Automatic removal of cookie banners on websites

Sunday 15th January, 2023 - 15:05

Rui Filipe Giesta Gonçalves
University of Luxembourg
Email: rui.giesta.001@student.uni.lu

This report has been produced under the supervision of:
Luis Leiva
University of Luxembourg
Email: luis.leiva@uni.lu

Abstract

This paper presents the third Bachelor Semester Project carried out by Rui Filipe Giesta Gonçalves with Luis Leiva as his tutor where the goal was to find the best way to identify cookie banners that often overlap the content of the website and remove them. In order to do so, a browser extension was created which automatically injects some JavaScript on webpages in order to fetch the cookie banners and properly removing them without impairing the performance of those webpages and breaking existing functionality. The scientific part of this project was to understand why these cookie banners exist and how they are implemented so that one could access and properly isolate them without affecting any other elements of the pages.

1. Introduction

Cookies are pieces of information that websites store on computers so that they can be accessed later to present the user with information customized to fit his needs. These pieces of information can be for example login status, preferred language, location, etc. Most of the times, cookies allow the website to recognize an existing customer when they return to that website and to give him a more convenient experience. However, in Europe, according to the General Data Protection Regulation (GPDR) [1] and the ePrivacy Directive (EPD), in order to use those cookies websites must receive users' consent. This regulation induced the proliferation of cookie consent pop-ups across all European websites that uses cookies, which is a burden for users as they need to accept or decline them, which impacts the browsing experience by reducing the usability of most of the websites.

The main goal of this project is to find the best way to identify these cookie banners that often overlap the content of the website and remove them by creating a browser extension compatible with Chrome that automatically injects some JavaScript that is executed after the entire load of the page in order to fetch the existent cookie banners and then finally remove them.

However, these kind of banners are implemented in many different ways as there is no mandatory method to do so. It is important to understand how this content is treated by the web pages and how one can automatically access and isolate it to preserve websites' functionality when removing them. Therefore, a further research on the General Data Protection Regulation (GPDR) and the ePrivacy Directive (EPD) as well as the general cookie banner content was conducted so that it would allow to properly understand what normally these banners have or is presented to the users to find something common between them so that one could automatically detect it. Nevertheless, three main aspects were considered, namely "Consent message", "Consent options" and "Consent policy", which are fundamental to identify the kind of content that the banners have.

Additionally, the implementation of the fetching and removing of those banners as well as the performance of the extension was also a fundamental aspect to consider as it is related to avoid breaking existing functionality and reducing the time needed to remove the banners so that the user has a better browsing experience. Therefore, a further exploration of the different methods and ways to implement it was mandatory which allowed to improve the extension's efficiency and performance.

2. Project description

2.1. Domains

2.1.1. Scientific.

Data Protection Regulation. Data protection can be defined as the process to safeguard personal information from corruption, compromise or loss. Nowadays, its importance has been increasing significantly as the amount of data created, stored and utilized keeps growing at a fast rate. Due to this, the European Parliament and Council of the European Union created the General Data Protection Regulation (GDPR) which regulates the "processing of personal data relating to

individuals of the European Union by any individual, company or organization" [1]. This regulation defines the types of data handlers, their privacy guidelines and also the breach notification, where all organizations must "report instances of data breaches" and "inform individuals impacted by the breach".

Web accessibility. Following on our previous discussion about identifying the cookie banner contents, isolating this kind of content is important not only to be able to properly remove them but also to not harm any other element of pages, assuring that the existing functionality remains untouchable and its performance is not impaired.

2.1.2. Technical.

JavaScript. JavaScript is a "lightweight, interpreted, object-oriented language with first-class functions and is best known as the scripting language for Web pages". It is a "prototype-based, multi-paradigm scripting language that is dynamic, and supports object-oriented, imperative, and functional programming styles" [2]. This language is fundamental to this project as it allows to retrieve the elements and its properties that need to be inspected and changed if needed, allowing to proceed with several computations such as contrast calculations, color model conversions and attribute setting.

Node.js. Node.js is an open source server environment and a back-end JavaScript runtime environment. It uses asynchronous programming and is able to perform several tasks, namely: generate dynamic page content, manage files on the server, collect form data and manage data on databases [3]. This runtime environment is used in this project by the screenshot application in order to evaluate one's approach to the cookie banner removal which allows the injection of the JavaScript code before taking the screenshot in order to compute the differences between before and after that injection.

2.2. Targeted Deliverables

2.2.1. Scientific deliverables. In this project, the question "How to detect non-essential content that is occluding essential parts of a website?" is answered by firstly defining what kind of content this is according to the General Data Protection Regulation (GDPR). Thus, it is defined what is the data protection and why there is such regulation as well as why it is a burden for the users that want a clean and better browser experience. Secondly, some ways of inserting this content into webpages are outlined considering three main aspects, namely "Consent message", "Consent options" and "Consent policy" which allowed to identify the common elements between the different implementations and find the best way to reach them. Lastly, understanding the best approach to successfully and efficiently inject the script to target the correct content and remove them was fundamental as well as deciding and using the best methods, properties and functions to improve its functionality without affecting the page's performance.

2.2.2. Technical deliverables. In this paper, FreeTheCookie, a browser extension compatible with Chrome and Firefox capable of automatically remove the cookie banners that often overlaps the original content of websites due to the General Data Protection Regulation, is developed. To do so, some JavaScript is injected that traverses all elements of the page and tries to find these kind of content and remove it. The extension starts in the very first node of the page and checks if that node has a shadow root or contains some of the specific keywords that are most common in this type of content. Then, if it does found, it will traverse all the child nodes of the node that had one of those keywords while checking if there exits an "accept button" and triggers a click event on them when they are found.

Lastly, in order to evaluate this approach, a screenshot application was used which takes a screenshot of the website before and after the removal of those banners to compute the differences between them.

3. Pre-requisites

3.1. Scientific pre-requisites

Before starting this bachelor semester project, some background knowledge about "cookies" and data protection is required in order to properly understand why there exists such banners and why it impacts the browsing experience of many users. Also, some basic knowledge about web accessibility is required as it is important not only to understand how this kind of content can impact the readability of websites but also to prevent breaking existing functionality when applying changes to it.

3.2. Technical pre-requisites

Before starting this bachelor semester project, some preliminary knowledge and familiarization with front-end web technologies is fundamental as it is the main domain of the technical part of this project since JavaScript is the main language of the extension as it is responsible to traverse all the elements of the page, fetch for the desired content and trigger the event to remove it. Thus, it is mandatory to have good knowledge and proficiency with this language in order to be able to complete the project and produce a reliable and optimized extension to return good results and reach the project goals.

4. Scientific Deliverable

4.1. Requirements

This scientific deliverable covers several aspects such as: Data Protection, the General Data Protection Regulation,

cookie consent and web accessibility. Its main scientific goal is achieved by answering the question "How to detect non-essential content that is occluding essential parts of a website?".

In order to answer this question it is important to firstly understand what typically kind of content overlap the content of a website and occludes it, namely the cookie consent banners. Therefore, it is strictly necessary to understand why the General Data Protection Regulation exists and how it works by defining its concept and what guidelines organizations should follow in order to comply with it. Also, it is important to acknowledge what is the Data Protection and why the GDPR is based on it and why this has become a burden to users when surfing through the Internet.

Secondly, it is fundamental to properly understand how the cookie banners are implemented on websites in order to be able to properly remove them. Therefore, a further research is required on the composition of those consent banners to define its content, namely: "Consent message", "Consent options" and "Consent policy".

Lastly, since the main purpose is to find a way to reach these banners, isolate them and remove them without interfering with the rest of the websites' content, it is important to explore the different methods and properties that can be used when injecting the script not only to preserve the websites' performance but also to properly reach the goal.

4.2. Design

This scientific deliverable was produced mainly under a detailed research made through the "General Data Protection Regulation" website and its white paper and the documentation of the JavaScript language in order to find all the information to support the understanding of the different aspects of this project such as: Data Protection, Data Protection Regulation, cookie consent banners implementation and web accessibility.

Data Protection. The Data Protection is important to be acknowledged as it is the main reason why the cookie banners exists due to the regulation created to protect users. Nowadays, Data Protection has becoming more and more important as there are a lot of personal data breaches that often have huge impacts on people's life and also organizations. Therefore, it was important to give its proper definition and how it is directly related to the General Data Protection Regulation and to the Fundamental Rights of the European Union.

General Data Protection Regulation. Similarly to the Data Protection, the General Data Protection Regulation is important to understand why there is the need for organizations to implement the cookie banners to comply with the regulation and protect users' data. Thus, there are some guidelines present in the GDPR that they must follow in order to properly comply with it and avoid further complications and that should be explored in context to this project. Regarding this, there are some of them that explain how data should be handled and what measures should be adopted in order to promote data privacy. Therefore, a further understanding on how organizations implement those measures is important for this project.

Cookie consent banners. As mentioned before, one of the main objectives of this project is also to find the best way to target the cookie consent banners, isolate and remove them. Thus, the composition of the cookie banners should be explored to define their main content so that one can find common elements between the different implementations and automatically remove the banners. Therefore a further research through the HTML and JavaScript documentation was also fundamental to find the best properties, methods or functions that should be used to reach that goal.

Web accessibility. As one of the main goals of the project is to properly remove the cookie banners without affecting any other elements of the page, it is important to make sure that when the script is injected the performance and the content of the page remains unharmed. Therefore, a detailed examination of the different ways of traversing the elements of the page and selecting those that are necessary to be removed was required as well as the different methods to improve the performance of the script.

4.3. Production

As referred before, this deliverable is based on some main related topics, namely the Data Protection, the General Data Protection Regulation, cookie consent and web accessibility. The first one concerns the protection of personal data and why it is important, as it is the main reason why the General Data Protection Regulation was created. The second one concerns the regulation that was created in order to promote the Data Protection as it regulates the processing of personal data. It is subdivided into two topics: data handling and data protection measures. These are important to define not only who processes the data but also how it should be done. The last topic concerns the presence of cookie banners, why they are needed in order to provide users' consent for the website to use their personal data and what is their content as it is important to understand what these banners include and how to target them.

4.3.1. Data Protection. As already mentioned before, data protection can be defined as the process to safeguard personal information from corruption, compromise or loss. According to the General Data Protection Regulation [1], "personal data" is related to "any information relating to an identified or identifiable natural person" who, in its turn, can be identified directly or indirectly by an identifier such as "a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person." The main goal of data protection is, therefore, to prevent any personal data breach which leads to "accidental or

unlawful destruction, loss, alteration, unauthorised disclosure of, or access to, personal data transmitted, stored or otherwise processed". According to the Article 8(1) of the Carter of Fundamental Rights of The European Union and Article 16(1) of the Treaty on the Functioning of the European Union, everyone has the right to the protection of their personal data, making data protection a fundamental right. Thus, it is mandatory to have the consent of the user when his or her data is going to be processed.

4.3.2. General Data Protection Regulation. Following our previous discussion about the data protection, since it is considered a fundamental right, the European Commission developed the General Data Protection Regulation which regulates the processing of personal data relating to every individual in the European Union by any individual, company or organization. Thus, any entity in the EU that collects, stores or processes someone's personal data must comply with this regulation, except, for example, "competent authorities for purposes of prevention, investigation, detection or prosecution of criminal offences or the execution of criminal penalties."

Data handling. According to the General Data Protection Regulation [1], there are two types of data handlers: "Processors" and "Controllers". A controller is "the natural or legal person, public authority, agency or other body which, alone or jointly with others, determines the purposes and means of the processing of personal data." The purposes and means of such processing are determined by Union or Member State law. A processor is "a natural or legal person, public authority, agency or other body which processes personal data on behalf of the controller." Thus, controllers define how data should be processed but also ensure that it complies with the regulation.

Data protection measures. In order to promote data privacy, this regulation imposes two data protection measures: "Privacy by Design" and "Privacy by Default". The first measure requires that organizations design procedures, policies and systems are complied with the General Data Protection Regulation from the initial stages of the product or process development as it "takes into account the context and scope of processing along with the implications." The second measure obliges data controllers to ensure that the personal data that is collected, stored or processed is "utilized only for specified purposes by implementing appropriate measures at the organization level."

4.3.3. Cookie consent. As already mentioned before, personal data also encompasses the presence of an online identifier that identifies a natural person. Cookies are a type of these online identifiers and being a part of personal data must require consent. Consent is defined as "any freely given, specific, informed and unambiguous indication of the data subject's wishes by which he or she, by a statement or by a clear affirmative action, signifies agreement to the processing of personal data relating to him or her" [1]. The General Data Protection Regulation combined with the ePrivacy Directive require that users must give consent before websites can collect cookies in their browsers. Therefore, every website based in Europe or with visitors from Europe, must have a cookie consent banner that informs about those cookies and users can choose whether to accept or decline them. However, this becomes a burden for users that want to have a clean and pleasant browser experience without having to deal with those banners all the time.

Consent message. In order to have a proper banner and to comply with the GDPR and the ePrivacy Directive, the cookie banner should include some text that informs users that the current website stores cookies and why that website uses them. It should be written in a clear and understandable language (according to its region or English for global).

Consent options. Since the main purpose of the cookie banner is to receive the consent from the user that the cookies may be used, then it is fundamental to give the user the option to accept, decline or even manage the cookie settings when they visit the website. Having the option to manage the cookie settings is important as it allows users to give different types of consent to specific cookie categories.

Consent policy. The cookie policy consists of a description of the website's use of cookies, their categories and how users can manage them. It is important when there are several types of cookies or when its use is more sensitive.

According to this kind of content found on most of the websites and since they have a lot of different implementations, the best way to target these cookie consent banners is to fetch for elements with the keywords "cookie" or "analytics" and then traverse their children to find the button to accept the cookie consent, which normally contains the keywords "accept", "agree", "allow", "consent" and "ok".

4.3.4. Web accessibility. Selecting and using the best ways to properly traverse through the different elements of the page, select the ones related to the cookie banners and remove them is also one of the main goals of the project. Therefore, understanding the best approach to successfully implement the automatic removal without interfering with any other elements is fundamental. Thus, after a further inspection on the different implementations by the most popular and visited websites, it was acknowledged that using a "TreeWalker" and the "includes" method is the best way to achieve the project's goal as it is efficient and performance-friendly.

To conclude, in order to detect the non-essential content that occludes essential parts of a website, one must first define that type of content, then find the common elements that appear on the different implementations by inspecting a good sample of different websites that implemented that content and, finally, target those common elements. In this particular case, to target the cookie consent banners one should look up for the keywords "cookie" and "analytics" which are the

most common words that appear on the consent messages that are present on those banners, which allows us to target and afterwards remove them by accepting the consent.

4.4. Assessment

As stated before, one of the main objectives of the scientific part of this bachelor semester project was to perceive how to automatically detect non-essential content that often occludes essential parts of a website. In order to do so, it was important to firstly define what is considered as "non-essential content" and how it is structured within a webpage. According to this and regarding this project, cookie consent banners are the main non-essential content that normally overlaps the content of a website. Therefore, it was fundamental to research the composition of those banners so that one could find any common elements between them as an efficient way to target that content automatically. This research turned out to be successful as it was clear to understand how the banners are composed and one was able to find the common keywords to select them automatically. Moreover, another main objective of the scientific part of this project was to find the best way to properly remove this type of content without impairing the performance of websites and avoiding removing something that was not intended to. Therefore, after a further research over the elements of the cookie banners, one found that the best way to properly and efficiently remove these banners was by accepting the consent. In order to test the effectiveness of this approach, the author ran the extension on the top 20 most visited websites in Europe, which are:

- Google.com
- Youtube.com
- Facebook.com
- Twitter.com
- Instagram.com
- Yahoo.com
- Whatsapp.com
- Netflix.com
- Tiktok.com
- Reddit.com
- Office.com
- Microsoft.com
- Linkedin.com
- Samsung.com
- Bing.com
- Weather.com
- Twitch.tv
- Discord.com
- Zoom.us
- Roblox.com

The goal of testing these top 20 most visited websites was to compute the percentage of successful removed cookies, which turned out to be 20/20 (100%). Thus, this main objective was also achieved and turned out to be successful as well.

The author spent some significant time researching, reading and choosing the various sources in order to select the best information that would match the objectives of this project and to acquire the required knowledge to answer the proposed question. The knowledge was acquired and the question was successfully answered and therefore the scientific goals of this bachelor semester project were all achieved.

5. Technical Deliverable

5.1. Requirements

5.1.1. Functionalities. These functionalities are the main features of the extension, that is, what the extension should do and what to expect to have or see when using it.

- **Cookie banner removal**: When active, the extension should check all the elements of the page and automatically detect and remove any cookie banners that websites may have. If there are no cookie banners, it should not apply anything to any other element. The extension will automatically accept the cookie consent.
- **Configuration file** : The extension should have a configuration file so that the users can include any keywords they feel like it would help removing future cookie banners not only for detecting the banners but also for their buttons.

5.1.2. Qualities. These qualities define how the extension should be and which attributes it should have to enhance its functionality and in consequence its performance and results. These attributes are fundamental for the extension to perform as desired and to reach the project's goals.

- **Usability**: The extension should be easy to install and use as well as its features should function well based on what one predicted. In order to improve its usability the extension automatically is turned on after installing and the user can enjoy the experience of browsing without any cookie banners popping on the screen.
- **Compatibility** : The extension should be compatible with both Firefox and Chrome browsers. It is intended that the users can use the extension while browsing through these two different browsers without having compatibility issues.

5.2. Design

For this project, a browser extension called **"FreeTheCookie"** was created which automatically removes the cookie banners that often overlaps the content of websites and becomes a burden for users since they have to keep accepting them every time they visit a new website and are prompted with them. This extension is compatible with both Firefox and Chrome browsers and it is intended to work on every website without impairing its performance and providing a great browser experience for the user. Therefore, in order to be able to have the best performance results and also successfully remove all the intended cookie banners, it was necessary to go through the JavaScript documentation to find the best methods and properties to be used.

First, the manifest file was created which is essential to give the extension its name and version and also the manifest version in order to be compatible with both Chrome and Firefox. In this file is also included the other files that contains the scripts to be injected so that when the extension is running they are injected and automatically fetches the cookie banners and removes them.

Then, the JavaScript file was created and the functions were developed under several steps. First, a main function was created to generate a TreeWalker object which will iterate over all the elements of the page while checking if they have any keywords previously defined as the most common ones: “cookie” and “analytics”. In case a node contains one of those words or it finds a shadowRoot it will call another function in order to go through all of its children. This main function uses a “while loop” with the condition that the next node is not “null” so that it will only stop when it reaches the final node. Also, it uses a “for loop” to try to match any word from the list with the current node.

Afterwards, it was created the recursive function that iterates over all the children of the current node that is being checked. If there is no children then it returns. Otherwise, using a “for loop” to go through every children, calls another function to check if that child node has a button. The function is recursive as for every child that it iterates it also calls itself to check if the current node also has children so that all the nodes are completely checked.

Later, one more function was created to check if the current node has a button and, using a “for loop”, tries to match that node with the “button” element or the “role=button”. If it finds a match, then, once more using another “for loop” tries to match that button with the keywords previously defined as the most common to accept the consent: “accept”,“allow”, “agree” and “consent”. All of the functions are related to each other and executed simultaneously after the main function is first executed.

Lastly, two more aspects were important for the better results of the extension as they were responsible for allowing the users to edit the keywords without having to edit the code itself and also to properly work on websites that require a bit more time to load due to their heavy components. Therefore, it was included in the manifest file a new content script JavaScript file called ”config.js” which includes the two main lists of keywords for the extension to work. Also, it was added a load event to set a timeout of 500 ms before executing the first function to let all the components load before executing the script.

5.3. Production

The production of the technical deliverable began with the definition of the main features of the extension such as: browser compatibility, JavaScript injection and functionality, detection and automatic removal of the cookie banners and possibility for the user to change or add keywords to be used by the extension to target those banners and their buttons. In order to help structuring the flow of the extension a flowchart was created (fig. 1) to help understanding how the extension is supposed to work.

The first step to start developing the extension was to create the manifest file which is a JSON file format that gives information about the extension to the browser and how it should behave when installed. In this file it is specified some basic metadata about the extension namely the name, description and version of the extension and also the content scripts which are the ones that are going to be injected when the extension is running. In order to have the scripts to be injected on every website it was used the “matches: [“<all_urls>”]” method which allows the extension to run on every URL (Uniform Resource Locator).

After completing the manifest file, the JavaScript file was created (“freethecookie.js”) in order to iterate over all the elements of the page and their children and trying to match them with some keywords previously defined and mentioned before. Therefore, after a further research on the JavaScript documentation it was decided that the best approach was to generate a TreeWalker which is an object that represents the nodes of the website and their position. This is important as it allows to go node by node and checking what is needed meanwhile.

Thus, a function called “main” was created which takes as argument the starting node, which in this case is the “document.body”. Then, the TreeWalker was created using the “createTreeWalker()” method and since all the nodes are important to track those cookie banners the node filter was set to “SHOW_ALL”. After defining the TreeWalker, a “while loop” was used with the condition that the next node is not “null” so that it will only stop when it reaches the final node as when there are no nodes the “nextNode” is set to “null”. Then, inside the loop, it verifies if the current node has any “shadowRoot” and if positive it will call the function to go through all the children of that “shadowRoot”. Otherwise, using the “textContent” property followed by the “toLowerCase()”, it extracts the text content of the node and converts it to lowercase and using a “for loop” it checks if any of the previous defined keywords are included in that text content using the “includes()” method. It is important to previously convert the text content to lowercase as this method is case-sensitive. If it is found then it will also call the function to go through all the children of that node as it is a potential node of a cookie consent banner. Nevertheless, if there is a match or a “shadowRoot” is found, the iteration still continues through the next node, due to the “while loop”.

Then, since it is important to go through every single element of the page because one cannot predict where the cookie banner can be found for every website, a function called “allchilds” was created which takes a node as input and returns and stops if that node does not have children. Otherwise, using a “for loop” and iterating over the length of that node’s children, for every child it calls the function to check if that node has a button which may be used to accept the consent. In order to be sure that every child is traversed,

the “allchilds” function is recursively called on the current child to check if that child has children until it returns when there is no more children.

Lastly, since the final goal is to accept the cookie consent banner, it was necessary to create the function “checkbtn” which takes a node as input and checks if that node has a button. If it matches with a button then it checks if that button includes one of the keywords defined earlier to trigger a click event on it. Therefore, a “for loop” was used to firstly check if there is a button element or an element with “role=[button]” using the “matches()” method. If a match is found, then the text content of that node is going to be extracted with the same previous property “textContent” and once more converted to lowercase. Then, using another “for loop” to iterate over the list of keywords that commonly appear on the buttons to accept the consent, it checks if the text content of the found button includes any of those keywords and if positive then it will trigger a click event on that button.

However, after testing the extension on the top 20 most visited websites in Europe, it was noticed that many of the websites consume some time loading all of its content due to heavy components. Therefore, a timeout needed to be added so that the extension would wait some time after the load event of the page. Therefore, an event was added using the “window.addEventListener” method with “load” as the type and using an arrow function to apply the “setTimeout()” method so that it would wait 500 ms before calling the main function on the body of the page for the first time and start iterating over the website’s nodes.

At last, since one of the functionalities expected for the extension was to be able for users to change or add keywords to the keywords list that was previously defined, a configuration file called “config.js” was created which contains the two main keywords lists used by the extension as two different variables. The first list called “wordlist” consists of the keywords that are used to try to find the nodes that are related to the cookie banners and the second list called “checklist” consists of keywords that are used to check if the button includes any of them in order to trigger the click event. However, in order to be able to access those variables in the main JavaScript file, this configuration file was added to the content scripts of the manifest file so that it would also be injected, setting the variables as global so that they can be used by the functions.

5.4. Assessment

The main objective of the technical part of this project was to create a browser extensive compatible with Firefox and Chrome browsers that would allow the user to browse through every website without being prompted with cookie consent banners. This extension was intended to automatically detect and isolate the cookie consent banners in order to properly remove them allowing the user to easily include more keywords that they feel that would help removing the infinite variations of those banners. Nonetheless, the extension shouldn’t impair the performance of the different websites nor breaking existing functionality.

The automatic removal of cookie banners of the extension was successfully achieved as it is possible to browse through many different websites without being prompted with those banners that often occlude the main content of the websites and oblige the users to keep accepting or refusing them which impacts users’ browser experience. However, there might be some limitations for some websites that are not optimized and take longer to load their content where the cookie banners are only available after the script is already injected, causing the failure on removing it automatically. Nevertheless, this situation is rare and nowadays most of the websites do not have this problem.

Additionally, the configuration file has been successfully implemented as well. This functionality is also very important for users that want the best browser experience possible and within usage can detect other keywords that sometimes are not very common but may impact their experience and they can add them to the keyword lists. Using a different file to allow this functionality turned out to be better as the user does not have to interfere with the main code of the extension so it avoids breaking the existing functionality.

The extension is easy to install and easy to use, as the changes are implemented right after enabling the extension and, therefore, the usability was achieved. Additionally, it is also compatible with Firefox and Chrome, thus the compatibility was also achieved.

Overall, all the functionalities and qualities have been accomplished and the extension works properly by removing all the cookie consent banners automatically and promotes a good browser experience for users. Therefore, all the main objectives of this technical part were achieved.

Acknowledgment

The author would like to express his gratitude to his project academic tutor (PAT) Luis Leiva for his time spent to help the author through the whole project as well as his kindness to always be available to answer any questions or assist during the development of the project. Additionally, would also like to thank the University of Luxembourg for providing access to resources and facilities that greatly assisted in the completion of this work.

Finally, the author would also like to thank his family, girlfriend, friends and colleagues for their support and love.

6. Conclusion

As already mentioned before, the main goals of the project were achieved as all of the requirements of the extension were achieved and the proposed scientific question was properly answered. The extension works properly and automatically removes the cookie banners from all websites without breaking any existing functionality and also allows users to even add more keywords to be checked within the purpose of targeting

those banners. Moreover, it was possible to understand why the non-essential content often overlaps the main content of the website and why it is a burden for users.

The author is satisfied with the results of the project and thinks that this third bachelor semester project was important not only to improve his knowledge of front-end technologies but also to better understand and enhance the possible relations between the different methods and properties used in the project. Moreover, it was also possible to understand why there are such regulations as the one studied in this project (GDPR) and why it is responsible for this kind of non-essential content. It allowed not only to improve the scientific perception of the importance of data protection but also to understand how this type of content is implemented on the different websites to push users to be aware of them and interact with them.

7. Plagiarism statement

I declare that I am aware of the following facts:

- As a student at the University of Luxembourg I must respect the rules of intellectual honesty, in particular not to resort to plagiarism, fraud or any other method that is illegal or contrary to scientific integrity.
- My report will be checked for plagiarism and if the plagiarism check is positive, an internal procedure will be started by my tutor. I am advised to request a pre-check by my tutor to avoid any issue.
- As declared in the assessment procedure of the University of Luxembourg, plagiarism is committed whenever the source of information used in an assignment, research report, paper or otherwise published/circulated piece of work is not properly acknowledged. In other words, plagiarism is the passing off as one's own the words, ideas or work of another person, without attribution to the author. The omission of such proper acknowledgement amounts to claiming authorship for the work of another person. Plagiarism is committed regardless of the language of the original work used. Plagiarism can be deliberate or accidental. Instances of plagiarism include, but are not limited to:
  1) Not putting quotation marks around a quote from another person's work
  2) Pretending to paraphrase while in fact quoting
  3) Citing incorrectly or incompletely
  4) Failing to cite the source of a quoted or paraphrased work
  5) Copying/reproducing sections of another person's work without acknowledging the source
  6) Paraphrasing another person's work without acknowledging the source
  7) Having another person write/author a work for oneself and submitting/publishing it (with permission, with or without compensation) in one's own name ('ghost-writing')
  8) Using another person's unpublished work without attribution and permission ('stealing')
  9) Presenting a piece of work as one's own that contains a high proportion of quoted/copied or paraphrased text (images, graphs, etc.), even if adequately referenced

  Auto- or self-plagiarism, that is the reproduction of (portions of a) text previously written by the author without citing that text, i.e. passing previously authored text as new, may be regarded as fraud if deemed sufficiently severe.

References

[1] European Parliament and Council of the European Union "General Data Protection Regulation" Available at https://gdpr.eu/tag/gdpr/

[2] Mozilla Firefox "About JavaScript" Available at https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript

[3] W3schools "Node.js Introduction" Available at "https://www.w3schools.com/nodejs/nodejs_intro.asp"

8. Appendix

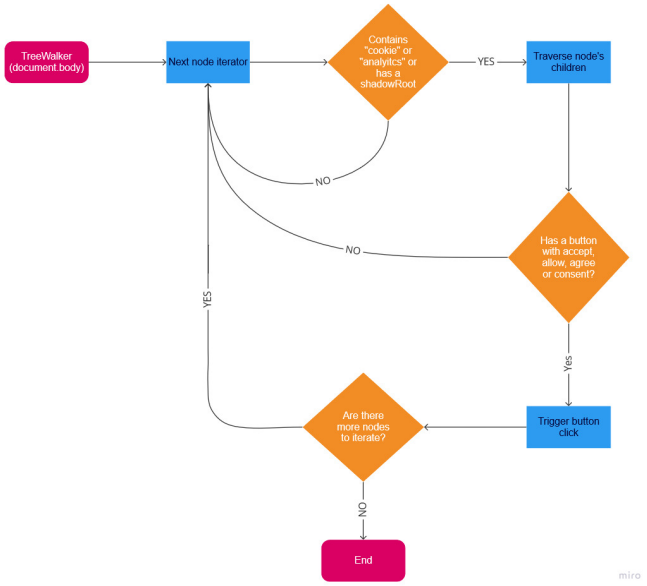

Fig. 1. Flowchart that represents the extension

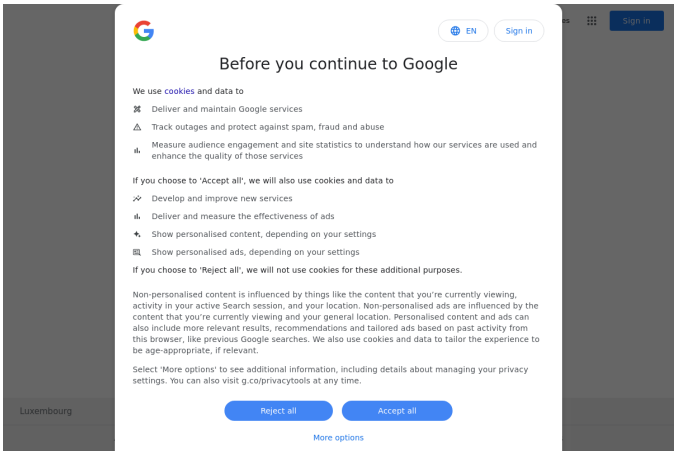

Fig. 2. Google.com - extension disabled

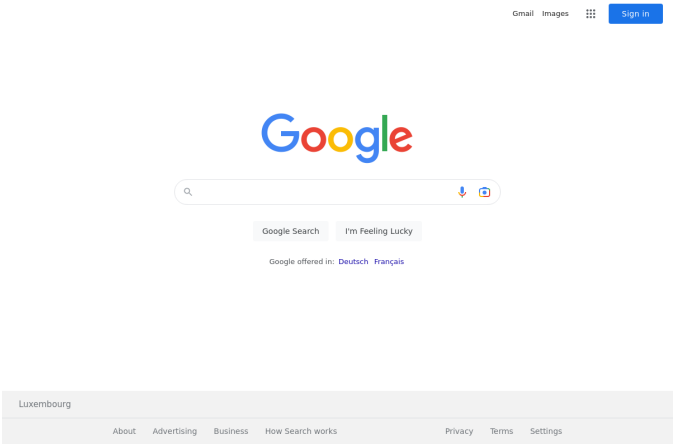

Fig. 3. Google.com - extension enabled

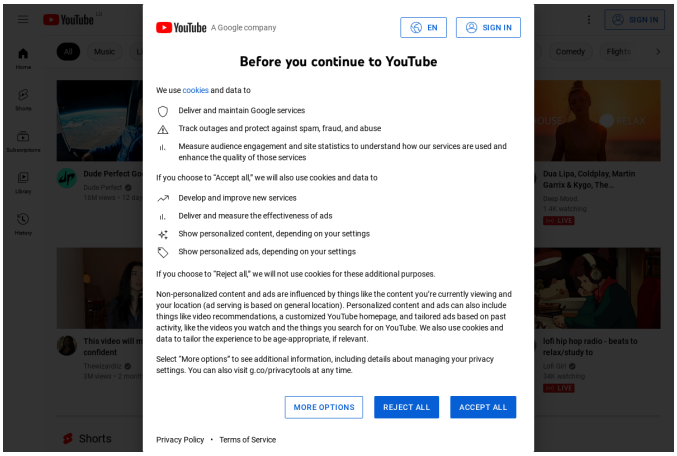

Fig. 4. Youtube.com - extension disabled

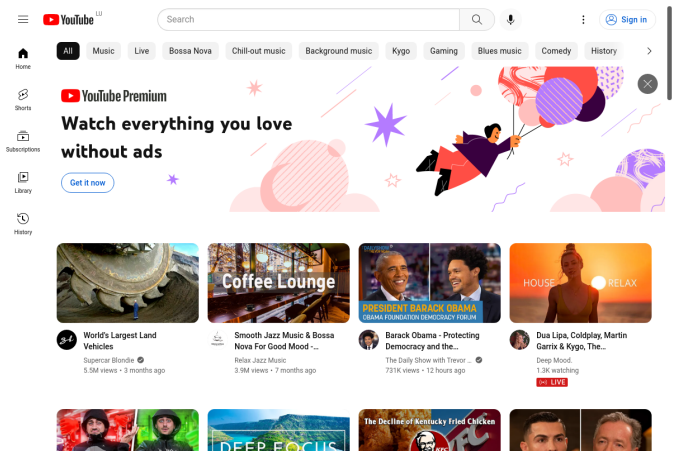

Fig. 5. Youtube.com - extension enabled

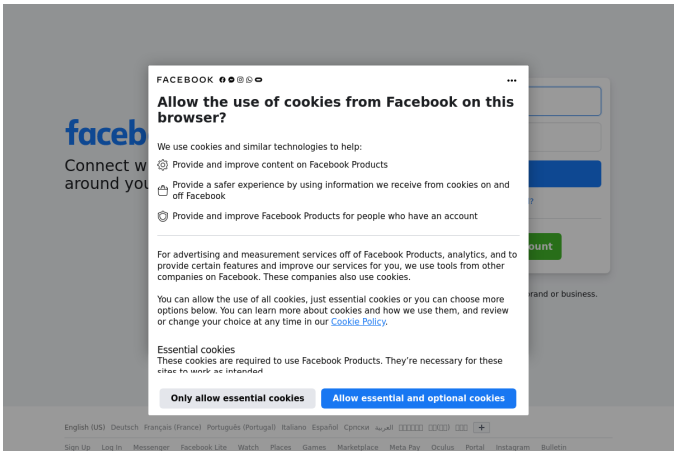

Fig. 6. Facebook.com - extension disabled

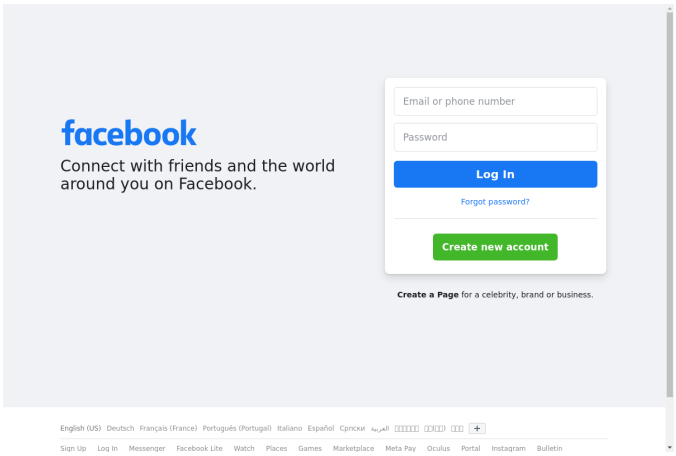

Fig. 7. Facebook.com - extension enabled

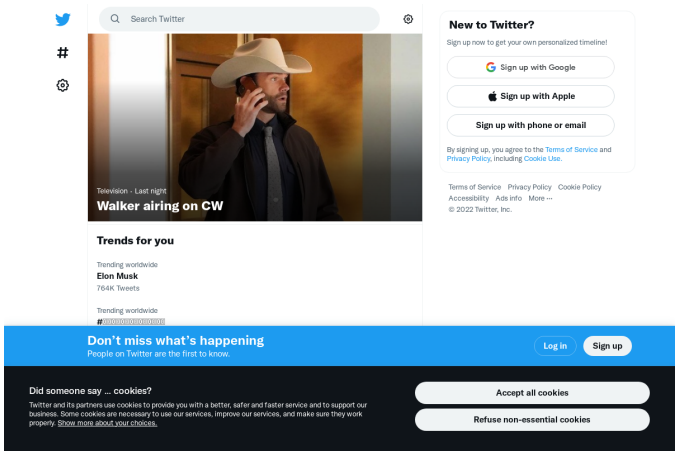

Fig. 8. Twitter.com - extension disabled

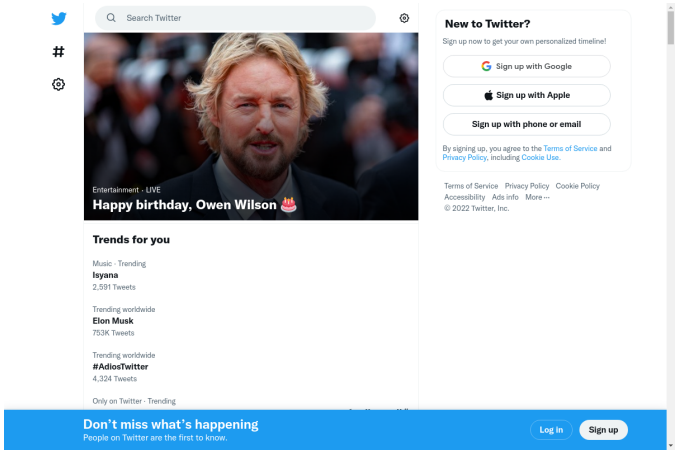

Fig. 9. Twitter.com - extension enabled

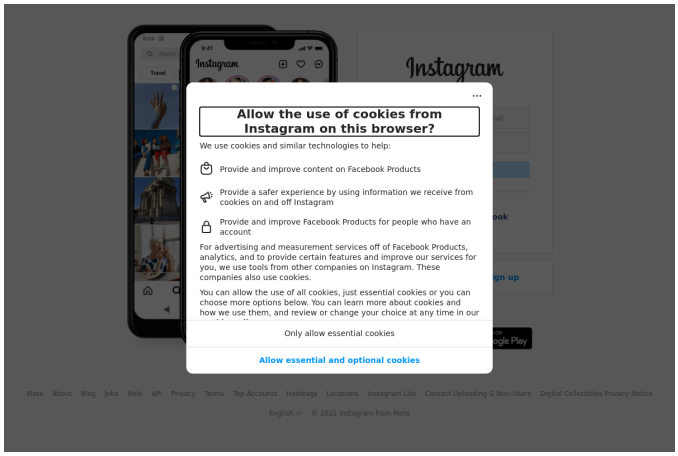

Fig. 10. Instagram.com - extension disabled

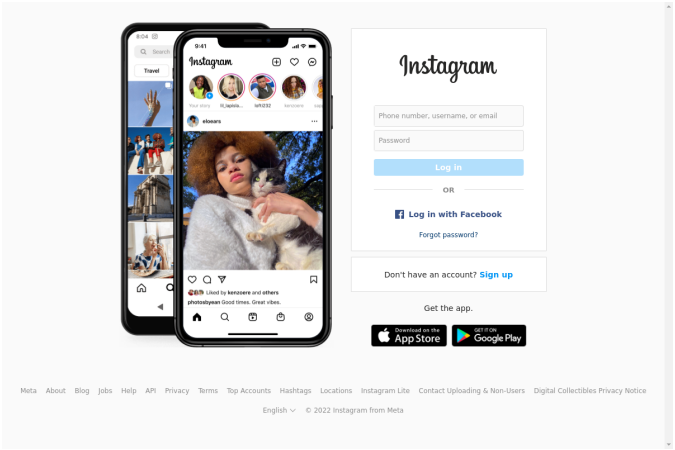

Fig. 11. Instagram.com - extension enabled

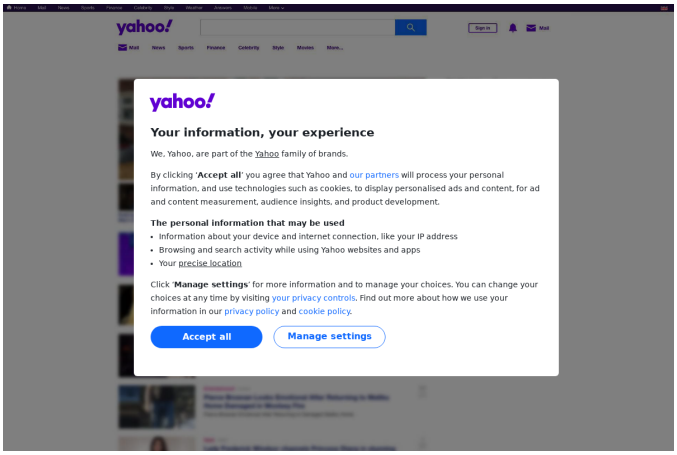

Fig. 12. Yahoo.com - extension disabled

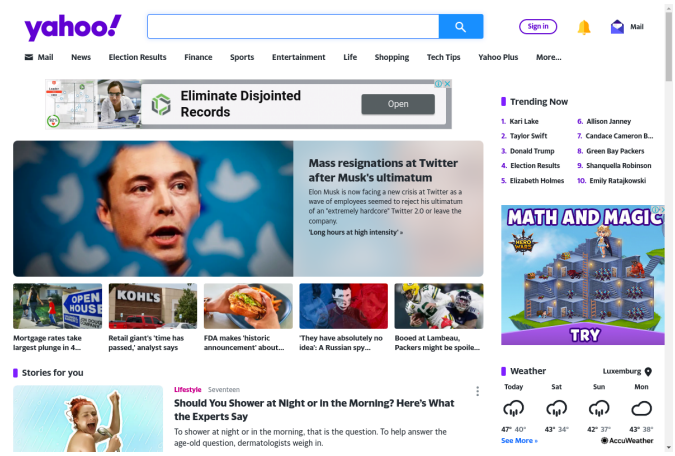

Fig. 13. Yahoo.com - extension enabled

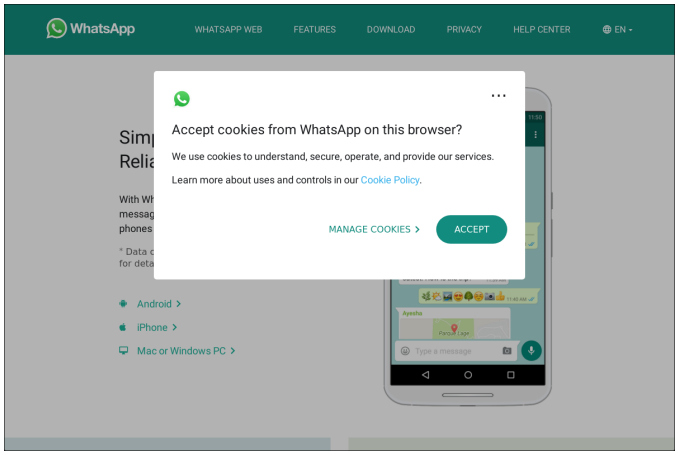

Fig. 14. Whatsapp.com - extension disabled

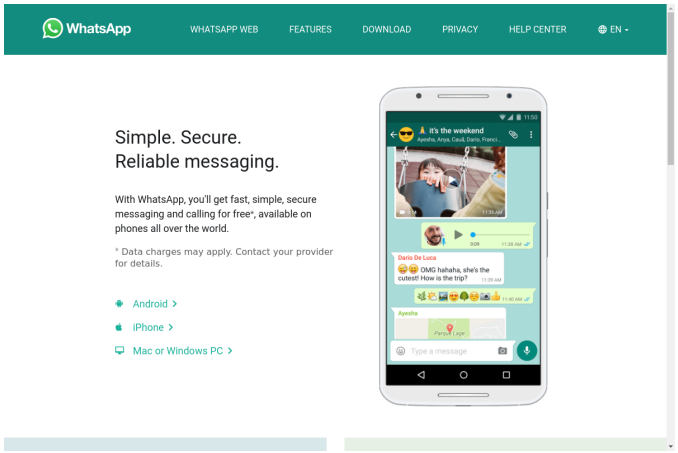

Fig. 15. Whatsapp.com - extension enabled

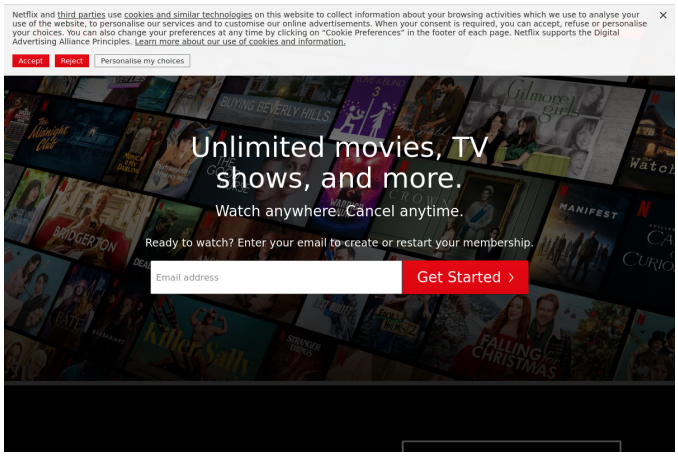

Fig. 16. Netflix.com - extension disabled

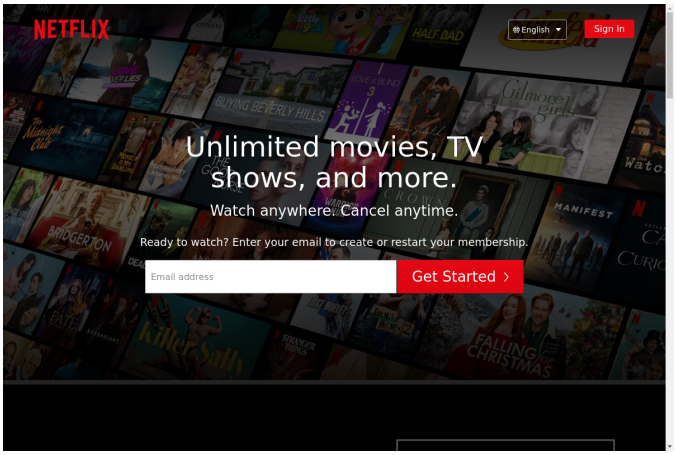

Fig. 17. Netflix.com - extension enabled

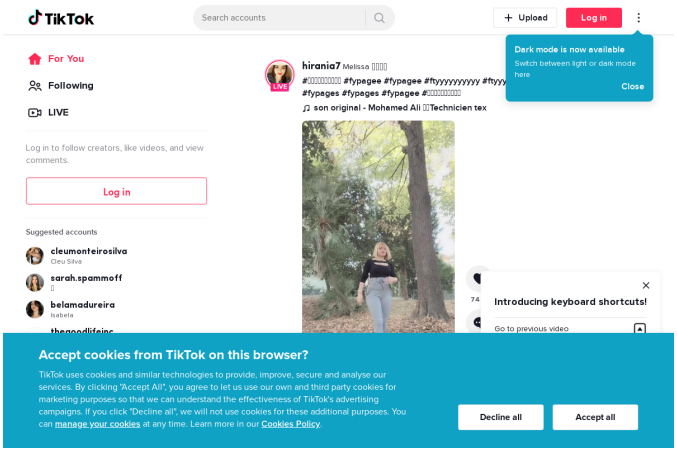

Fig. 18. Tiktok.com - extension disabled

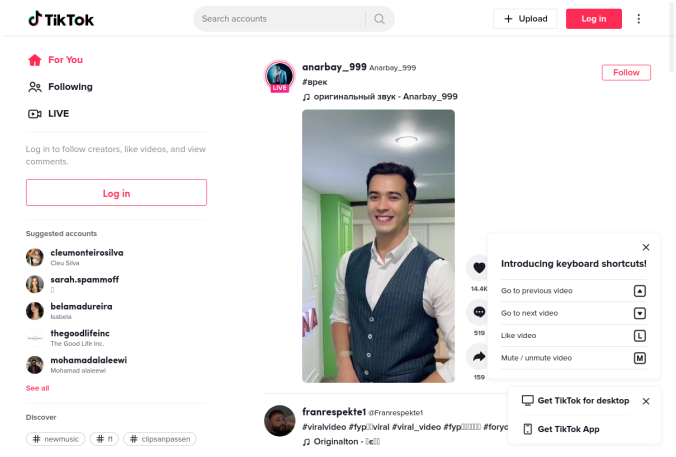

Fig. 19. Tiktok.com - extension enabled

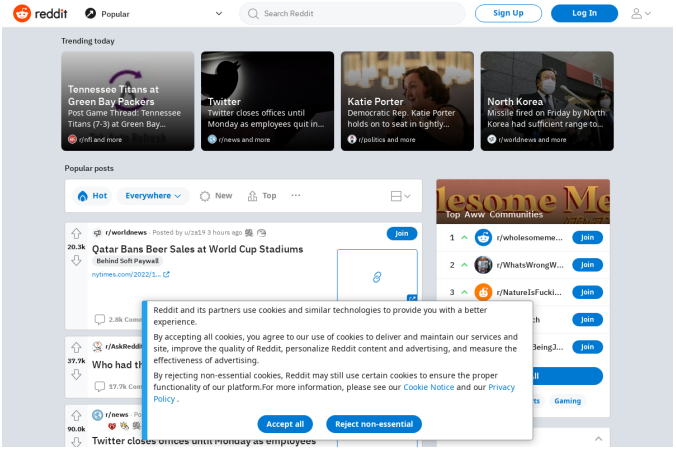

Fig. 20. Reddit.com - extension disabled

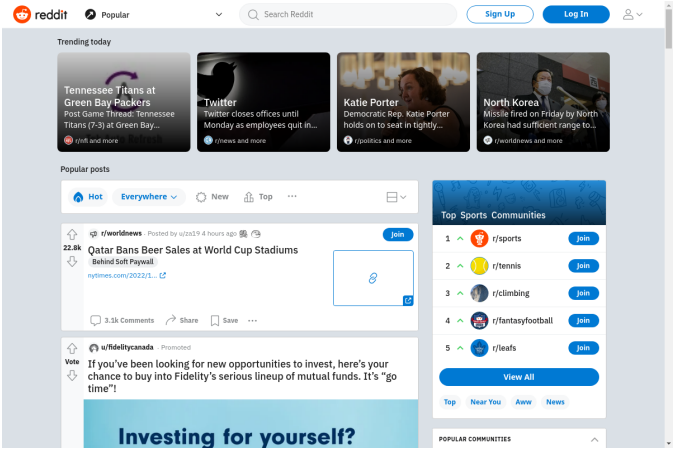

Fig. 21. Reddit.com - extension enabled

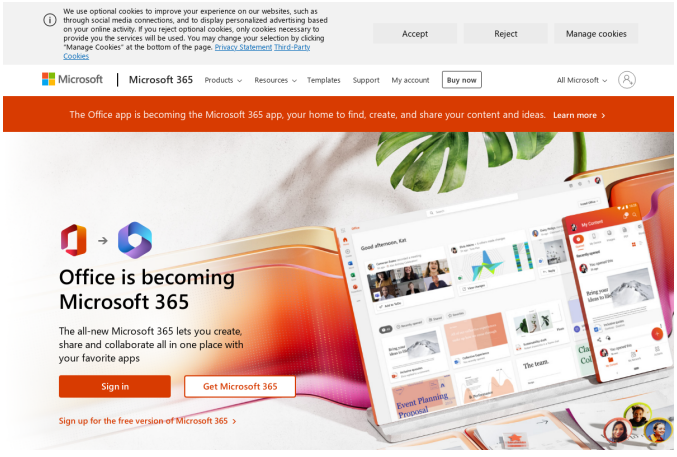

Fig. 22. Office.com - extension disabled

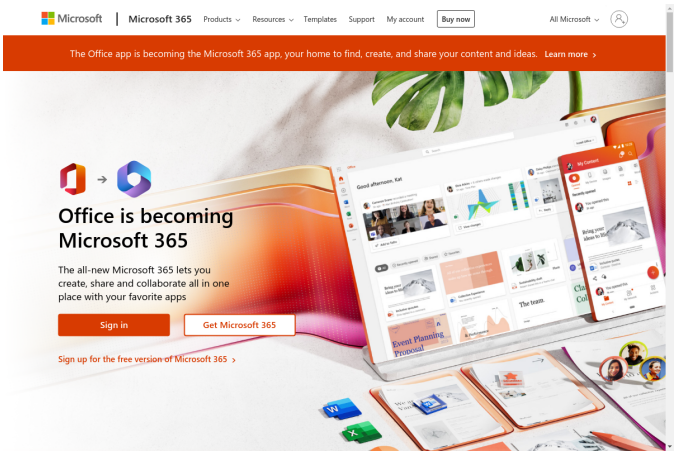

Fig. 23. Office.com - extension enabled

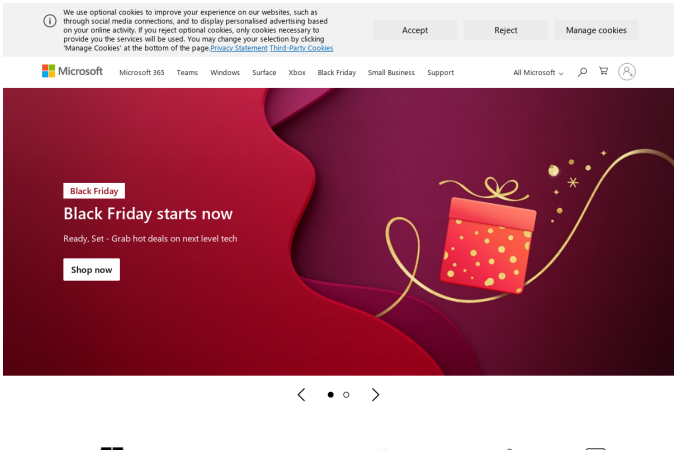

Fig. 24. Microsoft.com - extension disabled

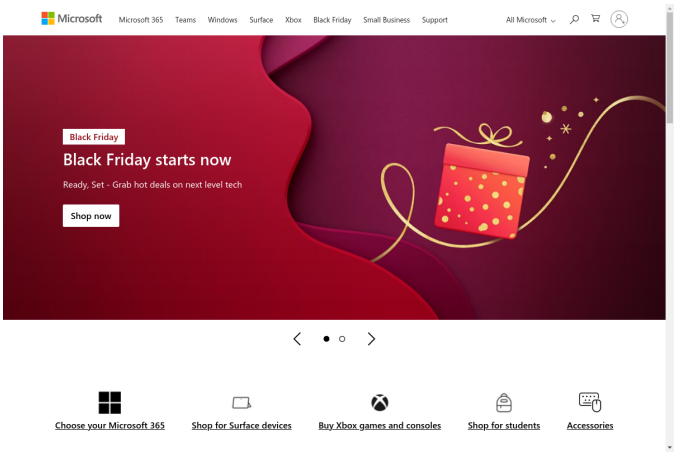

Fig. 25. Microsoft.com - extension enabled

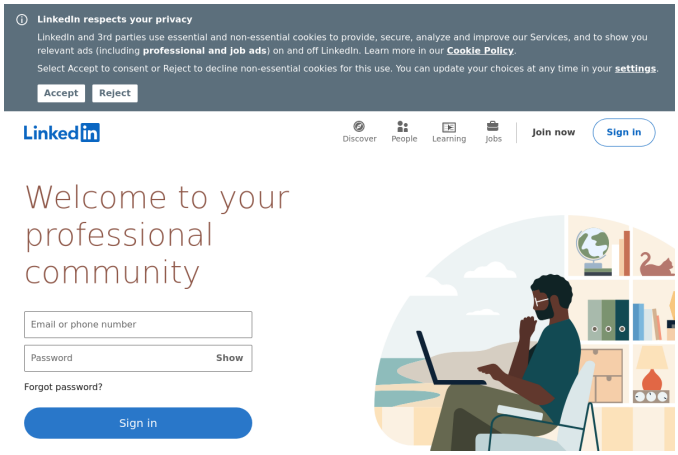

Fig. 26. Linkedin.com - extension disabled

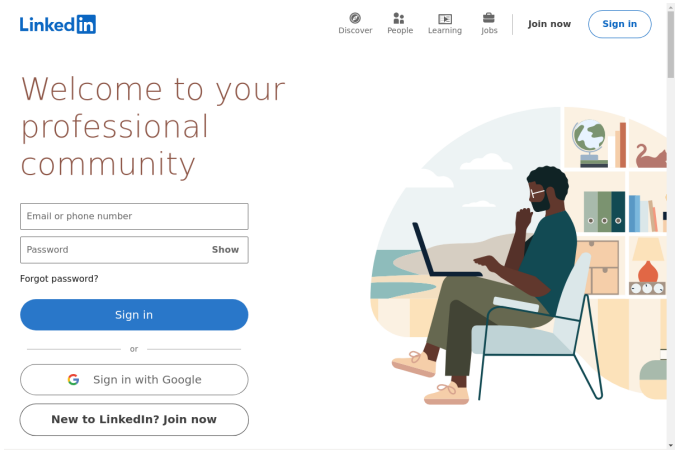

Fig. 27. Linkedin.com - extension enabled

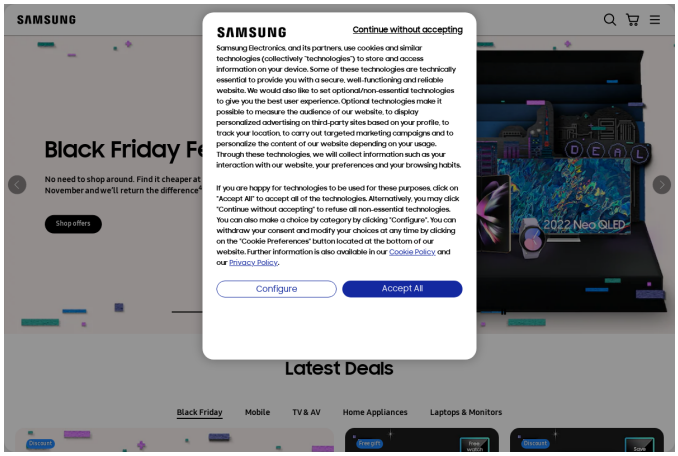

Fig. 28. Samsung.com - extension disabled

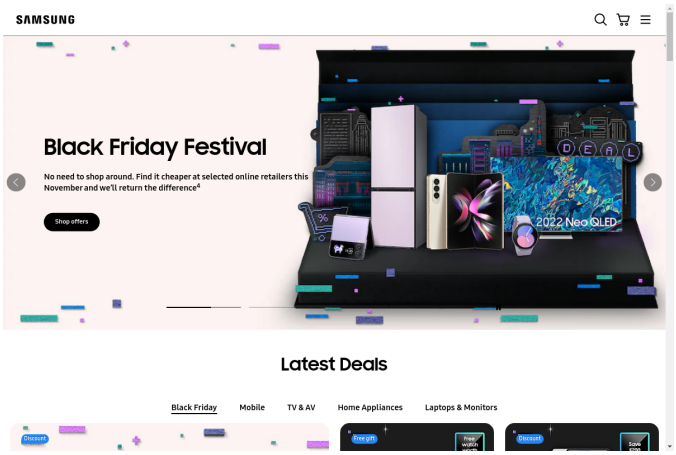

Fig. 29. Samsung.com - extension enabled

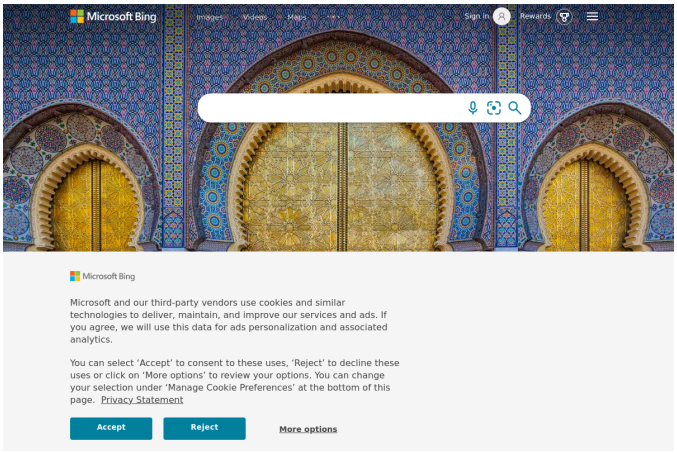

Fig. 30. Bing.com - extension disabled

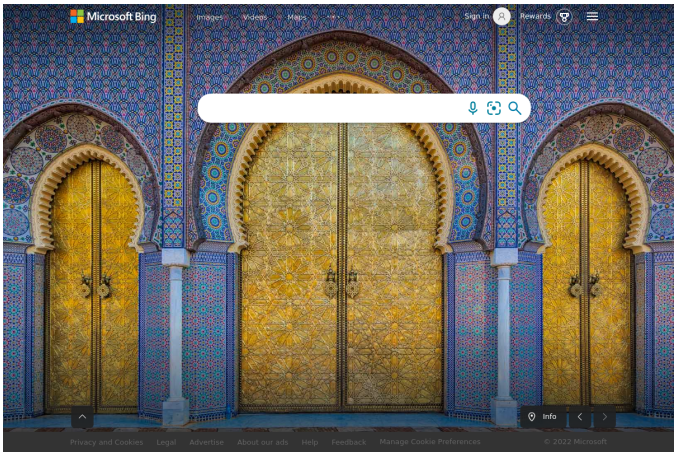

Fig. 31. Bing.com - extension enabled

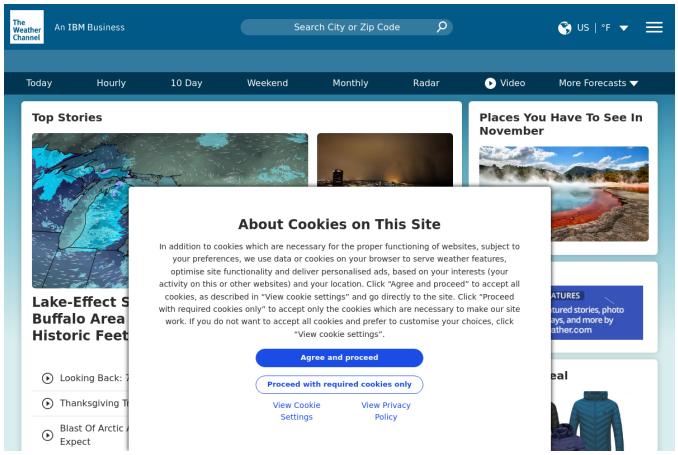

Fig. 32. Weather.com - extension disabled

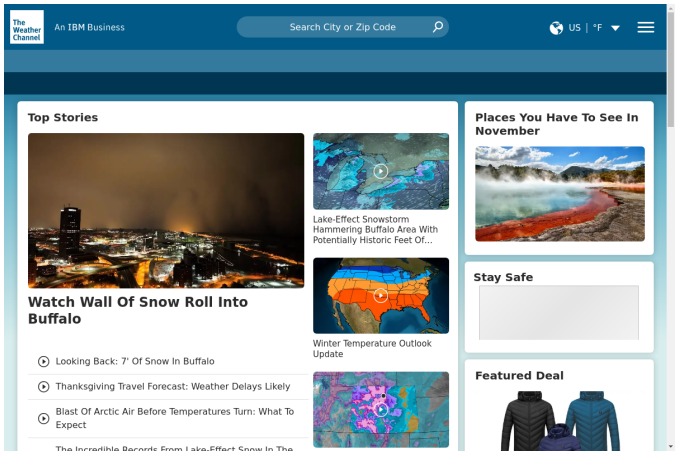

Fig. 33. Weather.com - extension enabled

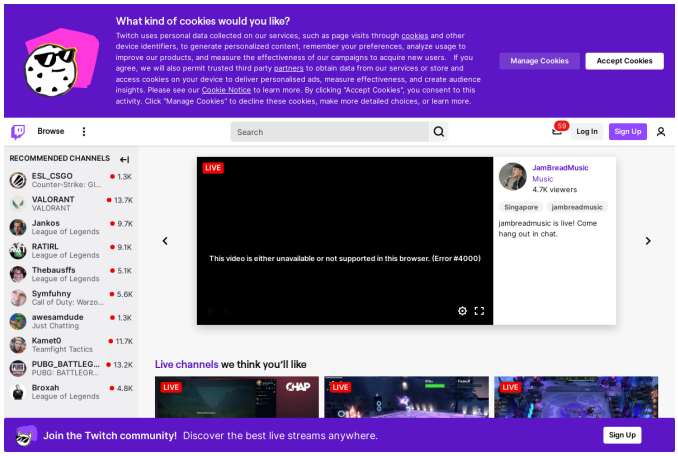

Fig. 34. Twitch.tv - extension disabled

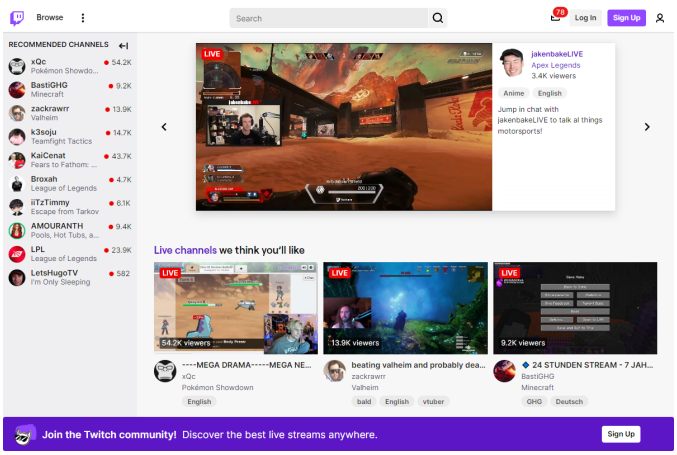

Fig. 35. Twitch.tv - extension enabled

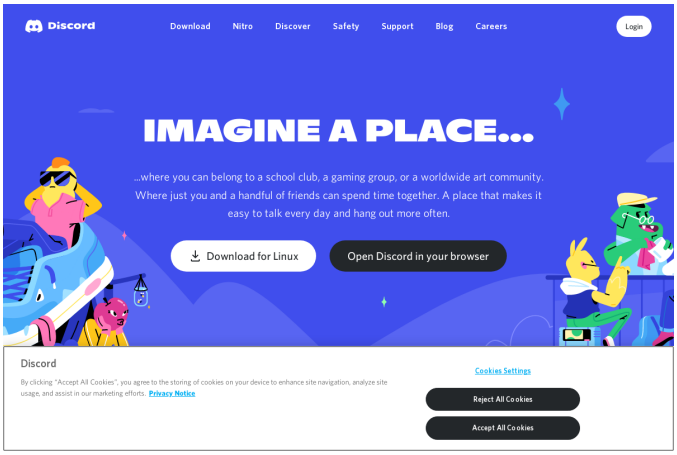

Fig. 36. Discord.com - extension disabled

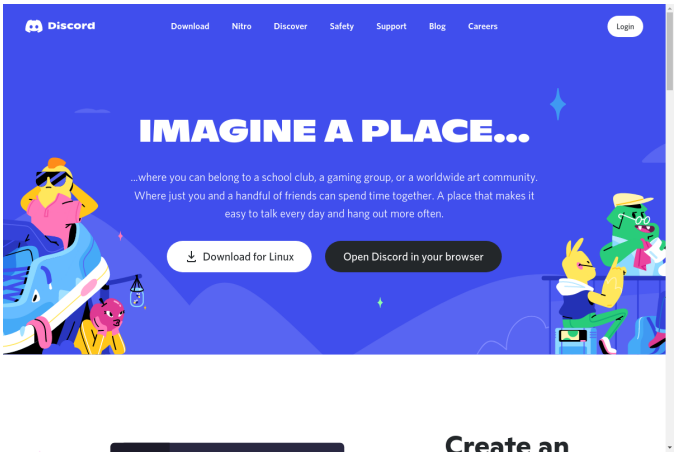

Fig. 37. Discord.com - extension enabled

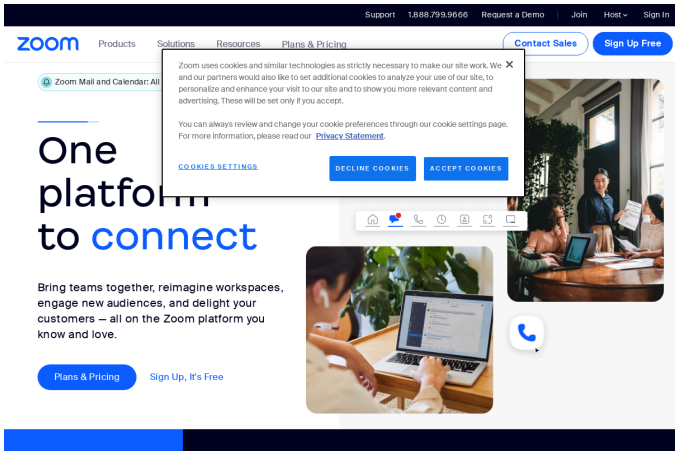

Fig. 38. Zoom.us - extension disabled

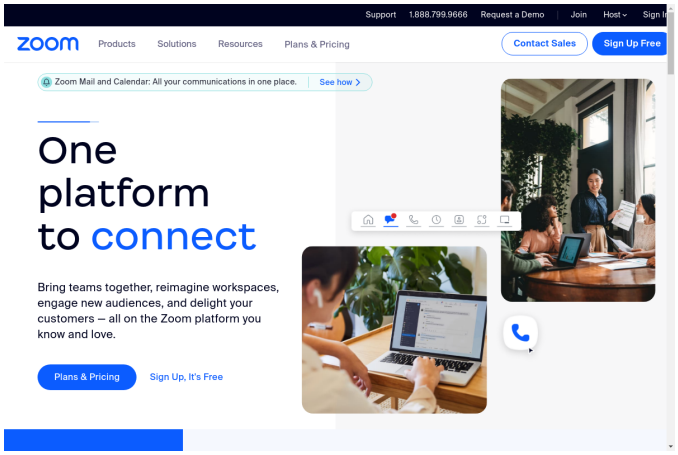

Fig. 39. Zoom.us - extension enabled

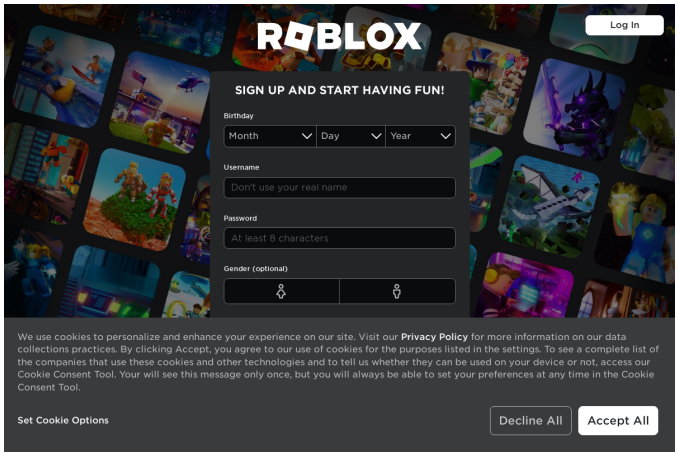

Fig. 40. Roblox.com - extension disabled

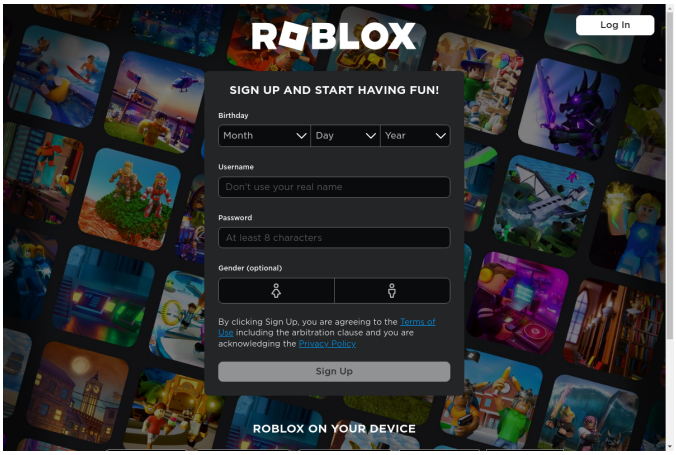

Fig. 41. Roblox.com - extension enabled

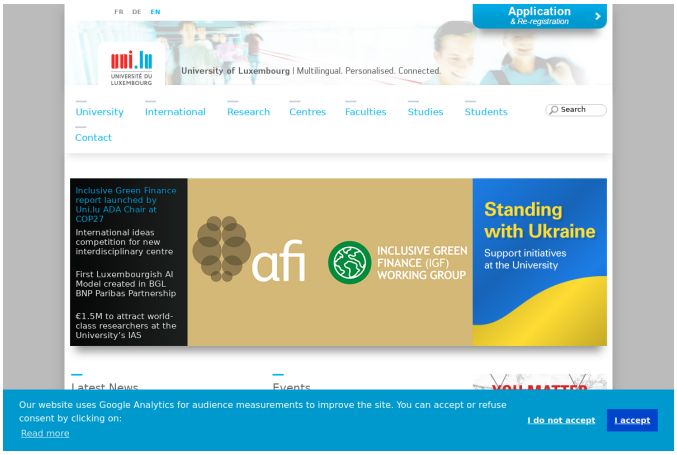

Fig. 42. wwwen.uni.lu - extension disabled

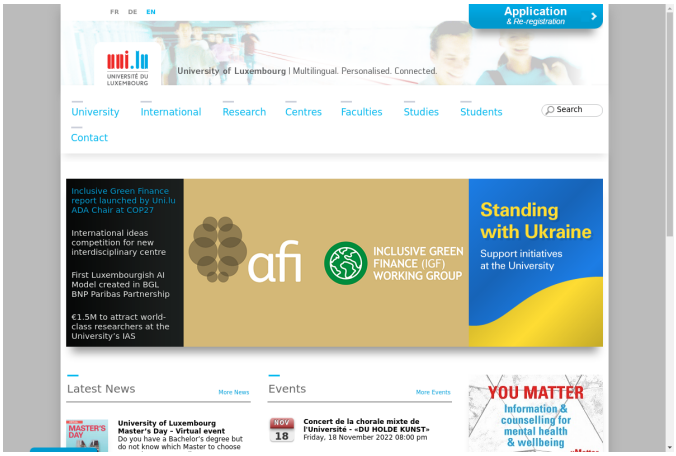

Fig. 43. wwwen.uni.lu - extension enabled