# Everybody wants ~~money~~ A SIMPLE WEBSITE

As a first-year student of Architectural Design in 2015, I was taught that architecture is a profession that has the magnetizing power to get various professional bodies all together around one table. The architect, the civil engineer, the electrical engineer, the master plumber, the contractor, the construction manager, the carpenter, the interior designer, the city hall staff—all of them sitting there, having a say in the matter. Architecture is a profession inviting enough to stir a democratic debate about various methods on how to build homes.

After three years of speculating what the best spatial design for a public debate could be, and finding irony instead of debate, I voluntarily blew up my unsettled dream to become an architect. Deep down, I knew it was never going to happen.

Although web development is often compared to architecture and considered as its digital practice, there are various reasons as to why I enjoy web development more. It’s true that both produce the same formats of either a decor or a public space where things (like life) can happen. It’s true that both thrive on simplicity. However, web development responds faster to urgency and its aesthetics, and is easier to break (down).

Whether a web developer’s chosen path is to build decors or public spaces, he will always aim to build *a simple website*: the umbrella term for any website that is technically actually quite complex.

A ‘simple’ website’s domain is quite straight-forward: it directs you to the subject’s acquired title plus the domain extension. A simple website has a (hamburger) menu, is immediately accessible by search engines, loads in two seconds or less, works the same across various devices, minimizes redirects, condenses images and media, is pretty (yes), mobile-friendly, and takes about six seconds of a user’s time to understand the story it wants to tell—exactly the time of an average user’s attention span.

All of these features necessary to get the user’s attention are in fact separate instruments called upon during the process of building a website. Underneath its surface, one ‘simple’ website is a combination of several products, technologies and frameworks that give an insight into the current state of the tech market; an endless ocean of specific-purpose plugins and frameworks, joining forces to catch the highly desired attention of the user. One bug on a simple website gives an insight into the product and plugin’s state: endlessly updating itself to survive among other similar tools on the market.

Because of the many available tools, as well as the industry’s fast-updating nature, a web developer is conditioned to choose which tools (programming language and back/front-end environments, additional frameworks, plugins) he most identifies with and that he wants to settle for in the long run. The pressure to settle for the best tools available and to stay updated with their updates makes a web developer think that, perhaps, making the most ideal product choices and gaining technical mastery over the chosen tools is in fact the essence of his practice.

But is this true, I wonder? At the same time as obtaining this mastery, the tech industry throws more tools and choices at the web developer, in case he wants to reconsider the tools he’s currently using. The worldwide community of web developers contributing to the open-source tech industry led to the dissection of a web developer’s role: Junior Web Developer, Senior Web Developer, Front-end Engineer, Front-end web developer, Back-end developer, Back-end web developer, Architect, Full-stack web developer, UX/UI Designer, Solution Architect, Creative Developer, Freelance Web Developer, Happiness Engineer; all of these happy little developers branched out on to the tree that is the tech industry to welcome, rate, choose, master, and promote incoming tools and technologies related to their respective roles.

Whether situated in a company or his bedroom, a web developer’s experience of life is defined by a set of time frames, each lasting about three to six months—depending on the client’s deadline to actually start making *money*. In a single time frame, a web developer is responsible for building as well as deploying a fully defined online identity ready to travel through the market and collect digits. When one time frame ends, a new one begins. A web developer’s life is a set of perfect circles, lined up in one single line.

Due to every web developer’s tight production schedule and the technical responsibility he must fit into a single time frame, his understanding of time is, out of all things, the least experimented with. Very little time in his practice is left to question the conditions of the user’s default role, next to proposing a new role that could be rich with context and the subject’s socio-political situation, class, and culture. For example, good questions could be: Why does an e-com website need to work 24/7? Why do my parents need to be able to read everything about my past projects? Why does a landing page count down the days until the opening and not the *closing* of a festival? After ten questions, the questions get absurder.

If there’s any extra time left in a web developer’s life, it is often used for keeping up with the tech industry, holding onto the acquired tech-darlings, or acquiring new ones.

Now, imagine how many web developers there are in this world (because I will refrain from specifying). Imagine how many of them build products, use products, promote products. Imagine what happens to the product not built solely around catching the user’s attention. Imagine one user’s story getting lost in the digital ocean, solely because they opted out of using a certain product. Now, imagine (digital) skies—for the sake of keeping it light.

Everybody wants to make money; that’s why everybody wants a *simple* website (be it for their own consumption or for representation). The users are fine with not understanding the specifics behind a simple website actually being rather complex because their own role on the market usually operates with similar complexities. The company/industry/commercial ecosystem they belong to has a website, it has its experts, it has a story, it offers specific service and maintains its specific nature through mysterious recipes. On the website, a user is a *client*. Accepting the default role of the user is the same as accepting being a client and client only. There’s more to this.

An ongoing desire for a simple website shrinks the user’s capacity to think in a complex and abstract manner—more specifically, to think on a reflective level. Oversimplified online engagements make for a ‘reading [that] becomes a process of elimination rather than deep engagement. Life becomes about knowing how not to know what one doesn’t have to know.’[[1]](#footnote-2) But, also, life becomes about what Geert Lovink calls techno-voluntarism in *Stuck on Platform*—a state in which we know how to communicate the received information; how to deliver it, how to present it, how to mix it, how to explain it, how to not let it move us into taking action, how to stop it from bringing us together, how to use it for overcoming paranoia and lack of trust in strangers. The consequence of passively consuming so-called simple websites is why everybody in the office suddenly is also a manager, managing information.

Due to these circumstances, it’s hard to tell whether building a social media platform is more ethical than building an artistic portfolio, or less. A social media web developer that gives the user the possibility to express their opinion online is not to be more respected than a web developer who decides on the aesthetics of the user’s reading experience. An artist’s Instagram account is neither more or less important than their official website. Imagine the skies, imagine the skies.

1. *Program or be Programmed*, Douglas Rushkoff [↑](#footnote-ref-2)