Z-INF

Introduction

The goal of this project is to teach everyone about online privacy and security, by securing their accounts and removing unnecessary data.

Website

The website start off by giving the user the ability to select which services they use, where the user are then taken to a seperate page with a total overview of all the services the user selected. On this seperate page, the user are then presented with steps to secure their account, steps to retake their privacy and what information each service collect. The services also display who owns each service, which has been added to make the user aware who actually owns the data the user provides them.

Technologies & tools

The website is built with a static-site generator, Jekyll[1], using HTML, SASS and JavaScript. There is no back-end, nothing is rendered from the server, everything the user sees is already generated and is only displayed by their browser. The data provided by the user is stored in `localStorage` in their browser and never sent to a server. Jekyll uses a template engine called Liquid[2], which makes it possible to populate HTML and Markdown from data-files instead of manually having to update vaules for each HTML or Markdown file.

SASS[3] is a CSS post-processor, that supports partials, nesting and much more. I chose to use this as it allows for more readability and sorting, since you can now split up the CSS into different files, while only needed to reference one in the HTML.

Jekyll has the ability to populate HTML and Markdown with data-files, stored as `JSON`, `YAML`, `CSV` or `TSV`. I chose to use `YAML` for the added benefit that it can often be easier to read and understand, as the syntax is based on colons, hyphens and identations. All the different information about each service is stored in `_data/service/`, where each service has their own `YAML` file, this file is then being used to auto-generate a HTML-page from a layout-template. This auto-generation is not built into Jekyll by default, and is provided by a plugin, `jekyll-datapage_gen`[4].

There are three pages that are using data-files; the front page, the "My services" page and the overview of all services that the website has data on.

The website has been made to look good on both desktop and mobile, which is made possible with media-queries[^5] in CSS. The website scales dynamically based on the viewport of the user, making it more accessible for everyone to use. The infomration provided is identical, as the HTML-markup is not changed, only some properties of the CSS is changed to accomodate the viewport of the user.

[1] Jekyll: https://jekyllrb.com/

[2] Liquid: https://github.com/Shopify/liquid/wiki

[3] SASS: https://sass-lang.com/

[4] `jekyll-datapage_gen`: https://github.com/avillafiorita/jekyll-datapage__gen

Why I chose the technologies & tools

The reason behind chosing these technologies and tools are simple: for a project of this scale and with a purpose like this, there is no reason to use any technology other than a static-site generator (you could also create everything manually with plain HTML). It could also be looked upon as contraditing to have a backend handling all the data the user provides, as this would add the need to have a privacy policy stating how the information provided will/can be used.

As everything is stored in the browser, the user has the ability to remove their information whenever they want and they are the ones that are in control of their own data instead of a third-party. A added benefit of a static-site generator, is that the size of the website quite small (=>100kB), which makes it very fast and does not have to rely on having a fast host.

Platforms/services

Similarities in terms of service

Similarities in privacy policy

What they collect

Breach/leaked data

Why it matters

Potential attacks