi tehdä muutakin kuin datan visualisointia. (Majorek J. 2021.)

D3.js on avoimen lähdekoodin JavaScript-kirjasto, joka käyttää HTML:ää, SVG:tä ja CSS:ää toteuttamiseen. (D3 Data-Driven Documents 2021 A.) Se on hyvä kirjasto visuaaliseen toteutukseen, mutta se on hankala oppia ja dokumentointi on hieman sekavaa navigoida.

D3.js:llä voi tehdä upeita datan visualisointeja, kuten kuvan 2 esimerkissä, mutta sen hankala käyttöönotto ei soveltunut tähän tutkimukseen. Tutkimukseen on tärkeää löytää kirjasto, joka olisi helpompi ottaa käyttöön.



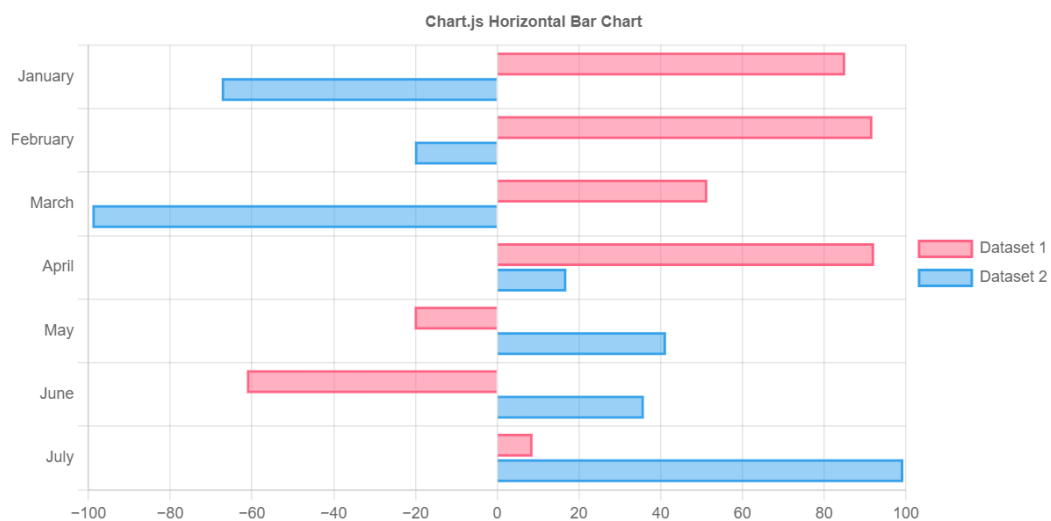
Kuva 2. Kuvakaappaus D3.js Bar chart race -kaaviosta. (D3 Data-Driven Documents 2021 B.)

2.3.2 Chart.js

Chart.js on todella suosittu avoimen lähdekoodin kirjasto, joka on saanut yli 60 000 tähteä GitHubissa. Se julkaistiin vuonna 2013 ja se on kasvanut siitä lähtien. Aktiivinen yhteisö pitää Chart.js:ää yllä. Sen miellyttävä kaavioiden visuaalinen ulkonäkö ja helppo käyttöönotto sopii hyvin datan visualisoinnin JavaScript-kirjastoksi. (Chart.js 2022 B.)

Chart.js on kevyt ja sillä voi rakentaa reagoivan kaavion HTML5 Canvas-elementtien avulla. Kirjasto on miellyttävän näköinen, kuten kuvan 3 kaaviosta voi nähdä, ja sen dokumentointi sekä esimerkit ovat selkeitä.

Kirjastossa on kahdeksan eri kaaviotyyppiä: tutkakartta, viivakaavio, ympyräkaavio, pylväskaavio, hajontakaavio, aluekaavio, kuplakaavio ja sekakaavio. Nämä riittävät peruskaavioiden tekoon, mutta D3.js:llä ja ApexCharts:lla saa tehtyä enemmän eri kaaviomalleja.



Kuva 3. Kuvakaappaus Chart.js Horizontal Bar Chart:sta. (Chart.js 2022 A.)

2.3.3 ApexCharts

ApexCharts on uusi, 2018 vuonna julkaistu, avoimen lähdekoodin kirjasto, jonka avulla kehittäjät voivat rakentaa interaktiivisia datavisualisointeja. Se sisältää useita eri kaaviotyyppejä, joita voi hyödyntää monissa eri datoissa. (ApexCharts 2022 A.)

ApexCharts:n dokumentointi on laaja sekä helposti luettava, ja kaavioiden muokaus tapoja löytyy useita. Melkein kaikki tarvittava löytyy suoraan ApexCharts:n omilta sivulta, mutta jos tarvittavaa koodia ei löydy sieltä, niin voi etsiä GitHubista.

ApexCharts JavaScript-kirjasto valittiin opinnäytetyöhön tutkittavaksi, koska se on helppo ottaa käyttöön, siitä löytyy paljon eri ominaisuuksia ja sen moderni ulkoasu, jota voi kätevästi muokata, sopii useisiin projekteihin. ApexCharts:sta kerrotaan lisää kolmannessa kappaleessa.

3 APEXCHARTS JAVASCRIPT-KIRJASTO

ApexCharts on moderni, MIT-lisenssin alla oleva, avoimen lähdekoodin JavaScript-kirjasto, jonka avulla kehittäjät voivat rakentaa interaktiivisia datan visualisointeja. Kirjasto tukee suosituimpia UI Frameworks eli käyttöliittymäkehelyksiä: Angular, React ja Vue.js. (ApexCharts 2022 A.)

3.1 Kirjaston historiaa

ApexCharts julkaistiin heinäkuussa vuonna 2018 ja se nimettiin Product Hunt -nimisellä sivustolla Päivän tuote no. 2:ksi. (ApexCharts 2022 A.) Kirjasto sai kuukaudessa noin miljoona latausta npm:llä, jolla kehittäjät jakavat muun muassa JavaScript-koodi paketteja. (Npm 2022.) Kirjasto on kasvanut siitä lähtien.

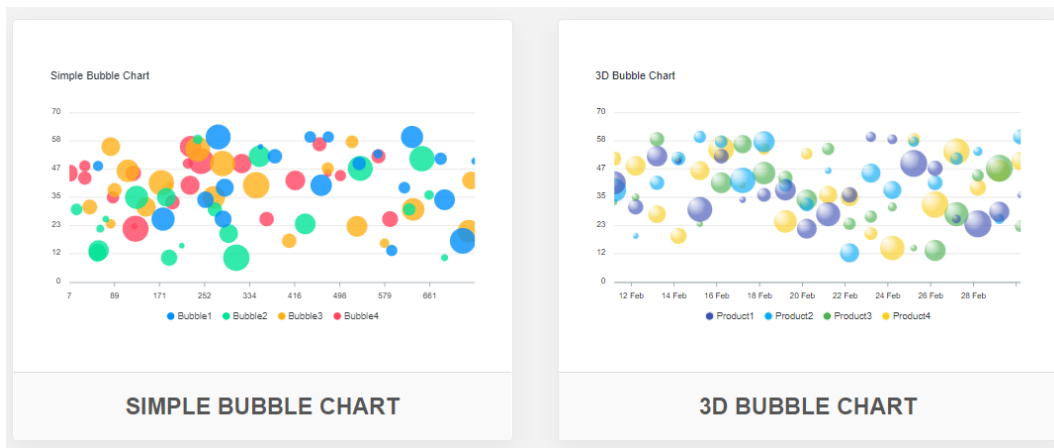
Kehitys tiimin takana ovat Juned Chhpa ja Brian Lagunas. Juned Chhipa on ApexCharts projektin perustaja ja hän on ammatiltaan front-end-insinööri. Hän toimii edelleen aktiivisena ApexCharts:n kehityksessä. Brian Lagunas on muun muassa Microsoft MVP, Xamarinin MVP, puhuja, kouluttaja ja Pluralsight-kirjailija. Hän omistaa ja ylläpitää Brian Prism-kirjastoa. ApexCharts projektissa hän korjaa ja varmistaa, ettei projektissa esiinny virheitä. (ApexCharts 2022 A.)

Vuosien varrella ApexCharts on pystynyt rakentamaan suuren yhteisön, joka osallistuu aktiivisesti projektiin. Tämä avoimen lähdekoodin yhteisö koostuu yli 100 kehittäjästä ja teknologian taitavista henkilöistä. (ApexCharts 2022 A.)

3.2 ApexCharts kaavio tyypit

ApexCharts:lta löytyy monia eri kaaviotyyppettä, joita voi muokata omanlaisiksi. JavaScript kaaviotyyppettä ovat: viivakaavio, aluekaavio, sarakekaavio, pylväskaavio, sekakaavio, kantama-aluekaavio, aikajana, kynttilänjalkakaavio, laatikkokaavio, kuplakaavio, hajontakaavio, lämpökartta, puukartta, ympyrädiagrammi, säteittäi-

nen pylväskaavio, tutkakartta, napa-aluekartta ja kojelaudat eli dashboards. Jokaisesta kaaviotyypistä löytyy useampia kaaviomalleja, josta näkyy esimerkki kuvassa 4, jossa esitetään kuplakaavion kaksi mallia.



Kuva 4. Kuvakaappaus ApexCharts kuplakaavion malleista. (ApexCharts 2022 B.)

3.3 Ominaisuudet

ApexChartsilta löytyy useita hyödyllisiä ominaisuuksia, joilla voidaan visualisoida dataa haluttuun tapaan. Tässä kappaleessa käsitellään kirjaston muutamia tunnetuimpia ominaisuuksia.

3.3.1 Responsiivisuus

Responsiivisuus tarkoittaa sitä, että verkkosivu skaalautuu laitteen näytön suuruuden mukaan. ApexCharts kaaviot voi muuttaa sopiviksi pöytätietokoneen-, tabletien- ja matkapuhelimien näyttöön. (ApexCharts 2022 F.)

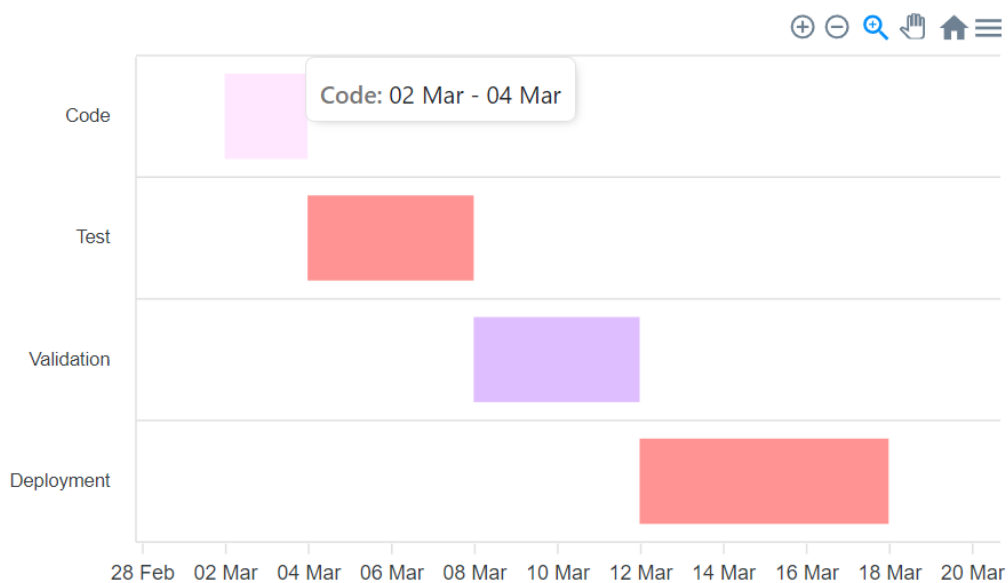
Nykyään suuri osa ihmisistä varmasti katsoo useimmiten verkkosivuja mobiililaitteella. Siksi on hyödyllistä, että JavaScript-kirjasto tarjoaa tämän mahdollisuuden kehittäjille.

3.3.2 Interaktiivisuus

ApexCharts kaavioista löytyy erilaisia toimintoja, joilla käyttäjä voi interaktivoida eli vuorovaikuttaa kaavioiden kanssa. Tämä tapahtuu kaavion oikeassa yläkulmassa olevan työkalu paneelin avulla, joka voidaan nähdä kuvassa 5.

Työkalu paneelissa on toiminnot, joita klikkaamalla voidaan zoomata, selata kaavion tietoja, panoroida, palauttaa kaavio oletus tilaan ja ladata kaavio SVG, PNG ja CSV-tiedostoina.

Huomioitavaa on myös, että kun kaavion tiedon osaa osoittaa hiirellä se tuo automaattisesti näkyviin ponnahtus laatikon, jossa lukee kyseisen kohdan tiedot ilman lisä koodausta. Esimerkin tästä voit nähdä kuvassa 5.

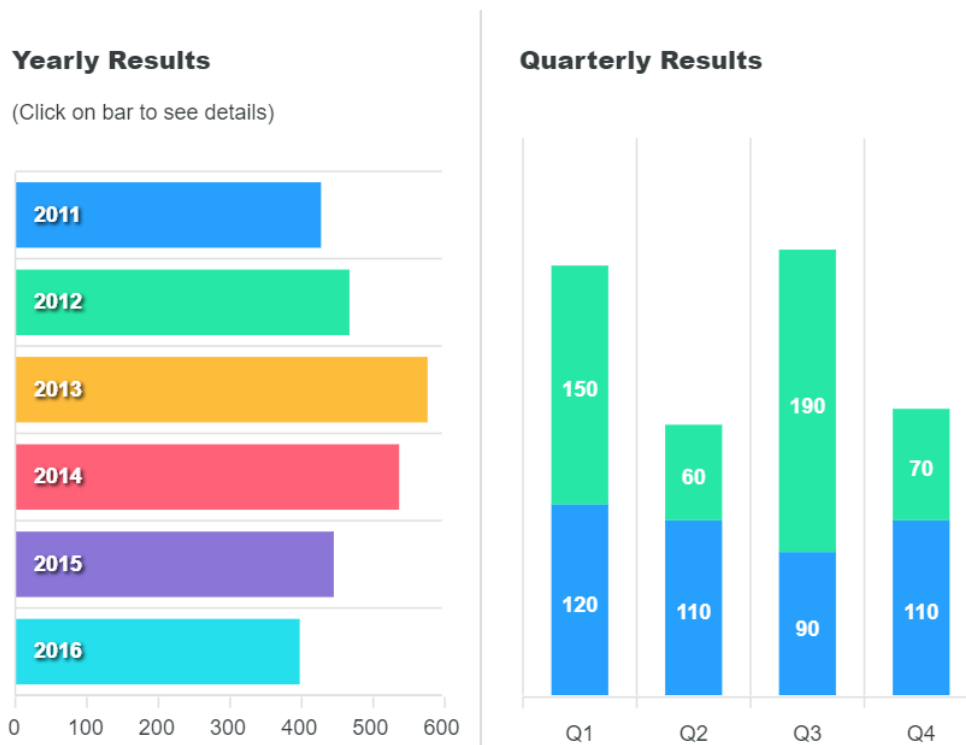


Kuva 5. ApexCharts dokumentin mallin avulla koodattu timeline, eli aikajana, kaavio.

3.3.3 Dynaaminen

ApexCharts-kaavioiden dynaamisella ominaisuudella voi tuoda tietoja eri valinnoista, kuten kuvan 6 vasemmalla puolen valinnoista nähdään, ja niistä voidaan

valinnan perusteella luoda kaavioita. (ApexCharts 2022 F.) Kuvassa 6 on klikattu arvoja 2011 ja 2012, jotka tekevät yhteisen kaavion kuvan oikealle puolelle.



Kuva 6. Kuvakaappaus ApexCharts dynaamisen kaavion esimerkistä. (ApexCharts 2022 E.)

3.4 Tiedon haku kaavioon AJAX-pyyntöillä ApexCharts:ssa

Usein kaavioon tarvittava tieto ei ole heti käytössä, vaan se pitää hakea esimerkiksi tietokannasta. Tällaisissa tapauksissa ApexCharts:illa voidaan tehdä tyhjä kaavio, jossa tieto-osa jätetään kirjoittamatta. Tarvittava tieto haetaan tekemällä AJAX-pyyntö käyttäen jQuery:n getJSON menetelmää. (ApexCharts 2022 G.) Tietokannasta tulevalle tiedolle voidaan sen jälkeen määrätä kaaviotyyppi, jonka mukaan se renderöityy verkkosivulle.

```
var url = 'http://my-json-serv-er.typi-
code.com/apexcharts/apexcharts.js/yearly';

$.getJSON(url, function(response) {
chart.updateSeries([{
```

```
        name: 'Sales',  
        data: response  
    })  
});
```

Yllä esiintyvässä koodissa `getJSON` käsky hakee tiedon url muuttujan avulla, joka on määritelty var url kohdan arvossa. ApexCharts käyttää koodissa esimerkkinä GitHubista löytyvää paikallista JSON-tiedostoa, joten kaavion data tulee kyseisestä tiedostosta.

Kyseinen JavaScript-koodi on suora esimerkki ApexCharts:n toteuttamasta `getJSON` menetelmästä. (ApexCharts 2022 G.) ApexCharts sivun dokumentin ohjeistuksen avulla koodia voidaan hyödyntää omien tietokantojen datan visualisoinnissa.

4 KAAVION TEKO JA MUOKKAUS APEXCHARTSILLA

Luvussa esitetään vaiheittain, kuinka tehdään yksinkertainen kaavio, jota voi helposti muokata, ApexCharts-kirjastoa käyttäen. Tutkimuksen tavoitteena oli löytää kirjasto, jota muutkin kuin kehittäjät voisivat hyödyntää. Koodissa käytetään samaa dataa kaikissa kaavioissa, jotta nähdään kuinka monta eri kaaviomallia voidaan tehdä samalla datalla. Viimeisessä kappaleessa näytetään, kuinka kaavion voi tehdä JSON-tiedostosta.

4.1 Kirjaston käyttöönotto

Ensimmäiseksi on otettava käyttöön ApexCharts JavaScript-koodi, jolloin kirjastoa voidaan käyttää verkkosivulla. Tämä toteutuu niin, että haetaan ApexCharts:n sivuilta koodi, jonka avulla voidaan käyttää kyseistä kirjastoa. (ApexCharts 2022 D.)

```
<script src="https://cdn.jsdelivr.net/npm/apexcharts"></script>
```

Yllä esiintyvä koodi lisätään HTML-koodin head-osioon, jolloin koodi ymmärtää hakea ApexCharts:n JavaScript tiedot. Alapuolen koodi esimerkissä, näkyy miltä head-osion tulisi näyttää tässä vaiheessa.

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>ApexCharts Malli</title>
  <script src="https://cdn.jsdelivr.net/npm/apexcharts"></script>
</head>
```

4.2 Yksinkertaisen kaavion teko ApexCharts:lla

Kun tarvittavat koodit on lisätty, niin kirjoitetaan HTML-koodin body-osioon seuraava JavaScript-koodi. Kyseinen koodi on otettu ApexCharts sivun "Get Started" -luvun "Creating Your First JavaScript Chart" -kappaleesta, ja muutettu haluttuun

muotoon. (ApexCharts 2022 D.) Koodi laitetaan script-koodin sisään, koska kyseinen JavaScript-koodi kirjoitetaan HTML-nimiseen tiedostoon.

```
<script>
  var options = {
    chart: {
      type: 'bar',
      height: 400,
      width: 600
    },
    series: [{
      name: 'Myynti',
      data: [100,259,300,350,400,200,230,160,270,390,400,589]
    }],
    xaxis: {
      categories: ["Tammi", "Helmi", "Maalis", "Huhti",
        "Touko", "Kesä", "Heinä", "Elo", "Syys", "Loka", "Mar-
        ras", "Joulu"]
    }
  }
</script>
```

JavaScript-koodi hakee bar-tyyppisen kaavion eli pylväskaavion ApexCharts-kirjastosta. Koodin series määrittää kaavion nimen sekä y-akselin arvot. Xaxis määrää arvot x-akselille. Kuvassa 7 huomataan, kuinka series-osion arvot näkyvät y-akselilla ja xaxis arvot näkyvät x-akselilla.

Valmiina oleva koodi pitää saada näkymään verkkosivulla, mikä tapahtuu seuraavasti. Var options -osion sulkujen ulkopuolelle tulee laittaa koodi, jolla määritetään ApexCharts kaavion luominen sekä id, jolla tunnistetaan kyseinen JavaScript-koodi. Tämän koodin jälkeen on laitettava käsky, joka toteuttaa kaavion renderöinnin. Kyseiset koodit esitetään kappaleen alla.

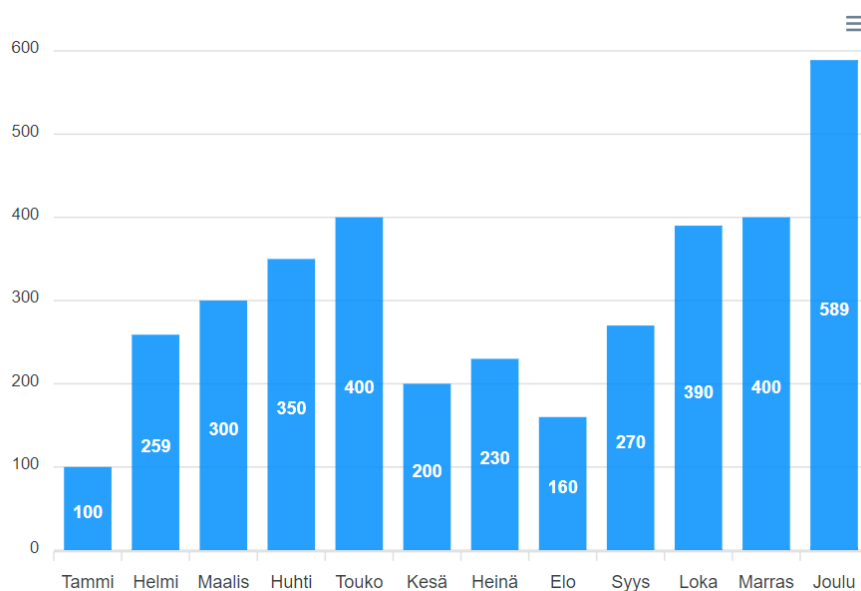
```
var chart = new ApexCharts(document.querySelector("#chart"), options);
```

```
chart.render();
```

ApexCharts koodin JavaScript osuus on valmis, mutta se ei näy vielä verkkosivulla. Tämä johtuu siitä, että id:tä ei ole vielä nimetty HTML-tiedostoon. Tämä tapahtuu niin, että script-koodin yläpuolelle kirjoitetaan HTML-koodilla id. Id:n nimi täytyy olla "chart", koska yllä mainitussa JavaScript-koodissa id:n tunnistava koodi on nimetty "#chart".

```
<div id="chart"></div>
```

Yllä näkyvä koodi kuvaa HTML-koodia, jolla haetaan ApexCharts JavaScript-koodi. Kun koodi on valmis ApexCharts kaavio renderöityy kuvan 7 mukaan verkkosivulle.



Kuva 7. ApexCharts pylväskaavio.

4.3 Kaavion muokkaaminen

Kaavion muokkaaminen käy kätevästi ApexCharts:lla, ja kaikki koodit sekä ohjeet löytää helposti ApexCharts:n omilta sivuilta. ApexCharts kaavioita voi muokata monella tavalla, mutta kappaleessa käydään vain muutama tapa monista tavoista muokata kaavioita.

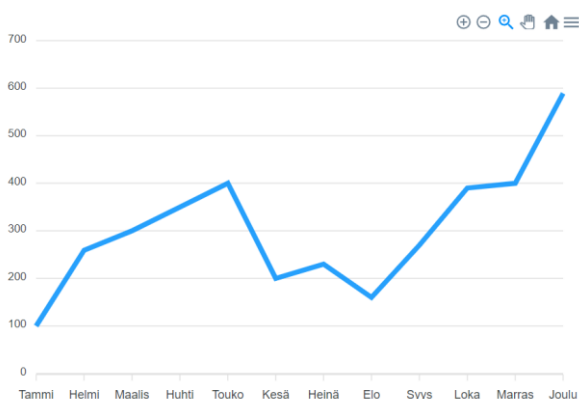
4.3.1 Erilaiset kaavio tyypit

ApexCharts sisältää monia erilaisia kaaviotyyppisiä ja -malleja erilaisille datoille. Kaikki kaaviot eivät käy kaikkiin datoihin, kuten nykyinen data ei soveltuisi ympyrädiagrammiksi, koska sen tulisi olla prosentteina ja eri muodossa JavaScript-koodissa. Siksi täytyy käyttää vain kaavioita, jotka sopivat projektin dataan.

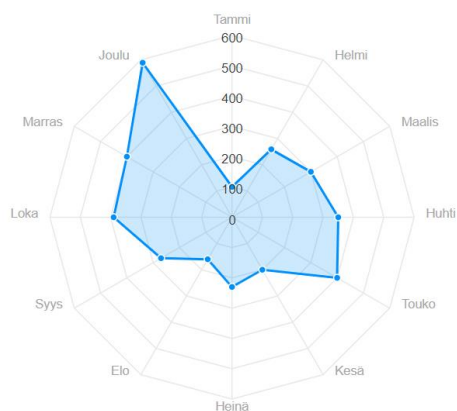
Kaavio tyyppin muuttaminen käy muokkaamalla ApexCharts JavaScript-koodin options kohdan chart-osuutta, alla näkyvän koodin mukaan. Chart osuuden "type" -niminen koodi määrää kaavion tyyppin. Alla olevassa koodissa type on määriteltä "bar" -nimiseksi kaavioksi, mutta jos se muutetaan "line" -nimiseksi sen ulkonäkö muuttuu viivakaavioksi, kuten kuvassa 8.

```
var options = {
  chart: {
    type: 'bar'
  }
}
```

Jatkamalla samalla mallilla kaavion tieto voidaan muuttaa radar eli tutkakaavioksi, kuin kuvassa 9, tai area eli aluekaavioksi, kuin kuvassa 10. Muut kaavio tyypit eivät sovellu koodin käsittelemään dataan. Jos koodi määriteltäisiin, esimerkiksi pie eli ympyrädiagrammiksi, kaavio näkyisi tyhjänä verkkosivulla.



Kuva 8. Kuva näyttää kaavion type: 'line' -koodilla



Kuva 9. Kuva näyttää kaavion type: 'radar' -koodilla



Kuva 10. Kuva näyttää kaavion type: 'area' -koodilla

4.3.2 Ulkonäön muokkaaminen

ApexCharts kaavioiden ulkonäön muokkaaminen käy helposti tutustumalla ApexCharts sivujen dokumenttiin. Mikä tekee ApexCharts:sta hyvän JavaScript-kirjaston on, että kaavion muokkaamiseen löytyy useita tapoja. Opinnäytetyössä käydään läpi vain osaa näistä muokkaustavoista.

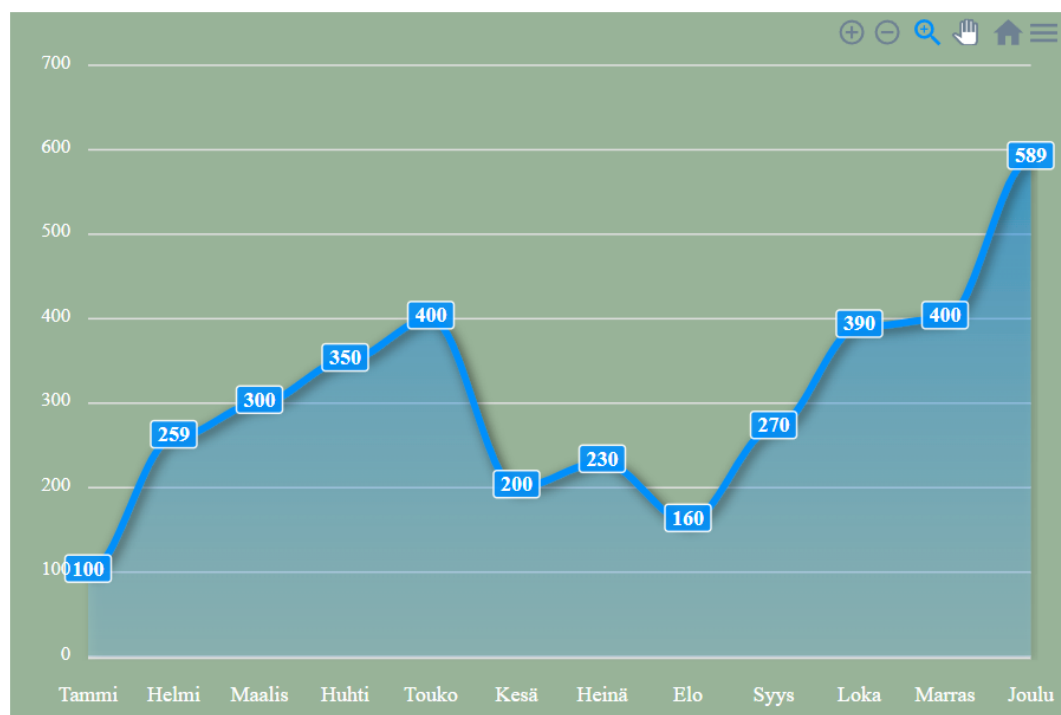
Kaavion muokkaaminen tapahtuu koodin options-osiossa, jonka sulkujen sisälle kirjataan kaikki, miltä kaavion halutaan näyttävän. Options-osion sisällä on chart-osio, jossa muokataan koko ApexCharts kaavion ulkonäköä, kuten taustaväriä, pituutta ja leveyttä, fonttia, fontin väriä tai varjostusta.

ApexCharts JavaScript-koodin height ja width -osiot merkkäavat kaavion pituutta ja korkeutta. Background-osion lisääminen muuttaa taustaväriä. Haluttu väri kannattaa kirjoittaa sen värikoodina. Värikoodit voi löytää, esimerkiksi W3Schools sivujen Color Pickerillä. (W3Schools 2022 A.)

Kaavion fontit määritellään fontFamily ja foreColor -osioissa. Ensimmäisessä muutetaan fontin tyyliä, ja seuraavassa fontin väriä, mikä muuttuu valkoiseksi kuvan 11 tavalla. Varjostus määräytyy ottamalla dropShadow menetelmän käyttöön,

joka tapahtuu kirjoittamalla sen sulkujen sisään ”enabled: true”, mikä tarkoittaa, että sen käyttö merkitään todeksi. Kappaleen alla nähdään tähän liittyvä koodi.

```
chart: {
  type: 'area',
  height: 400,
  width: 600,
  background: '#98b398',
  fontFamily: 'Times New Roman',
  foreColor: '#ffffff',
  dropShadow: {
    enabled: true,
    blur: 3,
    opacity: 0.3
  }
}
```



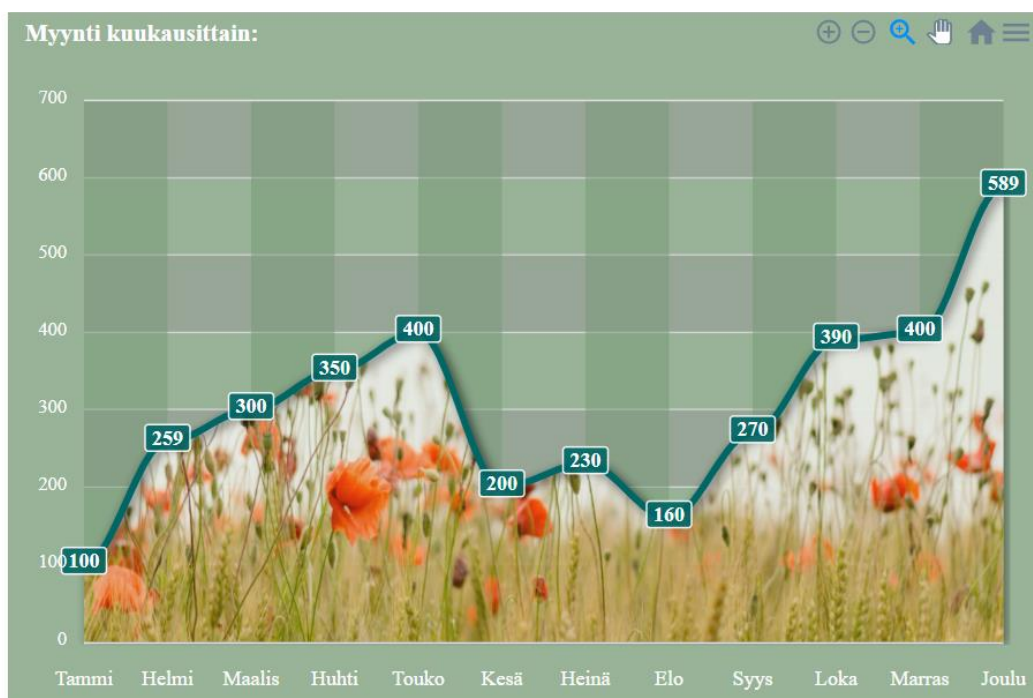
Kuva 11. ApexCharts JavaScript-koodin chart-osion muokkaamisen tulokset.

Kaavion datan väriä voidaan muuttaa kirjoittamalla colors-koodi options-osion sulkujen sisään. Värikoodin määrittäminen muuttaa dataviivan väriä kaviolla, mikä voidaan nähdä vertaamalla kuvaa 11 ja 12. Esimerkki koodissa dataviiva muutetaan vihreän väriseksi.

Kaavion ruudukon väriä muutetaan laittamalla värikoodit grid-nimisen koodin sulkujen sisään. Ruudukon rivien ja sarakkeiden väritystä voidaan määritellä erikseen row- ja column-osioissa, mikä esiintyy alla olevassa koodissa. Kaavio voidaan nimetä lisäämällä teksti title-osion sisälle, mikä on tässä tapauksessa ”Myynti kuukausittain”.

Fill-osio määrittää, miten kaavion data täytetään, esimerkiksi väreillä. Fill-osion sulkujen sisään on määriteltävä täytön type eli tyyppi, mikä tässä tapauksessa on image eli kuva. Type-osion jälkeen määritellään kuvan lähde, mikä lisätään src-osioon. Kuvassa 12 näytetään, kuinka alla esiintyvä koodi renderöityy verkkosivulla.

```
colors: ['#006666'],
grid: {
  row: {
    colors: ['#999999', 'transparent']
  },
  column: {
    colors: ['#759975', 'transparent'],
  }
},
title: {
  text: 'Myynti kuukausittain: '
},
fill: {
  type: 'image',
  image: {
    src: ['kukkia.jpg'],
    height: 400,
    width: 600
  }
}
```



Kuva 12. Kaavio, jossa on muokattu dataviivan väriä, ruudukon värejä ja kuvan lisäämistä dataan. Kaavion "kukkia.jpg"-kuva on otettu Unsplash-sivustolta. (Henry Be 2016.)

4.3.3 Muita kaavion muokkaustapoja

Fill-osiossa voidaan koodata myös funktio, jolla voidaan määrittää tiedon mukaan, millä värillä tai väreillä kaavio esiintyy. Esimerkki koodissa kaaviotyyppi muutetaan aluekaaviosta pylväskaavion, koska se soveltuu paremmin kyseiseen toteutukseen.

Alla esiintyvän koodin mukaan, fill-osion sulkujen sisään lisätään if-else-funktio, millä määritellään mitä tapahtuu tiettyjen arvojen mukaan tai kun näitä arvoja ei ole. Kaaviolle määrätään if-else-funktio, missä annetaan pylvään väriksi punainen, jos myynti on pienempi tai yhtä suuri kuin 200. If-else-funktiolla muokatun kaavion voi nähdä kuvassa 13.

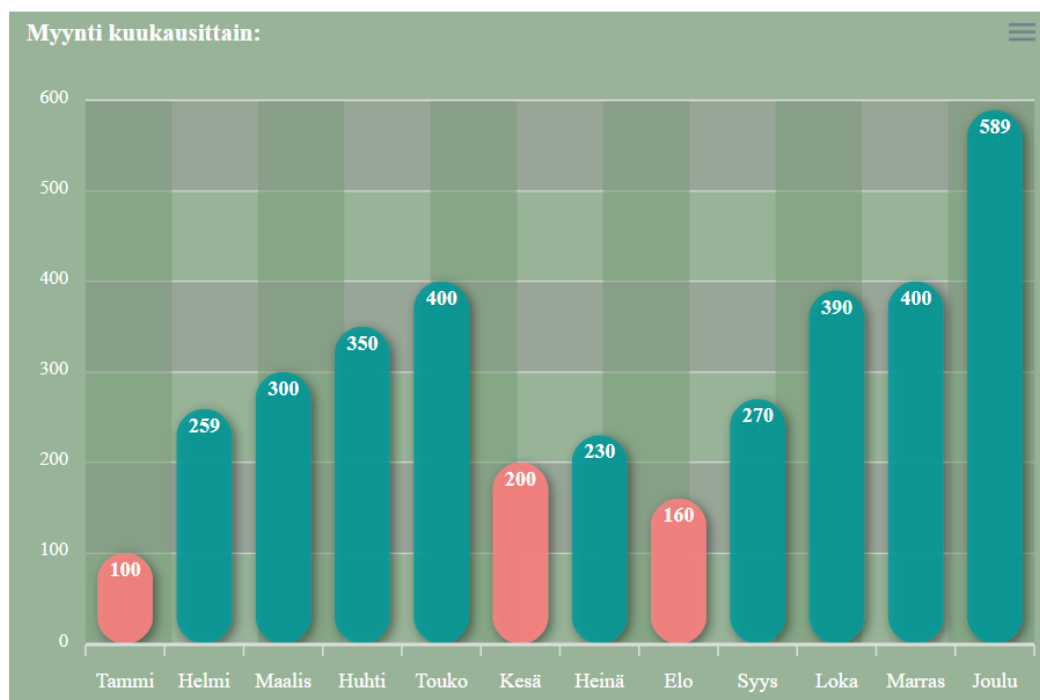
```
fill: {
  colors: [function({ value, seriesIndex, w }) {
    if(value < 200 || value == 200) {
      return '#ff8080'
```

```
        } else {  
            return '#009999'  
        }  
    }]  
},  
plotOptions: {  
    bar: {  
        borderRadius: 15,  
        dataLabels: {  
            position: 'top'  
        }  
    }  
}  
}
```

Pylväskaavion pylväiden muotoa voidaan, muokata ApexCharts:ssa "plotOptions"-nimisellä menetelmällä. Sen sulkujen sisään laitetaan bar-osio, johon listataan pylväisiin haluttavat toiminnot.

Pylväiden kulmien sädettä voidaan muokata borderRadius menetelmällä, jolloin kulmat voidaan pyöristää, kuten yllä olevassa koodissa on tehty. Tämän toiminnon lisäämisen vaikutuksen voi nähdä kuvassa 13.

DataLables menetelmällä voidaan muokata datan otsikointia, joka näyttää kyseisen datan arvon. Yllä olevassa koodissa datan arvot on siirretty pylväskaavion yläosaan. DataLables menetelmän lisäämistä koodiin voidaan analysoida vertaamalla kuvaa 7, jossa datan arvo lukee pylvään keskiosassa, ja kuvaa 13, jossa datan arvo on pylvään yläosassa.



Kuva 13. Kaavion muokaus if-else-funktiolla ja pylväiden arvojen sijoittaminen ylös.

4.4 Kaavion datan haku JSON-tiedostosta

Kappaleessa selitetään, kuinka kaavio voidaan tehdä silloin, kun käytössä on JSON-tiedosto. Tiedostossa käytetään samoja data-arvoja kuin edellisessä koodissa. Ensiksi on tehtävä JSON-tiedosto, mikä näkyy kuvassa 14. JSON-tiedoston haku toteutetaan getJSON menetelmällä, samalla tavalla miten esitetään opinnäytetyön luvun 3 kappaleessa 3.4.

HTML-koodiin on lisättävä koodi, mikä mahdollistaa jQuery:n käytön, joka lisätään koodin head-osioon. Ilman sitä tiedon haku ei toimi jQuery:lla. Alla näkyvän koodin voi löytää, esimerkiksi W3Schools verkkosivuilta (W3Schools 2022 B.) tai Google Developers verkkosivuilta (Google Developers 2022.)

```
<script src="https://ajax.goo-
leapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
```

Tämän jälkeen on tyhjennettävä JavaScript-koodin series-osio ja poistettava xaxis-osuus, sillä niitä ei tarvita, kun tiedot haetaan JSON-tiedostosta. Muuten käytetään samaa koodia, joka tehtiin luvun edellisissä kappaleissa. Ainut asia, joka muuttuu, on lisätä ApexCharts JavaScript-koodin loppuun, script-koodin sisään, AJAX-pyyttö jQuery getJSON menetelmää käyttäen.

Alla olevassa koodissa getJSON hakee "data.json" -nimisen JSON-tiedoston, joka esiintyy kuvassa 14. ApexCharts:n updateSeries menetelmä päivittää series-osion haetun tiedoston mukaan. Samassa menetelmässä määritellään kaavion nimi ja response, mikä vastaa getJSON fuction hakemalla JSON-tiedoston datan kaavioon.

```
$.getJSON("data.json", function(response) {  
    chart.updateSeries([{  
        name: 'Myynti',  
        data: response  
    }])  
});
```

Kyseinen koodi on sama kuin opinnäytetyön luvun 3 kappaleessa 3.4, mutta url-tiedostoa ei tarvita, koska tietoa ei haeta tietokannasta tai web-palvelimelta. Tiedon tulisi renderöityä verkkosivustolle saman näköiseksi kuin kuvassa 13, sillä kuvan ulkonäköä ei muutettu, vaan datan haun tyyliä.

```
[
  {
    "x": "Tammi",
    "y": 100
  },
  {
    "x": "Helmi",
    "y": 259
  },
  {
    "x": "Maalis",
    "y": 300
  },
  {
    "x": "Huhti",
    "y": 350
  },
  {
    "x": "Touko",
    "y": 400
  },
  {
    "x": "Kesä",
    "y": 200
  },
  {
    "x": "Heinä",
    "y": 230
  },
  {
    "x": "Elo",
    "y": 160
  },
  {
    "x": "Syys",
    "y": 270
  },
  {
    "x": "Loka",
    "y": 390
  },
  {
    "x": "Marras",
    "y": 400
  },
  {
    "x": "Joulu",
    "y": 589
  }
]
```

Kuva 14. Kuvakaappaus Visual Studio Code:lla tehdystä JSON-tiedostosta.

5 APEXCHARTS-KIRJASTON KÄYTTÖ DATAN VISUALISOINNISSA

Luvussa pohditaan ApexChartsin hyödyllisyyttä JavaScript-kirjastona, ja sen käyttöä datan visualisoinnissa. Käsitellään, kuinka käytännölliseksi kyseinen JavaScript-kirjasto osoittautuu, sekä onko se helppo ottaa käyttöön. Lopuksi kirjaston tutkimuksesta tehdään yhteenveto.

5.1 ApexCharts:n hyödyt datan visualisoinnin kirjastona

Tutkimuksen ideana oli etsiä datan visualisoimiseen soveltuva JavaScript-kirjasto. Työhön löydettiin JavaScript-kirjasto nimeltä ApexCharts. Vaikka se osoittautui uudeksi kirjastoksi, sen dokumentoinnin suuri määrä pisti sen etusijalle muista ehdokkaista.

ApexCharts tarjoaa laajan määrän kaaviotyyppejä, joista löytyy kaksi tai useampi malli kaavio. Kaavioiden malli esimerkkejä voidaan hyödyntää ottamalla esimerkiksi kyseisestä koodista ja, sitten muokata se omaan käyttöön sopivaksi. ApexCharts verkkosivujen esimerkkien avulla kirjaston käyttöönotto on helppoa.

Kaavion teko on helppoa käyttämällä ApexCharts verkkosivujen dokumentin kaaviomalleja. Kun haluttu kaavio on otettu käyttöön sitä, voidaan muokata. ApexCharts sisältää suuren määrän erilaisia menetelmiä muokata kaaviota. Kaavio voidaan, esimerkiksi muokata yrityksen värien mukaan ja käyttää sitä verkkosivuilla.

Välillä kaaviomallien haku erilaisiin tiedostoihin voi olla työlästä, sillä kaikki kaavioiden koodit eivät sovellu kaikkiin datoihin. Kuten luvussa 4, käytettyyn dataan soveltui vain ApexCharts:n 4 kaaviotyyppiä. Tämä voi tulla ilmi etenkin, kun täytyy hakea tietoa, vaikka tietokannasta.

ApexCharts:n ominaisuudet tuovat datan visualisointiin erilaisia työkaluja, joita voidaan hyödyntää. Ominaisuuksiin kuuluu kaavioiden skaalautuminen automaattisesti pöytäkone-, tabletti-, ja mobiilinäyttöihin, mikä on nykyaikana tärkeää verk-

kosivuilla. Kaavion interaktiivisuus työkaluilla voidaan, esimerkiksi zoomata ja ladata kaavioita PNG-muodossa, ilman erillistä koodausta. ApexCharts-kirjasto auttaa vähentämään ohjelmoinnin työtä, koska tietyt ominaisuuden tulevat kirjaston mukana.

5.2 Yhteenveto

ApexCharts-kirjasto soveltuu datan visualisointiin, koska se sisältää monia eri kaaviomalleja ja sen moderni ulkonäkö, jota pystyy muokkaamaan useilla tavoilla, toteuttaa miellyttävän näköisiä kaavioita. ApexCharts sisältää hyödyllisiä ominaisuuksia, jolloin kehittäjän ei tarvitse koodata niitä itse kirjastoon.

Sen käyttöönotto todettiin helpoksi ApexCharts:n oman dokumentoinnin avulla, jota kaiken tasoiset ohjelmoijat pystyvät hyödyntämään. Hieman vaikeammat kaaviomallit soveltuvat taas enemmän kehittäjän käyttöön, mutta peruskoodaamis taidon osaava voi myös hyötyä niistä.

Kokonaisuudessa ApexCharts on täydellinen datan visualisointiin tarkoitettu kirjasto yritysten ja henkilökohtaisten projektien verkkosivuille. ApexCharts on uusi JavaScript-kirjasto, joten voi olettaa, että se kasvaa entisestään. Tämä takaa sen, että kirjaston ominaisuudet ja JavaScript-koodi ovat aina nykyajassa.

6 JOHTOPÄÄTÖKSET

Opinnäytetyön tehtävä oli tutkia datan visualisointiin liittyviä JavaScript-kirjastoja ja, kuinka kehittäjät, yritykset ja asiakkaat hyötyvät niistä. Tarkoitus oli valita yksi näistä kirjastoista ja ottaa se lähempään tarkkailuun ja selvittää, miten kirjastoa käytetään.

Tavoitteena oli löytää laaja, helppo käyttöinen, avoimen lähdekoodin kirjasto, joista ApexCharts-kirjasto sopi parhaiten. Sen laaja dokumentointi antoi riittävästi lähteitä kirjaston ominaisuuksista ja kaavioista, joiden avulla saatiin hyvä käsitys sen käyttö mahdollisuuksista.

ApexCharts-kirjaston dokumenttia käyttäen tehty esimerkki tapaus datan visualisoinnista toteutettiin yleisimmillä kaaviomalleilla. Esimerkissä olisi voinut käyttää, monimutkaisempaa kaaviomallia, mutta tapauksen ideana oli esittää, kuinka monta kaaviotyyppiä voidaan tehdä samasta datasta. Koodia päätettiin esitellä vaiheittain useilla koodikirjoituksilla sekä kuvilla, joka soveltui hyvin datan visualisoinnin kuvaamiseen JavaScript-kirjastolla. Tarkoituksena oli tehdä datan visualisoinnin esimerkki, jota muutkin kuin kehittäjät voisivat hyödyntää.

Lähteet koostuvat lähinnä JavaScript-kirjastojen omista sivuista sekä blogi sivuista, joita käytettiin erilaisten kirjastojen etsimiseen ja vertailuun. Blogikirjoitukset voivat olla osittain asiantuntijan omia mielipiteitä, mikä voi vaikuttaa lähteen luotavuuteen. Mutta blogi lähteiden avulla saatiin kerättyä suuri määrä erilaisia kirjastoja, joista pystyttiin valitsemaan tutkimukseen sopiva kirjasto.

Tutkimuksessa tekijä tutustui datan visualisointiin JavaScript-kirjastolla, ja sai opitua lisää aiheesta. Datan visualisointi JavaScript-kirjastoa käyttäen oli uusi aihe tekijälle, joten opinnäytetyö oli oppimis- ja tutkimusprojekti. Eniten työssä selvitetiin ApexCharts-kirjaston käyttöä ja toteutusta.

Opinnäytetyössä opittiin asioita JavaScript-kirjastoista, datan visualisoinnista sekä ApexCharts:n käytöstä. Työn toivotaan antavan hyvän käsityksen siitä, mitä nämä käsitteet tarkoittavat ja, kuinka niitä voi hyödyntää erilaisissa projekteissa.

LÄHTEET

ApexCharts 2022 A. About us. Viitattu 17.11.2022. <https://apexcharts.com/about-us/>

ApexCharts 2022 B. Bubble Charts. Viitattu 17.11.2022. <https://apexcharts.com/javascript-chart-demos/bubble-charts/>

ApexCharts 2022 C. Dashboards/Modern. Viitattu 17.11.2022. <https://apexcharts.com/javascript-chart-demos/dashboards/modern/>

ApexCharts 2022 D. Docs. Viitattu 17.11.2022. <https://apexcharts.com/docs/installation/>

ApexCharts 2022 E. Dynamic Loaded Chart. Viitattu 17.11.2022. <https://apexcharts.com/javascript-chart-demos/column-charts/dynamic-loaded-chart/>

ApexCharts 2022 F. Features. Viitattu 17.11.2022. <https://apexcharts.com/features/>

ApexCharts 2022 G. Update Charts from JSON API & AJAX. Viitattu 17.11.2022. <https://apexcharts.com/docs/update-charts-from-json-api-ajax/>

Chart.js 2022 A. Horizontal Bar Chart. Viitattu 17.11.2022. <https://www.chartjs.org/docs/latest/samples/bar/horizontal.html>

Chart.js 2022 B. Latest. Viitattu 17.11.2022. <https://www.chartjs.org/docs/latest/>

D3 Data-Driven Documents 2021 A. Viitattu 17.11.2022. <https://d3js.org/>

D3 Data-Driven Documents 2021 B. Bar Chart Race. Viitattu 17.11.2022. <https://observablehq.com/@d3/bar-chart-race>

Google Developers 2022. Viitattu 17.11.2022. Hosted Libraries. <https://developers.google.com/speed/libraries>

Khan Academy 2022. What's a JS library? Viitattu 17.11.2022. <https://www.khanacademy.org/computing/computer-programming/html-css-js/using-js-libraries-in-your-webpage/a/whats-a-js-library>

Majorek J. 2021. Monterail sp. z o.o. 19 Best JavaScript Data Visualization Libraries. Viitattu 17.11.2022. <https://www.monterail.com/blog/javascript-libraries-data-visualization>

Npm 2022. About npm. Viitattu 17.11.2022. <https://docs.npmjs.com/about-npm>

Pengon 2022. Tiedon visualisoinnin hyödyt. Viitattu 17.11.2022
<https://blogi.pengon.fi/arkisto/tiedon-visualisointi-hyodyt>

Henry Be 2016. Unsplash 2022. Viitattu 17.11.2022. <https://unsplash.com/photos/licyiaPYGGI>

W3Schools 2022 A. HTML Color Picker. Viitattu 17.11.2022.
https://www.w3schools.com/colors/colors_picker.asp

W3Schools 2022 B. jQuery Get Started. Viitattu 17.11.2022.
https://www.w3schools.com/jquery/jquery_get_started.asp

XAMK. Data ja sen visualisointi. Viitattu 17.11.2022. <https://www.xamk.fi/data-opas-visualisointi/>