POLYTECH GRENOBLE

UGA Université Grenoble Alpes

Contribution au logiciel EDCampus

Élisa Beaugrand
Louis De Gaudenzi

INFO4

Introduction

In an era where everyone (or almost) is connected, ecology becomes a major issue. More and more people are mobilizing to save the planet. We often talk about transport, water consumption ... But one area seems to have been forgotten: digital.
Indeed, digital pollutes and a lot, to such an extent that its expansion could be strongly questioned in the coming decades.

We propose here to study the digital pollution generated by software. To carry out this study, we asked ourselves two questions:
How to assess the impact of software? And how to improve this software?

Our project was to report on the ecological impact of the EdCampus application.

We first have read the book "115 good practices" which lists 115 programming practices to adopt for eco-design.

We then analyzed the code using existing tools that allowed us to set metrics to know what are the parts of the code that are the most expensive in resources.

And finally, we programmed an application called ecoscore which gives a score of "good practices" to the application. The ecoscore program is universal, so it can be used for any application.

Code analysis

To analyze the code of the EdCampus application, we used existing tools. These tools are W3C for the quality of the code, GreenIT-analysis to know the ecological footprint of the application or even Audit Lighthouse to study the performances. In this section, we will only talk about the GreenIT-analysis and Audit Lighthouse results and not about W3C validation because the HTML files are all valid.

GreenIT-analysis

What is GreenIT-analysis?

GreenIT-analysis is a developer extension of Google Chrome. This extension is part of "Chrome DevTools" and is accessible from the Google Chrome console, by typing F12 or by "inspecting the element".

This extension measures the digital pollution generated by a web page according to various factors. Based on the results, a score is assigned to the web page. This score is a letter from A to G, A being the best score.

The rating criteria are based on some "GreenIT good practices" that can be found in book 115 (or our excel document which lists the 115 GreenIT good practices).

The good practices examined here are:

- Add expires or cache-control headers (> = 95%)
- Compress resources (> = 95%)
- Limit the number of domains (<3)
- Do not resize images in the browser
- Avoid empty SRC tags
- Outsource CSS
- Outsource js
- Avoid requests for errors
- Limit the number of HTTP requests (<27)
- Do not download images unnecessarily
- Validate javascript
- Maximum size of cookies per domain (<512 bytes)
- Minify CSS (> = 95%)
- Minimize the ds (> = 95%)
- No cookies for static resources
- Avoid redirects

- Optimize bitmap images
- Optimize svg images
- Do not use plugins
- Provide a CSS print
- Do not use standard social media buttons
- Limit the number of css files (<3)
- Use ETags (> = 95%)
- Use standard fonts

Other factors are also taken into account:

- The water consumption generated by the page loading (in cl)
- The size of the page (ko)
- The size of the DOM
- The carbon footprint of page loading (gCO2e)

All these parameters define an **ecoIndex** (between 0 and 100), and it is this ecoIndex which will determine the letter of the score.

How did we analyze the different scores?

We first connected to the EdCampus application under all the different possible profiles: a student profile, a tutor profile and an admin profile.

We have identified the web pages being common to these 3 profiles. The common pages are:

- The login page
- The notifications page
- Help page
- The general conditions of use
- Privacy page
- Pages concerning your profile
- Logout page

The tabs *files*, *tasks*, *calendar*, *workshops*, *deliverables* and *events* on the "project" page are common to tutor and student profiles.

Finally, several other pages are specific to each profile or have a different appearance depending on the profiles:

student	**tutor**	**admin**
Home page	Home page	Home page
Summary / publications / finance tabs of the "project" page	Summary tab of the "project" page	"Information and personalization" page
"New project" page	"New project" page	"Document templates" page
Summary / publications / finance tabs of the "project" page	"New Portfolio" page	"CSV imports" page
		"Course management" page
		"User management" page
		"Project management" page
		"Business management" page
		"Campus communication" page
		"Skills management" page

We then launched the GreenIT-analysis module on each page of the application and reported the results in different Excel documents:

- Score Green-IT analysis_fibre.xlsx found in the test_fibre directory. The content of this spreadsheet is also available in the different HTML pages of this file. These results were obtained with a broadband fiber connection (100Mb/s).
- Score Green-IT analysis_4G.xlsx found in the test directory_4G. The content of this spreadsheet is also available in the different HTML pages of this file. These results were obtained via 4G connection sharing (35Mb/s).

EdCampus is also available in English for most of its web pages. So there is an "English" score and a "French" score for each web page in the results. We would also have liked to obtain the results via an open wifi such as wifi-campus, but it is impossible due to quarantine.

Results of our analysis

Fiber connection

1. The page obtaining the best score / ecoIndex is the login page with an ecoscore B and an ecoIndex of 69. This score is the same whether the page is in English or in French. This page does not contain much information and is quite simple: a login form and the EdCampus logo at the top left. This score is surely linked to this simplicity.
2. The page with the lowest score / ecoIndex is the home page under the "tutor" profile in French. Its score is a G with an ecoindex of 5. The page size is enormous (more than 8000kb) and almost none of the good practices analyzed seem to be respected.
3. On average, the application to a D- ecoscore with an average ecoIndex of 36 for the English version and 44 for the French version.
4. The "admin" profile is the one with the lowest scores with an F for almost all of its own pages.
5. The pages common to the 3 profiles are those obtaining the highest scores.

4G connection

1. Overall, the application seems to have a lower score in 4G connection than in fiber connection. This is surely due to the fact that 4G consumes more electricity than a fiber connection (source: https://www.clubic.com/energie-renouvelable/actualite-874143-empreinte-carbone-numerique-fibre-optique-4g-extreme.html).
2. The best scores are obtained in the specific pages of the "administrator" profile. This result is astonishing when we know that these same pages obtain the least good results with a fiber optic connection.
3. The scores in pages in French and in English are almost equivalent each time.
4. On average, the application in French obtains an E score and in English an F score (which is bad, whether E or F).

Conclusion of the analysis

According to the results of the page-by-page analysis of the EdCampus application, via the GreenIT-analysis plugin, we can see that the code seems fairly well optimized but that the score depends on the type of connection used. To make the application more "ecological" according to the criteria of the plugin, you should avoid resizing images directly in the browser, avoid loading images that are not displayed on the page and think of outsourcing css and js. All these actions can lighten the application. Providing a print css, a printable

version of the documents online can also be an ecological action, since it allows a printing on paper optimized and therefore less expensive in natural resources.

LightHouse Audit

What is Lighthouse Audit?

Lighthouse is an audit extension for Chrome. Lighthouse is a very useful tool for those interested in the performance and quality of web applications. To generate an audit, it is possible to launch Lighthouse via Chrome DevTools, from the command line or as a Node Module. The principle is simple, Lighthouse takes as argument the URL of the application to be examined and generates a results report. The main criteria are performance, accessibility, progressive web apps, SEO. This tool is quite interesting because it does not only point to the failures of an application, but also furnishes documentation that explains how to resolve these failures.

Lighthouse on EdCampus

We ran the Lighthouse tool on the EdCampus application. We will report here about the audit report and try to deduce a possible ecological impact when it is not explicitly mentioned in the report.

Performance

The score of **0/100** is assigned to the EdCampus application in terms of performance. Given the results, the application seems very slow. With a fiber connection of more than 100Mb/s, the first contentful paint (text or image) to be loaded was loaded in more than 43s and the first meaningful paint appeared in 44,4s. The application content was visibly populated in about 48s. In a little more than 44s the main thread was quiet enough to handle input and a maximal first input delay a user could experience was 340ms. The statistics shows that the app was fully interactive in about 48s. These performances are way too slow and a slow loading means a bigger resource consumption. Lighthouse's advice for improving performance is to remove unused css, and eliminate render-blocking resources. These two actions could save loading times by around 50s. 321 resources were found to be static assets. An efficient cache policy with a long cache lifetime could speed up repeat visits on the app and by the way serve these static assets. The main-thread work could also be minimized by considering to reduce the time spent parsing, compiling and executing JS. It is estimated that 16s could be saved.

Accessibility

EdCampus scores a grade of **88/100** in terms of accessibility, which is a pretty good grade. The faults found here are mainly problems of contrast between the background and the text and referencing problems on links and labels. These problems can therefore be solved quite simply to make the application accessible to the greatest number of people. Nevertheless, these modifications will have no ecological impact.

Best practices

The score for the respect of best practices is **79/100** It seems that 325 requests were not served via HTTP/2, however it is known that HTTP/2 offers many benefits over HTTP/1.1, including binary headers, multiplexing, and server push. HTTP/2 serves the application resources faster with less data moving over the wire. It is a great and easy way to reduce the ecological footprint of the application. The function *document.write()* in invocated in a js. External scripts dynamically injected via *document.write()* can delay page load by tens of seconds for user with a slow connection. Lighthouse indicates that Chrome therefore blocks the execution of *document.write()* in many cases, meaning you can't rely on it (source: https://developers.google.com/web/updates/2016/08/removing-document-write). It is therefore useless to try to use this function and therefore consume resources.

Progressive Web App

The application does not work online and once again, app load is not fat enough on mobile networks. As said in the **Performances** part, the app began to be Interactive at about 48s.

Conclusion

This analysis highlights the fact that the app loading is too slow. The more the load is slow the more it consumes. So it is the main aspect to improve to respect a GreenIT approach.

Ecoscore

Ecoscore is a program that scans through a web app directory and analyses files of code.

It judges the app's eco-friendliness by looking at common recommended practices for resource saving.

It will give a score to the app depending on how much it respects these practices.

Requirements

- `python3`
- `nltk` package for `python3`: install with `pip3 install --user nltk`

Usage

Synopsis

`python3 ecoscore.py [OPTION]... DIRECTORY`

Description

Analyses and gives a score to the app located in `DIRECTORY`.

Mandatory arguments to long options are mandatory for short options too.

- `-v`, `--verbose` print a message for each file analysed
- `--help` display help and exit

Description

Ecoscore's analysis is based on the book *Éco-conception web : les 115 bonnes pratiques*, written by Frédéric Bordage.
Currently, it tests for several recommended practices the book gives :

PHP		**JS**	
Number in the book	Title	Number in the book	Title
64	Don't call functions in declaration of for loops	34	Avoid using try … catch … finally
69	Use simple quotes instead of double quotes	35	Use primitive operations
70	Replace $i++ by ++$i	38	Favour the use of anonymous functions
73	Never use SELECT * FROM	39	Use the string concatenator in an optimal way
		40	Prefer functions to strings in argument to setTimeout() and setInterval()
		41	Avoid for … in loops
		50	Use GET method for Ajax requests

We chose to write the program in Python, because of the simplicity of the language to scan through a directory and parse many files. We have been able to setup a working version of Ecoscore very quickly thanks to this aspect, and have chosen which recommended practices we could code easily.

Conclusion

This project allowed us to learn a little more about what is called "GreenIT". We believe that this eco-design approach will become a major challenge in the IT world in the coming years.

At the moment, few documentations/experiments have been carried out on the subject. We therefore based ourselves on the book "115 good practices" but we do not really know if these practices, once implemented, would be really effective against digital pollution. Likewise, certain practices, if they were functional, would make applications less accessible, with a very poor GUI in favor of very basic web pages. A balance must be found between GreenIT good practices and a pretty interactive application.
We would have liked our ecoscore application to be able to test many more GreenIT practices, but either we have not succeeded in implementing them, or they can already be tested via other applications with much better precision than what we could have provided.

Links

Airimag

https://air.imag.fr/index.php/Contribution_au_logiciel_EDCampus

Ecoscore-parser Gitlab repository

https://gricad-gitlab.univ-grenoble-alpes.fr/Projets-INFO4/19-20/16/ecoscore-parser

Documentation Gitlab repository

https://gricad-gitlab.univ-grenoble-alpes.fr/Projets-INFO4/19-20/16/docs

Demonstration video

https://gricad-gitlab.univ-grenoble-alpes.fr/Projets-INFO4/19-20/16/docs/-/blob/master/Presentations/Final%20Presentation/Demonstration%20Video.mp4