ASSIGNMENT 3

REST APIS

WHY HTTP IS MORE SAFER TO USE AS COMPARED TO HTTPS

1. SECURITY :

The first reason to migrate to HTTPS — using an SSL certificate — is your website security, as we mentioned. Especially for e-commerce, it is crucial to have that kind of **encrypted protection** when processing payment.

To other pages hosted in Word press, HTTPS brings more security to a login page. Imagine having yours and your visitors' data compromised by criminals.

2. CREDIBILITY:

In competitive markets like e-commerce, being safe isn't enough. You have to **look safe** too. Most users don't understand well the protocols and the meaning of encryption. But when they see the green lock before the navigation bar, they know what it means. It is a matter of credibility, professionalism, and trust.

3. PERFORMANCE :

A benefit HTTPS brings to sites, and is not really well known, is that the protocol can somewhat improve a page's performance — especially making it quicker to load.

4. SEO :

We've already mentioned that Google requires the switch from HTTP to HTTPS, but that is not the only thing that the secure protocol has in its favor when talking about Search Engine Optimization.

WAYS TO CONVERT HTTP TO HTTPS :

Step 1: Buying an SSL Certificate

It is possible to buy a certificate straight from your web hosting provider, though you can get it somewhere else if you see a better deal. Knowing that some of them even offer it for free when the user is purchasing a hosting plan.

There are different kinds of certificates, and the choice depends on your needs. For e-commerce and e-learning, it is recommended to buy those with **Organization Validation (OV)** — fully authenticated.

Step 2: Checking compatibility with your website's features

One of the most important steps when migrating is to make sure the site will keep functional and running properly after the change. To do that, you have to certify that all external features your pages need are provided under the same protocol.

All external dependencies should be able to work with HTTPS, such as social networks, Google AdSense, JavaScript, embedded video, certification labels, etc.

Step 3: Preparing the migration

The complexity of the process depends a lot on your website's size and the number of existing pages. A small site can be migrated at once.

But, if you have too many URLs, you can do that in parts. For example, starting with specific subdomains where more important content and features are.

You can also enable the HTTPS protocol without disabling HTTP until everything is good. In that case, you should use Canonical Tags to avoid duplicated content.

Just remember that you will end up losing some social media engagement metrics — sharing, likes, and others.

Another point to take into consideration is when you are planning to do it. **The best timing varies from market to market.** For e-commerce, it is a good idea to avoid doing that on holidays and other seasonal dates.

To most companies, weekends or extended days off are better because it means less traffic when doing the migration.

Anyway, prepare your team's mental strength and yours to deal with unexpected problems and delays. This guide serves exactly as a way to minimize those situations.

Step 4: Enabling HTTPS

After the planning phase, it is time to start and enable HTTPS on your website.

With the protocol up and running, and the correct implementation of all the setup needed, it will be already possible to access the pages via HTTPS. You need to check if the SSL certificate is correctly installed.

To do that test, you can leave HTTP and HTTPS running in parallel for about 5-10 minutes. If nothing is broken or acting strangely, you can conclude the transition.

Step 5: Updating features to HTTPS

The HTTPS is running after step 4? Great!

Now it is time to update internal links and features. The goal is to reorganize the website's architecture, making it as lean as possible to make the search engines' work easier.

That means **avoiding redundant redirecting**, for example. At this stage, you can check Canonical Tags, as mentioned. Google thanks you and your site's SEO too!

After that, it is also important to take care of external features. The update will guarantee that your site keeps or improves loading time. It will happen because, once again, you will be avoiding redundant redirecting.

After checking internal links and external features, you can still test the certificate's implementation on the server. The test allows you to identify possible tweaks to be made, like support restriction to specific browsers.

Step 6: Adding the new version of the site to Google Search Console

You may have noticed that Google Search Console considers addresses with and without "www" as two different sites, demanding some aspects to be verified on both. The same happens to HTTP and HTTPS.

Because of that, you will have to add the HTTPS version of your site to Search Console and verify it. Just follow the instructions given by the platform. By moving your website to a new property on Google Search Console, you will also need to update any configuration required.

And, of course, it is important to **create an XML Sitemap** for the new URLs with HTTPS. An XML Sitemap helps Google and other search engines to track and index your content quicker, which makes it a crucial step for your SEO.

But have this in mind: keeping the XML with the HTTP URLs is highly recommended. Instead of deleting it, just add the new XML Sitemap for HTTPS and keep both on Search Console.

Step 7: Enabling HTTP/2 and HSTS

After making sure the HTTPS protocol is working, you have at your disposal a new feature that speeds up the loading of content on your site — HTTP/2. Actually, most browsers already have support for the format, but only when the site is under HTTPS.

While HTTP/2 can improve your pages' loading speed, HSTS tries to **prevent redundant requisitions made to servers** that operate exclusively with HTTPS. That is an advanced feature that can be used by your webmaster — as it will probably be irreversible after installed.

Step 8: Redirecting HTTP to HTTPS

While and after migrating a website to HTTPS, it is important to map URLs that were accessed via HTTP and redirect them to the new protocol.