URI: urn:nbn:se:bth-17952

Google AMP and what it can do for mobile applications in terms of rendering speed and user-experience

Niklas Andersson
Oscar Bäck

June 3, 2019

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the bachelor degree in Software Engineering. The thesis is equivalent to 10 weeks of full time studies.

The authors declare that they are the sole authors of this thesis and that they have not used any sources other than those listed in the bibliography and identified as references. They further declare that they have not submitted this thesis at any other institution to obtain a degree.

Contact Information:
Authors:

Niklas Andersson	me.niklas.andersson@outlook.com
Oscar Bäck	oscarbackk@gmail.com

External Advisor:

Simon Nord, Prisjakt	simon.nord@prisjakt.nu

University Advisor:

Michel Nass	michel.nass@bth.se

Faculty of Computing	Internet: www.bth.se
Blekinge Institute of Technology	Phone: +46 455 38 50 00
SE-371 79 Karlskrona, Sweden	Fax: +46 455 38 50 57

1 Abstract

On today's web, a web page needs to load fast and have a great user experience in order to be successful. The faster the better. A server side rendered web page can have a prominent initial load speed while a client side rendered web page will have a great interactive user experience. When combining the two, some users with a bad internet connection or a slow device could receive a poor user experience. A new technology called Amplified Mobile Pages (AMP) was created by Google to help combat this issue.

The authors of this report gives an answer to if Google AMP could maintain the user experience while still contributing with a fast initial load speed for applications. To do this, we conducted an experiment through creating a Google AMP application and compared it to another application using a different rendering engine called Pug. We have also measured the metrics: page load time, speed index and application size between the two applications. To fully understand the AMP format, the authors conducted a literature study, to further strengthen their findings.

Google AMP is a great technology but it can still grow to become better. The format could increase the speed of a website, however the same result could be achieved without AMP if focus was set on writing a fast application. From the experiment, the authors concluded that Google AMP takes a great time to learn because of its own version of JavaScript through modules. The format also has a different structure than standard HTML. From the tests, a smaller application does not favor the implementation of AMP. We did however derive from the experiment and the literature study that bigger applications could benefit from the perks of AMP and could therefor be a potential choice for old and new applications.

Key Words: Google AMP, User Experience, Performance, Mobile Web Pages, Speed

Contents

2 Introduction

The basis of this paper is to evaluate some of the peaks and valleys of Google AMP (Amplified Mobile Pages). Google AMP is a web format using a newly created standard by Google [26]. One of the goals of the format is to amplify speed on web pages but foremost on mobile devices. Another goal which AMP strives to increase is the conversion rate through ads, which is the amount of users in percentage doing a certain desirable action [37]. Google AMP also provides a simple way to analyze your web page's statistics through available modules [26]. This paper contains a study where an AMP application is measured on certain aspects and then used in a comparison to a web application made with Pug. Pug is a simple rendering engine which makes it easier to create HTML structure and functionality [45]. The authors chose Pug as an engine because it was the default rendering engine for Express [17]. A study was made containing the complexity of creating a Google AMP web page for a developer, compared to a mobile-friendly web page.

2.1 Background

Prisjakt is a company which mainly lets their users compare price between products. They also provide price tracking and product comparison [42]. Around 2017 Prisjakt decided to swap the technology they were using to render their web page. From a static rendered stateless HTML page to a React web application which is rendered server side. When the website has rendered its initial load, the site is then being re-rendered with the help of JavaScript. They made this change since they wanted to make their product more interactive and to increase the overall functionality of the website. Even though the changes were beneficial for some users, other users were punished. This was because a big chunk of data and run time had to be downloaded the first time a user entered the site. In the early stages of the implementation, Prisjakt discovered that this was a large drawback for some of their users [52][10].

One of the more common use cases where this problem occurred is when a user searches for a product on Google. When the user navigates from Google to Prisjakt, they will be directed to Prisjakt's price list. The user then has to navigate themselves onto one of the stores which has got the product available. Since the content of the web site contains multiple images, and a lot of data has to be gathered, the product page might take several seconds to load. This is because of the chosen client-side rendering technique and a lot of data having to be downloaded at the initial load. If the user has either a slow or weak internet connection or a slow device, the site might be perceived as sub par.
Google AMP could be the solution that Prisjakt needs in order to achieve the best possible experience for all users. This includes the website being interactive as well as having a great initial load speed.

2.2 Purpose

The main purpose of this paper was to find out if Google AMP could improve the metrics: page load time, application size and speed index. In order to give Prisjakt the most accurate answer, their API and structure was used in the experiments performed.

Page load time is the time it takes for the entire content on the page to load. Application size is the size of the application. Speed index measures the average time until visible content is displayed on the site and will be analyzed with the help of Google Lighthouse, a tool for improving web pages [24][36][25].

One of the motives for the paper is to understand the level of difficulty for the technology to be learned and applied from a developers perspective. The authors conducted an experiment to provide an answer. This generated an idea revolving around whether the technique would be profitable for a product owner or a company to implement. Could the implementation be more or less profitable because of the long or short workload? Furthermore, the authors examined previously conducted experiments made by companies and developers. This was done in order to compare their results against the results of other experiments conducted.

This result was not only of importance to Prisjakt but also to all companies who face the same problem. A web page has to be interactive but also have a fast loading speed on all platforms, especially on mobile devices in order to be successful. If this technology shows a positive outcome, more product owners might want to embrace Google AMP as an addition to their product.

2.3 Contribution

The authors of this paper will contribute to a more well known understanding of Google AMP. More so, how the workload is for a developer to develop a web application using Google AMP. The paper also supplies a guideline for when Google AMP should be used and in which use cases.

3 Goal

The authors preliminary goals of this study was to find out more about what Google AMP is and if it could help Prisjakt in a positive way. This is achieved through raising the interactivity while still maintaining a good initial load speed for users on mobile devices. The authors set out to find what improvements AMP provided and what the underlying reason for those improvements were. We did also expose ourselves to the format through creating applications with AMP in order to further understand it.

3.1 Focus

The focus of the study and the report is how Google AMP works consuming an API compared to a normal website. The focus will also be to find studies prior to this paper and compare it with the experiments conducted in this report. This will be done in order to further strengthen the theories in the paper.

3.2 Research Questions

1. How is the user experience offered by Google AMP contrary to an ordinary website?

2. How does a Google AMP based website compare to an ordinary website in terms of development?

3. Will Google AMP offer any improvements compared to an ordinary website? The following aspects will be measured:

   (a) Page load time

   (b) Speed index

   (c) Application size

4 Method

To answer the research questions, three different studies were conducted.

RQ1 was answered by orchestrating a literature study. Within this study, the information gathered includes studies prior to this report. It includes opinions and thoughts from developers who have used Google AMP as a solution for creating fast web pages. The goal of this study was to compare the found information with the information we accumulated throughout our empirical studies. The authors chose this approach in order to obtain an understanding regarding the thoughts from developers' on the Google AMP format as a whole. From the collected discussions, the authors could then compile an overview from the mindsets and opinions regarding the technique.

To answer RQ2, the paper contains an experiment conducted by both authors. The experiment consists of four applications being created. Two web pages created using Pug and two created using the new Google AMP format [45]. To better understand the language and difficulty difference between the two technologies, both of the authors created one application using Pug and one using Google AMP.

During the development of these web pages, the authors observed and noted their work flow. This study was made in order for the authors to get an impression of the format themselves. From that study, they could later compile the data gathered describing the difficulty of implementing AMP. With the information, they could compare their findings with previously found results.

Furthermore, RQ3 was answered by comparing the Google AMP application and the Pug application produced in the study created to answer RQ2. An experiment was conducted to measure and compare the metrics: page load time, speed index and application size [36][25]. The authors observed the metrics given from the study and analyzed them, thus giving them information about which one of the applications that were the best.

The authors chose this method in order to analyze whether the speed of a Google AMP application would triumph over the speed of a Pug application. We chose not to analyze other metrics such as conversion rate since measuring that statistic would require the Google AMP application to be published in a production environment, and in our case, Prisjakt's production environment. Though the release of the AMP application on Prisjakt's domain was the original aim of the study, the authors had to narrow down the metrics studied due to time restraints on the paper. If the AMP pages would have been published on Prisjakt's servers, the statistics collected would not be valid enough due to the data not being able to be recorded for more than two weeks. In a discussion with Prisjakt, they explained to us that the data gathered during such a short duration of time would not be reliant enough to be compared with their current statistics.

4.1 Literature study

In order to gather the literature needed for the paper, the authors began by searching for scientific and published papers in different databases. The databases that were browsed through were Diva, Scopus, Google Scholar and IEEE. The databases was chosen since they have technical papers published. Most of the searches originated from Blekinge Institute of Technology's library. We began our searches from Blekinge Institute of Technology's library because the information published on these platforms is mostly peer reviewed and published as scientific papers and studies. The searches was made on 18th of April, 2019. The keyword that was used was *Google AMP* in all fields possible such as title, abstract etc. The restrictions the authors set were that the papers had to be a published study or paper and not published before 2015. The result of the search were 2650 hits. After sorting them out, removing articles and news publications and only allowing books, publications, encyclopedia, book chapters and essays, only 48 hits were left. From these hits, one book was found regarding how Google AMP works and how it was implemented, which would not give the authors any information regarding the research questions [39].

The rest of the hits were not relevant at all. On Google Scholar, using the same search restrictions, a study was found among 99 other hits [32]. The study contained information about how to use AMP as well as tests made with some of the metrics that the authors use in their tests for the applications. It also contained assumptions and opinions on AMP's place on the web. This was a great find since it gave the authors a great baseline for what their report should contain and what was already established. The authors could also cross reference the reports results with their results to strengthen the answer for their research questions. The rest of the hits were articles or content that were not relevant to the report. On Diva, Scopus and IEEE no matches were found concerning Google AMP at all. We changed our keyword to *Google AMP study* however the search results were still the same as from the first keyword.

The authors became aware that the databases used would not suffice for gathering information. We understood that since Google AMP is a new technology, few scientific papers or studies have had the time to be written favoring the authors goal of the report. Therefore we chose to loosen our restrictions and lower the criteria for valid references. We used the same keywords used before but allowed ourselves to look for information in articles found on the web and studies which were made by companies whom created studies as their field of work. The information gathered from articles were cross references with other articles to make sure the information was valid. Here, the authors found several studies. We found several articles where known developers described their impression of Google AMP as well as more thoughts from other developers. The authors chose these references only if the information found corresponded with information found in other articles from different sources. To find information regarding how AMP worked as a format, they navigated to the core of the information, Google AMP's website. There, they found information which told them how to create an AMP application. The exploration for more information continued by using the keyword *Google AMP good or bad*. A lot of articles where found discussing the thoughts of developers regarding the AMP format for the web. To complement the articles found, the authors used the keywords *Google AMP advantages* and *Google AMP disadvantages*.

The authors began by defining a reliable source as a published paper or study from a peer review database. Thereafter when not enough information could be gathered, they lowered their definition for a reliable source to web pages and articles with information which could be strengthened from different sources. Though the sources were less credible, the authors came to establish that their paper would still be of great interest since they themselves produced studies to strengthen the statements found through the literature study.

4.2 Observation and creation

In this chapter, the authors describes the approach for the empirical studies ushered. This is done in order for the reader to be able to recreate and reconstruct the experiment.

4.2.1 RQ2: How does a Google AMP based website compare to an ordinary website in terms of development?

To answer this, a study was conducted where the authors, independently, created one AMP application and one Pug application each. One began with the development of the Google AMP application whilst the other began with the Pug application. The choice of having a more well-known technology, such as Pug, opposed to AMP was to see if a new technology could stand a chance against a more well established technique. No restrictions were set when creating the Google AMP application. This was because the authors did not want any prior knowledge to AMP before developing the application. No communication was made between the authors on how they chose to implement Google AMP.

To better specify the functionality of the website, a discussion was held with Prisjakt regarding which content on their current product page that were relevant for the AMP page. Prisjakt themselves explained to the authors that they had several metrics they would have liked to see improvement on in the current system. Therefor, they wanted the following feature to be researched and implemented on the website:

- Display the product image

- Display the core properties of the product
- Display the following information about the retailer who sell the product
  - Retailer Image - logotype of the retailer
  - Stock - if the item is in stock, and if not, when and if it arrives
  - Price - both excluding and including shipping costs
  - External URL - link to the product page on the retailer website

Express was chosen as the foundation of the websites on Prisjakt's recommendation and with the help of Prisjakt's API, both authors worked toward mimicking Prisjakt's current product pages visuals [19].

The authors followed the specifications provided by Prisjakt while taking notes during the process until both websites had achieved the functionality and visuals needed. Thereafter we began creating a new page, doing the other application instead, which we previously had not done before. While most modules used were the same between the two Pug pages, there was a difference in the rendering library used in the AMP page. One of the two libraries were mustacheJS and the other one was Embeded JavaScript (ejs) [31][30].

Google AMP site	
Module (in common)	Version
es6-promise	^4.2.6
express	^4.15.2
isomorphic-fetch	^2.2.1
Module (difference)	Version
ejs	^2.6.1
Pug	^3.0.1

Table 1: Google AMP site modules used

Pug site	
Module	Version
body-parser	^1.18.3
cookie-parser	~1.4.3
debug	~2.6.3
es6-promise	^4.2.6
express	^4.16.4
isomorphic-fetch	^2.2.1
morgan	^1.9.1
pug	~2.0.0-beta11
serve-favicon	~2.4.2

Table 2: Pug site modules used

The authors took notes on how similar Google AMP was to already known and established web technologies, such as regular HTML and the time spent developing the site. Notes were taken on how much time was spent browsing documentation and forums, looking for assistance with problems that occurred. The reason for making these choices were that if a new technology shared the same logic and flow as previous technologies that a developer has used, it would be easier for the developer to understand and work with.

4.2.2 RQ3: Will Google AMP offer any improvements compared to an ordinary website?

To measure this, the authors analyzed the web pages created in regards to research question two, using the developer tools in Firefox, Chrome and Safari on different devices. To get a more realistic result, the application was published and hosted on a developer cloud called Digital Ocean [38]. The server chosen was located in Frankfurt which was one of the three server-centers in Europe available and close to the authors. The following aspects was analyzed:

- Page load time - the total time for the website to fully load all content
- Application size - the total size of the application when fully loaded
- Speed index - the time it takes for the website to display its first visuals

Page load time and speed index heavily impact user experience on the internet, as they both show important time stamps when the user can begin navigating the application. Application size is important because of the data plans used by mobile carriers around the world. A smaller application is often faster to load since less data needs to be transferred.

The first two metrics, page load time and application size, could be found in the developer tools in each browser tested. Having the time frame in mind, the authors decided to do five refreshes on each site without cache and took note of both the application size and the page load time seen in the developer tools. If the page load time deviated by more than 1000ms from the other results, the value was excluded and a new value was fetched. The highest value was peaking at above 3 seconds and we decided that a deviation of one third of the loading speed would be enough to be tagged as a wrong reading, as well as the largest deviation from average in the data gathered being 398ms. To measure the speed index of both applications, the authors chose to use an external application called Lighthouse [24].

Appendix A contains the devices used for the study. The units were chosen because they mirror devices used on a daily basis [50]. Each test was performed five times per browser and the average value of the five tests was used in the study. A summary of each test conducted can be found in appendix B.

4.3 Scope

This paper contains an empirical study, examining the difficulty of creating a Google AMP application. The experiment more specifically accommodates a JSON fetch from an API and placing the output according to Prisjakt's liking using Google AMP and Express. The output from the JSON fetch in question is the product the client has searched for. It also contains all the retailers that has the product in store as well as some technical description. In order to compare the application with like-worthy data, the authors additionally created another application using almost the same techniques, excluding Google AMP and using Pug as a rendering engine instead [35][45]. In order to get the best possible outcome from a small scale study, both of the authors conducted the same experiment. To make it even more trustworthy, one of the authors began with the Google AMP application and the other author with the Pug application. When they were done, they created the other remaining application. This experiment only covered the basics and was not aimed to be good looking, but the focus was on fetching the information from Prisjakt's API and render it with said technologies. Furthermore the scope covers another study comparing the created applications in the first study. The applications were compared with the help of the following metrics: page load time, speed index and application size.

Lastly, the scope of the literature study reviews the thoughts of companies and developers on Google AMP as a format and whether it is a preferable choice of technique. The literature study will also contain data gathered from earlier studies conducted. The data gained through the previous studies was compared with the information retrieved from the authors empirical study.

The study only covers the discoveries and features from Google AMP used in the experiments. Other features which Google AMP provides will not be included. The report only includes the aspects measured in the research questions. Other metrics will not be included.

5 Literature Review

This literature review will give an introduction to Google AMP but primarily focus on answering RQ1. It will also provide previous studies made which will be compared to the authors study covered in RQ3. Furthermore, a discussion will be held regarding the thoughts of Google AMP's impact on the web.

The goal of the review is to find other developers/companies who have implemented Google AMP and to record their results and thoughts in a knit review. The goal will also be to find prior studies which the authors can compare to their own.

5.1 Google AMP

This was said in a meeting with Chris Coyier, founder of CSS-Tricks, regarding Google AMP [12]:

> *Obviously, it's a big deal, I just haven't gotten there yet. Perhaps I will one day when it's clear I need to for a project [13].*

Google AMP is an open-source project which Google is in charge of. One of the main features of the format is the web components. AMP has replaced some of the standard HTML5 tags to their own version. An example of this would be the `img` tag that in AMP has changed to an `amp-img` tag.

This is an example of how an `amp-img` tag could look like [22].

```
<amp-img alt="A view of the sea"
  src="images/smiley.jpg"
  width="900"
  height="675"
  layout="responsive">
</amp-img>
```

This is an example of how an `img` tag using HTML5 standards could look like: [53]

```
<img
    src="smiley.jpg"
    alt="Smiley face"
    height="42"
    width="42"
>
```

Compared to its predecessor, the AMP tag is almost identical. In AMP, images must have a set width and height. This is crucial since AMP wants to know the layout of the page before it renders it. The reason is because AMP does not want any jumping elements on the page and render it instantly [4].

5.2 JavaScript and CSS

Another big part of Google AMP is its rules of coding. No extensive use of JavaScript is allowed except for limited amount in `style` tags. The same goes for CSS and the style is limited to a maximum size of 50 kilobytes. This means that all of the styling and inter-activeness has to be placed in the same file as in-line code. As AMP describes it:

> *JavaScript is powerful, it can modify just about every aspect of the page, but it can also block DOM construction and delay page rendering [4].*

This means that AMP only has JavaScript that is asynchronous which makes sure that data can be loaded during the already rendered web page in order to further lower the load speed. The only JavaScript that is available for the developer (unless used in an `iframe`), are within the components that Google has created for you. This is to make sure that everything is handled correctly and that it will not impact the performance in a negative way [4]. An example of this would be one of the modules that the authors are using in their study, more precisely the `amp-bind`. Here, AMP has created their own format for JavaScript:

```
<p
    [text]="'Hello ' + foo">Hello World
</p>
<button
    on="tap:AMP.setState({foo: 'amp-bind'})">Say "Hello amp-bind"
</button>
```

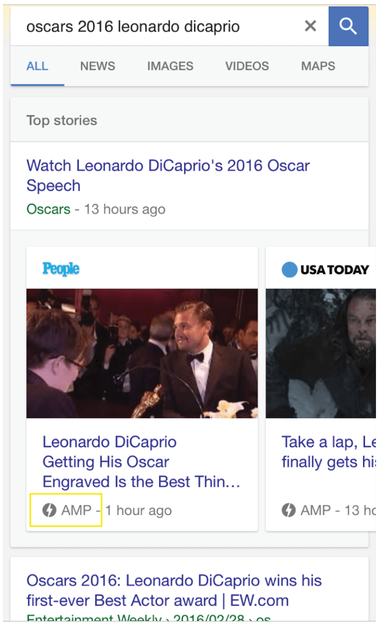
The text in the `<p>` tag has its static value of `'Hello World'`. When the `<button>` is pressed, the content of the `foo` variable is changed to `'amp-bind'` [1].

5.3 Caching of AMP web pages

One of the more interesting features that AMP has to offer is the caching functionality. Once implemented, when a user navigates to the site, the user might not find themselves on the websites host machine but instead on a cached version of the site on a server hosted by Google [13]. As described by AMP in a video on their website, the AMP cache is present in favor for the user [11]. It will make sure that all of the AMP pages that the developer has created are valid. It will also pre-render the page so that when it is viewed by a user, it will render instantly. Though the traffic will travel through Google's servers, AMP promises that with the correct use of their tags, the developer will get full control of the analytics and revenue gathered from the site [3].

5.4 Rich content

The AMP page might also, if available, be featured in the rich content carousel on Google Search. This means that for example, if a user searches for SEO (Search Engine Optimization) and the AMP page that the developers has created contains content that matches the search, it might be shown on top in the search result in a carousel (`Figure 1` [21]).



Watch Leonardo DiCaprio's 2016 Oscar Speech

Oscars - 13 hours ago

Leonardo DiCaprio Getting His Oscar Engraved Is the Best Thin…

AMP - 1 hour ago

Take a lap, Le
finally gets hi

AMP - 13 ho

Oscars 2016: Leonardo DiCaprio wins his first-ever Best Actor award | EW.com

Entertainment Weekly › 2016/02/28 › os

Figure 1: The top stories amp carousel.

5.5 Developers perspectives

In an article written by Neil Patel, an author who specializes in SEO, he states the importance of web page speed. Patel then presents AMP as one great solution for SEO and website speed problems. The only negative aspect Patel finds in what AMP provides and restricts the developer from doing, is the fact that you are not in control of the JavaScript. This might be the result of your page being lazy loaded which means that some content that might take longer to load can be rendered after the web page has had its initial load [40][55].

Several developers has found AMP's statement that sites load fast, true. The format does help the web page to load fast as well as having great visibility in the AMP carousel [14][46][54].

A large study on how and to what extent the format would benefit the companies was made covering over one thousand AMP websites [32]. The result was clear, AMP helps a great deal with minimizing the complexity of the website, mostly due to the strict standards that Google has set for AMP. Because of this, aspects such as conversion rate and speed gets improved as a side effect. The pre-rendering of the website through Google was also a huge participant in the improvement list. The study discovered that increased mobile data consumption could be the case for the perk of loading a pre-rendered site. The authors of the study also stated that the format was not world wide accepted:

> *The controversy surrounding AMP and its impact on the health of the Web is ongoing [32].*

While all of the features sound promising from Google's perspective, not everyone is on board with all of their thoughts.
Tim Kadlec, who is a web consultant and an author, understands the meaning of AMP [34]. He means that by receiving a fast loading page and have the optimization done for you, you sell yourself out. Through adapting to the standards of AMP such as only using AMP- based components as JavaScript and in-line CSS.
Additionally, opting to use this format, your site will be cached for free in order to further push the speed of the developers website.

Though the speed result is a great thing, Kadlec means that the same outcome could be achieved with any other Content Delivery Network (CDN). He continues by saying that a developer can develop a site with pre-rendering, caching and the minimal use of JavaScript and CSS in mind to receive the same result. That should also be the standard of coding according to Kadlec [33].
John Gruber, the creator of Markdown which is a plain text formatting syntax [56][57][29], shares the same thought.

> *The lock-in aspect makes no sense to me. Why would I want to cede control over my pages to Google? AMP pages do load fast, but if publishers want their web pages to load fast, they can just engineer them to load fast [13].*

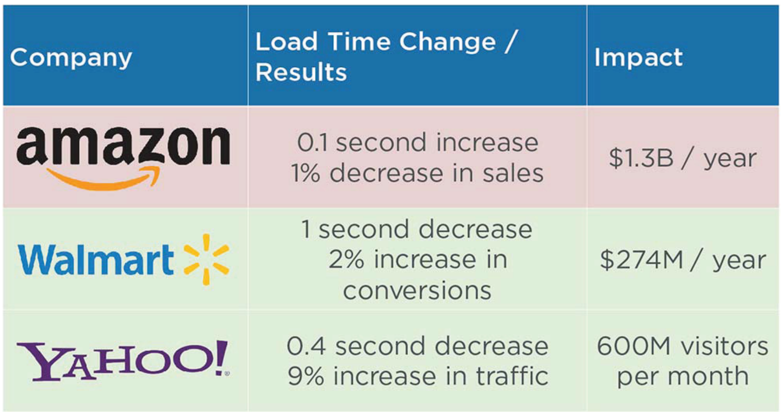
The overall feeling from the developers, mentioned in the literature review articles, is mutual. The benefits that AMP provides are positive but not to the cost of the lock down in their ecosystem and the stripping of style because the format requires you to. One other thing that weighs heavily on the downside for many of the developers is the change of HTML structure [44][20]. Daniel Meisler writes in his article that Google AMP does not only change the standards of HTML structure but also the core principles of the web [43]. Nathan Kotny, CTO of Rockstar coders says:

> *I'm hesitant to fully embrace the Google Ecosystem that takes as much control over my site as AMP [54].*

AMP does make your website load faster in almost all cases. Speed does increase the user experience and other metrics as well, even though everyone is not on board with the price. Referring to RQ1, Google AMP does offer as good or better user experience than most standard websites does. This can however depend on different aspects. One could be that AMP restricts and forces the developer to minimize the website and use caching. If the same mindset would be taken into account in every mobile website that is developed using techniques other than AMP, the same result could potentially be achieved. Though AMP is young, some authors from the articles mentioned in the literature review believe that it could grow into something better with time, while others straight up loathe the thought of having Google in that much control of the web.

5.6 Studies

Even though Google AMP mean to increase several aspects for a website such as conversion rates or traffic, they could be the positive outcome from the increased velocity on the web page. Eric Enge, CEO of Stone Temple and search engine consultant, has written an article where he states that just a few milliseconds of loading speed could impact a websites conversion [15][16]. An image Enge uses in the article shows just how delicate a few milliseconds could be (Figure 2).



Company	Load Time Change / Results	Impact
amazon	0.1 second increase 1% decrease in sales	$1.3B / year
Walmart	1 second decrease 2% increase in conversions	$274M / year
YAHOO!	0.4 second decrease 9% increase in traffic	600M visitors per month

Figure 2: Data from what speed can do on a big corporate website.

Enge performed a research study on companies where they converted more than 90 percent of their website to an AMP version. The implementation of AMP took about one month, which gave them time to implement styling, tracking and ads. One of the companies, Thrillist, saw an increased growth of 70 percent regarding traffic. 50 percent of those 70, came with the help of AMP. The study also included a fashion company and a ticket sale website. The fashion company had 11 days of implementation which covered the landing pages of the website to be converted to AMP. Their bounce rate was reduced by 40% [15]. The ticket website did gain, after converting 99 percent of their site to an AMP version, 100 percent raise in e-conversion [51].

Eric Enge wraps the study up by saying:

> *AMP can offer some really powerful benefits — improved site speed, better user experience and more revenue — but only for those publishers that take the time to implement the AMP version of their AMP site thoroughly, and also address the tracking issue in analytics so they can see the true results [15].*

In order to flatter the idea of using AMP even more, Google has on the AMP page, a list of several companies that has implemented AMP. It contains a short summary of the company and thereafter some of the aspect that were improved [5]. In a more deep going and thorough study, Google hired Forester, a leading global research and advisory firm to conduct a study.
Forester interviewed four companies who had been implementing the AMP format for their websites. From the results gathered in the interviews, Forester created a forecast which would predict in what way, the cost of the AMP implementation would affect the companies conversion rate. Using the forecast, Forester said that if a company who had 4 million visits a month with a 10 percent margin of the profit, the costs of the AMP implementation would be payed back at the six months mark. From there after, the conversion would continue to grow, as shown in Figure 3 [27].

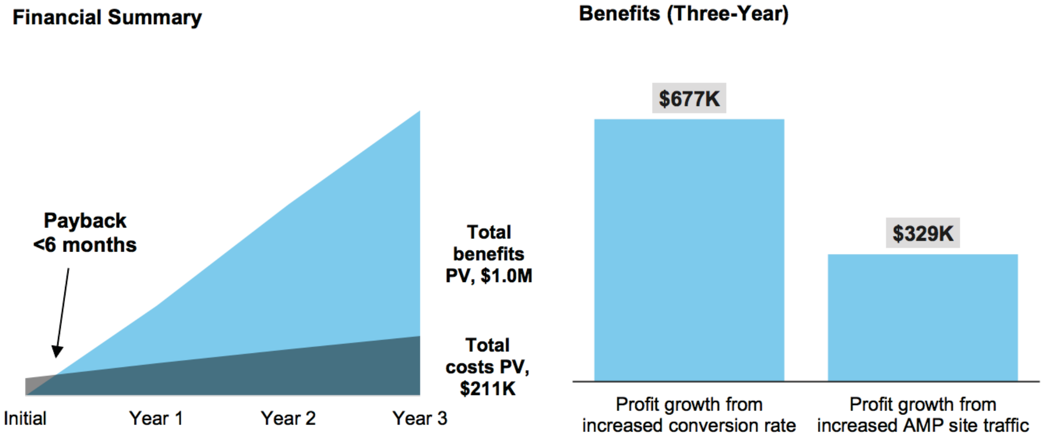


Figure 3: The forecast that Forester came up with.

Though many companies is presented as getting improved statistics, a study was conducted saying otherwise [8]. From this study, only one out of three of the websites in the study could see a clear difference in positive feedback. They came to a result that AMP could be a good thing, but for smaller companies it might be a risky format to implement. If they heavily weigh on the websites

monetizing, dedicating a lot of time for a format that only might improve your conversion rates is a bad bet [9].
To summarize all of the studies, most of them point towards the greatness of AMP's improvements on websites metrics. In order to achieve the best result with AMP, the format needs to be implemented correctly. If not done right, the format could decrease the perks of AMP and or even diminish the conversion rate. The underlying reason that many aspects was improved is the speed of the website that AMP has increased. Even though many of the studies show examples of conversion rate or traffic acceleration, speed is the recurring factor. Therefore the authors in this paper find their study to be of great importance. The study they perform will give an insight to what needs to be done in order to implement Google AMP and what the result could be, comparing it to a non AMP website.

6 Results

In order to provide an answer for the research questions, the authors decided to create three different studies. One of the studies were a literature study, while the other two were empirical studies. Throughout the experiments, the authors noted and recorded their work flow in a way so that they could compare the data gathered. This chapter compiles the data retrieved during each of the studies performed in the report. Each research question will first be summarized in order to give the reader a good overview of the results. The questions will then be thoroughly described to allow the authors to fully analyze the result.

6.1 Results summary

To give the reader a short summary of the results gathered, the research questions are shortly answered below.

6.1.1 RQ1: How is the user experience offered by Google AMP contrary to an ordinary website?

Most of the users of Google AMP did see an increase in user experience in form of speed, conversion rate and other metrics discussed in the literature review. Though this was the case, some argued that you could achieve the same result with another format. Some developers also disliked the amount of power Google got from having the users directed through Googles' cached servers.

6.1.2 RQ2: How does a Google AMP based website compare to an ordinary website in terms of development?

To create an AMP application, more time need to be spent compared to create a Pug application. This is because even though a lot of knowledge can be reused in the creation of the AMP application, the developer still needs to learn the new ways of approaching JavaScript and AMP's modules. Creating the AMP application will require more time even though the developer has prior programming knowledge within the area.

6.1.3 RQ3: Will Google AMP offer any improvements compared to an ordinary website?

The AMP application is six times larger that the Pug application where most of the size comes from the pre-loaded AMP library JavaScript files. Even though this was the case, the two applications almost had the same time to load, depending on what device and which browser you are using. Firefox is an exception where they do not seem to have implemented support for AMP. The metrics for the AMP application shows that Firefox is significantly slower than the other browsers when loading an AMP page.

6.2 Results extended

In order to give the reader the full version of the result, all of the results regarding the research questions will be displayed below.

6.2.1 RQ1: How is the user experience offered by Google AMP contrary to an ordinary website?

On today's web, websites has to load fast in order for the company behind the application to be successful. Even the slightest lowering in velocity can impact a big company with a huge loss in conversion rate.
Through looking on web pages which implemented the format on AMP's own website, all of the pages showed increased values on aspects such as traffic, conversion rate or web page velocity. From other studies conducted by other agencies, almost the same result were accumulated. Web pages that fully embraced the AMP format did get increased values in the metrics mentioned. There was however some smaller businesses which saw a diminishing amount of users reaching their site or a lower conversion rate. One study showed that only one out of three web pages who were using the AMP format saw a noticeable difference for the good.

When applying the format to a web page, it has to be done with great care. In order for AMP to give the application its full potential, the modules that AMP provides has to be used the right way. If not embraced fully, it could be the reason for negative outcome. Therefore, companies who rely on the income from their websites' ads could take a risk if choosing to implementing Google AMP.
When the different metrics for the web application did increase, so did also the speed. The authors found out that the underlying reason for that many of the aspects were improved, were because of the caching of websites on Google's domain but also the increased speed of the application.
Google AMP did for most of the applications that implemented the technology, improving their user experience by a increase of speed.

Though the format has a high success rate, not every developer were happy with AMP. Many of the sources found in the literature review, did see Google AMP as a threat for the web. This was because the format changed the HTML structure for how a standard website structure should look. Another huge dislike from the user base of AMP was that Google got too much control over the website. Since the application was cached on Google's servers, a lot of the data was handled through Google. The last discussion point was whether AMP was necessary at all. Developers within the SEO area of expertise saw that the same results could be achieved without Google AMP.
Even though many of the comments were negative, they were all agreeing in the sense of AMP increasing the speed and user experience of a web page if implemented correctly.

6.2.2 RQ2: How does a Google AMP based website compare to an ordinary website in terms of time to build?

To record and try to compare the results from the different web application formats, the results will be split into categories.

6.2.2.1 Setup

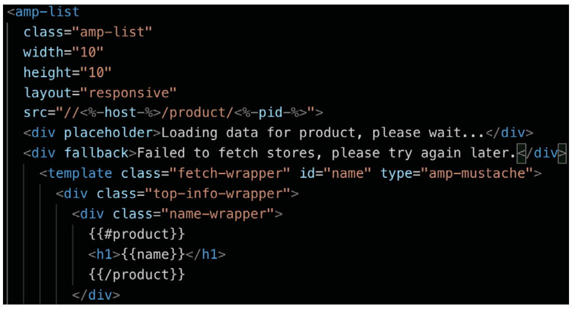
The Pug application was generated with the command `"express --view=pug amp-example-pug"`, using Express Generator [18]. Isomorphic-fetch and es6-promise modules were added to allow the application to fetch data from Prisjakt's API [6][41].

The AMP application was scaffolded from a Git repository located on Google AMP's tutorial website [28]. The same modules for fetching data in the Pug application was used in the AMP application since both applications share Express as their common backend. In order to handle data and display it on the page, the solution differed between the two authors. One began by using the mustacheJS rendering library and the other used ejs. For the final product both of the rendering libraries were used regarding to RQ3.

6.2.2.2 Tags

Pug uses standard HTML tags but is written as a stripped down version. When the website is rendered, the Pug engine translates the Pug template to normal HTML. An example paragraph in Pug is written as `"p Hello World"` which Pug in turn translates to normal HTML which is rendered as `"<p>Hello World</p>"`.

The tags for the Google AMP format is very close to standard tags of HTML. One of the tags that differ is `img` which is changed to `amp-img`. Other tags which also differs to standard HTML are tags connected to some of the Google AMP modules. This could for example be `amp-list` and `template` which, in the case of our studies, are used when retrieving data from a route in the Express backend. An example of this is displayed in Figure 4.



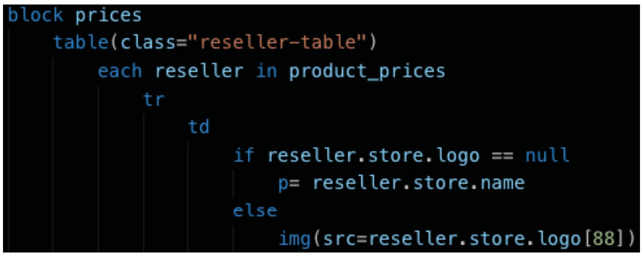
```
<amp-list
  class="amp-list"
  width="10"
  height="10"
  layout="responsive"
  src="//<%-host-%>/product/<%-pid-%>">
  <div placeholder>Loading data for product, please wait...</div>
  <div fallback>Failed to fetch stores, please try again later.</div>
    <template class="fetch-wrapper" id="name" type="amp-mustache">
      <div class="top-info-wrapper">
        <div class="name-wrapper">
          {{#product}}
          <h1>{{name}}</h1>
          {{/product}}
        </div>
```

Figure 4: Code snippet showing the amp-list tag

The `amp-list` src field is defined with ejs and is a link to the Express endpoint which retrieves the product stores and details. The information from the endpoint is then presented within the template tag with the help of mustacheJS tags: {{`#product`}} and {{`name`}}.

6.2.2.3 Structure and data handling

Pug provides tools to work with data, such as a for-each functionality. A good example of this can be seen in `Figure 5`.



```
block prices
    table(class="reseller-table")
        each reseller in product_prices
            tr
                td
                    if reseller.store.logo == null
                        p= reseller.store.name
                    else
                        img(src=reseller.store.logo[88])
```

Figure 5: Code snippet showing a each-loop in Pug

The data found in the product_prices is set by Express as a variable which Pug then accesses when rendering the template. The `each` function takes a variable and loops it, allowing the developer access to each item by a temporary variable. In this case, we store each reseller from `product_prices` in a `reseller` variable which then is used inside the block to render a row of information in a table.

The structure of the AMP application is very close to the standards of HTML. Things that differ are the way the developers displays data with AMP modules. A great example of this is displayed in figure 4. Here the data has to be rendered within an AMP tag called `template`. If a developer wants to display data dynamically using the AMP functionality, the `amp-list` tag has to be used. Otherwise, data can be gathered from the Express backend, sent via a variable,

and used in the AMP document with ejs. The same way as the Pug version does.
The CSS and the JavaScript is written in the same file as the AMP code. This could be a different experience from what most developers are used to, when splitting style and JavaScript into separate files.

6.2.2.4 Time

The Pug application took around 15 hours for one of the authors to develop, while the other took close to 14 hours. Much of the time was spent working with the data from the API. Both authors also had some prior knowledge of Pug which decreased the amount of time needed.

The time it took to build and create the AMP page after Prisjakt's wishes took roughly 36 hours for one of the authors and 28 hours for the other. This includes browsing the Google AMP documentation in order to understand the concept of the format as well as understanding the API. It does also contain the setup mentioned above.

6.2.3 RQ3: Will Google AMP offer any improvements compared to an ordinary website?

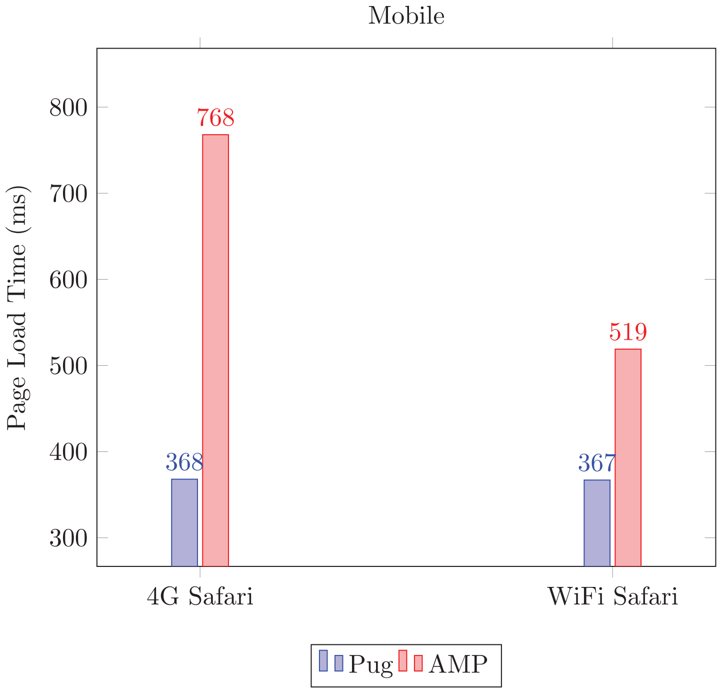
The authors discussed in which ways a common user visits a web page. They noted that the three major points of interest to test, were using different devices, internet connections and browsers.
Since the most commonly used devices to browse the internet are mobile devices, tablets and computers, the authors decided to pick one from each category [50]. The devices chosen were an iPhone 7 and an iPad 2017 as mobile devices. A desktop PC and a MacBook Pro laptop was chosen as the computer devices. These units were picked because they were available to the authors. Each of the devices specifications can be found in appendix A and the tests performed can be found in appendix B.

Different browsers were also tested to analyze if there was a difference in how the browsers managed AMP. Safari was the chosen browser for mobile and tablet devices, being the second most used browser on mobile and most used on tablets worldwide [48][49]. The tests performed on the mobile devices were analyzed using the developer tools in Safari on macOS, since developer tools for other browsers can not be accessed in a easy way. On desktop and laptop the authors opted to use Chrome, Firefox and Safari because they are the most, second most and fourth most used browsers [47]. Internet Explorer, the third most used browser, was excluded due to not officially being supported by Google AMP [2]. Safari was excluded on desktop because not being support in Windows [7]. The laptop tests were performed in Safari, Chrome and Firefox and the desktop PC tests in Firefox and Chrome.
The tests on the laptop, tablet and mobile were performed both using 4G and

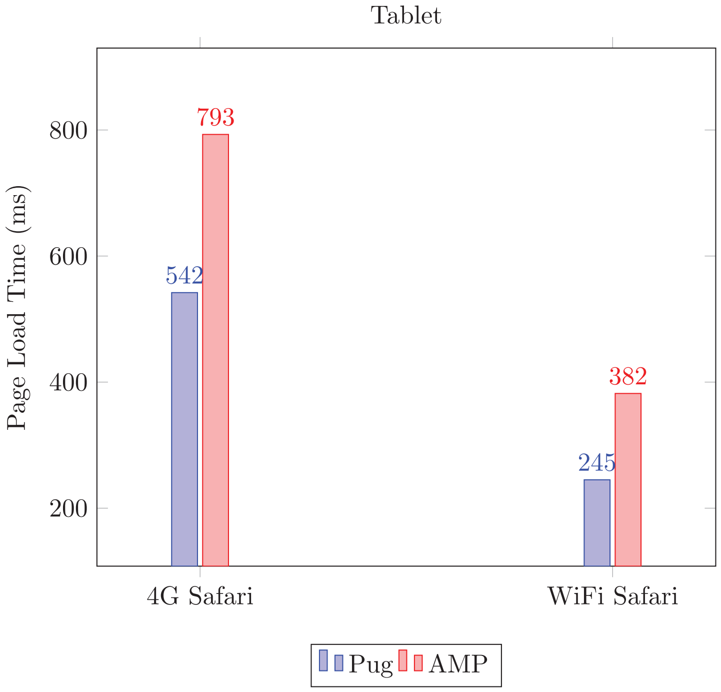
a WiFi connection while the only connection tested on the desktop PC was a wired connection. The different connections were chosen because they are used by both of the authors on a daily basis.

6.2.3.1 Page load time

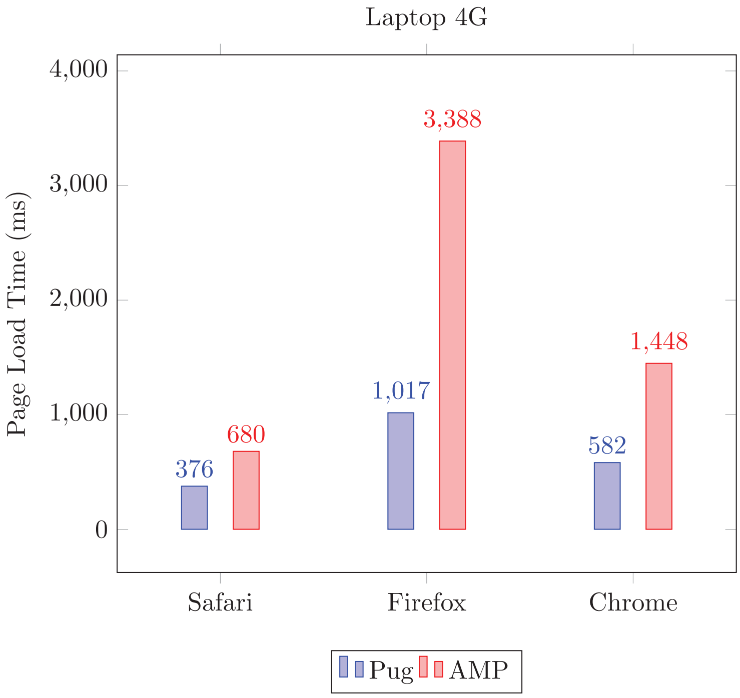
To get a good measurement, the authors ran a full refresh without cache five times per browser to get a good average measurement and remove any potential deviations. The lowest, average and highest value of each set of tests can be found in Appendix E.



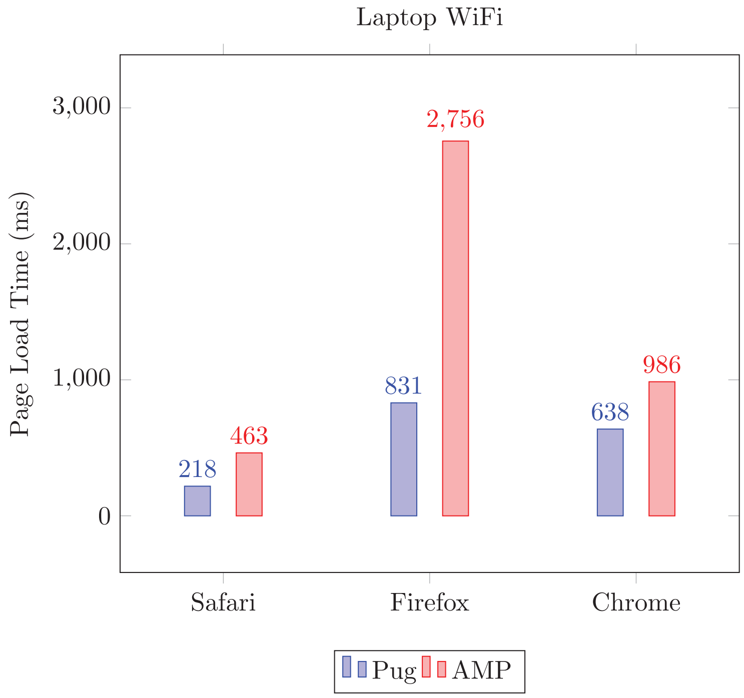
While testing on mobile devices, the load time while visiting the Pug page is almost the same for both 4G and WiFi, around 370ms. The AMP pages takes a bit longer to load, close to 770ms while using 4G and 520ms while using WiFi.



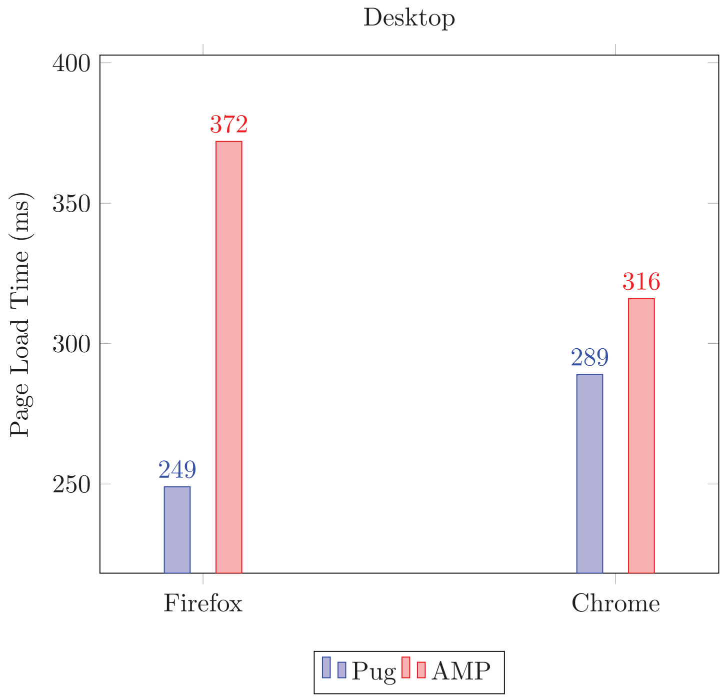
The tests performed on a tablet shows a larger percentage increase between the 4G and the WiFi results when compared to the measurements for the phone. While on WiFi, both pages loads faster on the tablet than the phone with the difference being 245ms compared to 367ms while browsing the Pug page and 382ms compared to 519ms on the AMP Page.



Looking at the 4G result from the tests performed on the laptop, the Pug page is still faster than the AMP page. However, the tests also show that there is a large difference between each browser where Firefox, at 3388ms, almost takes five times longer to load the AMP page when compared to Safari's 680ms. Chrome is faster than Firefox, but still takes more than twice as long as Safari does. The difference between Safari and the two other browsers is far less on the Pug page, where Firefox is 170% slower and Chrome is 54% slower than Safari.



The laptop tests performed with WiFi shares a similar result to the one performed with 4G, where the Pug page is still faster than the AMP page in all browsers.



The desktop computer loads both applications relatively quickly. The AMP page is however still slower than its Pug competitor, loading 49% slower in Firefox and 9% slower in Chrome.

6.2.3.2 Application size

Another aspect of the tests were to view the size of the web page through different browsers and devices.
The AMP page is 6 times larger than Pug, at 528.35KB compared to Pugs 87.65KB. More than 80% of the AMP application size comes from the external JavaScript needed for it to work, referenced as AMP JS in `Figure 6`. Shared data is the data which is present in both applications. This includes retailer logotypes, Prisjakt's logotype and other SVG images used on both sites. Store data contains information about each retailer, such as name, price, stock and link to their product page. HTML and CSS is the HTML structure of the website and its styling.
There was a small difference in size between the browsers when testing the AMP site. This could be a difference in the AMP library where different JavaScript could be needed for different browsers. While Safari and Google Chrome are closer to one another, at 528KB and 536KB, Firefox is 8.8% larger than the Safari application, at 575KB.

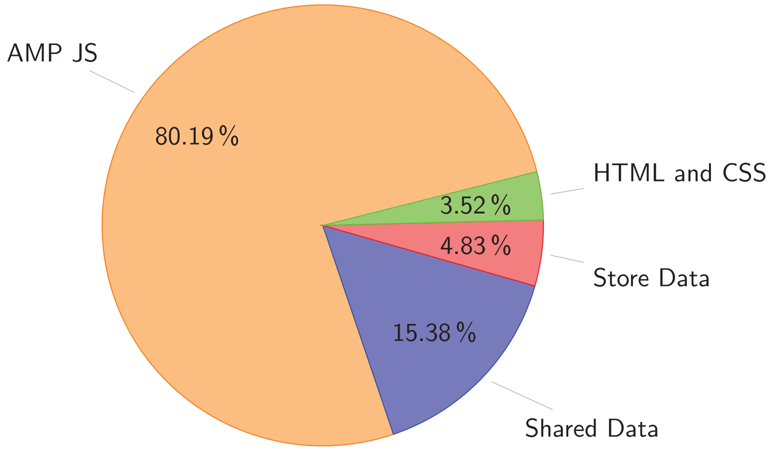


Figure 6: Parts of an AMP application size in percentage

The Pug page shares much of the result from the AMP page when looking at size. Safari comes in at 87.65KB while Firefox and Chrome has the same size, 88.94KB, roughly 1.47% larger than in Safari.

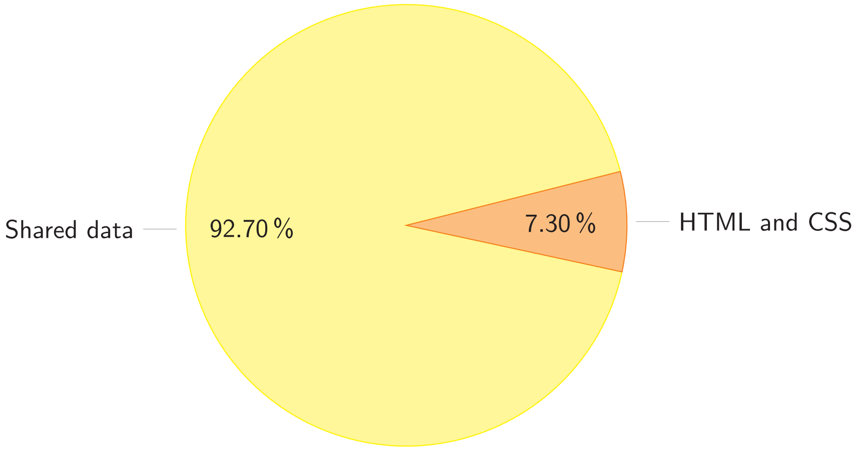


Figure 7: Parts of a Pug application size in percentage

6.2.3.3 Speed Index

When analyzing the result from the Lighthouse tests, both applications performed very well [24]. The Pug application is however slightly faster when looking at speed index, loading at 1.7s compared to AMP's 1.9s. The AMP page shows its first content faster than the Pug page but since it loads the retailer data after initial load, it finishes after the Pug application. The results of the AMP Lighthouse tests can be viewed in appendix C and the Pug tests can been seen in appendix D.

7 Analysis

This chapter analyzes the results and oppose them to the theories found in the literature study in order to answer the research questions.

7.1 RQ 1: How is the user experience offered by Google AMP contrary to an ordinary website?

Most of the studies found on Google AMP does point in the direction that the format improves the user experience. Some studies did however come to the conclusion that AMP gave a negative or close to no distinguishable positive outcome. Reasons for this could have been that they did not utilize or implement the format fully. It is possible to use the format with minor settings, however it could result in the application getting lower metrics in traffic and conversion rate even though the web page is faster.

The thoughts of Google AMP from companies and developers who had used the format differed. Some pointed out the greatness while other saw the features as something bad. Google AMP does provide a standard which the developer has to stick to. This does in return give the developer a fast web page because the standard limits the amount of styling and JavaScript that can be used. It does also allow the web page to be cached on Google's servers, for instant loading. This is seen both as a great feature or something that is redundant. Developers who are specialized within SEO and performance explains that the same outcome could be done without AMP. If a web page were to be created with performance in mind, there would be no reason to use Google AMP.

Google AMP does increase the user experience. It acts as a guideline for how a web page should be created. With the help of the strict structure, the application gets faster. The speed also increases since it is stored on Google's servers which makes the web page render almost instantly.

7.2 RQ 2: How does a Google AMP based website compare to an ordinary mobile-friendly website in terms of development?

The authors conclude that AMP and Pug were not so different in terms of difficulty to build. We both had prior knowledge with Pug which resulted in shorter development time. Neither of the authors had used AMP previously. When creating the AMP page, most of the time ended up being used on understanding the concept of AMP and on the modules that AMP had created in order to solve the problems that otherwise were solved by JavaScript.

There was a big adjustment for the authors in order to learn the new AMP format. Since the modules created by AMP were using its own kind of JavaScript, the new way of applying dynamic content took time. If the content would have been static, a lot of time could have been saved and the structure of the AMP and Pug pages would have been more similar.

Another reason that the AMP page had an increase in development time was because all CSS and JavaScript were written in the same file. Though the reason behind it is good, since it lowers the amount of files being loaded, this took the authors time to get used to.

To answer the research question, the AMP application did take longer time to build because of the mentioned factors above.

7.3 RQ 3: Will Google AMP offer any improvements compared to an ordinary mobile-friendly website?

While our tests shows that AMP performs worse than Pug, the result might not be the same for larger applications, as seen in some of the studies mentioned in our literature study. One way to find out if Google AMP would be a good implementation for a company, would be to find an isolated part of the web application. When rewritten as AMP, the isolated part could be run as a test in order to gather data. The company could then see if the metrics favored AMP and make a decision whether to rewrite the rest of the application to the format. Implementing Google AMP could be a risk for smaller companies who solely depend on the revenue from ads because the implementation could give a negative outcome.

7.3.1 Page loading time

When comparing the results between the two applications, the Pug application won in all speed tests. While Safari and Chrome were relatively close to each other, Firefox seems to lack support for the AMP format. Though the AMP application is slower than Pug in all tests, the AMP page is by no means a slow application, still loading at below 1.5s in Chrome and Safari. However, as mentioned in the literature review, the difference in speed could risk being a big loss in conversion rates.

7.3.2 Application size

Even though the AMP application developed is six times larger than the Pug application, the difference in size does not justify the speed. AMP in Safari takes at most 112% longer to load, Chrome 249% and Firefox peaks at 330% longer loading time. The experiment done were performed on a small scale. Since the AMP format suffers from a big overhead with the required JavaScript libraries, small websites may have a hard time seeing the benefits of using Google AMP when looking at the application size.

7.3.3 Speed index

The AMP site has a speed index of 1.9s, a good performance according to the Lighthouse report which gives it a green mark. The Pug application is, as in the two previous metrics better, having a speed index of 1.7s. For the user, speed index is the most relevant measurement of the three since a good speed index indicates that the page loads content constantly during its page loading time. The AMP application renders the base page and its styling faster than the Pug site, but since the Pug page loads the retailer data at the same time, it completes its render faster.

8 Conclusion

The AMP format is great but it can still become better. The majority of developers who have used AMP can see that it provided a web page with high speed, though the same results could be achieved without using the AMP format. The fact that users indirectly browse the web page through Google, which grants Google an even larger impact on the web, gives many developers a reason not to use the AMP format.

Through the experiments, the authors concluded that Google AMP has a higher learning curve than Pug. This is because AMP has got its own way of writing JavaScript through modules whilst no other JavaScript is allowed.
In the results, the AMP applications was slower than the Pug applications. AMP includes large JavaScript files which is needed in order for the format to be utilized. This gives smaller applications a hard time catching up with other technologies such as Pug, where the Pug application was 1/6 of the AMP application in size. The results gives an indication that a small web page is slower than a large application because it suffers from the loading and parsing of the AMP JavaScript files. Google AMP achieved almost the same score in speed index. It performs a bit slower than Pug and shows that if the application renders dynamic content instead of only static, it takes a longer time to fully render.

The authors came to an understanding that AMP could be a good choice and that a lot has been updated since its initial release in 2015. Though the format is published, a lot of new features are being developed and updated. Google AMP does not have a guideline for every company to follow in order to achieve a prominent result. Every potential user of Google AMP must spend resources and do research in order to find out if the format works for them and their application.

The general thoughts of Google AMP is both positive and negative, which it shares with all technologies. However after this report, the authors see a potential bright future for Google AMP.

9 Validity threats

Throughout the experiments the AMP application was slower than the Pug application even though many studies found through the literature study said otherwise. This could be because the authors implemented Google AMP in a bad way or because the web pages found in the study had larger application size. Another reason could be that the experiment did not include the caching functionality of AMP which could have decreased the loading time further.

10 Future Work

This paper revolved around discovering whether Google AMP could help companies improve the user experience of their websites. The research done could not have been possible when Google initially released AMP. This is because AMP is a continuously developed framework where Google does not only want to support static but also dynamic content. From only being utilized in articles to being able to be used in stores and other dynamic content websites.

Though this study only scratches the surface of Google AMP's components and especially initial load speed, many more can be discovered and explored. A list of new features that will arrive to Google AMP can be found on their website [23]. Among these features are auto-complete and payment components which could increase the amount of websites which can utilize Google AMP.

Since Google AMP is being developed everyday and new features are released, many more experiments can be done in the future when the AMP format has matured even further. The problems that occurs today might be non existent tomorrow while other issues have arisen.

Not only Google AMP can be explored further but the study as well. Since this report only had experiments on a smaller scale application, it would be of great interest to try out the AMP format on larger applications with the same tests. To further examine Google AMP and investigate its features, the application could also be published in a production environment in order to not only examine the speed metric but also conversion rate and traffic which was not done in this study. Another feature that also could be analyzed if the application were to be published, would be the caching on Googles' server. This could give a better understanding of Google AMP.

In our study, we compared a Google AMP application using ejs to an application using Pug. A more correct comparison might have been to compare an AMP application using ejs and another application using ejs instead of Pug.

11 Appendix

11.1 Appendix A - Device Specifications

Device	**Desktop PC**	**Laptop**
Device Name	#	Macbook Pro 13” 2015
OS/Version	Windows 10 - 1809	MacOS 10.14.3 Mojave
CPU	Intel Core i7 6700K	Intel Core i5 5257U
Memory	32GB - DDR4-2666Mhz	8GB - LPDDR3-1866Mhz
Disk Read Speed	550MB/s	1020MB/s
Disk Write Speed	140MB/s	490MB/s
Network Connection (Download/upload Mbit/s)	Ethernet - 96/95	WiFi 5 (802.11ac) - 50/45 4G phone sharing - 8/7

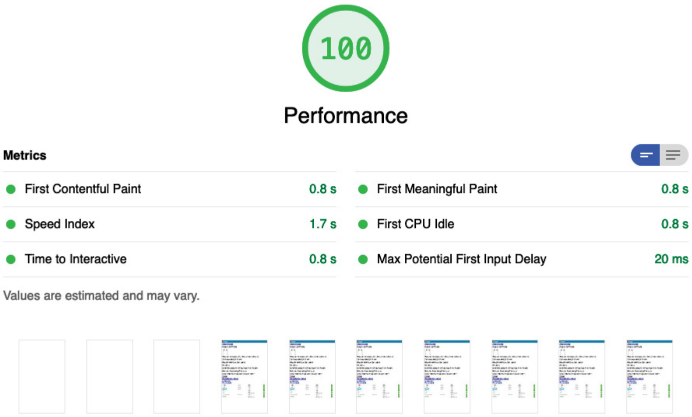
Device	**Tablet**	**Phone**
Device Name	iPad 2017	iPhone 7
OS/Version	iOS 12.3 (beta)	iOS 12.2
CPU	A9	A10 Fusion
Memory	2GB	2GB
Disk Read Speed	640MB/s	430MB/s
Disk Write Speed	72MB/s	40MB/s
Network Connection (Download/upload Mbit/s)	WiFi 5 (802.11ac) - 70/91 4G phone sharing - 17/17	WiFi 5 (802.11ac) - 62/47 4G mobile network - 8/18

11.2 Appendix B - Tests Performed

AMP Application			
Device	**4G**	**Wireless**	**Wired**
Desktop PC			Firefox x5
			Chrome x5
Device	**4G**	**Wireless**	**Wired**
Laptop	Safari x5	Safari x5	
	Firefox x5	Firefox x5	
	Chrome x5	Chrome x5	
Device	**4G**	**Wireless**	**Wired**
Tablet	Safari x5	Safari x5	
Device	**4G**	**Wireless**	**Wired**
Mobile	Safari x5	Safari x5	

Pug Application			
Device	**4G**	**Wireless**	**Wired**
Desktop PC			Firefox x5
			Chrome x5
Device	**4G**	**Wireless**	**Wired**
Laptop	Safari x5	Safari x5	
	Firefox x5	Firefox x5	
	Chrome x5	Chrome x5	
Device	**4G**	**Wireless**	**Wired**
Tablet	Safari x5	Safari x5	
Device	**4G**	**Wireless**	**Wired**
Mobile	Safari x5	Safari x5	

11.3 Appendix C - Pug Lighthouse Report



100

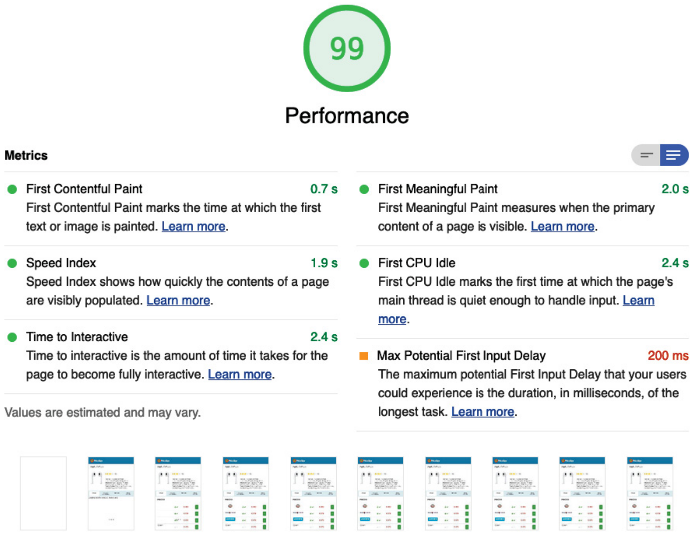
Performance

Metrics

● First Contentful Paint	0.8 s	● First Meaningful Paint	0.8 s
● Speed Index	1.7 s	● First CPU Idle	0.8 s
● Time to Interactive	0.8 s	● Max Potential First Input Delay	20 ms

Values are estimated and may vary.

11.4 Appendix D - AMP Lighthouse Report



99

Performance

Metrics

● First Contentful Paint 0.7 s
First Contentful Paint marks the time at which the first text or image is painted. Learn more.

● First Meaningful Paint 2.0 s
First Meaningful Paint measures when the primary content of a page is visible. Learn more.

● Speed Index 1.9 s
Speed Index shows how quickly the contents of a page are visibly populated. Learn more.

● First CPU Idle 2.4 s
First CPU Idle marks the first time at which the page's main thread is quiet enough to handle input. Learn more.

● Time to Interactive 2.4 s
Time to interactive is the amount of time it takes for the page to become fully interactive. Learn more.

■ Max Potential First Input Delay 200 ms
The maximum potential First Input Delay that your users could experience is the duration, in milliseconds, of the longest task. Learn more.

Values are estimated and may vary.

11.5 Appendix E - Speed Measurements

Test	Lowest	Average	Highest
Pug - Mobile 4G Safari	354ms	368.4ms	390ms
AMP - Mobile 4G Safari	671ms	768.4ms	1000ms
Pug - Mobile WiFi Safari	387ms	367ms	496ms
AMP - Mobile WiFi Safari	406ms	532.6ms	696ms

Test	Lowest	Average	Highest
Pug - Tablet 4G Safari	374ms	541.8ms	767ms
AMP - Tablet 4G Safari	704ms	793ms	900ms
Pug - Tablet WiFi Safari	212ms	244.8ms	319ms
AMP - Tablet WiFi Safari	315ms	382.4ms	435ms

Test	**Lowest**	**Average**	**Highest**
Pug - Laptop 4G Safari	285ms	376ms	677ms
Pug - Laptop 4G Firefox	872ms	1017ms	1190ms
Pug - Laptop 4G Chrome	500ms	582.4ms	675ms
AMP - Laptop 4G Safari	598ms	680ms	874ms
AMP - Laptop 4G Firefox	3250ms	3388ms	3490ms
AMP - Laptop 4G Chrome	1050ms	1448ms	1770ms

Test	**Lowest**	**Average**	**Highest**
Pug - Laptop WiFi Safari	186ms	218.4ms	245ms
Pug - Laptop WiFi Firefox	800ms	831.2ms	929ms
Pug - Laptop WiFi Chrome	460ms	638ms	1070ms
AMP - Laptop WiFi Safari	421ms	463.4ms	522ms
AMP - Laptop WiFi Firefox	2640ms	2756ms	2900ms
AMP - Laptop WiFi Chrome	896ms	986ms	1100ms

Test	**Lowest**	**Average**	**Highest**
Pug - Desktop Firefox	241ms	248.6ms	258ms
Pug - Desktop Chrome	262ms	289ms	302ms
AMP - Desktop Firefox	323ms	372.2ms	402ms
AMP - Desktop Chrome	285ms	316.2ms	340ms

References

[1] Google AMP. Amp bind. https://amp.dev/documentation/components/amp-bind?referrer=ampproject.org. Accessed: 2019-04-20.

[2] Google AMP. Amp supported browsers. https://amp.dev/support/faq/supported-browsers. Accessed: 2019-05-30.

[3] Google AMP. How amp pages are cached. https://amp.dev/documentation/guides-and-tutorials/learn/amp-caches-and-cors/how_amp_pages_are_cached. Accessed: 2019-05-10.

[4] Google AMP. How it works. https://amp.dev/about/how-amp-works. Accessed: 2019-05-10.

[5] Google AMP. Success stories. https://amp.dev/success-stories/?referrer=ampproject.org. Accessed: 2019-04-15.

[6] Matthew Andrews. isomorphic-fetch - npm. https://www.npmjs.com/package/isomorphic-fetch. Accessed: 2019-05-20.

[7] Apple. Update or reinstall safari for your computer. https://support.apple.com/en-us/HT204416. Accessed: 2019-05-30, Published: 2018-09-24.

[8] CHRIS BREAUX and BRADLEY DOLL. Amped about amp? https://cdn2.hubspot.net/hubfs/3794894/PDF/Chartbeat\%20AMP\%20Study_082318\%20.pdf. Accessed: 2019-05-10.

[9] CHRIS BREAUX and BRADLEY DOLL. Research study: Only 1 in 3 publishers see a clear traffic boost from amp. http://blog.chartbeat.com/2018/08/23/research-study-1-3-publishers-see-clear-traffic-boost-amp/. Accessed: 2019-05-10.

[10] Benjamin Burkholder. Javascript seo: Server side rendering vs. client side rendering. https://medium.com/@benjburkholder/javascript-seo-server-side-rendering-vs-client-side-rendering-bc06b8ca2383. Accessed: 2019-05-04, Published: 2018-06-08.

[11] The AMP Channel. Why amp caches exist. https://www.youtube.com/watch?v=n8n7fj60lds. Accessed: 2019-05-21, Published: 2017-04-06.

[12] Chris Coyier. Css-tricks. https://css-tricks.com/author/chriscoyier/. Accessed: 2019-05-10.

[13] Chris Coyier. Need catch amp debate. https://css-tricks.com/need-catch-amp-debate/. Accessed: 2019-04-23, published: 2017-03-17.

[14] Jayson Demers. Should you be using accelerated mobile pages amps. https://www.forbes.com/sites/jaysondemers/2017/12/01/should-you-be-using-accelerated-mobile-pages-amps/#3eab35132710. Accessed: 2019-04-25.

[15] Eric enge. Amp digital marketing 2018. https://moz.com/blog/amp-digital-marketing-2018. Accessed: 2019-04-24, Published: 2017-11-21.

[16] Eric enge. The complete guide to amp. https://www.stonetemple.com/the-canonical-guide-to-amp/. Accessed: 2019-04-24, Updated: 2019-02-14.

[17] Express. Using template engines with express. https://expressjs.com/en/guide/using-template-engines.html. Accessed: 2019-05-30.

[18] ExpressJS. Express application generator. https://expressjs.com/en/starter/generator.html. Accessed: 2019-05-19.

[19] Node.js Foundation. Express website. https://expressjs.com/. Accessed: 2019-05-05.

[20] Scott Gilbertson. Google amp is bad bad bad. https://www.theregister.co.uk/2017/05/19/open_source_insider_google_amp_bad_bad_bad/. Accessed: 2019-04-23, Published: 2017-05-19.

[21] Google. Amp carousel. https://developers.google.com/search/docs/guides/about-amp. Accessed: 2019-05-10.

[22] Google. Amp-img. https://amp.dev/documentation/components/amp-img?referrer=ampproject.org. Accessed: 2019-04-12.

[23] Google. Google amp roadmap. https://amp.dev/community/roadmap?referrer=ampproject.org. Accessed: 2019-05-04.

[24] Google. Google lighthouse. https://developers.google.com/web/tools/lighthouse/. Accessed: 2019-05-16, Updated: 2019-05-01.

[25] Google. Speed index documentation. https://sites.google.com/a/webpagetest.org/docs/using-webpagetest/metrics/speed-index. Accessed: 2019-05-05.

[26] Google. What is amp? https://amp.dev/about/websites. Accessed: 2019-03-23.

[27] Forester Google AMP. forester study. https://blog.amp.dev/2017/11/29/the-total-economic-impact-of-amp-across-publishers-and-e-commerce/. Accessed: 2019-04-15.

[28] googlecodelabs. Codelab: Advanced interactivity in amp. https://github.com/googlecodelabs/advanced-interactivity-in-amp. Accessed: 2019-05-21, Last updated: 2018-02-09.

[29] John Gruber. Markdown. https://daringfireball.net/projects/markdown/. Accessed: 2019-05-07.

[30] Open Source https://github.com/mde/ejs. Embedded javascript website. https://ejs.co/. Accessed: 2019-05-06.

[31] Open Source https://github.com/mustache. Mustache website. https://mustache.github.io/. Accessed: 2019-05-06.

[32] Byungjin Jun. Amp up your mobile web experience. http://aqualab.cs.northwestern.edu/wp-content/uploads/2019/02/AMP-Mobicom-2019.pdf. Accessed: 2019-04-25.

[33] Tim Kadlec. Amp and incentives. https://timkadlec.com/2015/10/amp-and-incentives/. Accessed: 2019-05-07, Published: 2017-10-08.

[34] Tim Kadlec. Markdown. https://timkadlec.com/. Accessed: 2019-05-07.

[35] Mozilla. Rendering engine - mdn glossary. https://developer.mozilla.org/en-US/docs/Glossary/Rendering_engine. Accessed: 2019-05-04, Published: 2019-03-23.

[36] Amos Ndegwa. What is page load time? https://www.maxcdn.com/one/visual-glossary/page-load-time/. Accessed: 2019-05-05, Published: 2016-05-03.

[37] Jakob Nielsen. Conversion rate. https://www.nngroup.com/articles/conversion-rates/. Accessed: 2019-05-16, Published: 2013-11-24.

[38] Digital Ocean. Digital ocean about. https://www.digitalocean.com/about/. Accessed: 2019-05-18.

[39] Ruadhan O'Donoghue. Amp: Building accelerated mobile pages. https://learning.oreilly.com/library/view/amp-building-accelerated/9781786467317/cover.xhtml. Accessed: 2019-04-18, Published: 2017-10.

[40] Neil Patel. The definitive guide to accelerated mobile pages (amp). https://neilpatel.com/blog/the-definitive-guide-to-accelerated-mobile-pages-amp/. Accessed: 2019-05-10.

[41] Stefan Penner. es6 promise. https://github.com/stefanpenner/es6-promise. Accessed: 2019-05-20.

[42] Prisjakt. About prisjakt. https://classic.prisjakt.nu/info.php?t=about_company. Accessed: 2019-05-16.

[43] Dan Puben. Google amp not good thing. https://danielmiessler.com/blog/google-amp-not-good-thing/. Accessed: 2019-05-07, Published/Updated: 2019-05-07.

[44] Dan Puben. Why amp is bad for your site and for the web. https://medium.com/@danbuben/why-amp-is-bad-for-your-site-and-for-the-web-e4d060a4ff31. Accessed: 2019-05-04, Published: 2018-02-28.

[45] Pug. Pug website. https://pugjs.org/api/getting-started.html. Accessed: 2019-04-12.

[46] Yvo Schaap. Google amp reviewed as a developer. https://hackernoon.com/googles-amp-reviewed-as-a-developer-19734198c9b1. Accessed: 2019-04-23.

[47] statcounter. Desktop browser market share worldwide - april 2018 to april 2019. http://gs.statcounter.com/browser-market-share/desktop/worldwide/#monthly-201804-201904-bar. Accessed: 2019-05-19.

[48] statcounter. Mobile browser market share worldwide - april 2018 to april 2019. http://gs.statcounter.com/browser-market-share/mobile/worldwide/#monthly-201804-201904-bar. Accessed: 2019-05-19.

[49] statcounter. Tablet browser market share worldwide - april 2018 to april 2019. http://gs.statcounter.com/browser-market-share/tablet/worldwide/#monthly-201804-201904-bar. Accessed: 2019-05-19.

[50] Statista. Which is the most important device you use to connect to the internet, at home or elsewhere? https://www.statista.com/statistics/387447/consumer-electronic-devices-by-internet-access-in-the-uk/. Accessed: 2019-05-18.

[51] Stone Temple. Event ticket center study. https://www.stonetemple.com/amp-case-study/event-tickets-center/. Accessed: 2019-05-10.

[52] Juan Vega. What exactly is client side rendering and how it is different from server side rendering. https://medium.freecodecamp.org/what-exactly-is-client-side-rendering-and-hows-it-different-from-server-side-rendering-
Accessed: 2019-05-04, Published: 2017-02-28.

[53] W3School. W3schools img tag. https://www.w3schools.com/tags/tag_img.asp. Accessed: 2019-05-10.

[54] Jeremy Wagner. Google amp. https://www.rockstarcoders.com/google-amp/. Accessed: 2019-04-23.

[55] Jeremy Wagner. What is lazyloading. https://developers.google.com/web/fundamentals/performance/lazy-loading-guidance/images-and-video/. Accessed: 2019-05-10.

[56] Wikipedia. John gruber. https://en.wikipedia.org/wiki/John_Gruber. Accessed: 2019-05-07, published: 2019-04-29.

[57] Wikipedia. Markdown. https://en.wikipedia.org/wiki/Markdown. Accessed: 2019-05-07, published: 2019-05-03.