By now I would have lots of data in front of me to **hack for weeks, even months**. However, since my first initial look was only to understand how things work and now I want to dig deeper, after going through subdomains, the last step in this section is to go back through the main web application again and to check **deeper** on how the website is set up. Yes I mean, **going through everything again**. Remember my intentions are to spend as much time as possible on this website learning everything possible. **The more you look, the more you learn**. You can never find anything on your first look, trust me. You will miss stuff.

For example, on a program I believed I had thoroughly tested I started revisiting various features and simply viewed the HTML source of endpoints I found and actually discovered they used a unique .JS file on each endpoint. These contained specific code for this endpoint and sometimes developer notes as well as more interesting endpoints. **On my first initial look I did not notice** this and was merely interested to know what features were available, parameters used etc. In an “explain like i’m 5” , I was too busy looking at Burp! After discovering this common occurrence on the target, I spent weeks on each endpoint understanding what each .js file did and I soon quickly built a script to check **daily** for any changes in these .js files. The result? **I was testing features before they were even released** and found even more bugs. I can remember one case where I found commented out code in a .js file which referenced a new feature and one parameter was vulnerable to IDOR. I responsibly reported the bug and saved this company from leaking their user data **before they released the feature publicly**.

I learnt to do this step last because sometimes you have **too much information** and get confused, so it's better to understand the feature & site you're testing first, and *then* to see how it was put together. Don't get information overload and think “*Too much going on!*” and burn yourself out.