MSc in Computing - Team Project

Final Report

Team Name: Shopping Tricks Finder

Lan Zeng D19124089
Alan Dowley D19127378
Jordan Donnelly D20125012
Yunpeng Liu D20123758
Yufei Su D18130277
Quanwei Sun D19126879

Andrea Curley
Damian Gordon
Paul Kelly

Technological University Dublin

22/12/2021

Introduction

Dark patterns are design elements that webpages use to mislead, obscure, coerce and/or deceive users of that website into making choices that they would not normally make, without these patterns being present (McNealy, 2021). As dark patterns are becoming ever more prevalent throughout the internet, on all types of web pages, it is very likely that the vast majority of people who use the internet have encountered some form of a dark pattern, whether they realize it or not.

Currently, if you are not well informed about dark patterns, there is not a reliable way to identify dark patterns on webpages and even people who are well informed and knowledgeable enough about dark patterns, to be able to spot them on webpages, might occasionally fall victim to one of these patterns.

The aim of this project is to provide people with the education and awareness of what dark patterns are and how they are being used to trick and manipulate people. To accomplish this, there are two main components of the project. The first is a dark patterns detection tool called 'Shopping Tricks Finder' which will allow the users to detect and highlight six different types of dark patterns on webpages and the second is an educational website, "www.dpexplained.com", where users can find out more information about the different types of patterns there are, see real-world examples of patterns on webpages, as well as educative videos on dark patterns.

The report begins with a description of who the target users are and what problem this project will solve for them. This is followed by why this project is being developed, why a chrome extension was chosen and the technical problem it is trying to tackle. Other similar systems that have tried to address this problem are also discussed within this section. The next section describes in detail the technical components of our system, explaining the front-end, back-end technologies used and how the data will be collected and stored. Section 4 will consist of all the users' evaluations that were carried out throughout the development lifecycle. Section 5 will be the evaluations of the machine learning models that were tested throughout the development. Finally in section 6, the conclusions, future work and main challenges of the project will be discussed in detail.

1. User Scenario

1.1 Identify Target Users

In terms of looking for target users, according to (Siting et al., 2012), the observation method is used as an experimental method to determine a large user group. Then focus on small target groups through user surveys. In these user surveys, two methods were used to refine the categories of target users, namely questionnaires and interviews.

The first step is to find a broad target audience based on the paper and the detection features of our extension. During the next step, a user survey was created to narrow down the user's group, firstly using questionnaires. The result shows that people aged 18-44, especially those aged 18-34, were more willing to detect dark patterns. Among these people, 85.94% of those who noticed the dark patterns said they would like to use a detection software/tool. In addition, 69.84% of those who wanted detection tools preferred to use browser extensions. StatCounter's global browser market share (Browser Market Share Worldwide, n.d.) shows that Google Chrome was used by 65.15% of the population. So, those people using Chrome as a browser can be considered as the user group.

To sum up, the users are mostly of the age 18-44, who have the habit of using Google extensions, have noticed some Dark pattern-related content, and often visit English shopping websites. In addition, to find more target users, the interviews were held to find out the special user groups that are not considered in the user questionnaire. These users would be people who have a vast knowledge of dark patterns already but could still use the tool to help them detect and find patterns on webpages, without having them manually look through the page itself.

Three user personas were created according to the refined information above, which represented our target users. Two of the personas are the young and older average users, meaning that they have never heard of dark patterns or might have come across the term before but know very little about them. The third user persona is our expert user, who is very knowledgeable about dark patterns. The three personas can be seen in figure 1.

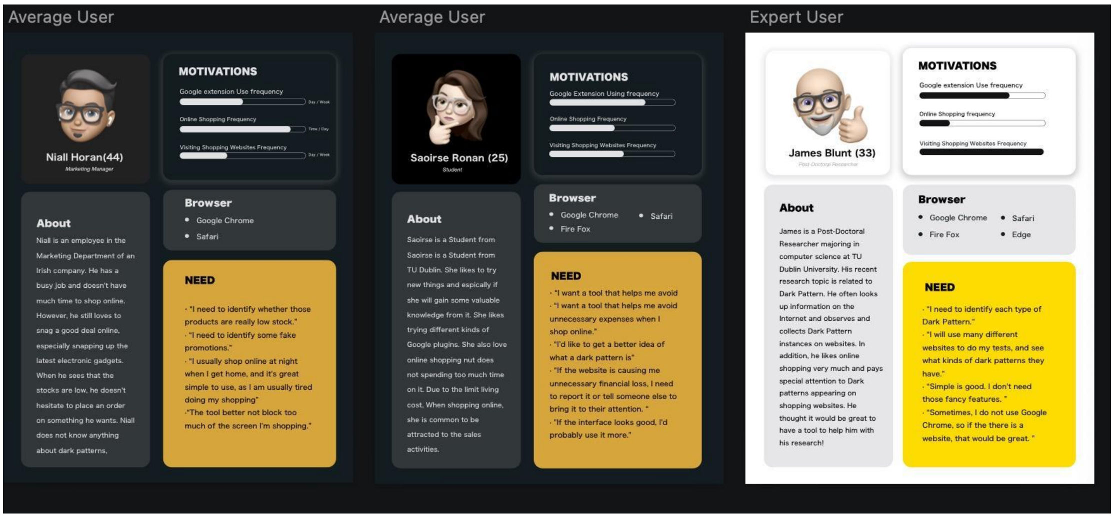


Figure 1 - user personas-elder average user, younger average user, and expert user

1.2 Importance of Target Users

For a product like ours that was developed from scratch. Finding target users is an absolute necessity. The target user can help design the Persona. to be more specific, it can help to assign attributes like names and ages, etc. to the persona according to the target user's needs. According to the characteristics of each type of user, the user survey continues to uncover the needs and aggregate them together (Ferreira et al., 2015). In our extension development process, all work begins with an analysis of the needs of the target user. These requirements are to guide building use cases and design the entire system architecture. In addition, in order to design products that allow users to easily achieve their goals, the expectations of the target users are also important. Figure 2 shows the direction of design for the extension.

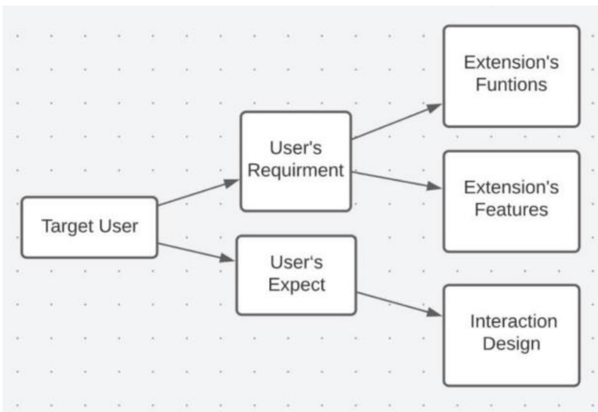


Figure 2 - The importance of target user

Question 6: Which will you choose as a detection tool? [Single choice questions]

options	Subtotal	proportion
Browser Extension	54	68.35%
Website	25	31.65%
Number of valid people filled in this question	**79**	

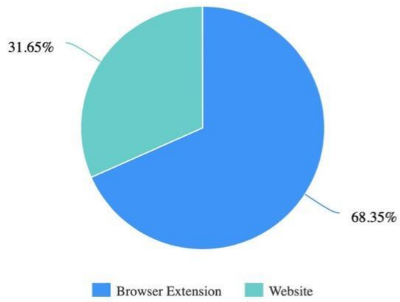


Figure 3 - Example of user's needs to guide the development

1.3 Problems Solving for Target Users

There are two main problems encountered by the target users. One is the lack of knowledge and inability to recognize dark patterns. The other one is the lack of suitable auxiliary identification tools.

Dark Patterns are widely found on shopping websites, according to research (Brignull, 2019) and it causes users to suffer loss in many aspects (Mathur et al., 2021). Meanwhile, a paper also mentioned that some dark patterns potentially could be avoided if people can pay enough attention to them (Maier & Harr, 2020). The result of the questionnaire shows that 71.43% of the users who had been harmed by Dark Pattern had never heard of Dark Pattern, let alone recognize it. In addition, although many people saw the Dark Pattern sentences before, they did not avoid causing damage because they were unaware it was a trap. The system can highlight Dark Patterns on the webpage, which assists users to find and avoid Dark Patterns. In addition, other dark pattern detection software available to users cannot effectively detect Dark patterns on web pages.

The average user journey of the extension's main functions is shown in figure 4.

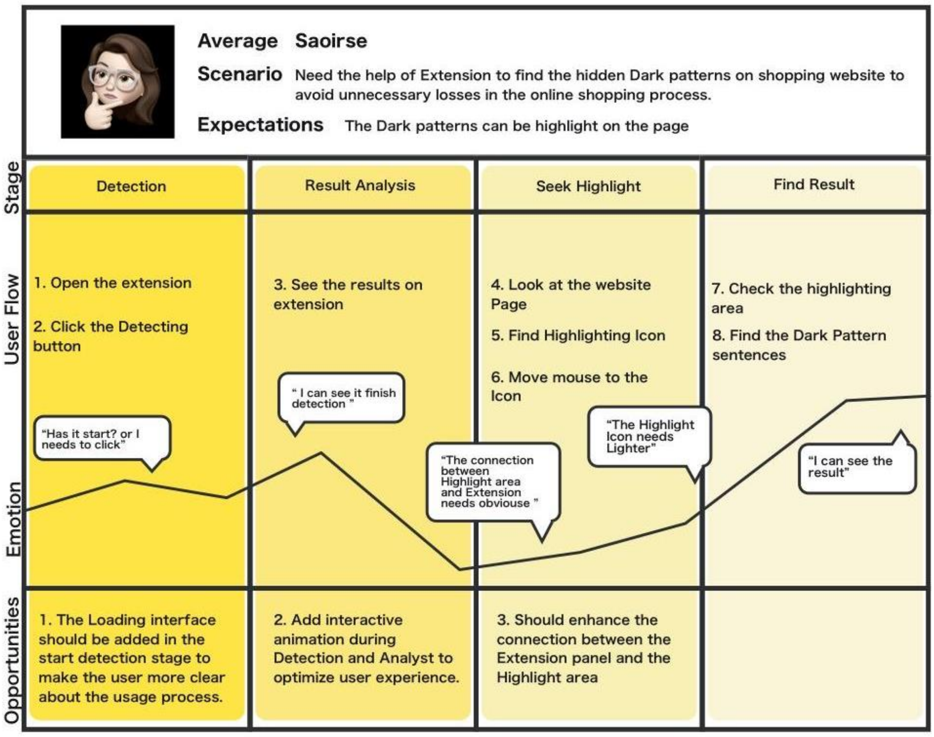


Figure 4 - Main functions of average user journey

2. Technical Problem

2.1 Why we built Shopping Tricks Finder

Due to the vast technological advances of websites, more and more people are using the internet, be it casual browsing of websites or online shopping, as a part of their everyday lives. This means that the number of victims of Dark patterns is always rising. A user should be able to go shopping online without being deceived or tricked unbeknownst to themselves. Dark patterns are design elements, on webpages, that deliberately obscure, mislead, coerce and/or deceive website visitors into making unintended and possibly harmful choices. No user wants to get tricked by a website that they trust enough to input their credit/debit card information into.

The impact that dark patterns can have on people falls into a wide range, with some patterns being more severe and manipulative than others and certain types of websites being able to get away with using dark patterns more than others. Dark patterns can cause confusion, annoyance, potentially cost the user more money than they realize, or even go as far as lowering their own self-confidence and second-guessing themselves.

The results of an extensive comprehensive literature review and systematic analysis of websites, conducted in (Parrilli, D. M., &Hernández-Ramírez, R. 2020), conveyed that most design scholars agree that dark patterns are to be considered unethical. Although many people agree that they are unethical and manipulate people, there are no laws or acts which prevent websites from using dark patterns to trick and coerce their users.

Surveys were conducted using potential users to get a better understanding if:

1) People had encountered dark patterns before?

Question 2:

Have you ever noticed similar phrases, to the ones shown below, on any websites? [Single choice questions]

options	Subtotal	proportion
yes	63	79.75%
no	16	twenty-five percent
Number of valid people filled in this question	**79**	

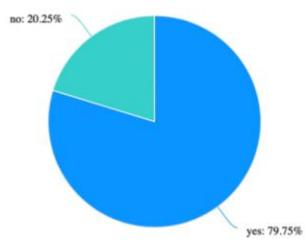


Figure 5 - Statistical results of people had encountered dark patterns

2) People had been affected by these patterns before?

Question 3:

Have any of these phrases made you do things that you didn't mean to, like buying more things th an planned or signing up for something you don't need. Single choice questions]

options	Subtotal	proportion
yes	56	70.89%
no	23	29.11%
Number of valid people filled in this question	**79**	

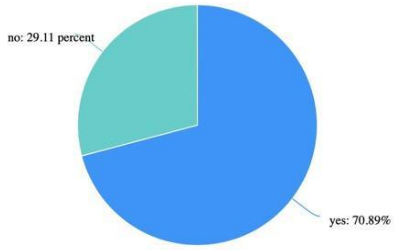


Figure 6 - Statistical results of people had been affected by the dark patterns

3) People knew what dark patterns were?

Question 4: Did you know these things are tricks called Dark Patterns? [Single choice questions]

options	Subtotal	proportion
yes	24	30.38%
no	55	69.62%
Number of valid people filled in this question	**79**	

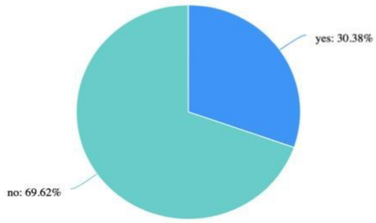


Figure 7 - Statistical results of people knew dark patterns or not

4) People would use a system that would help them to detect these dark patterns on web pages?

Question 5:

If there is a tool that could help you detect Dark Patterns and protect you from unknowingly bei ng manipulated by them, will you like to try it? [Single choice questions]

options	Subtotal	proportion
yes	63	79.75%
no	16	twenty-five percent
Number of valid people filled in this question	**79**	

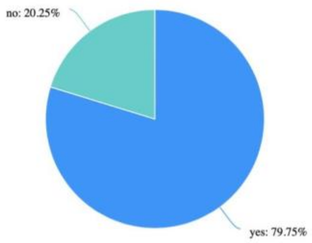


Figure 8 - Statistical results of people 's willingness of using detection tools

From the results of the survey conducted, it was very evident that a lot of people had encountered and been affected by dark patterns, while not knowing what dark patterns were and they were also wanting/willing to try some sort of tool or system that would help them to not fall victim to dark patterns.

For the reasons just explained, we decided to build a system that would detect and highlight the dark patterns that may be present on a webpage, for the user. This system will be available for all age groups, will be free, and will be easily accessed through a chrome extension, meaning it will be less intrusive to the user. It will also be customizable to the user, by allowing them to choose the dark patterns that they want to highlight.

The highlighting of the devious dark patterns will allow the user to be aware of when he/she is being tricked or deceived. It will also provide information on how the dark pattern is considered a dark pattern which will educate the user, hopefully allowing them to recognize dark patterns in the future. This system will assist the user in various ways:

1. Financially – It will highlight when the user might be tricked into adding extra items to their baskets or being pushed into buying the more expensive option.

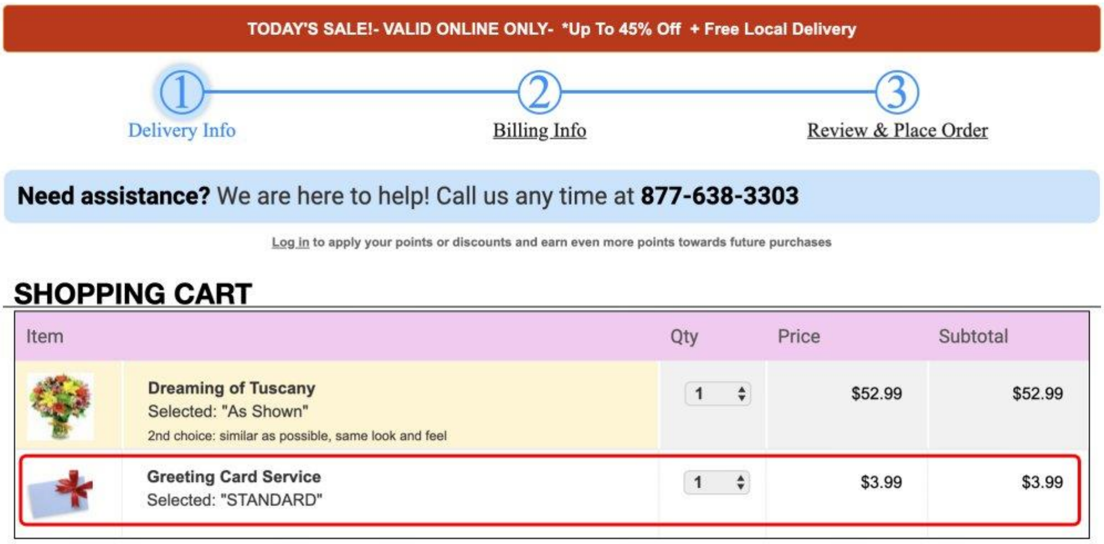


Figure 9 - Example of Dark Pattern on a Website

2. Emotionally - It will highlight the urgency of low stocks and fake countdowns and remind the user to relax and that more than likely the stock and countdown is fake.

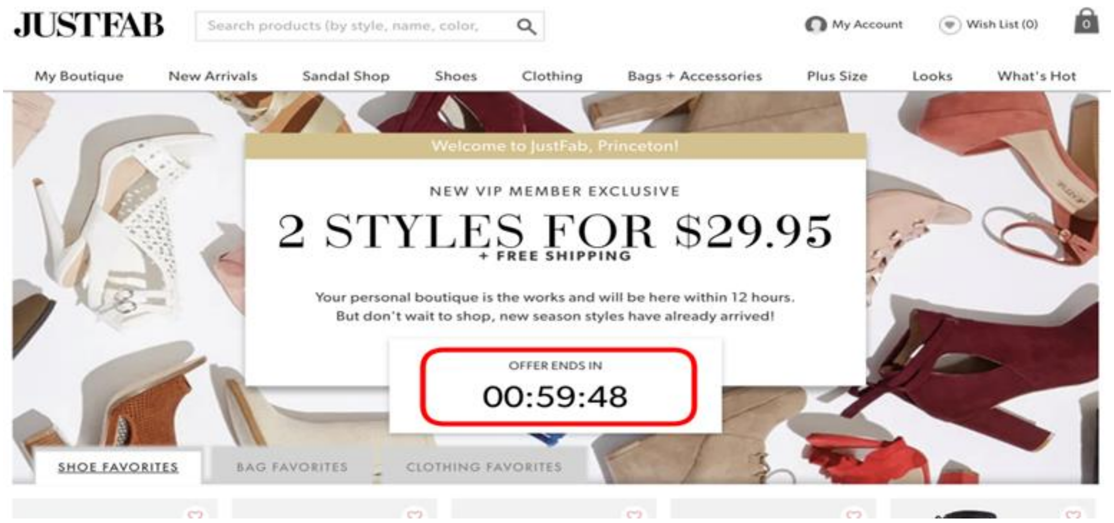


Figure 10 - Example of Fake Countdown Dark Pattern

2.1.1 Why did we build it as a chrome extension?

We had two main choices when it came to building a system that detects and highlights dark patterns on webpages. The first option we had was to build it as a web page, where the user inputs the URL of any webpage, which then would be checked by our system. The second was to build it as a chrome extension, in which case the user could just simply run the extension on any page that they visit, without having to go to a different webpage and input a URL. However, the problem with building the system as a chrome extension was that we are immediately cutting out every single person who does not use Google Chrome as their browser of choice.

From research we found that as of September 2021, over 64% of people used Google Chrome as their daily browser, with Safari in second place with just over 19%. Having the convenience of being able to run the detector on any webpage without having to navigate to a specific webpage to use it, will encourage more people to actively use our extension as they shop online. Another big reason to develop a chrome extension, over a webpage, is that there is a ready platform, the chrome web store, in which the extension can be readily available for people to download and use.

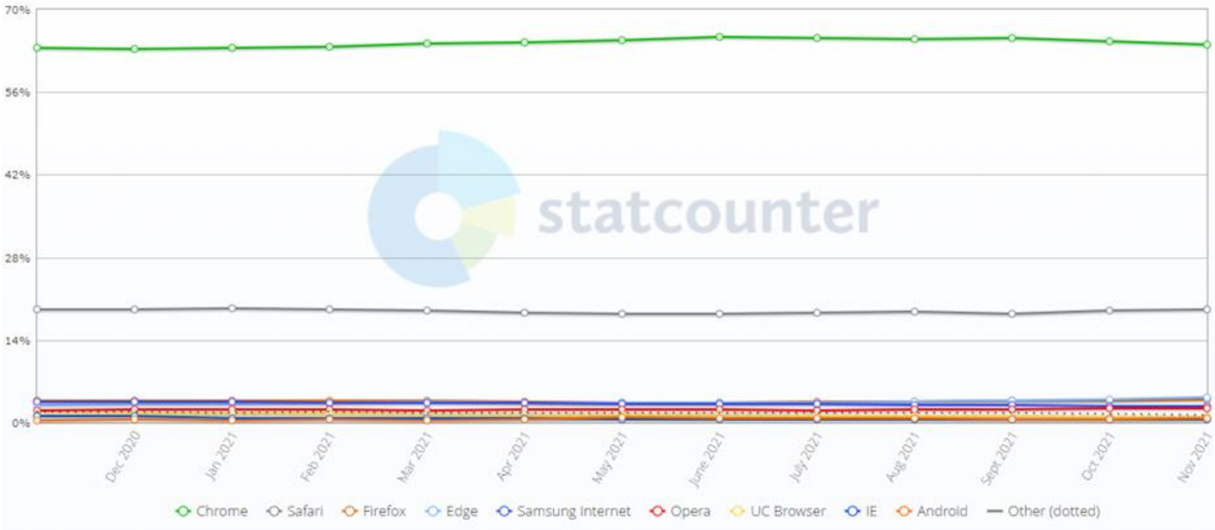


Figure 11 - Result of browser usage statistics

Within the survey, we conducted about people's knowledge of dark patterns, we also asked them in what form (website or extension) they would use a detection tool, with the majority of the responses being for an extension.

Question 6: Which will you choose as a detection tool? [Single choice questions]

options	Subtotal	proportion
Browser Extension	54	68.35%
Website	25	31.65%
Number of valid people filled in this question	**79**	

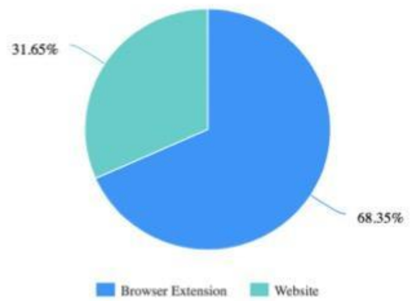


Figure 12-Statistical results of people's willingness of detection tool

We decided that these two factors of convenience and a ready platform to put the extension to, outweighed the fact that we would be potentially cutting out a lot of users by only having it a chrome extension and would overall help us to get the extension out to users and then provide them with a more enjoyable and hassle-free experience when using the system.

2.2 Review of Similar Systems

2.2.1 "Dark Patterns" Extension

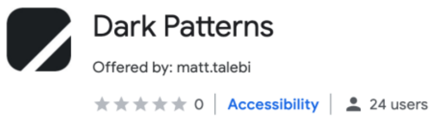


Figure 13-Chrome Extension Info

This Chrome extension aims to find out the dark patterns on the user's website. The pros and cons of this system can be seen in Table 1.

Table 1 - Pros and cons of similar system

Pros	1. It has a simple and clear side window UI design. 2. It has an automatic detection feature.
Cons	1. It fails to list out the dark patterns it can detect in detail. 2. During the usage of this extension, it failed to detect dark patterns on any of the websites, showing "Great news!" on every website, while there are dark patterns that can be easily observed manually. 3. It has an "Add Dark Pattern" feature, which directs users to a new window to fill out a form reporting a new dark pattern found on the website. Four out of five questions of the fields are prompt for text input, asking for the description of the dark patterns. This function, on the one hand, is time-consuming for users to report a dark pattern, on the other hand, makes it difficult for the developer to locate where the reported dark pattern is on the page.

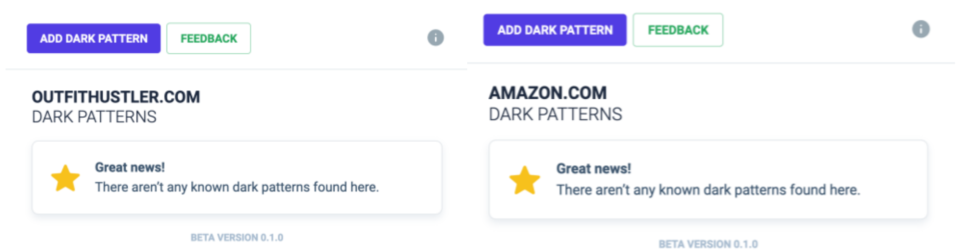


Figure 14-Examples of the detection results of the extension

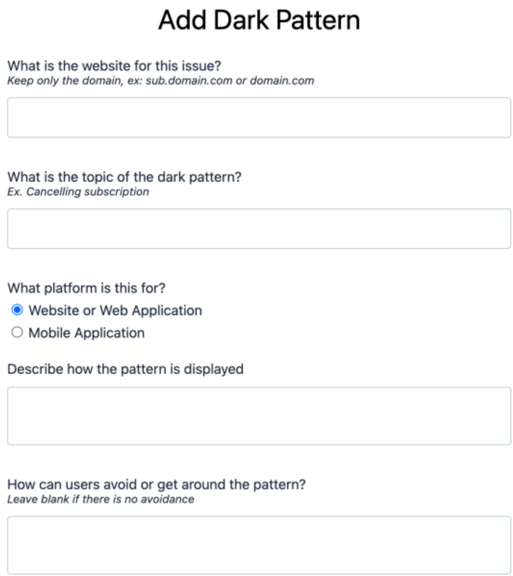


Figure 15 -Report page of the extension

2.2.2 "Dark Pattern Detector" Extension

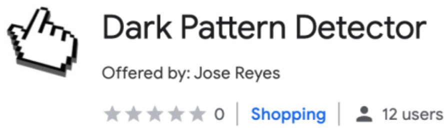


Figure 16 - Chrome Extension Info

This Chrome extension aims to detect the dark patterns on the user's website and warns users about them. The pros and cons of this detector can be seen in Table 2.

Table 2 - Pros and cons of similar detector

Pros	1. It has an automatic detection feature. 2. A pop-up window will show to notify and educate the user if any dark patterns are detected on the page & domain. 3. The highlighting feature on the page is easy for users to locate the dark patterns.
Cons	1. It also fails to list out the dark patterns it can detect in detail. 2. Users must close the pop-up window in the middle of the web page to continue shopping, which can be annoying for some users. 3. After installing this extension, the only dark pattern it detected and highlighted is called "Urgency" by itself,

	which is actually a special offer (debatable to be classified as a dark pattern). 4. The Report function on the pop-up window is not functional, giving no reaction when clicking on it.

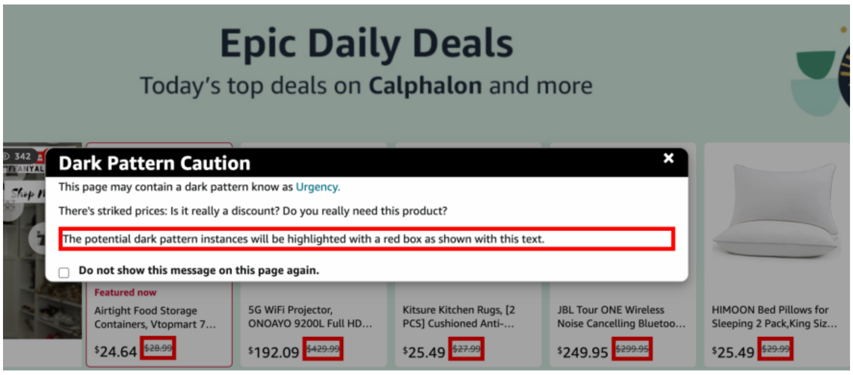


Figure 17 - The pop-up window when dark patterns found on the webpage

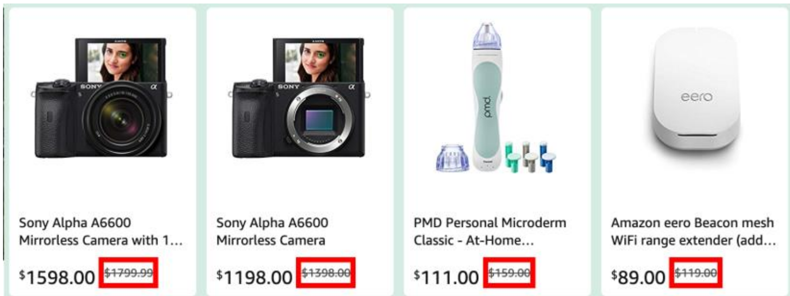


Figure 18 - The highlighting function of the extension

2.2.3 Analysis of Comparisons

For both available extensions, the automatic detection feature is a good design, and the report function of the "Dark Patterns" is a good idea for collecting more potential dark pattern information with the help of the users. The highlighting function of the "Dark Pattern Detector" is a good design, and the pop-up window can help alarm and educate the users about the potential harm from the detected dark patterns.

The problems of both extensions is that the detection results are compromised, as the "Dark Patterns" exentension is not detecting any actual dark patterns, and "Dark Pattern Detector" is only detecting special offers which is actually not a dark pattern. At the same time, although the pop-up window used to alarm users is a good idea, it is not the best design to put it in the middle of the webpage and force the users to close it before they can continue shopping.

2.3 Core Technical Problems

The core technical problems in the project include 3 aspects:

1. Dark Pattern Detection:
   (1) Identify which dark pattern types are possible to be detected for this project.
   (2) Identify the number of models that need to be developed and what algorithms should be used for training the models.
   (3) Find public and related benchmark datasets to train machine learning models for dark pattern detection.
   (4) Find the correct direction to enhance the dataset for improving the model performance.
2. Dark Pattern Highlighting:
   (1) How to accurately locate the recognized dark pattern text. As HTML is a language similar to XML, it can only be located through TAG. This means that if two texts on the page (one is a dark pattern and the other is not) have the same label, then it will end up highlighting both labels at the same time.
3. System Architecture:
   (1) Allow communication from all aspects of the project (machine learning model, API, and Chrome extension).
   (2) Appropriate technologies that each aspect of the project can use and implement.
   (3) Recognize the data to be sent and how it is handled, as well as the appropriate responses that need to be returned.
   (4) Identify what data is to be stored and why the data type is to be stored.
   (5) Select a secure and affordable cloud service provider that provides services that would assist in developing the project.
   (6) Nominate a reliable approach to deploying the API and machine learning model.

2.3.1 Definition of Dark Patterns

Dark patterns are design elements on webpages that deliberately obscure, mislead, force, or deceive website visitors into making unintended and possibly harmful choices. (Harry,2010) According to Curley et al. (2021), the definition of each dark pattern is shown in Table 3.

Table 3 - Dark Pattern List

Dark Pattern Type	Description
Sneak into basket	There will be extra items added into the shopping basket when the user is purchasing a product.
Hidden Costs	There will be an extra cost when entering the purchase pages.
Trick Questions	When purchasing a product, there will be an additional item added into the basket and shown in the purchase page. like a warranty.
Misdirection	The design purposely distracts the user's attention on other items.
Confirmsharming	It tries to guilt users into opting into something. Users may feel uncomfortable by declining it.
Disguised Ads	There will be a fake 'download' button. Once a user clicks it, it will lead the user to another website as an advertisement.
Roach Motel	When a user is subscribing to a service, it is hard to find the button to unsubscribe.
Forced Continuity	When a user's free trial of a service ends, their credit card quietly starts

	charging without warning, and it's hard to cancel it.
Privacy Zuckering	Tricking users into sharing more information than they intended to.
Price Comparison Prevention	Retailers make it hard for users to compare the price of two items by putting them into different bundles.
Bait and Switch	The user starts to do one thing, but a different, undesirable thing happens instead
Friend Spam	The product asks users for their email or social media permissions to spam all their contacts
Fake activity	There may not be a fully truthful statement when the pages show the number of people in activity.
Fake Reviews	There will be some fake reviews and testimonials
Fake Countdown	In the shopping website, there will be a countdown times to add urgency to a sale
Ambiguous Deadlines	In the shopping website, the items be on sale with limited amount of time

Low Stock Messages	The website may show the items are in low stock
Deceptive High Demand	The website may show the items are in high demand

The detection strategy of these dark patterns has been mentioned in some research (Curley et al., 2021). In this project we can detect in total six types of dark patterns. The rest of the dark patterns can't not be implemented due to the limited time, complexity of detection, and necessarily manual detection, as shown in Table 4.

Table 4 - Technical Problem of Detecting each type of Dark Patterns

Dark Pattern Type	Technical Problem
Sneak into basket	Must detect manually or need a reminder rather than warning, as sometimes it is necessary to add the extra item (e.g. taxes or free gift)
Hidden Costs	Must detect manually or need a reminder rather than warning. It is required to compare the origin cost in the first webpage with the final cost in the second webpage. Same reason as above: tax
Trick Questions	Requires dataset for using Natural Language Processing for double negation detection
Misdirection	Must detect manually. The CSS elements need to be compared.

Disguised Ads	Google has already taken over advertisements in websites. Now these download advertisements have the ads mark.
Roach Motel	It is hard to detect even manually as the unsubscribe button may be in the different pages, or it does not exist at all.
Forced Continuity	It can be detected after the free trial, require long time. (One - three months)
Privacy Zuckering	Different Websites have different implemetations on this dark patterns.
Price Comparison Prevention	It is hard to highlight the items the user would like to compare. Maybe a manual highlight tool is more helpful.
Bait and Switch	Detecting this pattern is extremely challenging as there is such a significant variation in how the pattern is implemented on different sites.
Friend Spam	It can be detected by analysing the requirement for sending email and analyse the HTML if the site asked for email or social media permissions.

2.3.2 Dark Pattern Detection Strategy

There are six dark pattern types detected in this project, Fake Activity, Fake Countdown, Fake Limited-time, Fake Low-stock, Fake High-demand, and Confirmshaming. The detection strategy for these six dark pattern types is shown in Figure 19. There are two models that need to be developed for these 6 dark pattern types. The first model is for the first 5 dark pattern types, as they can be directly predicted based on only the text. Another model is just for the Confirmshaming dark pattern detection. This is because Confirmshaming dark patterns content is very similar to normal content on the website, which is not dark patterns, which can heavily increase the misclassification rate if detecting the Confirmshaming from text extracted from the whole HTML. However, one thing unique about Confirmshaming is that it mostly appears on the buttons and links, therefore the text on the buttons or links can be extracted first to be the pre-processed data for just the Confirmshaming detection model to detect.

Pattern Type	Detection Strategy
Fake Activity	(1) Gather the text content of the HTML on the webpage. (2) Apply Natural Language Processing in machine learning / deep learning to achieve fully automatic detection of these 5 pattern types, based on the text only.
Fake Countdown	
Fake Limited-time	
Fake Low-stock	
Fake High-demand	
Confirmshaming	(1) Gather the text of the buttons and links on the webpage. (2) Use Natural Language Processing in machine learning to achieve fully automatic detection based on text only.

Figure 19 - Dark Pattern List and Detection Strategy

The detection process is shown in Figure 20. To detect dark patterns on the webpage, the HTML of the webpage needs to be analyzed first for text extraction. A specific tag type will be assigned to each text string extracted and the pre-processed data will be ready for detection. This project will detect both texture and image data. To detect texture in an image, Optical Character Recognition (OCR) is required. OCR is the technology that allows users to detect, analyze and extract texture data from the image file (Woodford, 2021). If image detection is enabled (when OCR is on), the image detection will be conducted first to extract text from the source images and add the extracted text back to the pre-processed dataset for detection. If image detection is not enabled (when OCR is off), the detection will be directly conducted after data pre-processing.

There are two stages included in the dark pattern detection process after data pre-processing: (1) Dark Pattern Detection stage; (2) Dark Pattern

Classification stage. The first stage includes the detection of the 6 dark pattern types in the project, within which it can be further divided into 2 parts, one part for the 5 dark pattern types of detection, and the other part just for the Confirmshaming dark patterns detection. The second stage is the classification of the 5 dark pattern types, conducted on the results from the 5 dark pattern types of detection. Finally, the dark pattern classification results and the Confirmshaming dark pattern detection results are combined to be the final detection result for returning.

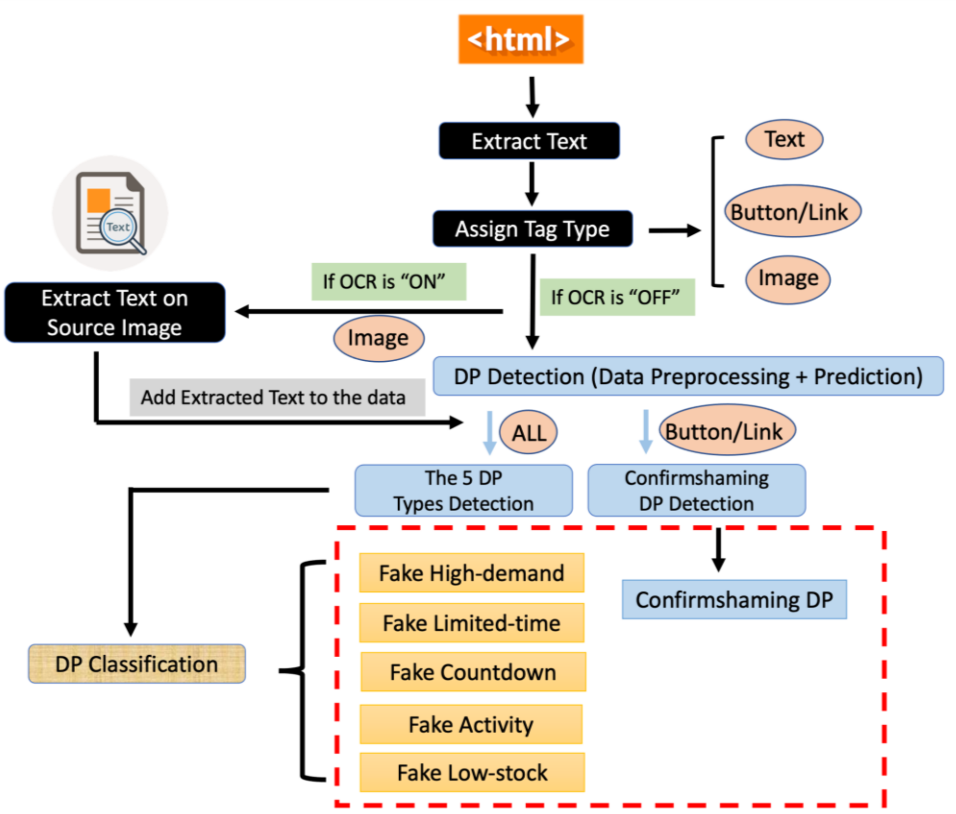


Figure 20 - Real-time Dark Pattern Detection Process

Therefore, there are 3 models that need to be developed for detection purposes:

1. Model 1, for the 5 dark pattern types of detection, which are Fake Activity, Fake Countdown, Fake Limited-time, Fake Low-stock, and Fake High-demand.
2. Model 2, for Confirmshaming dark pattern detection.
3. Model 3, for the 5 dark pattern types of classification.

To train 3 models for detection, public and reliable benchmark datasets containing the six dark pattern types are required. One-sided data is not enough to train the models, therefore a large amount of data of normal content (those content that is not dark patterns) needs to be gathered to enrich the dataset for training.

2.3.3 Dark Pattern Highlighting Strategy

In the initial stage, all detections are based on text. So, the basic strategy is to first extract all the text on the page and at the same time give each text a unique identifier, and then detect each text one by one. If the text detected hits Dark Pattern, its parent HTML element will be highlighted.

2.3.3.1 Technique Alternatives

First, the HTML of the page needs to be obtained through a certain strategy, and then extract all the text from the HTML. There are two strategies to achieve this function: 1. Build a website, the user can enter the URL to be detected in this website, and then the server crawls the HTML of the URL. The problem with this strategy is that the crawled HTML may be different from the page visited by the user, such as if logged in, where the IP belongs, which will affect the accuracy of the detection result. 2. Develop a Chrome Extension. The extension includes a "detect" button, which gets real-time HTML when the user clicks it. This strategy can ensure that the content to be detected each time is the content that the user is accessing.

Secondly, the detection results should return to the page and be highlighted. If the web page strategy is used, the detection results should be embedded into the website in the form of an iframe. The problem with this strategy is that the content of the iframe is not under the control of the website. Once the page is returned, it can only be displayed and cannot be modified. The chrome extension has the authority to modify the content of the page. It can also customize the returned results to meet the functional requirements. However, the disadvantage is that because the extension is based on the Chrome PC browser, mobile phone users and other browsers' users will be excluded.

The chrome extension was finally selected as the current implementation strategy. In the future, the web version of detection will be developed to cover all user groups.

2.3.3.2 Text Extraction

Regarding the realization of the text extraction function, from a technical point of view, the browser side, node side, and python side are all feasible. The problem on the browser side is that the tracking of the data will be lost. The backend can only get the text from the text. Another issue is the difficulty of detecting other Dark Patterns, such as Sneak into Basket, Hidden Cost, which will increase sharply. On the python side, there is a lot of work on machine learning that needs attention. In the end, the function is implemented on the node side.

2.3.3.2 Highlight Strategy

Xmldom.js and Xpath.js are used to implement the highlight strategy. Xmldom is used to convert the HTML string from the browser to HTML DOM, and XPath is used to extract the text in HTML dom. An optional strategy that can be implemented is to add a unique class name to each element when the user enters each page, but this is an overly aggressive strategy. Once the page has too many HTML elements, it will affect page performance. In the end, this strategy was abandoned.

The strategy used is to track all the parent HTML tags of the text and record these tags together with the text to be detected in an object. After that, a request will be sent to the python side. This request contains all the above tags in an array. After the detection on the python side is completed, the python side will return the detected dark patterns according to the previously requested data structure. After the node receives the response and completes tasks such as diary recording, the result is returned to the browser. The browser locates the target texts through the tag and highlights them.

For some pages, due to the page structure, the different text will have the same tag structure. To solve this problem, the second condition of text content matching is also added. Only when the tag structure and text content are the same as the target element Highlight.

2.3.3.3 Compatibility

Link and button detection functions are added, and the above strategies can meet the demand. The image detection (OCR) function is added too. The above strategy also applies to this function. The difference is that the text content is replaced with the URL of the image.

2.3.4 Designing a System Architecture

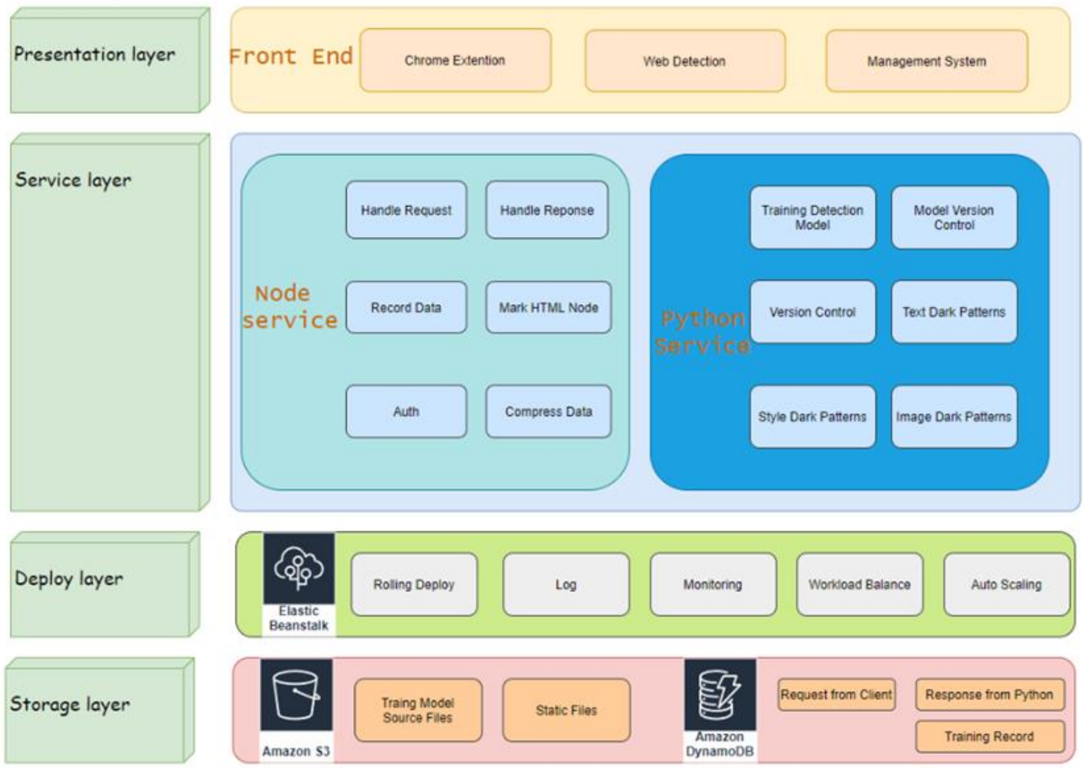


Figure 21 - System Architecture Diagram

The system is separated into 4 layers, which is the presentation layer, service layer, deploy layer and storage layer. The presentation layer mainly refers to the frontend system, which consists of a chrome extension, a website, and a management system. In the service layer, the node service and python service are present. The Node service handles the requests from the presentation layer, and then gives the responses to the presentation layer. The Python service's main function is to detect the dark patterns. In the deploy layer, AWS Elastic BeanStalk is used to do continuous integration, the code is automatically deployed online via this platform. The Storage layer is used to store all the data which includes AWS S3 that stores static files and AWS DynamoDB that stores the models, datasets, and reports.

3. Technical Solution

The overall system will be divided up into three main subsystems.

1) A chrome extension called 'Shopping Tricks Finder',
2) An educational website 'dpexplained.com'
3) A management system.

Shopping Tricks finder is a tool to detect and highlight dark patterns on webpages with each type of pattern having its own unique colour to help users to be able to distinguish between the various types of patterns. This tool will help to stop users from falling victim to these patterns while raising the users' awareness and knowledge of dark patterns.

The educational website 'dpexplained.com' will provide users with more in-depth knowledge of what dark patterns are, what types and categories of patterns there are, and give real-world examples of each type all to help the user gain a better understanding of dark patterns.

The Management system is a system to help manage the reports generated by the users of the chrome extension and the website. The management system will help to add new data to our data set, which in turn will help improve the overall accuracy of the chrome extensions detection.

3.1.1 Functionalities Design

3.1.1.1Card Sorting and Mind Map Design

Before designing low-fidelity prototypes, card sorting can quickly identify core functions and help find the underlying logic between functions according to the user's needs. Meanwhile, it allows us to stand in the user's perspective and understand how users operate and think about the extension. The results of the classification can lay a preliminary foundation for the information structure of the plug-in and have an important guiding role in the design of the prototype.

First, all possible potential user needs were listed. This includes the needs like "users can launch the extension with a shortcut key" that are not closely related to the main functionality of the extension. These broad requirements then need to be addressed by card sorting. All members of the team discussed each area, as participants in Card Sorting. This process verifies the user's way of thinking and the user's mental model, helping to discover more of the user's logic. Finally, based on the sorting, it is easy to determine the functionality that users expect on each page. On the side window, for example, the functionality the user expects to see the details of the detection result, submit the report, or set the detection tool, see figure 22.

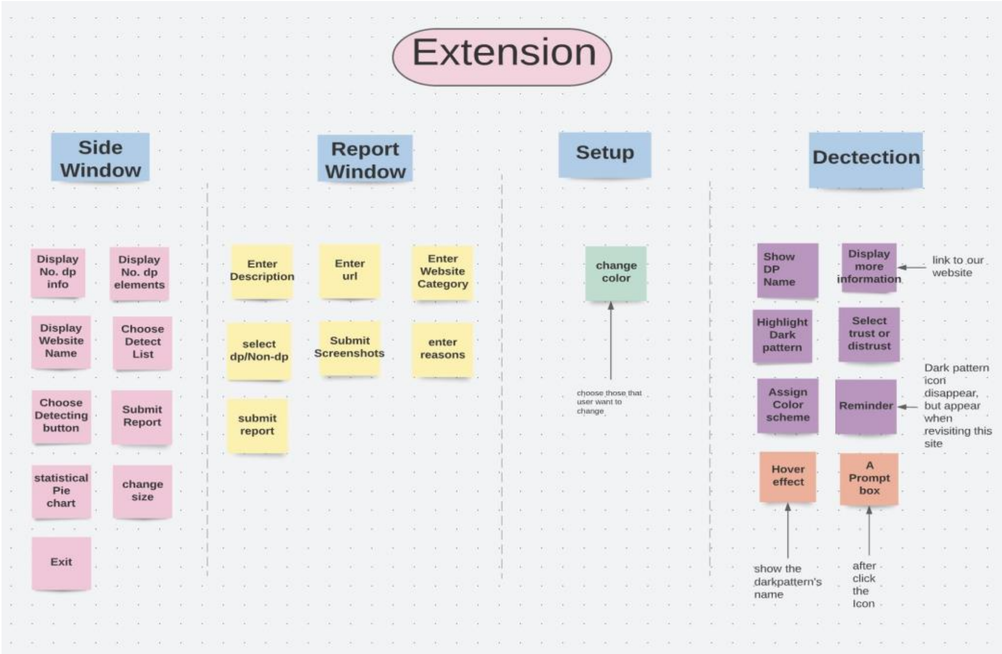


Figure 22 - Card Sorting

Mind maps are used to help card sorting to further organize the logic of users, which is a great help to design the interaction between each page of the extension. In the initial stage of extended design, Mind Map can be used as an important reference for the design of interactive functions of information interfaces at the beginning, see figure 23.

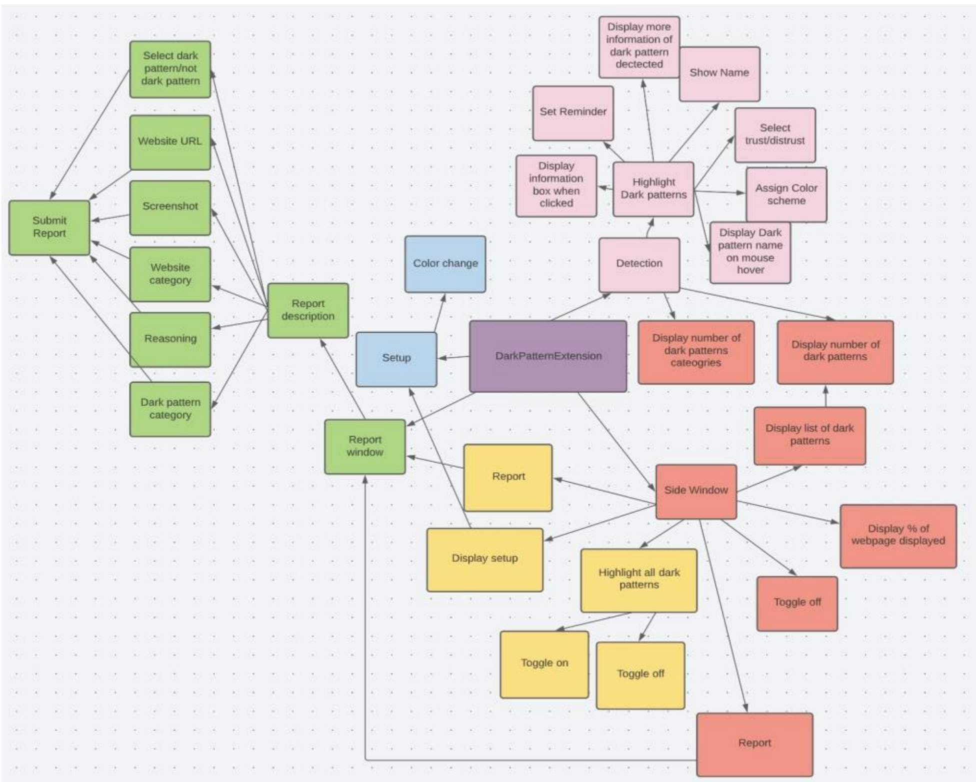


Figure 23 - Mind Map

3.1.2 Detection and Highlighting Feature

The extension's main feature will be detecting and highlighting dark patterns on web pages. Once the user clicks the 'Detect Dark Patterns' button or has 'Auto Scan' turned on and they visit any web page, the extension will take the HTML of the page, removing all the scripting tags, and send it to our back end, where its machine learning models will identify whether or not there are dark patterns in the HTML and then if so, what is the type of the pattern. This data will then be sent back to the front end. This data will consist of each dark pattern found, the string of that dark pattern, the location of it on the page, and what pattern it is. This data will be used to 1) highlight the pattern on the page and 2) highlight it in the correct colour based on what type it is. Each dark pattern type will be highlighted in its own unique colour and have its own unique icon beside it as well, see figure 24. This will allow the users to be able to identify what type of pattern it is easy as well as make it stand out from the web page it's on.

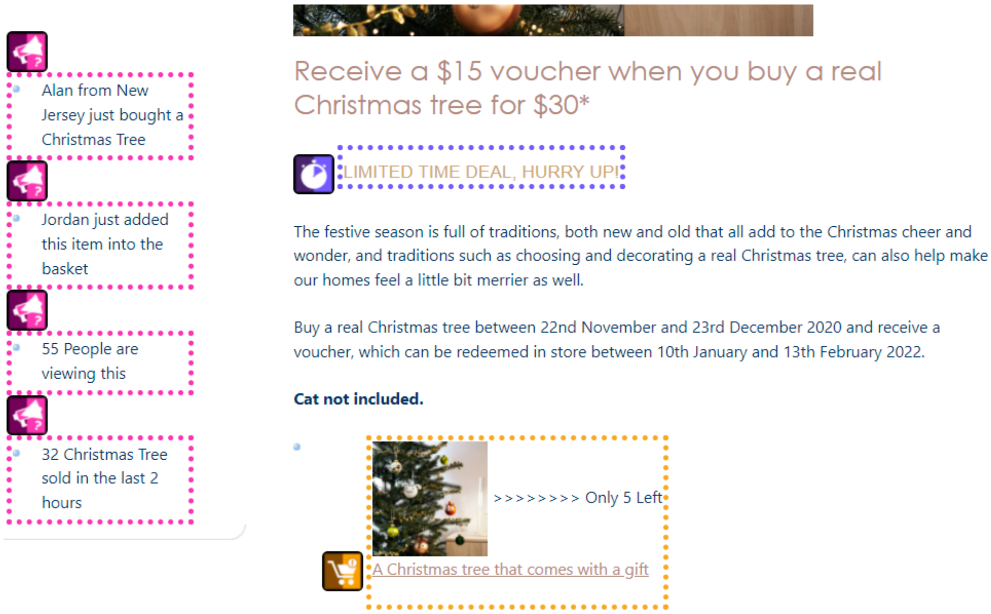


Figure 24-Highlighting icon

3.1.3 Report Feature

Within the chrome extension popup, there will also be a report page, for users to report any patterns that they think are undetected or mis-detected. A user can make a report by clicking the "Help Us" tab and will be prompted with the interface in figure 25. The user is required to fill out for part of the information, the sentence or Keywords of Dark pattern, this is the phrase or what makes the user think it's a dark pattern. The second part is the type of dark pattern, all the possible options are listed in a dropdown box. Next part is the description as to why the user believes it is a dark pattern, and lastly a checkbox to tell us if the dark pattern has been mis-highlighted or not highlighted at all. Once a makes a report, it is sent to our management system, where we can screen and review each of the reports made. Once a report is accepted as a new pattern, it will eventually be added into the dataset used to train our machine learning models, thus improving the overall detection of the system. This gives the user a sense of responsibility and becomes a symbiotic relationship, by the user sending in reports of undetected dark patterns it helps create a more accurate system for them.

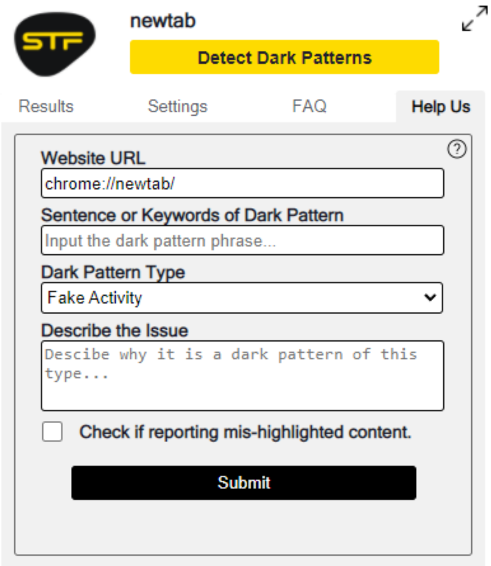


Figure 25 - Interface of the Help Us tab

3.1.4 Customizable Settings

Users will also have a settings page, within the popup of the extension. From this page, the user will be able to configure certain options within the extension. Examples of some of these options are turning on and off auto-detect and choosing which patterns do not get detected on the page and whether they want OCR enabled. The level of customizability makes the system more desirable for a variety of users as different will have different concerns when it comes to what dark patterns they want to be detected.

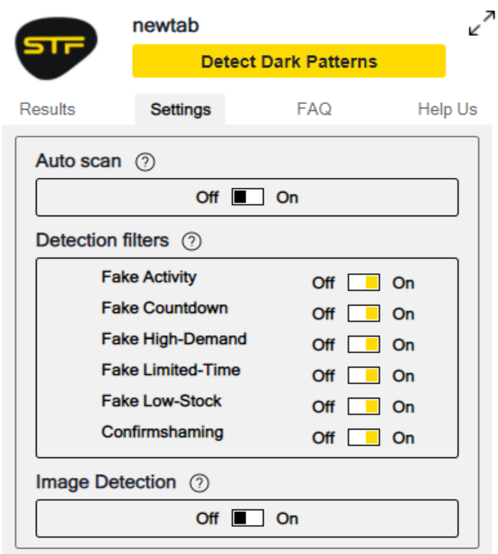


Figure 26 - Interface of the Settings

3.1.5 Website for Dark Pattern Introduction

The website side of the system has two vital functions. The first function is used as an educational resource for the user. As proven through the user evaluation, the average person does not know what a dark pattern is. So, the website is designed to help them learn what dark pattens are. It contains all sorts of information on dark patterns like, what exactly a dark pattern is, the impacts they can have, the categories that they are categorized into, in-depth descriptions of each of the patterns, along with examples of each and a clear idea of which patterns the system can detect. All this information will answer many questions related to dark patterns that a user may have. The website also contains the reporting functionality from the chrome extension, each team member's email in case a user has questions, a video tutorial of how to use the extension, and a link as to where to install the chrome extension on the chrome store.

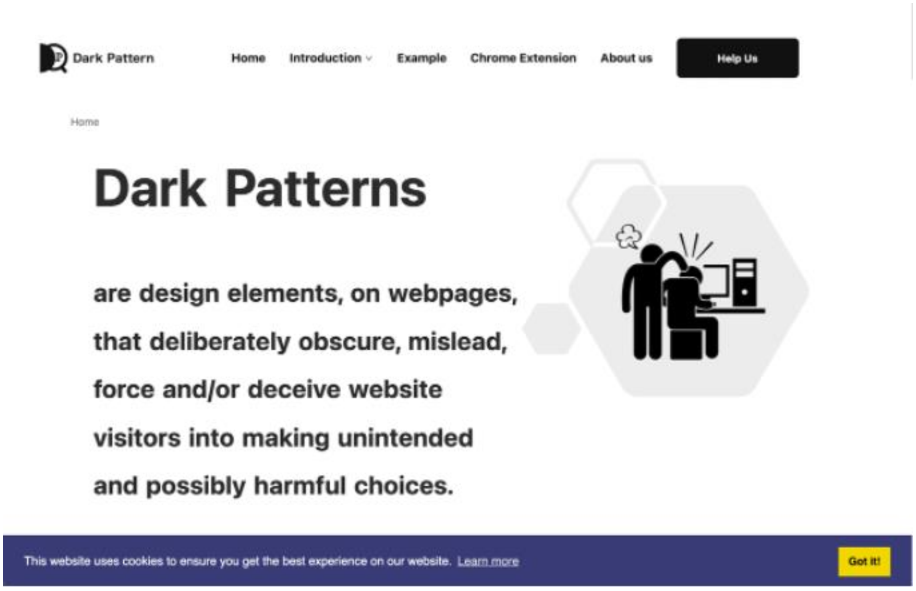


Figure 27 - Website homepage

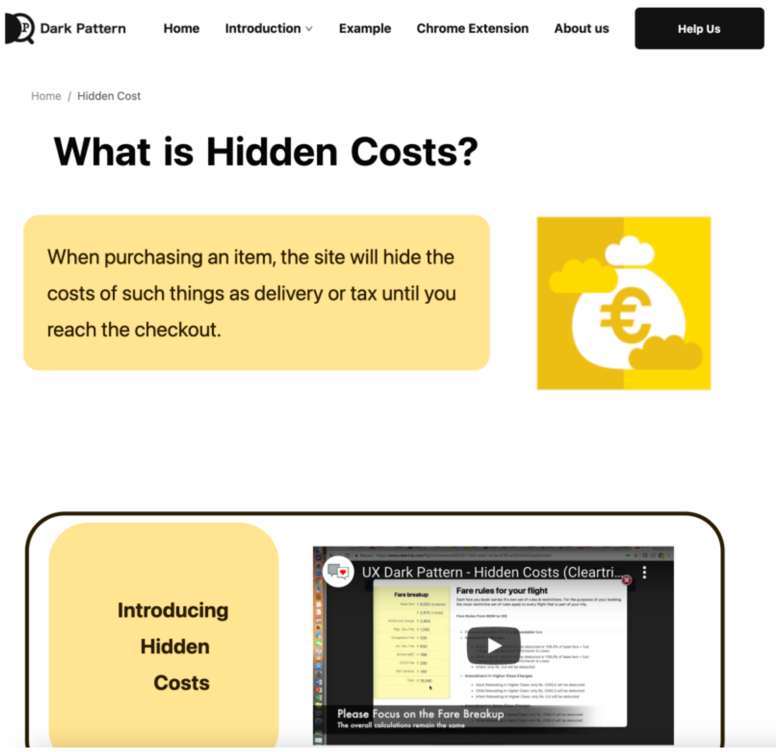


Figure 28 - Website Hidden Cost Introduction Page

It also contains a page filled with all the possible dark patterns the system can detect, this can be used as a reference point or tutorial point for the user to familiarize themselves with the extension.

Figure 29 - Website Testing Page with Dark Pattern Examples

The second function of the website is the administration/management side. The management side is used to screen all the reports made by users, retrain the training models, download all the versions of the datasets.

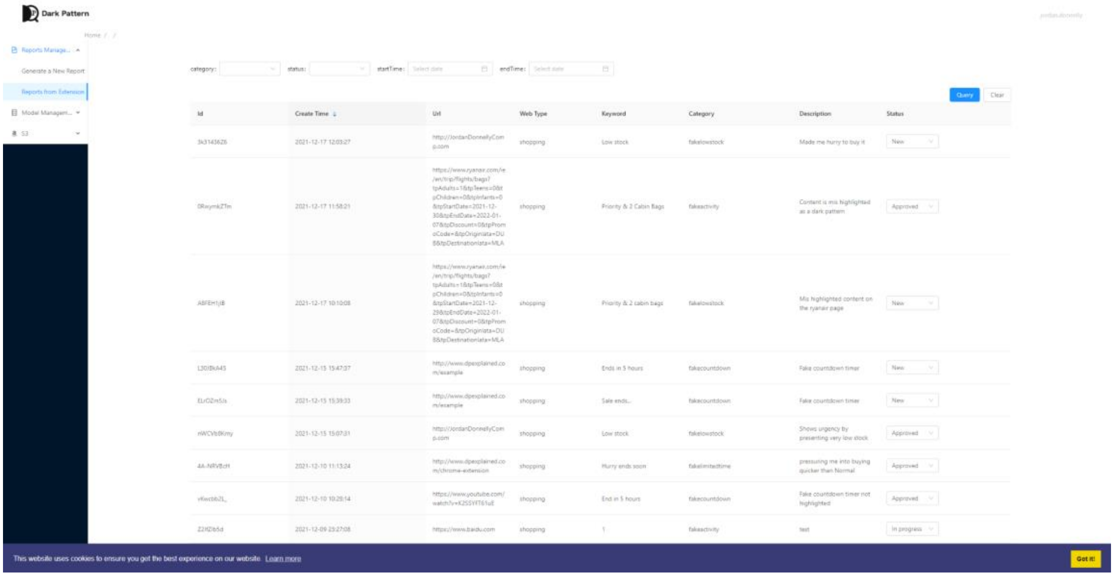


Figure 30 - Management System on the Website

3.2 System Diagram

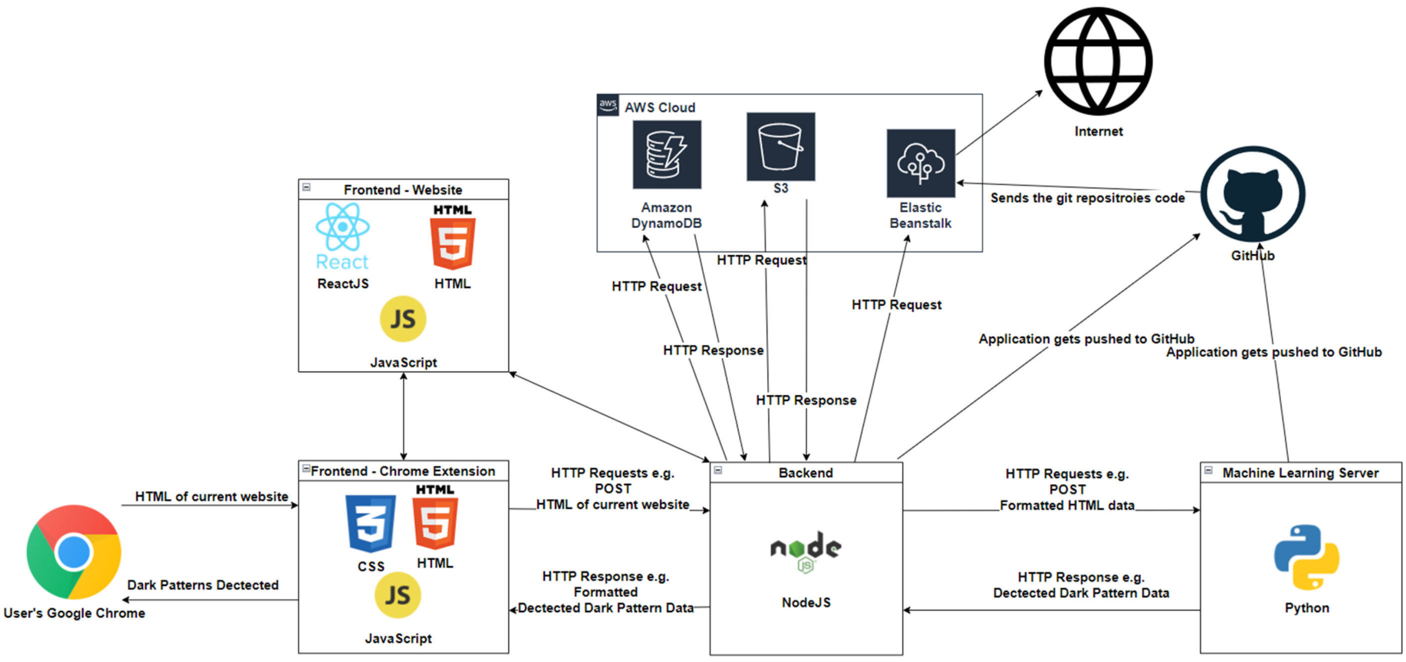


Figure 31 - System Architecture Diagram

Our system can be broken down into three main categories of components:

1) Front-end (Chrome Extension - Website - Management System)

2) Back-End (Node API - Cloud Services - OCR)
3) Data (Machine Learning Model)

3.3 System Components

3.3.1 Frontend - Chrome Extension

The front end of the system will be built as a chrome extension, using JavaScript, HTML, and CSS. Extensions are made up of many different, but cohesive components. (*Getting Started*, n.d.). Our front-end architecture can be broken up into four main types of components within the system. The four types being, the manifest, the background script, content scripts, and popup scripts.

3.3.1.1 The Manifest

The manifest.json file is the only compulsory file within chrome extensions and is read by Google Chrome upon loading it into Chrome. This file contains all the metadata about the extension, which files (such as content-scripts, background scripts, and popup scripts) are registered, what external libraries are being used, and the permissions that the extension needs to function. The manifest version that our system uses is manifest version 3.

3.3.1.2 Background Script

The background script is the main hub of the chrome extension, acting as a link between all the other components of the extension, allowing for message and data passing between them. The background script is entirely composed of JavaScript and contains a lot of the logic used within the extension. The background script works in context to the extension itself and cannot talk to or modify web pages. Users of the extension will not directly interact with the background script either.

3.3.1.3 Content-scripts

Content scripts work in context to the webpages, having access to the DOM, being able to retrieve data from webpages, modifying them, and injecting content onto them. Content scripts do not work in context to the extension itself and must communicate with the background script or a popup script to get data from the extension. Content scripts are composed of JavaScript files and CSS files which are used to interact with and modify the web pages. Content scripts cannot communicate with one another and need to pass messages through the background script if needed to do so.

3.3.1.4 Popup scripts

Popup scripts are used to build the popup UI that appears once the extension icon is clicked. The popup UI acts very similarly to a regular HTML page, having JS and CSS files that it interacts with. The popup UI communicates with the background script or a content script through its JS files and cannot interact with the webpage directly either.

3.3.1.5 Chrome APIs

Chrome extensions have access to many APIs specifically designed for extension development, which is used for the facilitation of message and data passing between the different components of the extension and any external system that the extension needs to communicate with, as well as many other functionalities that the extension might require. A full list of the APIs an extension has access to can be found on the chrome developer website (*API Reference*, n.d.).

3.3.1.6 UI development lifecycle

We began developing our UI through a content script and injecting it onto the web page itself. This allowed the UI to appear and disappear from the side of the user's screen and we also planned on having a small toolbar, which the main UI could collapse back into. However, while developing the UI in this manner, we realized that it was not going to be suitable for a long-term solution as we were running into too many problems with it. A list of these problems is outlined below:

- As we were injecting the UI onto the webpage, it was becoming part of the DOM and therefore would be sent as part of the encoded HTML to the backend, which would eventually cause text to be highlighted on it.

- With the UI being injected onto the webpage, it would inherit default CSS options or classes that the specific webpage, was opened on, had. This meant that if some class from the webpage was not overwritten, it would affect the layout or look of the UI, so it would nearly look slightly different on every webpage it was opened on

- Building a UI through a content script, as it is not an HTML page, is very difficult and as we were getting more data back to the front end, it was becoming increasingly more difficult to manage and display the data in the way we would have liked.

- We also quickly realized that the toolbar was going to serve no real purpose and was just going to cause the same problems as the main UI if we continued to develop it in the same way.

Due to these reasons, we decided to scrap the toolbar altogether and come up with a better solution to display our UI. Next, we looked at keeping the side window style of the UI but now displaying it as an HTML webpage but through an iframe (see the WAVE evaluation tool extension UI). This would allow us to develop the UI as an HTML page with JS and CSS files, making it easier to display the data and handle user interactions a lot easier.

Again, this solution did not end up being an appropriate or easy way of developing and displaying our UI, mainly since it made the message passing within the extension very difficult and unreliable as the page was outside the scope of the extension and had to be used as an external resource, so again this way of developing the UI was scraped.

After these two options had not turned out that they were going to be a viable long-term solution for our UI, we decided to display the UI as a popup that comes out of the extension icon on the top right of Google Chrome. This would not allow us to display the UI as a side window, as it would be more in the middle right of the screen, but it cut out all of the previous problems that we were having. It was no longer being injected onto the webpage, so we didn't have to worry about it inheriting styles from the page or it is passed, to the back end, as part of the encoded HTML. The popup for an extension is developed like a regular HTML page, meaning it became a lot easier to develop the UI and display the data we wanted on it and lastly, as it is within the scope of the extensions, message passing became a lot easier between it and the other components of the extension.

3.3.1.7 Final Front-End Architecture Diagram.

The diagram below shows how the different components in the front end interact with each other, the webpages that the extension will be used on, how the user interacts with the system, and which component interacts with our node server.

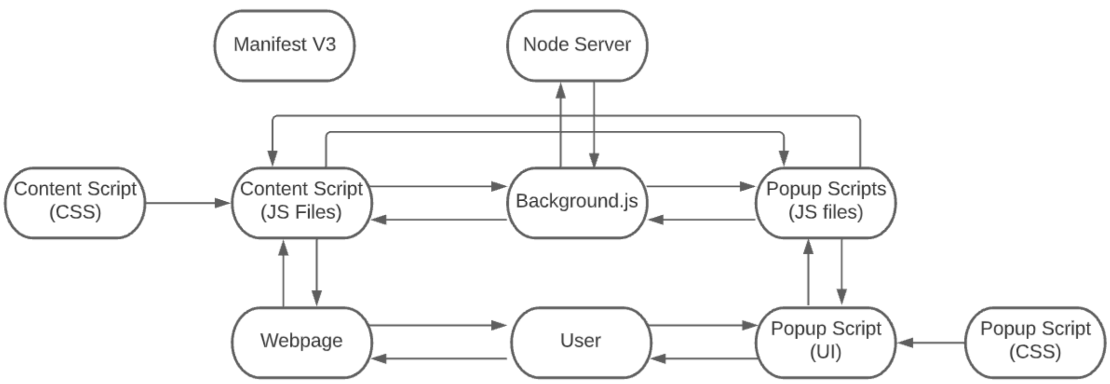


Figure 32 -Front-end architecture diagram

The screenshots shown in figure 33, show the design for the popup of the chrome extension. Each of the three different pages within the popup UI is shown in the mockups.

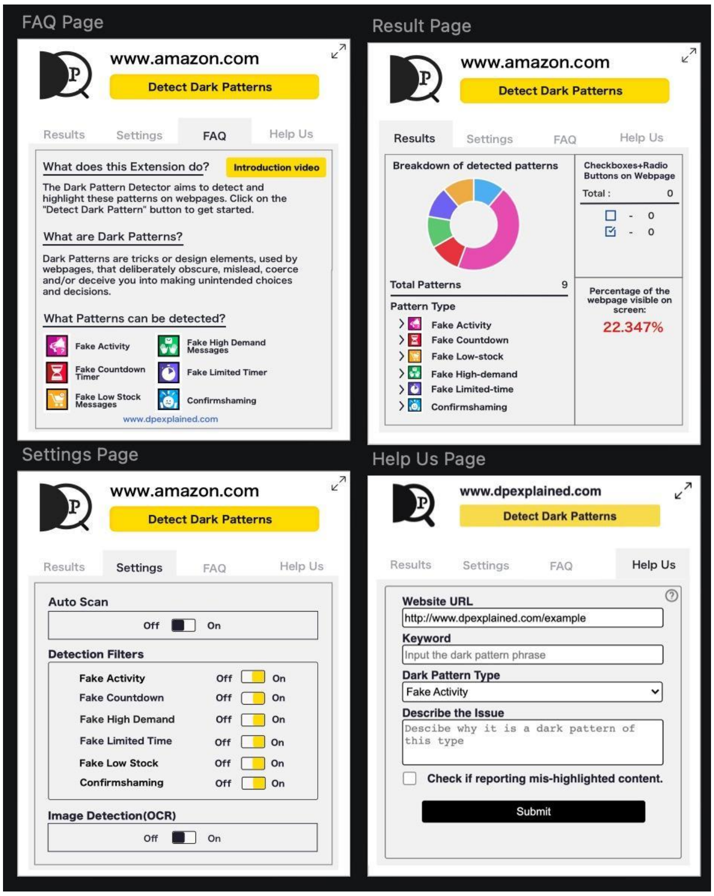


Figure 33 – Extension screenshots

Figure 34 shows the mockup design for the highlighting icon. Each different type of pattern will have its own icon which will be injected onto the page, to help the users see and identify the pattern.

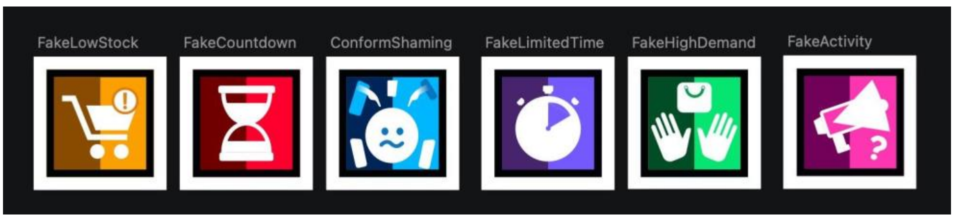


Figure 34 - Extension highlighting icon

3.3.2 Frontend - Portal Website

The educational website dpexplained.com will be built using the JavaScript library, React.js. React.js and Antd.js were chosen to develop the website as it allows for the creation of dynamic and responsive websites with relative ease of development. Webpack is used to bundle the static files.

The portal website is mainly designed for users who can have an overall understanding of dark patterns. It has 5 components: Home page, Introduction pages of each dark pattern, Example page, About Us page, and Report to Us page.

3.3.2.1 Home page

The home page shows an overview of what dark pattern is, what this website can do, and what categories dark pattern has.

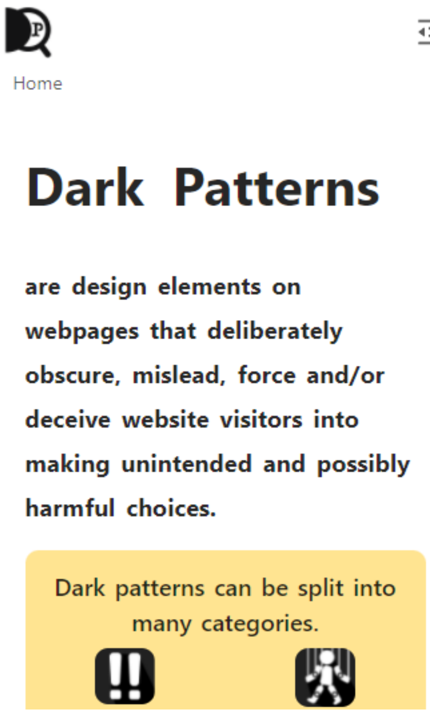


Figure 35 - Phone version of homepage

3.3.2.2 Introduction pages

These are some detailed pages to show each dark pattern. It explains what this dark pattern is and gives some examples of this kind of dark pattern.

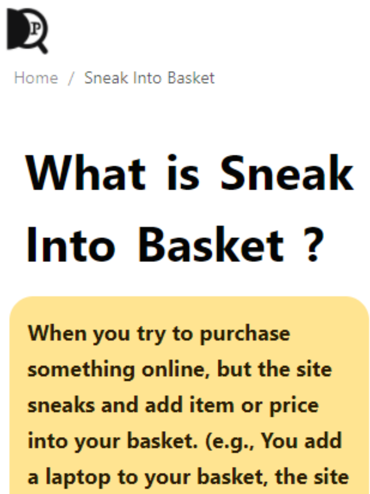


Figure 36 - Phone version of dark pattern introduction page

3.3.2.3 Example page

The example page shows some typical dark patterns. Users can also use the Chrome extension to detect this page to see how the extension works.

Figure 37 - Dark pattern example page

3.3.2.4 Chrome extension page

Chrome extension page has a brief introduction of how to use the chrome extension. A download button also shown on this page.

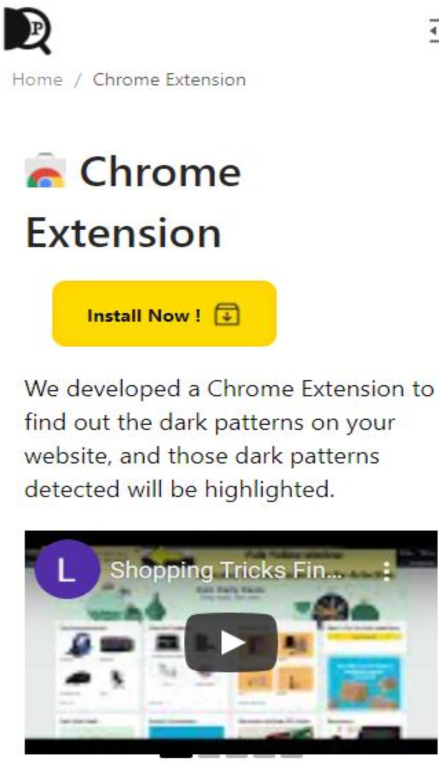


Figure 38 - Phone version of chrome extension page

3.3.2.5 About Us page

The about us page introduces the team members and their goals of doing this project.

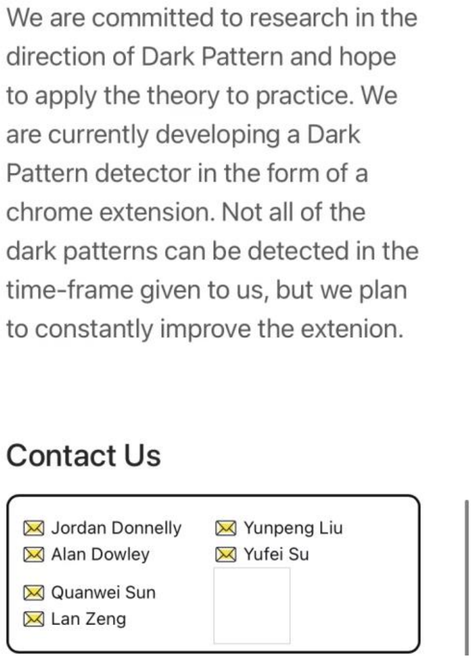


Figure 39 - Phone version of about us page

3.3.2.6 Report to Us page

The report to us page is designed for users reporting new dark patterns that can't be detected by chrome extension so far. Users can input Website URL, Sentence or Keyword of Dark Pattern, Dark Pattern type, and Describe the Issue to help developers improve the accuracy of models.

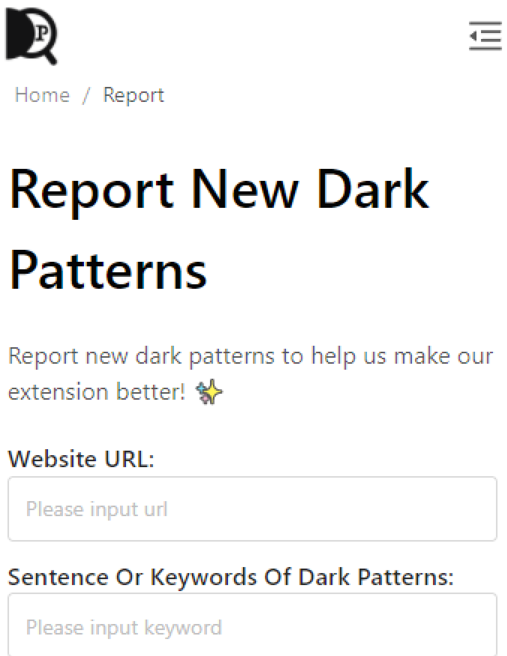


Figure 40 - Phone version of report page in website

3.3.3 Frontend - Management System

The management system and the portal website are in the same project and share the same build tools. The difference is the management system requires access permission to enter.

3.3.3.1 Login page

This system is only used for team members, so the register function is removed after some user accounts are created.

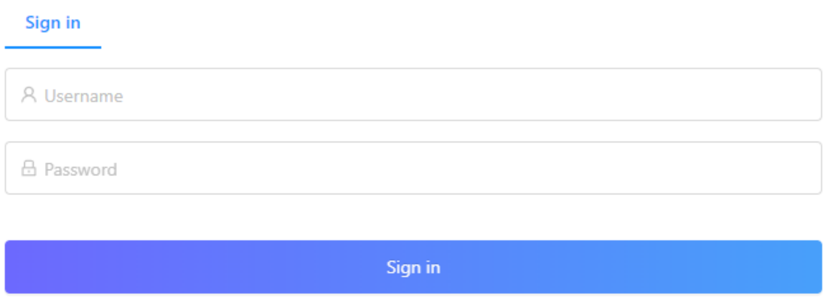


Figure 41 - Management system login page

3.3.3.2 Generate new report page

The Generate new report page shares the same components as which on the portal website.

3.3.3.3 Reports list page

The report list page shows all the reports from the chrome extension and website. It provides basic filtering functions that can filter category, status, start time, and end time. The Query and Clear the Filter option functions are provided.

In the list of reports, auditors can change the status of each report. The status includes new, in progress, approved, declined. “New” means new report. “In progress” means this report needs other auditors’ review. “Approved” means the report can be added as new data in the model dataset. “Declined” means this report is aborted.

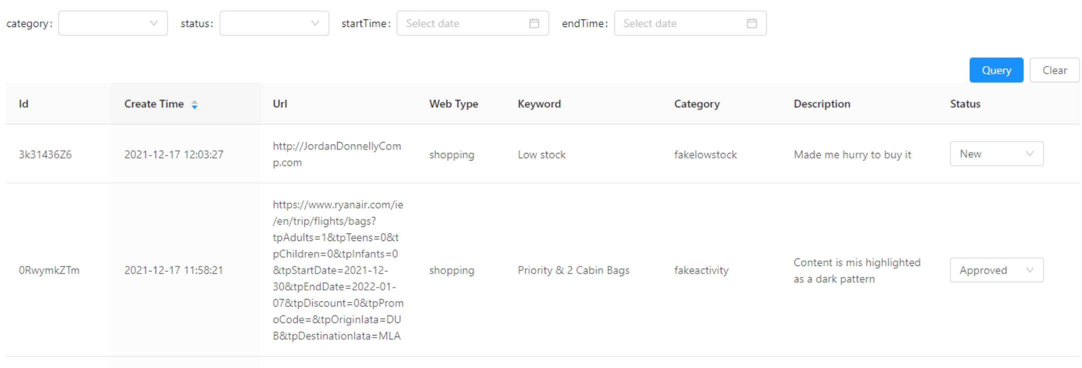


Figure 42 - Report list page

3.3.3.4 Model test page

The Model test page is to detect if one text or image is a dark pattern.

Figure 43 - Model test page

3.3.3.4 Auto training page

When the Generate New Model button on this page is clicked, the node side will generate a new model for the python side's usage.

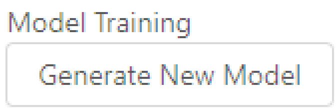


Figure 44 - Auto training page

3.3.3.5 Model list page

The Model list page shows all the new trained models from the auto training page. The python side developers can download the models they want to update their model.

Name	Size	LastModified	Link
V5DarkPatterns.csv	215506	2021-11-25T12:57:50.000Z	Download
V6DarkPatterns.csv	215774	2021-11-26T12:53:13.000Z	Download
V7DarkPatterns.csv	119749	2021-11-26T12:59:31.000Z	Download
V8DarkPatterns.csv	119749	2021-11-26T13:01:57.000Z	Download

Figure 45 - Model list page

3.3.4 Backend and Cloud Services

3.3.4.1 Node JS + DynamoDB

The node service is mainly based on the Express framework and webpack-dev-middleware is also used to ensure that the front-end and back-end code can be developed at the same time. It uses @aws-sdk/client-dynamodb as a dependent library to interact with DynamoDB. Xpath + xmldom + shortid is used for html recognition and preliminary data filtering and uniqueID is added to the tag to reduce the size of the request. Jsonwebtoken is used to encrypt user passwords. Json File is used to read AWS key and secret and pm2 is used to start and manage the node services.

3.3.4.2 Python Server

The python service is mainly responsible for detecting dark patterns, so it is a very thin layer of service. It uses the Flask framework and implements the controller function to interact with the node service. In terms of detection functions, Pandas and Sklearn are mainly used for model training and data detection, and Joblib is used to load the models. If you want to implement image detection and style detection later, you need to introduce other components.

3.3.4.3 AWS SDK

Throughout the application, AWS services are being used. From deployment to database storage to file storage. AWS is the most popular cloud service provider and provides all the necessary services to meet our systems requirements.

3.3.4.4 AWS ELASTIC BEANSTALK

AWS Elastic Beanstalk is the service that is responsible for deploying and hosting both the system's servers. It hosts both the python server and the node server. Having these hosted makes it much easier for users to be able to use them and for the developers to be able to test them. The code of the servers gets uploaded from GitHub, the "origin/main" branch only. All the API calls are then used using the URL of the deployed servers. The hosted URL of the website is http://www.dpexplained.com.

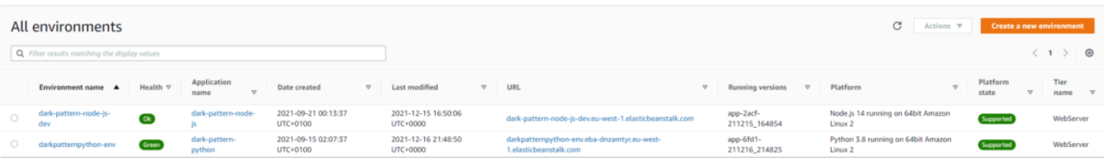


Figure 46 - The two services hosted on ElasticBeanStalk

3.3.4.5 AWS DynamoDB

DynamoDB is a NoSQL database that is fast, scalable, and secure. DynamoDB accepts JSON format. DynamoDB is considered scalable as it claims to be able to support up to 20 million requests per second. Also, the availability of DynamoDB claims to be 99.99%. This can help ensure that our system sustains reliability (Amazon Web Services, 2021). Due to the above benefits, it was decided that DynamoDB will be the primary database. A table is made to store the datasets used to train the machine learning models. Since AWS services tend to work with each other easily, the dataset can then be exported from the database to retrieve the newest, bigger dataset.

DynamoDB is also used to store the reports made by the users and the admin user information used for the management side of the system. All this data can be easily accessed and changed through the back-end node API.

Dataset	Active	Pattern_String (String)	-	0	Provisioned (5)	Provisioned (5)	DynamoDB Standard	Default
Report	Active	id (String)	-	0	Provisioned (1)	Provisioned (1)	DynamoDB Standard	Default
User	Active	Id (String)	-	0	Provisioned (1)	Provisioned (1)	DynamoDB Standard	Default

Figure 47 - The 3 different databases used: Dataset, Report, User

Dataset
Expand to query or scan items.

Items returned (300)

Pattern_String	Pattern_Type:
Only 10 left in stock	Low-stock Message
Only 26 Left	Low-stock Message
Don't miss out, only 2 left in stock	Low-stock Message
24 people have purchased this wine today	Activity Notification
Carol in St. Louis, Missouri bought Original USB Charging Anti-Theft Backpack	Activity Notification
James in Cashion, Oklahoma just bought Glossy Black Brick Geode / Amber Gradient Fast Lanes	Activity Notification
only 2 left	Low-stock Message
Only 1 item left	Low-stock Message
Sale ends soon	Limited-time Message
167 piece(s) left for this promo	Low-stock Message
Ends in 1 Days, 4 Minutes, 44 Seconds	Countdown Timer
60% OFF ENDS IN DAYS HOURS MINUTES SECONDS	Countdown Timer
Only 1 Left in Stock. Buy Soon!	Low-stock Message
Only 3 available	Low-stock Message
LIMITED TIME DEAL!	Limited-time Message
1day21:04:58	Countdown Timer
Only 2 copies left	Low-stock Message
50,000+ bought this	Activity Notification
6,798 sold	Activity Notification
Hurry up! Only 11 items left	Low-stock Message
2,862 sold	Activity Notification
436 users are currently active on our website Just now	Activity Notification
1 person is looking at this item.	Activity Notification

Figure 48 - 300 items stored in the dataset database.

3.3.4.6 AWS S3

AWS S3 is a file storage service provided by AWS. Objects can be stored in a "bucket" and accessed through a web interface or the AWS SDK. AWS S3 is used in our system to store the dataset CSV files. Since the files are stored here and not locally, it increases the reliability of the system as it prevents corrupted files or accidentally deleted files being used. Since the datasets will be stored in different versions, e.g., "V1, V2", this allows the training model to revert to older datasets in case a new one causes an error. These files have been made public so anyone can read and use them.

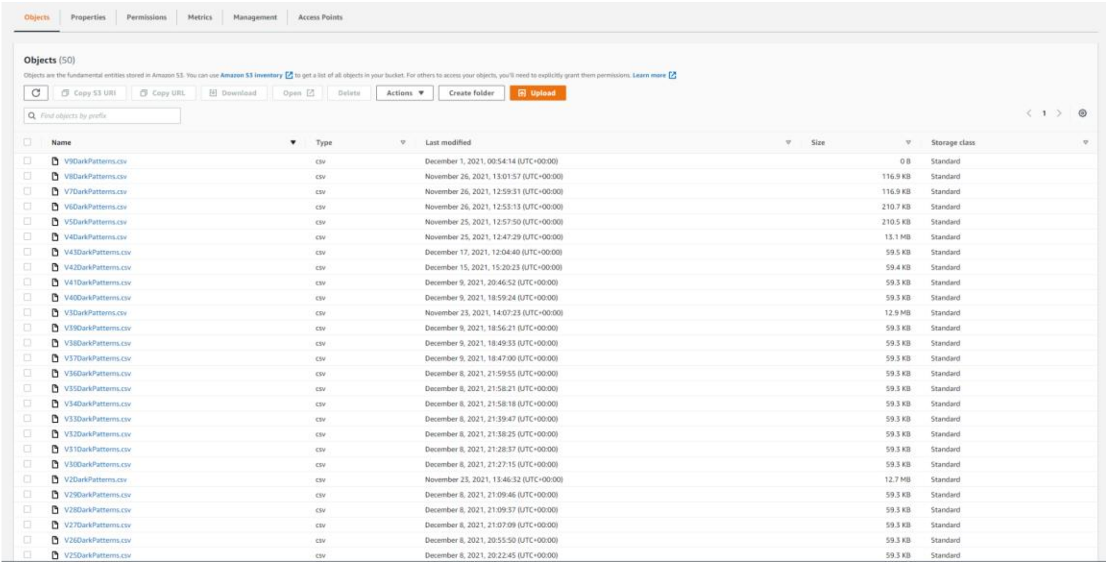


Figure 49 - All the different versions of the dataset files are stored in an S3 bucket.

3.3.4.7 Optical Character Recognition (OCR)

Optical Character Recognition is the technology that allows users to detect, analyze and extract texture data from the image file. The deep learning models for OCR to detect the text from images are Convolutional recurrent neural network (CRNN) which is combined with convolutional neural network (CNN) and Recurrent neural networks (RNN), and long short-term memory LSTM. In the early version of OCR in extension, the CRNN was used to detect the texture in images. However, it can only detect simple pictures with a pure color background. Once the background is colorful or has some noise items, the result will be impacted badly. In the version with LSTM, the current solution is using OpenCV to read the image file directly, then grayscale, threshold or blur it directly before detection.

3.3.5 Data

3.3.5.1 Data Sourcing

There are four parts of the data source for this extension to collect and make use of. Firstly, we found a benchmark dataset called "dark-patterns.csv" from in the research of Mathur et al. (2019), which contains the detected patterns strings and the pattern type/category classification information, and a dataset called "normie.csv" from a dark pattern Chrome extension project on GitHub, which contains the strings of normal content on the website. Secondly, we need to enrich our dataset for training models with more text strings from website content. Thirdly, we need to get real-time HTML on users' current page using JavaScript for the text classification process to detect potential dark patterns. Lastly, we found a website with gathered examples regarding Confirmshaming dark patterns, and the examples there are used to train and evaluate the Confirmshaming detection models.

The data source for training dark pattern detector are given in Table 5.

Table 5 - Data Sources for training dark patterns

Data Source Description	***URL of the Data Source / Method***
Dark Pattern Content dataset from Journal	https://github.com/aruneshmathur/dark-patterns/tree/master/data/final-dark-patterns
Normal Website Content dataset from GitHub project	https://github.com/NicholasTung/dark-patterns-recognition/tree/master/train_classifier
Additional Normal Website Content data and Dark Pattern Content data from Web Scraping result	Collected by Web Scraping
Confirmshaming Dark Pattern Examples from Dark Pattern website	https://confirmshaming.tumblr.com/

3.3.5.2 Data Collection

(1) Data for training dark pattern detector:

(a). Detecting Five Types of Dark Patterns (Exclude "Confirmshaming"):

The dark pattern content dataset and normal website content dataset can be downloaded from their public GitHub repository. The extra dataset containing normal website content and dark pattern content is obtained by using a web scraper built with a python script using the selenium package, to collect additional normal content and dark pattern strings. The three datasets are combined and re-labeled for the model training. The example of the organized dataset used for training is shown below, containing two columns including "Pattern String" which is used for listing website content strings, and "Classification" which is used for labeling target value, where "0" indicates the string is a dark pattern, and "1" indicates the string is not a dark pattern:

Pattern String	classification
FREE SHIPPING ON ORDERS OVER $100!	1
SOME EXCLUSIONS APPLY - LEARN MORE	1
HAVE A QUESTION? - CONTACT US	1
WELCOME TO 034MOTORSPORT!	1
SHOP AUDISHOP VOLKSWAGENPERFORMANCE SOFTWARE034 APPARELLOCAL SERVICENEWSRESOURCES	1
ADJUSTABLE SOLID REAR SWAY BAR, 8J/8P AUDI TT/TTS/TTRS & A3/S3/RS3, MKV/MKVI VOLKSWAGEN R32 & GOLF R	1
MORE VIEWS	1

Figure 50 - Dataset Example for training / evaluating dark pattern detection

(b). Detecting Confirmshaming Dark Pattern:
The original Confirmshaming dark patterns strings can be extracted from the dataset in the research of Mathur et al. (2019), then combined with the normal content from the web scraping data to form the dataset for training the Confirmshaming detection model. Later the dataset can be enriched from the examples from the Confirmshaming Example Website. The example of the organized dataset used for training is shown below, containing two columns including “Pattern String” which is used for listing website content strings, and “Classification” which is used for labeling target value, where “Dark” indicates the string is a Confirmshaming dark pattern, and “Not Dark'' indicates the string is not a Confirmshaming dark pattern:

Pattern String	classification
SUBSCRIBE	Not_Dark
Nope, I don’t care enough.	Dark
NO THANKS. I DON'T LIKE TO SAVE MONEY	Dark
Increase My Conversions	Not_Dark
Sign Me Up!	Not_Dark
No thanks, I don't care what my cat eats.	Dark

Figure 51 -Dataset Example for training / evaluating Confirmshaming dark pattern detection

(2) Data for obtaining real-time website content:
The real-time HTML on users’ current page is collected through the extension side using JavaScript. Then the JavaScript is used to manipulate the obtained HTML, extracting content string for dark pattern detection and the corresponding tags for locating the dark pattern strings on the webpage.

(3) Data for training dark pattern classifier:
The dataset for training and evaluating the dark pattern type classifiers is extracted from the research of Mathur et al. (2019), by selecting the rows only containing any of the 5 dark pattern types. Later the dataset is enriched by using the web scraping data. The example of the organized dataset used for training is shown in figure 52, containing two columns including “Pattern String” which is used for listing website content strings, and “Pattern Type” which is used for labeling target value which contains 5 types of dark pattern.

Pattern String	Pattern Type
Only 2 left	Low-stock Message
Only 3 left	Low-stock Message
9 people are viewing this.	Activity Notification
5338 people viewed this in the last hour	Activity Notification
Crystal Li in Flushing, United States purchased a	Activity Notification

Figure 52 - Dataset Example for training / evaluating dark pattern classifiers

(4) User's Report Data:
The data collected from the user's report is like those used to train the dark pattern detection model. The content of the report includes "Keyword" which is "Pattern String" in the dataset for training, "URL" which is automatically filled in, "Pattern Type" which is multiple choice, and "Description" which is collecting user's opinion on the reported content. Once a report has been submitted successfully it must go through a screening process before getting stored into the dataset to prevent data that may be harmful or have a negative effect on model training.

3.3.5.3 Data Storage

The real-time HTML data is not stored in our dataset, it will be directly passed to the machine learning models after initial formatting in JavaScript. There are two parts of the data stored in our database.

(1) Dataset for training dark pattern detector:
The dataset is stored in an AWS DynamoDB database, the initial storage of the database was made manually using S3 to store the dataset as a CSV then uploaded to DynamoDB. This can then be used to develop a newly updated dataset that will be stored in S3.
Whenever the detection model needs to be trained again, the data from the database is exported and stored in S3 and as a CSV file, and the URL is displayed, to easily retrieve the said file.
Each time a new dataset is exported to S3, there is a naming convention applied to it, "V1" for the 1st version, "V2" for the second, and so on and so forth. The reason for this is to implement version control, if a new dataset causes an error or even worse accuracy then it is easy to revert to the older one.

(2) User's Report data:
The user report data is also stored in its own DynamoDB database. The screening process of the data is done on the management side of the system. In the management system, all reports are listed, with their status, "New, In Progress, Approved and Declined". The status can be changed, and if it is changed to approved then the report data gets stored into the dataset database for future auto-training.

3.3.5.4 Difficulties and Choices

1. *Limited Data*

   The main difficulty is the limited data to use for training and evaluation. For the starting point, we only had the pure dark pattern dataset from the research of Mathur et al. (2019), and a normal content dataset with a few dark pattern instances inside. In total there were less than 2000 rows of examples after removing the duplicates to train the models, which was not enough.

   Therefore, a python script for web scraping was used to obtain more textual content from various shopping websites. The web scraped data was firstly examined by the current detection model to check if there were any real dark patterns or misclassified dark patterns (which are not dark patterns, but normal content instead) in the web scraped data. The normal content and the misclassified dark patterns were merged and labeled to be "Not Dark Pattern", while the real dark patterns in the web scraped data were added into the "Dark Pattern" target group.

2. *Overfitting caused by dataset*

   At first, we were using the whole dataset from the research of Mathur et al. (2019) for Pattern Category detection, however, the overfitting happened where during the training and validation the precision could be higher than 90% but on the real-world data, the precision went lower than 25% for some websites. The variance in accuracy indicates the low training error and high-test error.

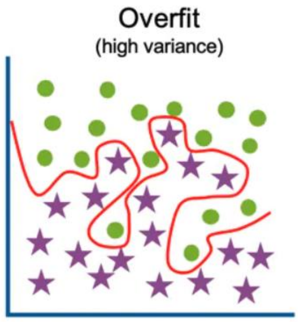


Figure 53 - Overfitting Illustration (Indicates low training error and high-test error)

   The dataset was examined and compared with the prediction results; we found the problem regarding the noises in the dataset. On the one hand, not all the dark patterns can be detected based on the text only, and when we used the machine learning models trained based on text-only to detect them all, there would be large amounts of misclassifications. Therefore, we select the dataset with only the 5 pattern types from the original dataset that can be fully automatically detected by the machine learning models to train the models. On the other hand, there were huge amounts of text noises that can be misleading for models to learn from them, for these words we later manually checked and removed them from the training dataset.

3. *Underfitting caused by dataset*
   For the model training of Confirmshaming detection, underfitting happened for Bernoulli Naive Bayes model, as the dataset was highly imbalanced, where it was around 1:47 ratio in the training dataset. From the accuracy perspective, the model could reach around 98% by just predicting everything into one class, which was "Not Dark '', so the models chose the easiest way to learn from the data, by giving all the prediction results to be Not Dark Pattern. The recall in both training and testing was near 0, which indicated high training error and high-test error.

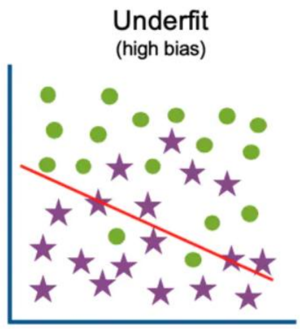


Figure 54 - Underfitting Illustration (Indicates high training error and high-test error)

   To prevent the Bernoulli Naive Bayes from choosing the easiest way to do prediction, the dataset was balanced using both duplicating strategy and SMOTE (Synthetic Minority Oversampling Technique). Then the Bernoulli Naive Bayes started learning from the real distribution of the data, although overfitting was observed later the evaluation dataset.

3.4 Key Technical Contributions

1. The extension (Detection purpose)
   A Google Extension that can detect and highlight six different dark patterns on shopping websites. It allows users to choose to detect all kinds of dark patterns, types, or parts of them.

2. Introduction Video (Illustration purpose)
   The video on YouTube introduces how to use the extension in detail. Users can follow the video to experience all functions in extension.

3. Website (Educational purpose)

The dark pattern website contains an introduction to all dark patterns. Users can get more detailed information about dark patterns. It also includes the link about installing the extension, report function, example page, and introduction video.

4. Example Page for testing Extension (Guide purpose)
   The example page is on the introduction website. It provides all six dark pattern samples in texture and images for users to use to detect.

The biggest key contribution was to be able to detect dark patterns. Shown in table 6 is a list of the dark patterns which can be detected.

Table 6 - Dark Patterns that can be detected

Pattern Type	***Description***	***Detection***
Fake Activity	Informing the user about other people's activity on the website, including the behavior of purchasing, viewing, visiting, etc., which may not be truthful. (e.g., "3 people are viewing this item now")	(1) Gather the text content of the HTML on the webpage. (2) Apply Natural Language Processing in machine learning / deep learning to achieve fully automatic detection of these 5 pattern types, based on the text only.
Fake Countdown	Using a countdown timer to alert users that a discount or deal is about to expire, which only purposely creates urgency for the purchase (e.g., "sale ends in 12h20m33s")	
Fake Limited time	Giving users the impression that a deal or sales only for a limited amount of time or is about to expire soon, without stating a specific deadline. (e.g., "sale ends soon", "only available for a limited time")	
Fake Low-stock	Informing users about the limited availability of a product, making it more desirable to users. (e.g., "only 2 items left in stock")	

Fake High-demand	Informing users that the product is in high-demand and will sell out soon, thereby making it more attractive to users. (e.g., “this item is in high demand”, “selling fast”)	
Confirmshaming	Invoking language and emotion (shame) to convince users not to make a certain choices or guilt tripping users into opting into something. (e.g., “No thanks, I don’t want to save.”)	(1) Gather the text of the buttons and links on the webpage. (2) Use Natural Language Processing in machine learning to achieve fully automatic detection based on text only.

4. User Evaluation

User evaluation is an integral part of any product or system design. It allows designers to get a better understanding of current design issues and helps designers to reflect on the needs and wants of the users. All while, attempting to shine a light on the context, stimuli, environment factors, time factors, and a lot more (IshΔn, 2019). User evaluation and usability testing can be done in many different forms, from a simple questionnaire about the product to letting the users use it and gathering their reactions and thoughts throughout the process.

In this part, a description of what components of the system are being evaluated. Then, what user evaluations have been done on the system and the different types of user evaluations used each time are discussed. Next, the design of the user evaluation experiments, going through how they were set up, how data was collected, the users, and how data was analyzed, is talked about. The results of each user's evaluations will follow from that, where it is shown the quantitative and qualitative results and discuss some of the findings as well. Lastly, will be the conclusion, which is an overview of the findings as well as thoughts on what could of been done differently.

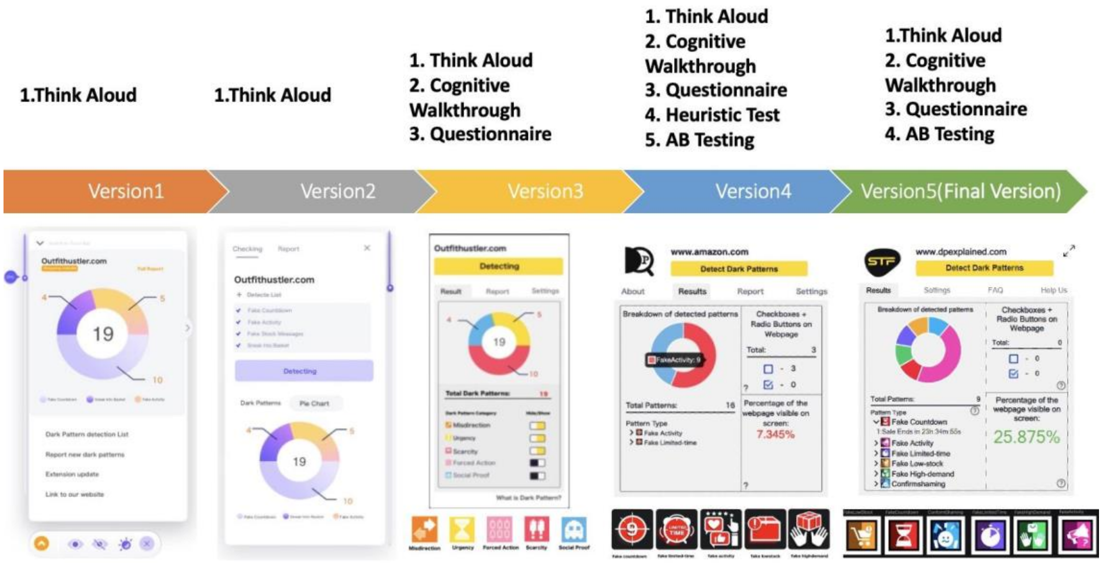


Figure 55 - Image of user evaluation timeline

4.1 Evaluation Scope and Perspectives

Below is a list of all the components of the system which were evaluated throughout the development lifecycle and ones that were evaluated at the end to see how accurately they perform across a variety of webpages.

1) The UI - Multiple evaluations of the UI happened throughout the development lifecycle. This evaluation is very important to develop a UI that is intuitive, easy to use, and accessible to all different types of

users. Features on the extension should be easy and obvious to use and doing multiple evaluations uncovers all the problems that might be overlooked throughout the development.

2) Checkbox and Radio Button Scanner - This function on the extension needs to be evaluated, to see what accuracy it is, as not every website develops its checkboxes and radio buttons in the same manner. Therefore, it will be important to test this function across a variety of websites to get an estimate of how accurate it will generally be.

3) Machine Learning Models - The machine learning models were each evaluated, to find which one gave the most accurate results for detecting dark patterns on webpages. This is very important to maximize the entire accuracy of the extension's detection capabilities.

4.2 Evaluate Usability

Five user evaluations were carried out over the course of the project. The first and second evaluations are evaluating a High-fidelity prototype, which was held within the group and only let one or two users participate. The processes are single, using only cognitive walkthrough methods. The last three evaluations are for the formal version, the processes are similar, using both cognitive walkthrough and questionnaires. However, in the first evaluation, we did not consider any expert users. In the second experiment, expert reviews and AB tests are included to improve the testing.

4.2.1 Methods Overview and Usability Metrics

4.2.1.1 Proposed Questions

The questions below are those used in the formal evaluation, and they will be present them from two aspects: Cognitive Walkthrough and Questionnaire.

Cognitive Walkthrough Questions

For user evaluation, three different testing methodologies are applied. Cognitive walkthrough, questionnaires, and heuristic evaluation. In terms of Cognitive walkthrough, three questions, were put forward to the testers, which are as follows.

1. Is the user able to locate the area to complete the task?
2. Is the user able to make the correct action without assistance?
3. Does the user associate the correct action with the correct expected result?

Observers collected data on these three questions as a basis for evaluating whether the user could complete the task.

In the questionnaire evaluation, the questions needed to be collected are divided into two categories; one is open-ended qualitative questions, and the other is quantitative questions, using a Likert Scale or star rating. Using qualitative questions helps to assess the whole system more comprehensively, while quantitative data can show the severity of the problem.

The test subjects were five ordinary users and two experts, who first tested the application via 8 user tasks and then were presented with 12 open-ended qualitative and quantitative questions. The user tasks are listed below:

User Task:

1. How would you find more information on Dark Patterns?
2. How would you detect Dark patterns on a webpage?
3. How would you report a dark pattern on a webpage?
4. How do you turn on/off Autoscan?
5. How do you turn on/off the detection of certain types of patterns?
6. How would you locate highlighted patterns on webpages?
7. Can you find what each icon represents?
8. Where do you find information about checkboxes and page location, in the extension?
9. How would you do image detecting?

Questionnaire Qualitative question

Qualitative questions directly identify major usability issues in the interface.

1. What is your opinion on the design and colours used within the extension?
2. Would you change the layout or design of any part of the extension if you had the chance?
3. Do you understand what each icon means and how it relates to the pattern type?
4. Do you understand what needs to be filled out in each section of the report page? Why or why not?
5. Do you understand what each feature in the extension does?
6. Is there too much or too little information displayed in the extension?
7. What do you think is the best thing about the extension?
8. What do you think is the worst thing about the extension?
9. Do you have any suggestions on how the extension could be made more user-friendly?
10. What other features would you like to see in the extension?

11. Do you find this extension useful?
12. With the help of this extension, do you know what a Dark Pattern is now?
13. Would you like to use this extension or recommend it to your friends and families?

Questionnaire Quantitative questions

1. The text in the extension is easy to read and understand. (Likert Scale 1-5)
2. The colour used across the extensions UI, make it easy to read the information in the extension. (Likert Scale 1-5)
3. The extension has the same feel throughout its entirety. (Likert Scale 1-5)
4. Detection data presented in the extension is always in the same manner across different web pages. (Likert Scale 1-5)
5. The amount of information on a single page of extension is appropriate (i.e., “About” page, “Result Page”, “Report” Page, and “Settings” page). (Likert Scale 1-5)
6. Rate the quality of the information displayed in the UI of the extension. (Star Rating 1-5).
7. Rate the relevance of the information displayed by the extension. (Star Rating 1-5).
8. Rate consistency of extension design style. (Star Rating 1-5)
9. Rate the visual clutter of the user interface of the extension. (Star Rating 1-5).

According to Dewi et al., 2020. Cognitive drills are used to capture data about the implementation of extended functionality, time consumption, and so on. Questionnaires were used to obtain user satisfaction, ease of use, readability, accessibility, visual perception, data consistency, etc. User experience data from cognitive exercises and user evaluation questionnaires can be combined to suggest improvements. Heuristic Evaluation techniques discover usability problems in design from an expert's perspective.

4.2.1.2 Experimental Method

4.2.1.2.1 Overview

The user experience evaluation of the Dark Pattern Extension is an evaluation study that integrates qualitative and quantitative data and adopts a mixed method. The mixed method is a combination of quantitative and qualitative methods, in which the collection of experimental data is divided into the following stages:

1. Determination of user tasks and questionnaire.

2. Establishment of experimental steps
3. Data collection
4. Data analysis
5. Put forward suggestions for improvement. At this stage, conclusions are drawn based on data analysis results, and suggestions are put forward for the improvement of the extension application interface.
6. Implement the change plan.
7. Feedback results are obtained from AB testing.

<u>User Task Evaluation:</u>

To test the functionality of the extension application using a cognitive walkthrough approach. In this approach, users are provided with information about performing pre-set user tasks, and the tester records user responses and fills out forms. These tasks are the actual actions of the user when using the plug-in.

At the end of each task, the recorder answers 3 default questions and completes the recording form. The questions are as below:
1. Is the user able to locate the area to complete the task?
2. Is the user able to make the correct action without assistance?
3. Does the user associate the correct action with the correct expected result?

If the answer to all four questions is yes, the mission is called a "success story." If the answer to one of the four questions is no, the mission is called a "failure story" of that mission. Finally, performance was evaluated according to the task completion rate table provided by the observer.

User Experience Questionnaire Evaluation:
There are 9 questions to measure user experience from six aspects: Readability, Data correlation, Accessibility, Visual aesthetics, Visual overload, and satisfaction. The questionnaires will be distributed to the participants after the Cognitive Walkthrough. The questionnaire consists of the above 8 questions. The score for each availability scale consists of the interval scale score. Participants' responses to the questionnaire were submitted and stored in a Google form.

4.2.1.2.2 Data Collection
During the survey, record their behavior or reaction from three directions as mentioned in the Cognitive Walkthrough questions above. Cognitive Walkthroughs are used to evaluate the usability of a product (Dalrymple, 2018), by focusing on the level of easiness and the length of time taken for users to complete their tasks.

Table 7 - Data collection for Cognitive Walkthrough (3 Answers & Time taken)

	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8
	Task Completion / Time Taken							
User 1	y/y/y/5s	y/y/y/6s	y/y/y/15s	y/y/y/9s	y/y/y/5s	s/s/s/15s	s/s/s/5s	y/y/y/5s
User 2	y/yn/3s	y/y/n/3s	y/y/y/13s	n/n/n/23s	n/n/n/30s	s/s/s/3s	y/n/n/13s	y/n/n/3s
User 3	y/y/y/7s	y/y/y/8s	y/y/y/7s	y/y/y/7s	F/17s	s/s/s/7s	n/n/n/7s	y/y/y/17s

In this evaluation test, there will be mixed quantitative and qualitative questions in the survey. The first nine questions are quantitative questions. And the last are qualitative questions. The feedback, suggestions, and issues given by users from the questionnaire will all be qualitative. The Likert scale of 5-point agreement scale for users to score: (1) Strongly disagree; (2) Disagree; (3) Neither agree nor disagree; (4) Agree; (5) Strongly agree, was used. The metrics which are measured in the questionnaire are shown below:

Readability score: It tells how easy for users to read the information in our Dark Pattern detection extension.
Data relevance: According to Hes (2017), data relevance is the extent to which data answers or gives insight into the question of the target user. It measures if relevant dark pattern data, is provided.
Accessibility: Accessibility allows users of all abilities to understand, use and interact with our extension (Kuar,2018). It should be made for all users with different backgrounds to use our product.
Consistency: Consistency needs to be achieved both in terms of visual experience and design. It’s important to provide users with consistent experience in the extension.
Visual Weight:
Visual overload: Options and colourful icons, charts may overload the user's visual sensor and confuse them.
Visual aesthetics: A successful user interface will make users enjoy using our extension. It defines a design’s pleasing qualities and makes them more tolerant of usability issues. (International Design Foundation)

The relationship between evaluation indicators and question index is shown in Table 8.

Table 8 - Relationship between evaluation indicators and question index

Evaluation Indicators	Question Index
Readability score	Q1
Data relevance	Q6 and Q7
Accessibility	Q2
Consistency	Q3, Q4, and Q8
Visual weight	Q5 and Q9

4.2.1.2.3 Selected Subjects

It is desired that the system can be used by all sorts of people ranging from Dark Pattern experts, everyday shoppers, those who have fears of being deceived, etc. Due to this, 3 kinds of users for our system, were evaluated:

- People like the personas: These are users that share traits with our created personas. Selected users, regular online shoppers.
- Common person: This is no person, but someone who does not share many traits of our created personas. Relatives, friends, and classmates.
- Experts: These are considered experts in the field, whether it be experts in UX or Dark Patterns. Lecturers.

Without evaluating such users, it would be immensely difficult or near impossible to reach a sufficient level of usability throughout the system that is desired.

These are the representative samples because they match the target user groups. The idea is to allow as many people as possible to use our system and for it to benefit them, regardless of if they have knowledge of dark patterns or not. A Cognitive Walkthrough was used for evaluation in the first half of development. The main issues that the users listed were as follows:

- There is no information on what a dark pattern is.
- The highlighting feature was too dull to see.
- The users had to search to find the highlighted dark pattern.

These tests were conducted with people who are like the personas and common people. The evaluation of these results allowed us to change the UI of the chrome extension to become more user-friendly. We added a page dedicated to informing the user as to what a dark pattern is. We changed the colour and border of the highlighted dark patterns to be clearer and added a feature that would allow the user to be directed to any of the selected dark patterns.

In the second half of the development, we would conduct surveys and questionnaires as well as cognitive walkthroughs with the 3 user groups listed above. The testers are sourced through relatives, friends, classmates, lecturers, housemates, and those who are frequent online shoppers.

4.2.1.2.4 Data Analysis

Data Analysis of Cognitive Walkthrough

The data collected from the Cognitive Walkthrough tasks are in 2 parts, and the measuring criteria are shown in Table 9.

1. The task completion status, according to the response of the 3 Cognitive Walkthrough questions mentioned in Section 2.1.
2. Time taken for each task.

Table 9 - Measuring Criteria of Cognitive Walkthrough

Metric Measured	Measuring Criteria (Success)	Measuring Criteria (Failure)
Task Completion Tasks are either assigned "Success" or "Failure" for each user based on whether the user can complete the task.	All the responses towards the 3 Cognitive Walkthrough questions are "Yes".	Any of the responses towards the 3 Cognitive Walkthrough questions is "No".
Time Taken	≤10s	>10s

Table 10 shows an example of data collected from the Cognitive Walkthrough, manipulated according to the measuring metrics. The example table shows the result of three users doing 8 tasks in the Cognitive Walkthrough, to test the extension functionalities. Each task is assigned two values, representing 'Task Completion Status' and 'Time Taken'. For example, "S/5s" means the task is completed successfully and the time taken for the task to complete is 5 seconds.

Table 10 - Example of data collected from Cognitive Walkthrough

	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8

	Task Completion / Time Taken							
User 1	S/S	S/S	S/F	S/S	S/S	S/F	S/S	S/S
User 2	F/S	F/S	S/F	F/F	F/F	S/S	F/F	F/S
User 3	S/S	S/S	S/S	S/S	F/F	S/S	F/S	S/F
Task Score	66.7%	66.7%	100%	66.7%	33.3%	100%	33.3%	66.7%
Time Score	100%	100%	33.3%	66.7%	33.3%	66.7%	66.7%	33.3%

'S' = 'Success', 'F' = 'Failure'

As shown in the example table, 48 attempts (24 Task Completion Status attempts + 24 Time Taken attempts) in total are observed to perform the tasks. Among all the attempts, 32 were successful, and 16 were failed. All the success points contribute to the success rate and all the failure points contribute to the error rate.

According to this example, the success rate of Cognitive Walkthrough would be: 32/48 = 66.7%. The error rate of Cognitive Walkthrough would be: 1-66.7% = 33.3%.

Data Analysis of Questionnaire

The data collected from the responses of quantitative questions in the questionnaires, which were presented to users after the Cognitive Walkthrough, is analyzed according to 5 metrics, Readability, data relevance, accessibility, consistency, and Visual Weight. Each of the 9 quantitative questions is mapped with a certain metric as shown in Table 11. For each quantitative question in the questionnaire, the Likert scale with a 5-point agreement scale is applied.

Table 11 - Measuring Criteria of Questionnaire (Quantitative Questions)

Metric Measured	Question Index	Measuring Criteria (Success)	Measuring Criteria (Failure)
		Calculated based on the average score of the related questions	

Readability	Q1	≥4	<4
Data Relevance	Q6 and Q7	≥4	<4
Accessibility	Q2	≥4	<4
Consistency	Q3, Q4, and Q8	≥4	<4
Visual Weight	Q5 and Q9	≥4	<4

Table 12 shows an example of data collected from the questionnaire, each metric parameter is assigned 2 values, one is the Success Status, and the other is the average score of the parameter. For example, "F (2)" for 'Data Relevance' for 'User 2' means the average score of question 6 and question 7 is 2, and this is defined as a 'Failure' according to the measuring criteria in Table 11.

Table 12 - Example of data collected from Questionnaire (Quantitative Questions)

	Readability (Q1)	Data Relevance (Q6+Q7)	Accessibility (Q2)	Consistency (Q3+Q4+Q8)	Visual Weight (Q5+Q9)
User 1	F(2)	F(1)	F(3)	F(2)	F(1)
User 2	F(1)	F(2)	S(4)	F(3)	F(3)
User 3	F(4)	F(1)	F(3)	F(1)	S(5)
Average Score	2.3	1.3	3.3	2	3
Success Score	33.3%	0%	100%	33.3%	66.7%

'S' = 'Success', 'F' = 'Failure'

As shown in the example table, 15 attempts in total are organised according to the measuring criteria. Among all the attempts, 2 were successful, and 13 were failed. All the success points contribute to the success rate and all the failure points contribute to the error rate.

According to this example, the success rate of the Questionnaire would be: 2/15 = 13.3%. The error rate of Cognitive Walkthrough would be: 1 - 13.3% = 86.7%.

The final success rate is the average of the success rate of Cognitive Walkthrough and the success rate of Questionnaire, which would be (66.7% + 13.3%) / 2 = 40%, and the final error rate is 1 - success rate = 60%.

Data Analysis of Think Aloud

Data included in the Think Aloud are:

1. Qualitative results from the questionnaire
2. Issues the users meet during the Cognitive Walkthrough
3. Feedback and suggestion users gave during and after the Cognitive Walkthrough

4.2.1.2.5 Practical Setup

1. The first step of the experiment is to find users that represent our user personas created early in the project. Having a good selection of different types of users would give us the most helpful feedback.

2. The users doing the experiment could either use a group member's computer or install it onto the personal computer, with the help of the group member carrying out the experiment.

3. Each experiment will then be carried out online, on one or more types of shopping websites, such as Amazon.com or Outfithustlers.com. The experiment can be carried out at any location, provided there is internet access.

4. Each user will be given a list of tasks that they are too complete using the extension. No other help will be given unless they request it. If help is requested, the task they are on and the help they asked for will be recorded by the tester. The user will work through each of the tasks provided and the tester will record how long it took them to complete each one, their reaction towards it, any obstacles they faced, and any comments they made along the way.

5. Once the user completes each of the tasks, they will then be asked to fill out an online questionnaire about the experience they just had using the extension. This questionnaire is designed to give us both quantitative and qualitative data, from the user's feedback.

4.2.2 Usability Evaluation for High-Fidelity Version 1 and 2（Within the group）

In User evaluation of Version 1 and Version 2, the main evaluation methods were used as Think Aloud. Meanwhile, four user tasks were designed for the basic functions of this plug-in. Team members are the primary participants, acting as users to finish User tasks to evaluate existing high-fidelity prototypes.

4.2.2.1 High-fidelity prototype of Version 1

There are three main user interfaces in this release. The first interface is the main panel, the upper part of which is the Result Area, which contains the hide and show of the Dark Pattern list. There is a toggle button at the right end of the Result Area to switch to Pie Chart. The second part is the selection menu, including Dark Pattern Detection list, report, An Extension Update button, link to the website. The second interface is the toolbar, which contains Settings, Report, Hide and show function, expand panel, and an exit button. The third is a single Popup of the Report interface. All of these are shown in figure 56.

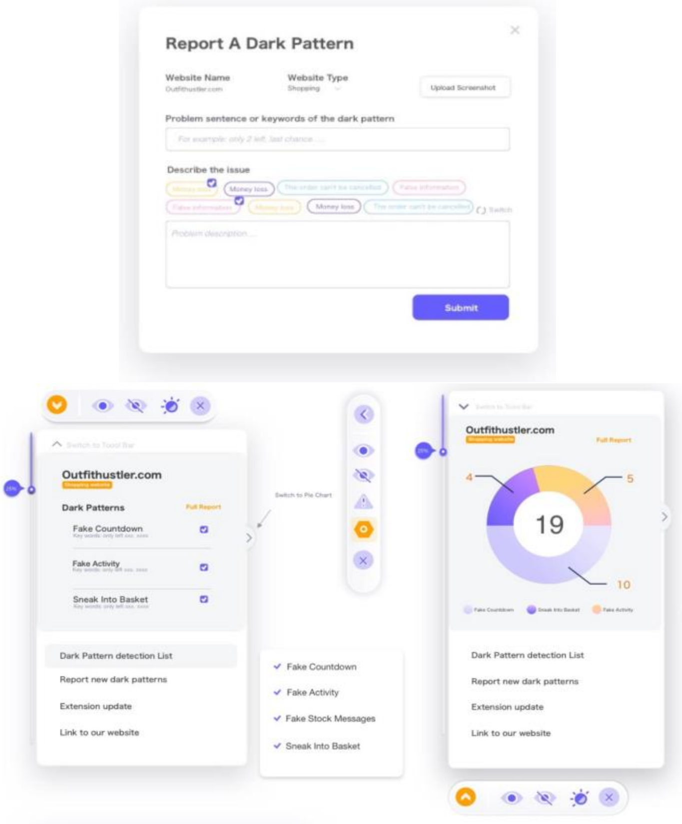


Figure 56 - High-fidelity prototype of version 1

4.2.2.1.1 Think Aloud

The user task of version 1 of the high-fidelity prototype requires the user to perform four tasks, including:

At the same time, users were asked to "Think Aloud" while performing the task. The data in the following table contains the problems that users encounter while performing user tasks.

1. How would you find more information on Dark Patterns?
2. How would you detect Dark patterns on a webpage?
3. How would you report a dark pattern on a webpage?
4. How do you turn on/off the detection of certain types of patterns?

Table 13 - Think aloud of version 1

	Issues
Functionality	1.No detection button 2. Seems like Sidebar is a bit redundant; many of the functions are repetitive. 3. There is no need for the light and dark mode 4. Full report link is not needed 5. There is no need to tell people this is the shopping website 6. report page needs to add in the extension
Layout	1. The switch buttons of the result page are unclear 2. Link to our website needs to change to other places
Content	1. The Extension Update option is not required.

4.2.2.2 High-fidelity prototype of Version 2

There are two user interfaces in version2. The first interface is the Checking Page, which contains the Detection button, Detection List, and Result Area. The Result Area is still divided into two interfaces. One is a pie chart for intuitive analysis, and the other is a "Hide and Show function of dark patterns." Another interface is the "Report Page".

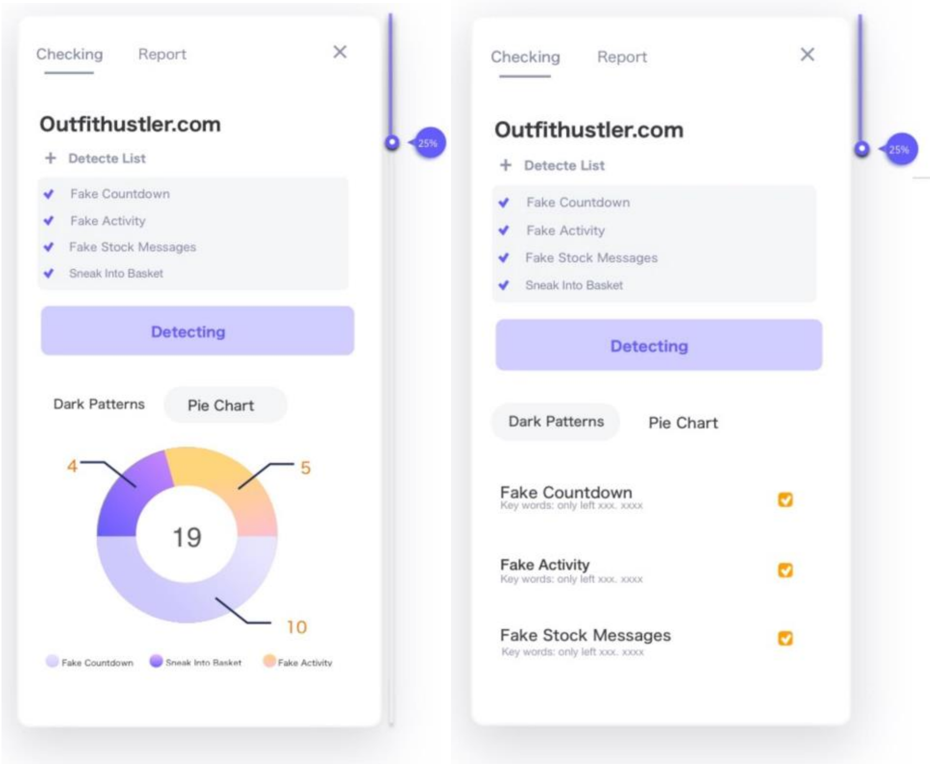


Figure 57 - High-fidelity prototype of version 2

4.2.2.2.1 Think Aloud

The high-fidelity prototype of the version 2 test method is similar to version 1, requiring users to perform four tasks, including:

1. How would you find more information on Dark Patterns?
2. How would you detect Dark patterns on a webpage?
3. How would you report a dark pattern on a webpage?
4. How do you turn on/off the detection of certain types of patterns?

At the same time, users were asked to "Think Aloud" while performing the task. The data in the following table contains the problems that users encounter while performing user tasks.

Table 14 - Think aloud of version 2

	Issues
Functionality	1. Needs a Setting page 2. Page percentages are not needed at the moment, it is better not to add them next to it. 3. needs some highlight icons to highlight dark patterns on the webpage.

Layout	1. The results area is too small. 2. There is no need to paginate the results display on page 3. The color of the extension is so light, it needs to be changed.
Content	1. The display area of the Result page and the Detection List need to be selected. They serve the same purpose

4.2.3 Usability Evaluation for Version 3

In version 3, three evaluation methods were designed and used, which were Think Aloud, cognitive Walkthrough and Questionnaire. The detailed user tasks are formulated for each extension function at this stage. Meanwhile, test steps are also designed to make the process more standardized. In addition, various forms and questionnaires for recording user data have been designed to record user reactions and collect user feedback. The task success and failure rates are evaluated in the results processing phase. At last, ABTest is used to test the feasibility of the improvement scheme.

4.2.3.1 High-fidelity prototype of Version 3

There are three main user interfaces in version 3. The first one is the Result page. Including pie charts and Detection Filter, the second interface is Report Page. The third interface is Settings. This includes the Auto Detection option, page highlighting area colour selection and dashboard mode selection.

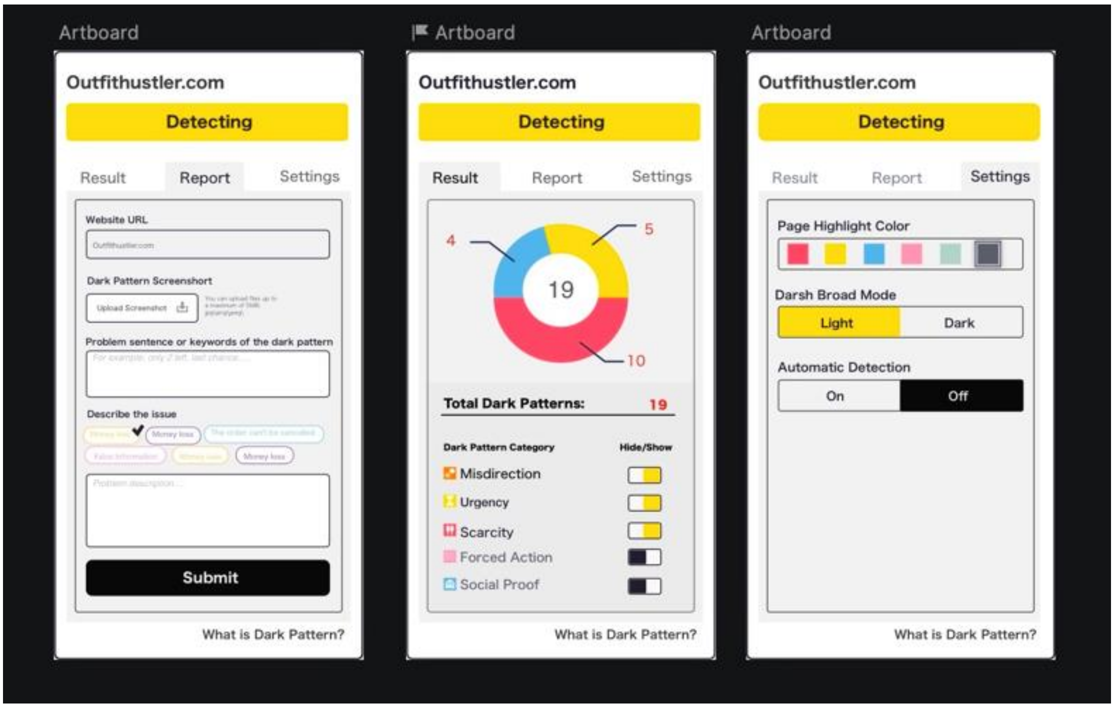


Figure 58 - High-fidelity prototype of version 3

Figure 59 - Highlighting icon of version4

4.2.3.2 Cognitive Walkthrough

For the Cognitive Walkthrough for extension version 3, 6 tasks were conducted by users, mostly the same as the 8 tasks in Version 4, only removing 2 tasks that were developed with Version 1. The tasks are shown as below:

1. How would you find more information on Dark Patterns?
2. How would you detect Dark patterns on a webpage?
3. How would you report a dark pattern on a webpage?
4. How do you turn on/off the detection of certain types of patterns?
5. How would you locate highlighted patterns on the webpage?
6. Can you find what each icon represents?

Table 15 - Performance results obtained through Cognitive Walkthrough

		Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
	Task Completion / Time Taken						
User1		F/F	S/S	F/F	S/S	F/F	S/S

User2		S/S	S/S	S/F	F/F	F/F	S/S
User 3		F/F	S/S	F/F	S/S	F/F	S/S
User 4		S/S	S/S	F/F	S/S	F/F	F/F
User 5		F/S	S/S	S/S	F/F	F/F	S/F
User 6		F/S	S/S	F/F	S/S	F/F	F/F
User 7		F/S	S/S	F/F	S/S	F/F	S/S
User 8		S/S	S/S	S/S	S/S	F/F	S/F
Task Score		37.5%	100%	37.5%	75%	0%	75%
Time Score		75%	100%	25%	75%	0%	50%

From the result of the tasks, we could see at least 3 functionalities need to be improved, the issues are as below:

1. "Introduction" of Dark Patterns on the extension is not enough, users found it hard to see the link to the website. (Task 1)
2. The "Report" function on the extension, firstly the name itself is misleading to the users, should change it into something like "Report a Dark Pattern", secondly users found it difficult to understand what should be filled in in each field. (Task 3)
3. Most of the users can't locate the highlighted patterns on the webpage at all. (Task 5)

Performance results: 96 responses in total (48 Task Completion Status responses + 48 Time Taken responses) are organized according to the measuring criteria. Among all the responses, 52 were successful, and 44 were failed. All the success points contribute to the success rate and all the failure points contribute to the error rate.

The success rate of Cognitive Walkthrough for Version 4 would be: 52/96 = 54.2%. The error rate of Cognitive Walkthrough would be: 1- 54.2% = 45.8%

For the questionnaire, 9 quantitative questions are presented to users, and 5 metric parameters are used for result evaluation, the results are shown in Table 16.

Table 16 - Results obtained from questionnaire (quantitative)

	Readability (Q1)	Data Relevance (Q6+Q7)	Accessibility (Q2)	Consistency (Q3+Q4+Q8)	Visual Weight (Q5+Q9)
User 1	S(4)	F(3)	S(4)	F(3.3)	F(3.5)
User 2	F(3)	S(4)	S(4)	S(4.3)	S(4)
User 3	S(4)	F(2.5)	F(3)	S(4)	S(4)
User 4	F(2)	F(2.5)	F(3)	S(4.6)	F(3)
User 5	F(2)	F(2)	S(4)	S(4)	F(3.5)
User 6	S(4)	F(2.5)	S(4)	F(3.3)	S(4)
User 7	F(3)	F(2)	S(5)	F(3.3)	F(3.5)
User 8	F(3)	F(1.5)	S(5)	F(3.3)	F(3)
Average Score	3.1	2.5	4	3.8	3.6
Success Score	37.5%	12.5%	75%	50%	37.5%

From the result of the questionnaire, most of the designs need to be adjusted to improve the usability, especially data relevance:

1. Data Relevance needs to be improved the most, what we presented on the extension UI doesn't really match what users expected.
2. The link to the website for more information and the text content in the extension need to be adjusted to improve readability.
3. The overall visual design is not eye-catching enough to attract users.

Questionnaire results: 40 responses in total are organized according to the measuring criteria. Among all the responses, 17 were successful, and 23 were failed. All the success points contribute to the success rate and all the failure points contribute to the error rate.

The success rate of Cognitive Walkthrough for Version 4 would be: 17/40 = 42.5%. The error rate of Cognitive Walkthrough would be: 1- 42.5% = 57.5%.

The final success rate is the average of the success rate of Cognitive Walkthrough and the success rate of Questionnaire, which would be (54.2% + 42.5%) / 2 = 48.4%, and the final error rate is 1 - success rate = 51.6%.

4.2.3.3 Think Aloud

Data (in Table 17) included in the Think Aloud are:

1. Qualitative results from the questionnaire
2. Issues the users meet during the Cognitive Walkthrough
3. Feedback and suggestion users gave during and after the Cognitive Walkthrough

Table 17 - Results obtained from Think Aloud

	Issues
Functionality	Needs some prompt beside the pie chart for the explanation, not hover effect
	Hard to connect pie chart with the detected result
	What is this extension used for? Add the introduction about dark pattern
	Add a function that makes the detection to be automatic
Layout	Highlighting icon colour is too light to observe on the webpage.
	Hard to connect Highlight icons with the information's on the side window together
	Put a number to the pie chart for the explanation
	Pie chart is not important, don care about it
Content	Question 3 and Question 4 on the report page is not clear

	“What is the Dark Pattern” looks not clear and not clickable
	The category title is too small make it bigger
	What do those small icons mean? User doesn’t understand
	Confusing about the Icon colour on pie chart
	Not sure what does the highlighting area do
	“Number of total Dark Pattern” on the result page needs to be bigger

4.2.3.4 Results of AB Test

The AB test (Survey content is attached in Appendix) is conducted by 11 participants and the results are organized from the total of 11 participants. Among the responses, the first 6 responses (Group 1) are collected from the users who haven’t used our extension at all, and the last 5 responses (Group 2) are collected from the users who have used our extension and went through the Cognitive Walkthrough with us.

Among all the responses, 16.7% of the Group 1 (who haven’t tested our extension) and 100% of the Group 2 (who have tested our extension) preferred the modified version of the Introduction page (Version 2), which is adding an “About” page to the interface. In total, 54.5% of participants preferred the modified version.

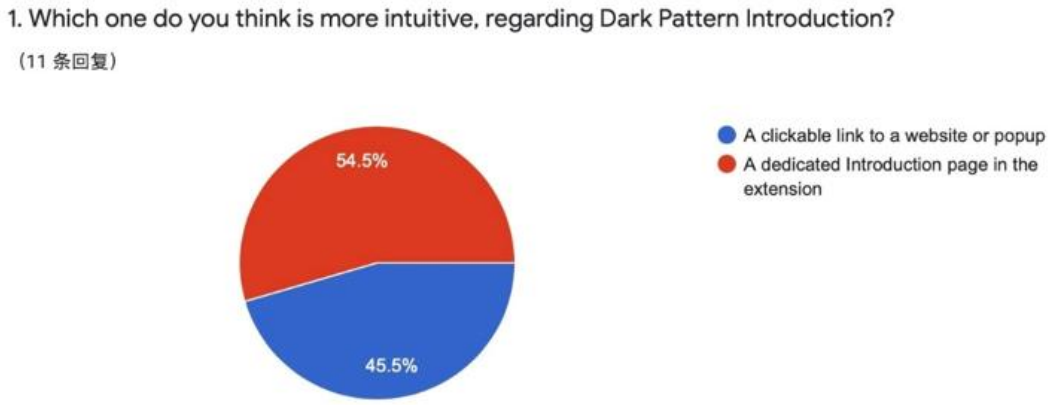


Figure 60 - Statistic Results of question1

Among all the responses, none of the users in Group 1 (who haven’t tested our extension) and 100% of the Group 2 (who have tested our extension) think the modified version of UI (Version 2) provides more clear and detailed information about the detected Dark Patterns. In total, 45.5% of responses preferred the modified version.

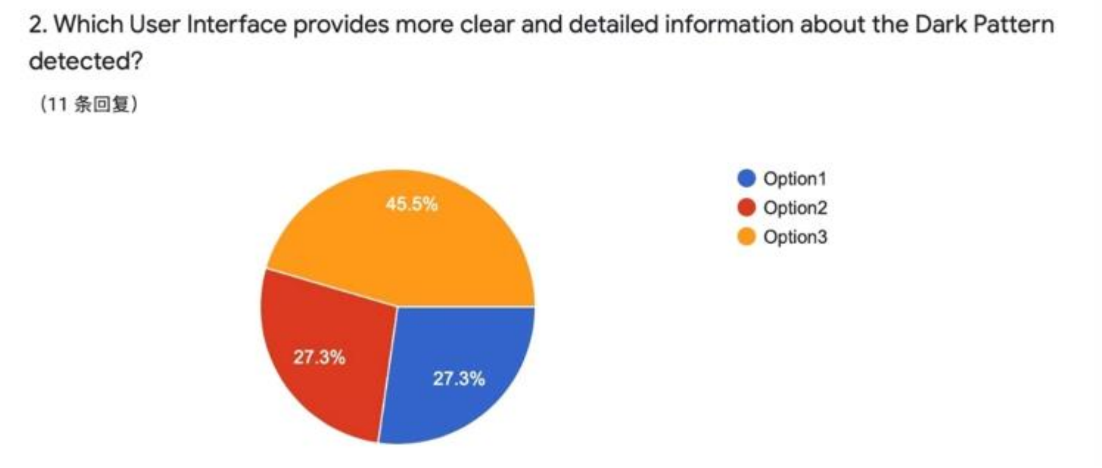


Figure 61 - Statistic Results of question2

Among all the responses, 33.3% of the Group 1 (who haven't tested our extension) and 100% of the Group 2 (who have tested our extension) preferred the modified version of the Report page (Version 2). In total, 60% of participants preferred the modified version.

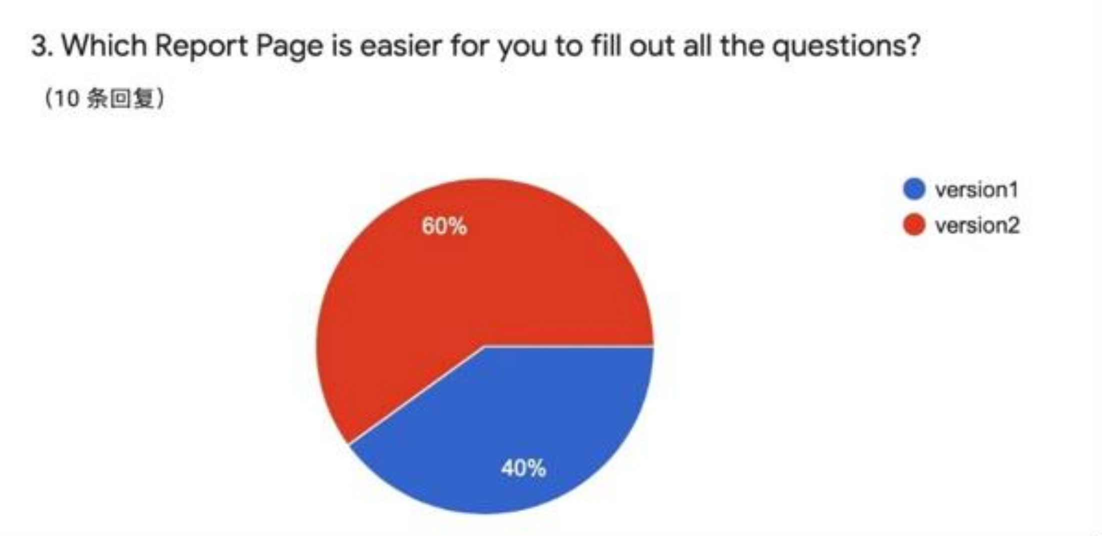


Figure 62 - Statistic Results of question3

Among all the responses, 50% of the Group 1 (who haven't tested our extension) and 100% of the Group 2 (who have tested our extension) preferred the modified version (Version 2), which is to put the filter on the settings page. In total, 72.7% of participants preferred the modified version.

4. Which page do you think is more reasonable for the settings of detection filtering?

（11 条回复）

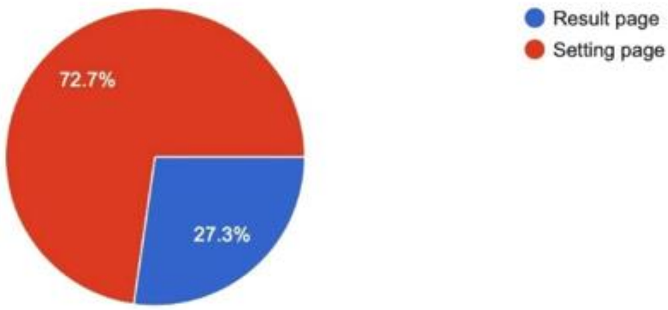


Figure 63 - Statistic Results of question4

Among all the responses, 66.7% of the Group 1 (who haven't tested our extension) and 100% of the Group 2 (who have tested our extension) preferred the modified version of the highlighting icons (Version 2). In total,81.8% of participants preferred the modified version.

5. Which Highlight style is more eye-catching?

（11 条回复）

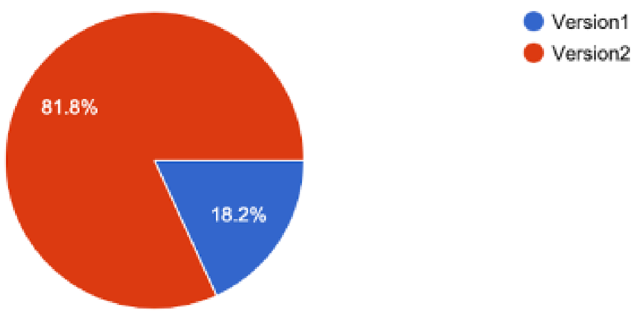


Figure 64 - Statistic Results of question5

Among all the responses, 66.7% of the Group 1 (who haven't tested our extension) and 100% of the Group 2 (who have tested our extension) thought the revised version was better. Overall, 81.8% thought we should build stronger links between panels and Highlighting areas.

6. Which version of UI provides a closer connection between the results on the UI and the highlighting on the web page?

（11 条回复）

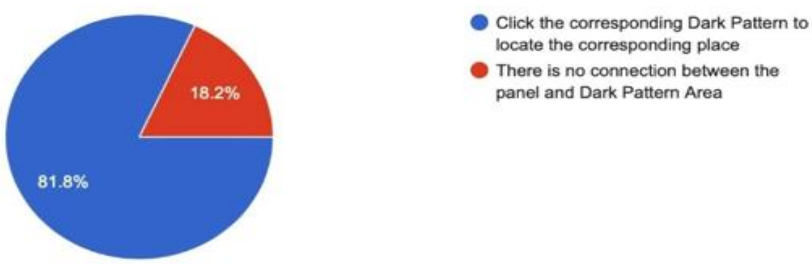


Figure 65 - Statistic Results of question6

4.2.4 Usability Evaluation for Version 4

In version 4, some improvements were made to the testing methodology based on version 3. In this phase, there are five test methods. They are Think Aloud, Cognitive Walkthrough, questionnaire, Heuristic and AB tests. Think Aloud and Cognitive Walkthroughs are conducted similarly as before. In the questionnaire, the questions were continued to be refined into more detailed quantitative and qualitative questions. The expert user reviews were also conducted and used the results to guide improvement. In addition, the AB test includes the interface's new and old designs, new and old logos, and Highlighting Icon's contrast and colour selection. The result processing is basically similar to version 3.

4.2.4.1 User Interface of Version 4

There are four user interfaces in this version of the interface, and one is About Page, which provides a brief introduction to the plugin and Dark Pattern. The second page is the Result Page, which contains three fields. Dark pattern monitoring area, checkboxes, radio button detection results, and current page position. The last page is the Settings Page, containing Detection Filter and Auto Scan options.

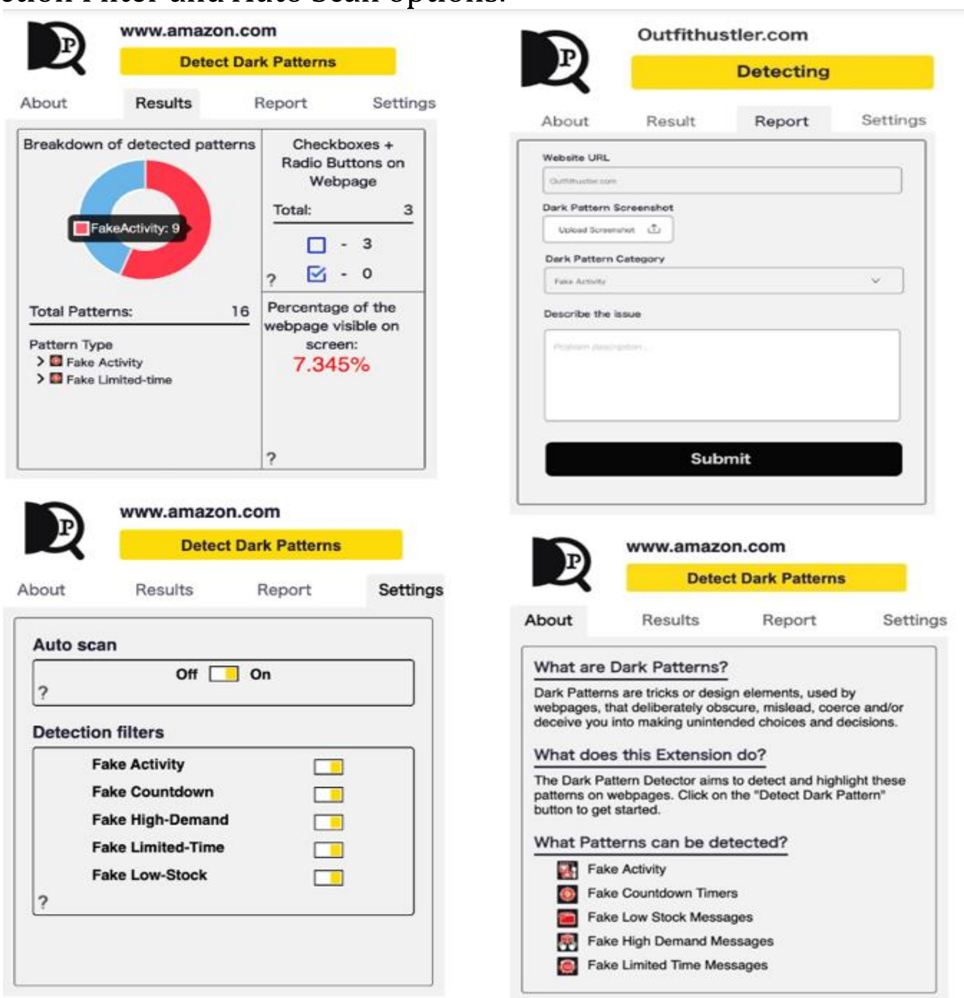


Figure 66 - UI of version 4

Figure 67 -Highlighting icon of version 4

4.2.4.2 Cognitive Walkthrough

For the Cognitive Walkthrough for extension version 4, 8 tasks were conducted by users. The tasks are shown as below:

1. How would you find more information on Dark Patterns?
2. How would you detect Dark patterns on a webpage?
3. How would you report a dark pattern on a webpage?
4. How do you turn on/off Autoscan?
5. How do you turn on/off the detection of certain types of patterns?
6. How would you locate highlighted patterns on webpages?
7. Can you find what each icon represents?
8. Where do you find information about checkboxes and page location, in the extension?

Table 18 - Performance results obtained through Cognitive Walkthrough

	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8
	Task Completion / Time Taken							
User 1	F/S	S/S	S/S	S/S	F/F	F/S	F/F	S/F
User 2	F/S	S/S	S/S	S/S	F/S	F/F	F/S	S/S
User 3	F/S	S/S	F/F	S/S	S/S	F/F	S/S	S/S
User 4	F/S	S/S	F/F	S/S	S/S	S/S	S/S	S/S
User 5	F/S	S/S	F/F	S/S	S/S	S/S	S/S	S/S
Task Score	0%	100 %	40%	100 %	60%	40%	60%	100 %
Time Score	100 %	100 %	40%	100 %	80%	60%	80%	80%

From the result of the tasks, we could see at least 3 functionalities need to be improved, the issues are as below:

1. Most of the users don’t have the patience to read the text description of dark patterns on the ‘About’ page of the extension. (Task 1)
2. The “Report” function on the extension, again, users found it difficult to understand what should be filled in in each field as in Version 1, and it took users quite long to fill all the fields and submit the report. (Task 3)
3. Users don’t know how to locate the highlighted dark patterns according to the results on the extension ‘Result’ page. (Task 6)

Performance results: 80 responses in total (40 Task Completion Status responses + 40 Time Taken responses) are organized according to the measuring criteria. Among all the responses, 57 were successful, and 23 were failed. All the success points contribute to the success rate and all the failure points contribute to the error rate.

The success rate of Cognitive Walkthrough for Version 4 would be: 57/80 = 71.3%. The error rate of Cognitive Walkthrough would be: 1- 71.3% = 28.7%.

4.2.4.3 Questionnaire (Quantitative)

For the questionnaire, 9 quantitative questions are presented to users, and 5 metric parameters are used for result evaluation, the results are shown in Table 19.

Table 19 - Results obtained from questionnaire (quantitative)

	Readability	Data Relevance	Accessibility	Consistency	Visual Weight
User 1	S(4)	F(3)	S(4)	F(3.7)	F(2)
User 2	S(5)	S(4.5)	S(4)	S(4)	S(4)
User 3	S(5)	S(5)	S(4)	F(3.3)	S(5)
User 4	S(5)	S(4.5)	S(5)	S(5)	F(3.5)
User 5	S(4)	F(3)	S(4)	F(3.3)	F(3.5)

Average Score	4.6	4	4.2	3.9	3.6
Success Score	100%	60%	100%	40%	40%

From the result of the questionnaire, some of the designs still need to be adjusted to improve the usability:

1. Consistency needs to be improved, for example, when turning on the auto-scan, how to tell the user that our extension is still working.
2. In terms of Visual Weight, the pie chart was not very attractive for the users, however, it occupies more than ¼ of the result page.

Questionnaire results: 25 responses in total are organized according to the measuring criteria. Among all the responses, 17 were successful, and 8 were failed. All the success points contribute to the success rate and all the failure points contribute to the error rate.

The success rate of Cognitive Walkthrough for Version 2 would be: 17/25 = 68%. The error rate of Cognitive Walkthrough would be: 1- 68% = 32%.

The final success rate is the average of the success rate of Cognitive Walkthrough and the success rate of Questionnaire, which would be (71.3% + 68%) / 2 = 69.7%, and the final error rate is 1 - success rate = 30.3%.

4.2.4.4 Think Aloud

Data (in Table 20) included in the Think Aloud are:

1. Qualitative results from the questionnaire
2. Issues the users meet during the Cognitive Walkthrough
3. Feedback and suggestion users gave during and after the Cognitive Walkthrough

Table 20 - Results obtained from Think Aloud

	Issues
Functionality	Detection process took too long without any timer, the user thought it crashed, there should be a timer.
	User doesn’t know the result is clickable to be directed to the dark pattern location on the page.
	Don’t know what each dark pattern means in the “settings” page, suggest a hover effect on each of them with a picture pop-up for the explanation.

	Should have a “help/FAQ” function regarding how to use the extension.
	Perhaps a way to contact developers to report issues and or bugs.
	Can every webpage be detected automatically? Or just one website page
	Connect the Dark Pattern Introduction Website with the content on the “About” page.
Layout	The pie graph is not wanted by the user, which occupies the most space on the result page.
	Move the About page back or change it to the scroll down menu
	Logos of the dark patterns are too small and need to be clearer in the extension
	Layout of the reports page and the results page need to be better
	Bigger icons for dark pattern results, explanation it
	Change the icon colour or its style, the red and black colour icon makes the user anxious.
	Result should be on the first page, don’t want to see “About” every time when initiating the extension
	Though the ‘Report’ tab meant a report of the detection. i.e., ‘Results’
	Did not realize there was a drop-down menu under each detected type of pattern
	Move the link for the DP detector website further up the page it is on. Change 'report' to 'report dark pattern'.
	The fact that if there are no checkboxes on the page, that section of the "Results" tab is blank, and it doesn't say "No buttons are detected on this page".
	I like the size of the extension, but I would like if it had a control bar on the top with an “X” on the top right to

	close the window, and in the middle of the control bar, a piece of text saying “[A][A][A]” which could make the window (and text and diagrams) bigger, and a lot bigger.
	The layout on the reports page and results page needs to be better
Content	Bigger fonts needed in the extension (for detection results), or replaced with image/gif when hovering on the text
	The “?” should be beside each heading, not the left corner of the box
	No idea what “Detection Filter” means in the “settings” page
	Change 'report' to 'report dark pattern'
	Colour needs to be unified
	Don't know what the words below pattern type stand for, so better to mark them directly, or tell the customer the truth
	Report page questions use need to apply better explanations on what to fill out in these sections
	Why not just write about rational consumption? Instead of just using the icon
	Highlight area the user wants something similar to iPhone/Android warning window, a grey exclamation mark inside a yellow triangle sign.

4.2.4.5 Expert Users Review

Jakob Nielsen's ten interaction design principles are used in Heuristic Evaluation. During this period, two experts were evaluators and made evaluation suggestions from 10 aspects according to 10 principles. In the end, each problem is grouped according to Nielsen's Severity Rating.

Table 21 - Nielsen's Severity Rating

Severity Ratings	
Rating	Definition
0	NO PROBLEM
1	Cosmetic Problem
2	Minor Usability Problem
3	Major Usability Problem
4	Usability Catastrophe

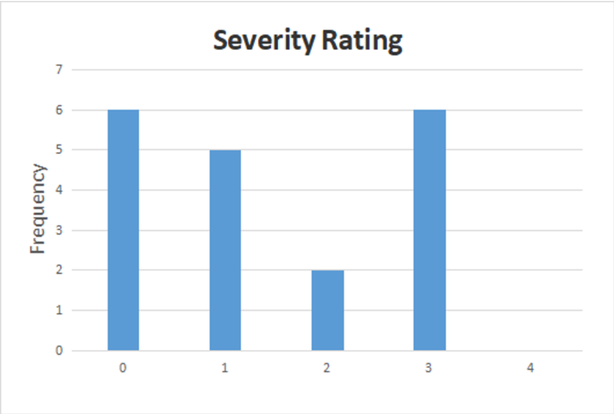


Figure 68 - Frequency Severity Rating in Dark Pattern Extension

As can be seen from the figure above, the most serious problems are at the severity level (3), with a total of six. They respectively violate Nielsen heuristics usability principles of "Visibility of system status", "Recognition rather than recall", "Flexibility and efficiency of use", "Help and documentation", these four principles. In addition, severity level (2) has two problems, which belong to the "Match between system and the real world (real user's habit)" and "User control and freedom". There are also five problems which are Cosmetic Problems.

Table 22 - What each problem is at each severity level.

Severity Ratings	Nielsen heuristics	Problems
3	Visibility of system status	Once auto-detect is on, patterns are being detected even when going on to new

		pages? There is nothing indicating this.
		Difficulty to see what the icons are.
		Difficult to see where patterns are actually on the page
	Flexibility and efficiency of use	The idea of being able to see exactly where dark patterns are is a good one but not accurate at the moment.
	Recognition rather than recall	Provides me with no more information on dark patterns. Need to connect to the information website.
	Help and documentation	Website needs to be incorporated to educate the user.
2	Match between system and the real world (real user's habit)	Slider buttons are good. Should have labels on them.
	User control and freedom.	The ">" symbol at the start of each pattern isn't obviously a menu option to see more detail, maybe change to [>] ??
1	Consistency and standards	Is the website consistent with the add-on?
	Match between system and the real world (real user's habit)	Extension defaults to the "About" tab, it gets boring after a while.
	Flexibility and	I'd like if I could move

	efficiency of use	around elements on the “Results” tab, or replace the Pie Chart with a Bar Chart
	Aesthetic and minimalist design	Lots of blank space in the “Report” tab
	Help and documentation	Good “About” tab, but I would like a 50-second video showing the key features of the extension

4.2.4.6 Logo Colour Testing

To give the user a better understanding of the meaning of each icon, the introduction of each black pattern was added in the question and all options are set to color ICONS. There was a total of 60 users taking the test. Select the first three tall colors for statistics.

Basically, except for the highlighting icon of Fake Activity and Fake Low Stock, the final color for other highlighting icons is the color with the highest percentage. Many people thought that Fake Activity was best for red followed by pink, and as red Fake Countdown accounted for a much larger percentage of 47.37%. For Fake Activity, pink was chosen as the highlighting icon color with a percentage of 21.05%. In addition, for Fake LowStock, yellow and orange were both heavily represented at 22.81 percent and 19.3 percent, respectively. But some people think orange and yellow look the same. Orange was picked as the color for the highlighting icon color because it has a clearer pattern than yellow.

Table 23 - Logo Color Testing Result.

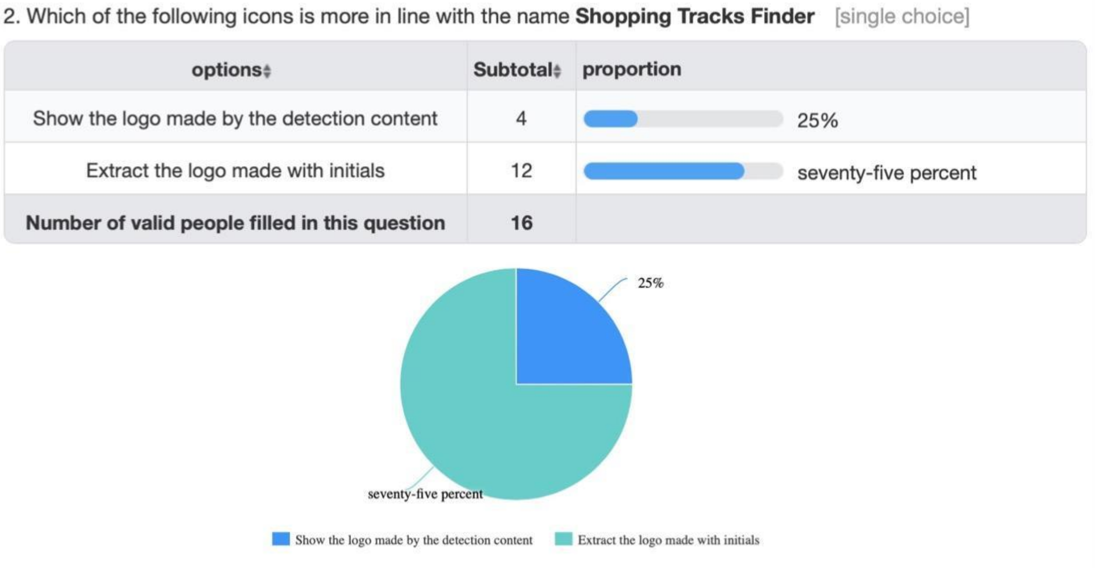
Fake Countdown	Fake Activity	Fake Low Stock	Fake High Demand	Fake Limited-Time	Confirm Shaming
Red 47.37%	Red 22.81%	Yellow 22.81%	Green 21.05%	Purple 21.05%	Blue 29.82%
Orange 22.81%	Pink 21.05%	Orange 19.3%	Blue 17.54%	Pink 21.05%	Purple 17.54%
Yellow 12.28%	Orange 19.3%	Green 15.79%	Orange/Pink 12.28%	Green 19.3%	Green 15.79%

Red	Pink	Yellow	Green	Purple	Blue

4.2.4.7 Result of AB Test

This AB Text contains a comparison of the versions 4 and 5. Including ICONS, user interface, and plug-in work prompt. Meanwhile, the two versions of the logo were compared.

Of all the responses, 31.25% of the testers felt that version 4 (black and red ICONS) was more obvious and better suited to Highlight Icon than version 5. On the contrary, 68.75% of people think that the improved version 5 is better suited as a Highlight Icon. Overall, the improved version 5 has more support.



2. Which of the following icons is more in line with the name **Shopping Tracks Finder** [single choice]

options	Subtotal	proportion
Show the logo made by the detection content	4	25%
Extract the logo made with initials	12	seventy-five percent
Number of valid people filled in this question	**16**	

Figure 69 - Version 4-AB Test-logo choosing

Of all the responses, 25% of the testers thought the version 4 logo was more suitable for Shopping Tracks Finder and 75% thought STF was more suitable for the logo. So, a new version of the icon can be adopted.

3. Which interface looks clearer?
[Single Choice]

options	Subtotal	proportion
A	6	37.5%
B	10	62.5%
Number of valid people filled in this question	16	

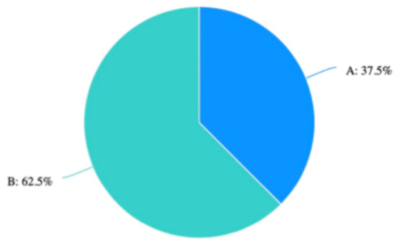


Figure 70 -Version 4-AB Test-Interface comparison

Of all the responses, 62.5% said the improved version 5 user interface is clearer. Overall, the improved version 5 provides more support.

4. Which prompt box is clearer? [Single choice questions]

options	Subtotal	proportion
1	1	6.25%
2	15	93.75%
Number of valid people filled in this question	16	

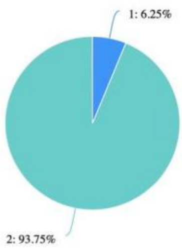


Figure 71 - Version 4-AB Test-prompt box comparison

Of all the responses, 93.75% said the improved prompt box of auto-detection is better than the previous version.

In summary, the results show that all the improved versions are more suitable for the needs of users than the previous version.

4.2.5 Usability Evaluation for Version 5

This version of User Evaluation uses four approaches. Think Aloud, Cognitive Walkthrough, questionnaires, and AB tests. The testing process and content is similar to the version 4.

4.2.5.1 User Interface of Version 5

Version 5 had four main user interfaces. The content of the Result page is the same as Version 4. Image Detection is added into the second Setting page, and the third user interface is FAQ, including common questions, an introduction video and links to our website. The fourth user interface is the Help Us interface which is the same as the Report page of version 4.

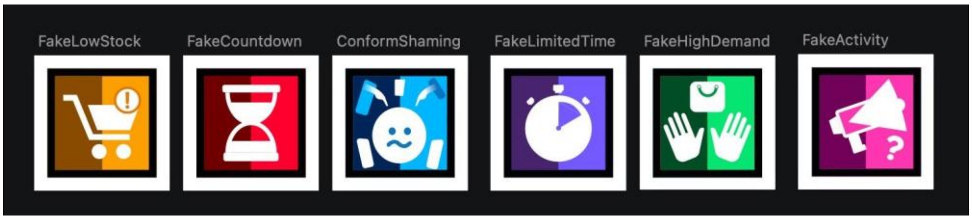


Figure 72 - Highlighting icon of version 5

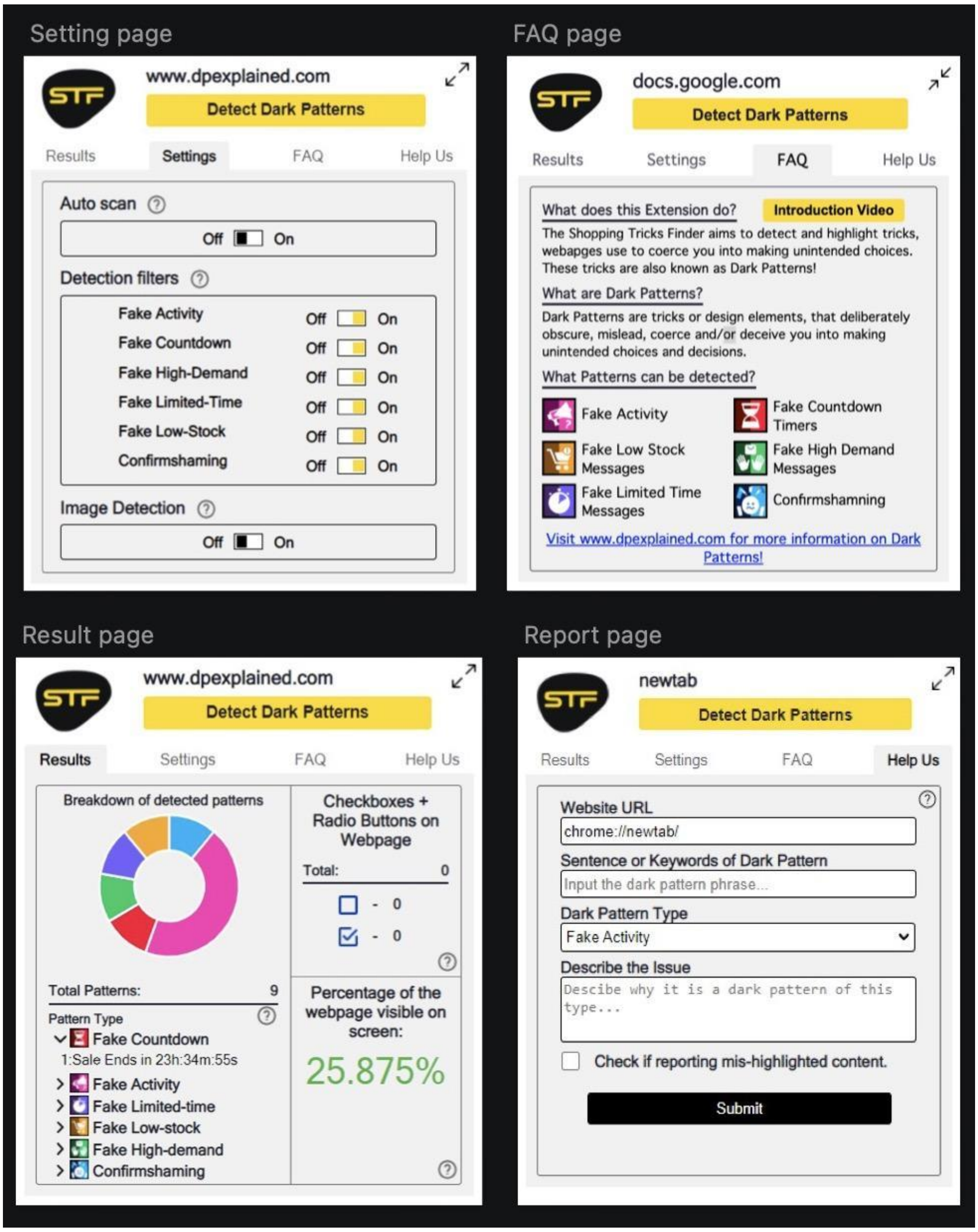


Figure 73 - User Interface of version 5

4.2.5.2 Cognitive Walkthrough

For the Cognitive Walkthrough for extension version 3, 8 tasks were conducted by users. The tasks are shown as below:

1. How would you find more information on Dark Patterns?
2. How would you detect Dark patterns on a webpage?
3. How would you report a dark pattern on a webpage?
4. How do you turn on/off Autoscan?
5. How do you turn on/off the detection of certain types of patterns?
6. How would you locate highlighted patterns on webpages?
7. Can you find what each icon represents?
8. How would you do image detecting? (Image detection = OCR)

Table 24 - Performance results obtained through Cognitive Walkthrough

	Task1	Task2	Task3	Task4	Task5	Task6	Task7	Task8
	Task Completion / Time Taken							
User 1	S/S	S/S	S/S	S/S	S/S	S/S	S/S	S/S
User 2	S/S	F/F	S/F	S/S	S/S	S/F	F/F	S/S
User 3	S/S	S/S	S/S	S/S	S/S	S/S	S/S	S/S
User 4	S/S	S/S	S/S	S/S	S/S	S/S	S/S	S/S
Task Score	100%	75%	100%	100%	100%	100%	75%	100%
Time Score	100%	75%	75%	100%	100%	75%	75%	100%

According to the task results, users' responses are almost positive, but there are also small issues are as below:
1. Some users who are new to this extension have some hesitation in recognizing the detection button. (Task 2)
2. When locating Dark Pattern, some participants seem to have not found the function that can automatically locate the Pattern. (Task 7)
3. The user shows that the above problems are solved by referring to the instructions of the extension.
Performance results: A total of 64 responses (32 task completion status responses + 32-time occupancy responses) were organized according to

measurement criteria. Of all the responses, 58 were successful and 6 were unsuccessful. All points of success affect the success rate, and all points of failure affect the error rate.
For version 2, the success rate of cognitive drills was 58/64 = 90.6%. The error rate of the cognitive exercise was 1-71.3% = 9.3%.

4.2.5.3 Questionnaire (Quantitative)

For the questionnaire, 9 quantitative questions are presented to users, and 5 metric parameters are used for result evaluation, the results are shown in Table 25.

Table 25 - Results obtained from Questionnaire

	Readability	Data Relevance	Accessibility	Consistency	Visual Weight
User 1	S(5)	S(5)	S(5)	S(5)	S(4.5)
User 2	F(3)	S(4)	S(4)	S(4)	S(4)
User 3	S(5)	S(4.5)	S(4)	S(4.6)	S(5)
User 4	S(4)	S(5)	F(3)	S(4.6)	S(4.5)
Average Score	4.25	4.625	4	4.55	4.5
Success Score	75%	100%	75%	100%	100%

Based on the results of the survey, some designs need to be tweaked to improve usability. For example, improve the contrast of page colors.

Questionnaire results: A total of 20 responses were organized according to measurement criteria. 18 cases were successful, and 2 cases failed. All points of success affect the success rate, and all points of failure affect the error rate.

Version 2 has a success rate of 18/20 = 90%. The error rate of cognitive drills is: 1-90% = 10%.

The final success rate is the average of the success rate of cognitive exercise and questionnaire, i.e. (90.6% + 90%)/ 2 = 90.3%, and the final error rate is 1 - success rate = 9.7%.

4.2.5.4 Think Aloud

Data (see Table 26) included in the Think Aloud are:

1. Qualitative results from the questionnaire
2. Issues the users meet during the Cognitive Walkthrough
3. Feedback and suggestion users gave during and after the Cognitive Walkthrough

Table 26 - Think aloud results.

	Issues
Functionality	
Layout	1. All good. The yellow button can be bigger. 2. Maybe the switch button can show one option (on or off) for the user at a time. 3. It would be perfect if the detection button were bigger
Content	1. I was a little bit confused about the pie chart at first. but then understand

4.3 Evaluation of checkbox scanner feature

Within the extension there is a function that looks for checkboxes and radio buttons on webpages. It gets how many are present on the page and how many of them are checked and unchecked. On sites such as Amazon, this feature works well and can accurately tell how many checkboxes/radio buttons there are and how many are checked/unchecked. However, on some other sites, this feature does not work as accurately as it would be hoped. This is because on some webpages checkboxes and radio buttons are not created in the traditional way using an ‘input’ tag, rather sometimes being clickable images and icons. Having these checkboxes as images severely impacts the accuracy of this feature, as it would be extremely difficult to identify an image that is a checkbox over an image that is just a regular image.

To test this feature, a random page on each of the websites seen in the ‘Website Domain’ column was chosen. The page chosen had to have either checkboxes or radio buttons present on the page. The number of checkboxes and/or radio buttons were then counted on each page before the extension was run on each of the pages. The results are shown in table 27.

Table 27 - Results of checkbox scanner testing across various webpages.

Website Domain	Number of checkboxes and radio buttons counted (Manually)	Number of checkboxes and radio buttons found by scanner	Accuracy	Reason (if applicable)
Littlewoods	62	0	0%	Every radio button was a pseudo-element (::before) of a <a> tag.
Amazon	23	23	100%	
Esty	22	22	100%	
Aliexpress	72	72	100%	
Argos	28	0	0%	All of the radio buttons were a pseudo-element (::after) of an <a> tag.
Pennys	20	20	100%	
Boots	8	8	100%	
River Island	74	74	100%	
Total	309	219	70.8%	

From evaluating the checkbox and radio button scanner on a variety of webpages, it produced an accuracy of nearly 71%. However, from viewing the results we can see that this is heavily influenced by sites where the scanner was not able to find any of the checkboxes at all, as the way they were developed made them impossible to distinguish as radio buttons. The success of the scanner is completely reliant on the way the buttons and checkboxes are coded in the HTML of the page and if coded in a way that makes them undetectable as checkboxes or radio buttons, it will fail completely.

4.4 Evaluation of extension detection speed

Table 28 - Results of Extension detection speed testing across various webpages.

Website Domain	Time (texture detection) /sec	Time (texture & image detection) /sec	Number of Images on the Webpage
Littlewoods	1.3	8	57
Amazon	1.6	23	170
Ebay	1.5	6	36
Aliexpress	1.5	27	180
Shein	2.2	16	60
ASOS	1.6	8	23
Boots	2.6	11	64
River Island	1.6	6	9
Super Dry	1.2	7	12
Boohoo	2.1	7	9

The speed test for the extension is shown above. The project picked 10 websites' homepage to test extension's speed with OCR and without OCR. According to the result of the test, time of texture detection is about 1.6 seconds. For the texture and images detection, the more images on the website, the more time it will cost to detect. And the size of images also will affect the speed of the extension, ASOS has 23 images in its website, but it almost costs the same time as in the River Island which has only 9 images.

4.5 Development Versions and Feature Implementations

Table 29 - Versions and features implemented in each version.

Version Number	Version Features	List of Implementations
Version 1	New Features 1. Toolbar design	1. A side window style UI + Toolbar, injected onto the

	2. Hide and show dark patterns on webpage function design 3. Dark Pattern Detection list filter design 4. Results displaying area design 5. Exit function design 6. Report function design 7. Automatically get the website name design 8. Website location bar design 9. Link to educational website design 10. Extension update design	page through a content script. 2. Created a node server that we knew would be needed in the future in order to connect all the services. It is used as a middle ground connecting the chrome extension to the machine learning server and handles and formats all requests and responses.
Evaluation method: Think aloud		
Version 2	New Features: 1. Panel paging design 2. Embedded Report page design 3. Detection button design Existing Features: 1. Hide and show dark patterns on webpage function 2. Dark Pattern Detection list filter 3. Results displaying area 4. Exit function 5. Report function	1. Implemented a Node Server that would hold the backend APIs. 2. Created a S3 bucket that would hold all the versions of our datasets. Functions implemented in the node server to modify this bucket. 3. Established a secure hosting for both the Node server and Python server on AWS ElasticBeanStalk.

<table>
<tr><td></td><td>6. Website location bar
7. Automatically get the website name</td><td></td></tr>
<tr><td colspan="3">Evaluation method: Think aloud</td></tr>
<tr><td rowspan="4">Version 3</td><td rowspan="4">New Features:
1. Embedded Setting page
2. Highlighting icon colour setting
3. Auto-detection function

Existing Features:
1. Dark Pattern Detection list filter
2. Report page
3. Detection button
4. Results displaying area
5. Report function
6. Automatically get the website name</td><td>Frontend Implementation</td></tr>
<tr><td>1. UI to be in the popup of the chrome extension.</td></tr>
<tr><td>Backend Implementation</td></tr>
<tr><td>1. Established two new databases on DynamoDB, one to store the dataset and the other to store the reports made by users.
2. Formatted tags that were returned from the Python service, to be easier to get location of the dark pattern.</td></tr>
<tr><td colspan="3">Evaluation method: Think Aloud + Cognitive Walkthrough + Questionnaire</td></tr>
<tr><td rowspan="4">Version 4</td><td rowspan="4">New Features:
1. Embedded About page
2. Highlighting icon colour setting
3. Checkbox and Radio button detection function
4. Page location percentage area
5. Detect pattern type</td><td>Frontend Implementation</td></tr>
<tr><td>1. FAQ section, checkbox scanner and percentage of page visible added.</td></tr>
<tr><td>Backend Implementation</td></tr>
<tr><td>1. Created an API function in the Node server that would allow users to create a report, the form is filled out on the website and chrome</td></tr>
</table>

	6. Instructions next to each function Existing Features: 1. Dark Pattern Detection list filter and Auto detection in Setting page 2. Detection button 3. Automatically get the website name 4. Results displaying area with dark pattern type list and details 5. Report function in Report page	extension. The form data is then saved in the AWS DynamoDB report database.
Evaluation method: Think Aloud + Cognitive Walkthrough + Questionnaire + Heuristic Test + AB Testing		
Version5	New Features: 1. Locating detected patterns on page 2. Image detection function 3. Change About page to FAQ 4. Change Report to Help Us 5. Panel enlargement and shrinkage function 6. Adding Confirmshaming dark pattern detecting Existing Features: 1. Dark Pattern Detection list filter and Auto detection in Setting page 2. Detection button	Frontend Implementation 1. UI size increased so more info and features can be displayed. Backend Implementation 1. Created an even more efficient way to get the tags of the HTML of the detected dark patterns, this now uses the tags and parent tags of the dark pattern. 2. Implemented the OCR function by adding the functionality into the Python services and added an API function in the Node server to be able to call the OCR function. When called, all images on the page get passed as a parameter to the OCR function. .

<table>
<tr><td></td><td>3. Automatically get the website name
4. Results displaying area with dark pattern type list and details
5. Checkbox detection result area
6. Report function on report page.
7. Instructions next to each function</td><td></td></tr>
<tr><td colspan="3">Evaluation method: Think Aloud + Cognitive Walkthrough+ Questionnaire + AB Testing.</td></tr>
</table>

5. CRISP-DM

During the project, there were multiple versions of models developed to find the best models to deploy. In the end, 3 different models were deployed for the Shopping Tricks Finder Chrome Extension to use, they are shown as the list below:

1. **The 5 Dark Pattern Types Detection** model (Fake Countdown, Fake Low-stock, Fake Limited-time, Fake High-demand, Fake Activity)
2. **Confirmshaming Dark Pattern Detection** model
3. **The 5 Dark Pattern Types Classification** model (Fake Countdown, Fake Low-stock, Fake Limited-time, Fake High-demand, Fake Activity)

Some of the main model versions are listed in the timeline graph (Figure 74). At first, pattern category detection and classification models were used in the project. But later, the models were found performing badly on the real-world data, and the problem was that not all the dark pattern categories can be detected based on the text only. Therefore, the detection range was narrowed down to 8 dark pattern types, and further to 5 dark pattern types. After the detection and classification models of the 5 dark pattern types were improved to an acceptable level, by removing the noises and further enriching the dataset, Confirmshaming dark pattern type was then being developed.

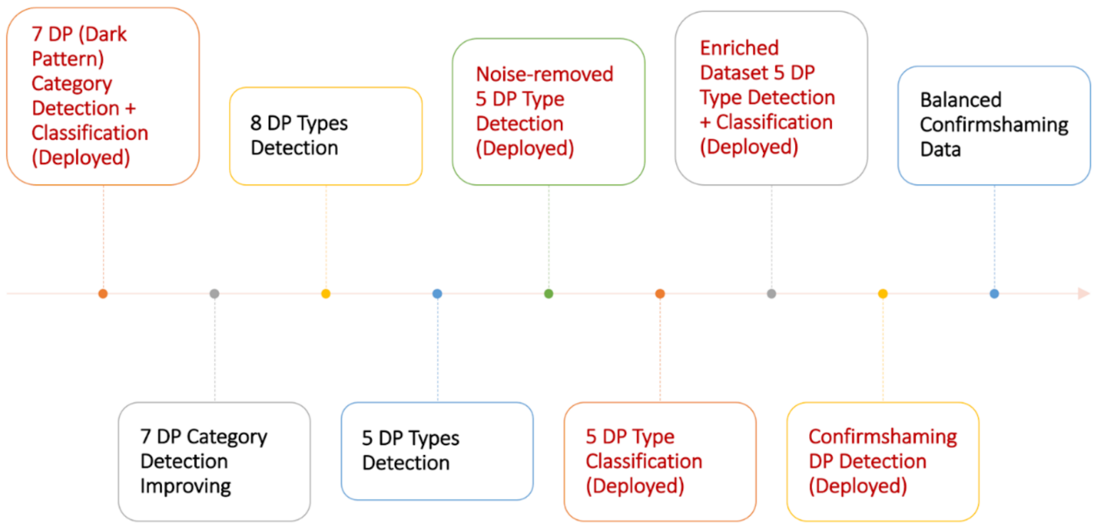


Figure 74 - Model Versions Development

5.1 Dark Pattern Detection Model Evaluation

5.1.1 Evaluation Metrics

During the model training, the dark patterns detection models are evaluated on the dataset we created, with a training and validation split being 0.7/0.3, which means 70% of the dataset will be used for training, and the other 30% of the dataset will be used for validation. The percentage regarding the distribution of target values ("dark pattern" vs "not dark pattern") will be the same in the training validation dataset.

Therefore, the dark pattern detection problem is transformed into a binary classification problem, where the examples in the dataset are divided into 2 target classes: positive (indicating the content is a dark pattern) and negative (indicating the content is not a dark pattern). There are four kinds of results for the prediction results of binary classification (TP, TN, FP, and FN) when we perform a comparison between the actual and predicted classifications. A 2×2 confusion matrix can be formulated from the outcome of the four kinds of results.

1. TP: True Positive — the actual classification is positive, and the predicted classification is positive — the actual dark pattern is correctly detected.
2. TN: True Negative — the actual classification is negative, and the predicted classification is negative — the actual normal is correctly classified.
3. FP: False Positive — the actual classification is negative, and the predicted classification is positive — the actual normal content is mistakenly classified as a dark pattern.
4. FN: False Negative — the actual classification is positive, and the predicted classification is negative — the actual dark pattern is mistakenly classified as normal content.

		Predicted condition	
	Total population = P + N	**Positive (PP)**	**Negative (PN)**
Actual condition	**Positive (P)**	**True positive (TP)**	**False negative (FN)**
	Negative (N)	**False positive (FP)**	**True negative (TN)**

Figure 75 - 2×2 Confusion Matrix

The parameters used for dark pattern detection model evaluation are precision, recall, and F1. The reasons are:

(1) What matters more to the users is whether all dark patterns are detected, which can be reflected by the value of "Recall". At the same time, it matters to the users whether all the returned dark patterns are truly dark patterns, which can be reflected by the value "Precision".

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

Figure 76 - Calculation of Recall and Precision

(2) The number of dark patterns on the website is much smaller than the amount of normal content on the website. So is our dataset for training and testing, and the F1 score is the most suitable evaluation parameter when the data is imbalanced, it considers both precision and recall.

$$F1\ Score = \frac{2 * (Recall * Precision)}{Recall + Precision}$$

Figure 77 - Calculation of F1 Score

5.1.2 Models Comparison

5.1.2.1 The 5 Dark Pattern Types Detection Model

5.1.2.1.1 Model Versions

There are in total 9 versions of the dataset used to train models for detecting the 5 dark pattern types (Fake Activity, Fake Low-stock, Fake Limited-time, Fake High-demand, Fake Countdown), they are shown in Table 30.

Table 30 - Versions of the Machine Learning Models for the 5 Dark Pattern Types Detection

Version	Descriptions and Modifications
V1: Original Dataset — full "dark_pattern.csv" + full "normie.csv"	The first version of the models was trained with the V1 dataset which was a simple merge of two datasets. This model is trained with the "Pattern Category" as the target feature.
V2: Enriched Dataset — V1 + misclassification data from results	1. Used the models trained with dataset V1 to test on the real-world data and gathered the misclassification data from the results to enrich the dataset. 2. Used the enriched dataset to retrain the models in V1 and got new models for V2.
V4: Filtered Dataset — including 8 pattern types from "dark_pattern.csv" + full "normie.csv"	1. Found the problem with models trained in V2 that the precision was quite low, as not all the dark pattern categories can be detected based on text only. 2. Filtered the dataset in the original "dark_pattern.csv" to contain only 8 pattern types, including "Fake Activity",

	“Confirmshaming”, “Pressured Selling”, “Trick Questions”, “Fake Countdown”, “Fake Limited-time”, “Fake High-demand”, and “Fake Low-stock” 3. Trained the new models with the filtered dataset of “dark_pattern.csv” and the original “normie.csv”.
V5: More Filtered Dataset — including 5 pattern types from “dark_pattern.csv” + full “normie.csv”	1. Found the problem with models trained in V4 that the precision was low because of a certain type of error — normal content regarding the checkbox message or normal advertisement message. 2. Comparing the misclassification results with the dark pattern strings used to train the models, found the reason for the problem — “Trick Questions” and “Pressured Selling” dark pattern types. 3. As the key to detecting “Trick Questions” is to detect double negative sentence structure, and this should be left for a separate model to do this. At the same time, the “Pressured Selling” dark pattern content is very similar to the normal content on ordinary shopping websites, which increases the false positive rate largely. 4. “Confirmshaming” dark patterns normally appear on buttons and links, so better to separate them from the others which can use only text to detect without any additional filtering. 5. Therefore, the “Trick Questions”, “Confirmshaming”, and “Pressured Selling” pattern types are removed from the dataset. 6. Trained the new models with the further filtered dataset of “dark_pattern.csv” and the original “normie.csv”.
V6: Remove Noise Dataset — V5 - remove noises from the dataset + re-labelled some tags in	1. Found the problem with models trained in V5 that the overfitting happened because of a certain type of error — for example, all the products with size information would be misclassified as dark patterns.

“normie.csv”	2. Comparing the misclassification results with the dark pattern dataset, it was found the problem was the words in dark pattern strings that were noise for model training. For example, “Tom from the US bought 55 inches curved BenQ monitor 3 hours ago”, the “55 inches”, “BenQ monitor”, “US” are all noises, what is useful to train the models is just the structure: “someone bought something some time ago.” But models learned from the noises and then they were regarding everything with “US” as a dark pattern. 3. Therefore, we removed the noises manually in the dataset to prevent the models from learning from these noises, by deleting the detailed information in the dark pattern strings. 4. Re-labelled some tags in “normie.csv” where the dark pattern types were not within the 5 Pattern Types listed in V5. 5. Trained the new models with the noise removed dataset.
V7: Lowercase Dataset — train the model with all data transformed to lowercase in V6	1. Found the problem that overfitting still existed, e.g., anything with “limited”, “only”, “low” these kinds of keywords that frequently appeared in the training dataset would be regarded as dark patterns. 2. Added a lot of filtering conditions to remove the normal content in the HTML before sending it to dark pattern detection models, to reduce the risk of misclassification. 3. As there were too many possibilities of the words if they were case sensitive, therefore, to reduce the number of filtering conditions, we tried this new version, by transforming all the text into lowercase during data processing before sending to the detection models. 4. Trained the new models with the lowercase dataset.
V8: Noise Free	1. The models trained with the V6 dataset

Dataset — V6 - (remove) more noises	performed better than the previous models, and also better than the models trained with lowercase data. Therefore, more noise was removed from the dataset V6. 2. Trained the new models with the further noise removed dataset.
V9: Enrich Trying Dataset — V6 + (add) web scraping data from 6 website pages.	1. Found the problem with the models trained in V8, after removing more noises, the models all went worse than before. Therefore, the noises we removed in V8 should be left there to train the models. 2. The problem with the previous models was the low precision, which indicates the poor capability of classifying the normal content into their own group — Not Dark Pattern. 3. Therefore, we need more normal content data in the dataset to train the models. As in the real-world data, the number of normal contents is much higher than the number of dark patterns. 4. Used the python script for web scraping 6 website pages and went through the web scraped data and re-labelled them. Merged the re-labelled dataset with the dataset of V6. 5. Trained the new models with the enriched dataset.
V10: Enriched Dataset — V6 + (add) web scraping data from 16 website pages.	1. Found that the models trained with enriched dataset in V9 worked the best so far, so it seemed enriching the dataset based on V6 was a correct direction to go for improving the model performance on real-world data. 2. Used python script for gathering web scraping data, obtained the data from 15 website pages. 3. Used the models trained in V9 to help separate normal content and the dark patterns in the scraped data. 4. Re-labelled the web scraping data and merge it with the dataset in V6. 5. Trained the new models with the further enriched dataset.

5.1.2.1.2 Models Evaluation Results

1. Bernoulli Naive Bayes

The models trained before version 9 all have serious overfitting, having great performance on the training and validation data, but performed poorly on the real-world data. The recall value was higher above 0.9 for all the models except the one trained with V10-HO, while the precision was below 0.9 all the time. From this, we could see the Bernoulli Naïve Bayes models worked well to detect actual dark patterns but were likely to wrongly regard the normal content as dark patterns.

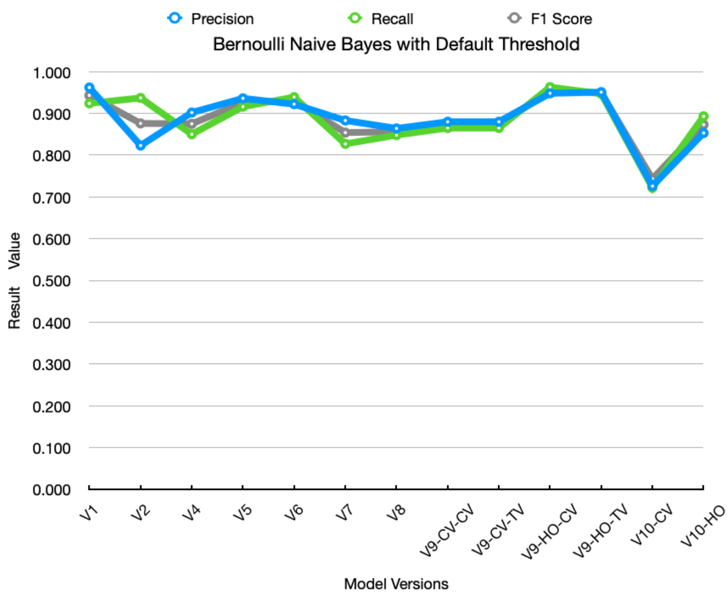


Figure 78 - Validation Results during training (Precision, Recall, and F1 Score)

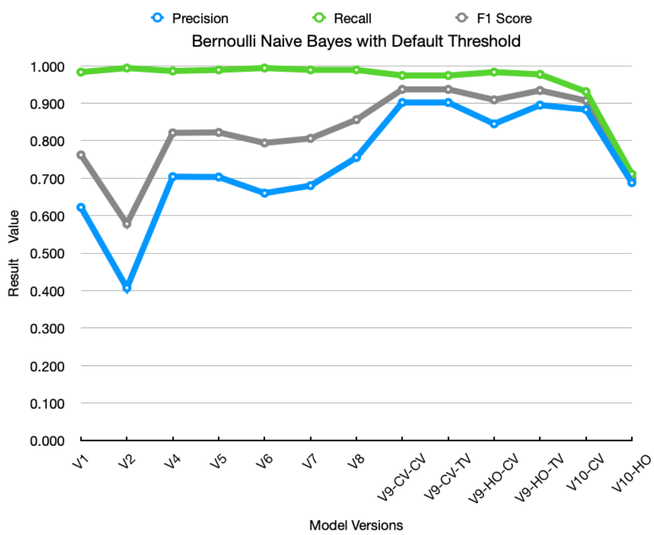


Figure 79 - Testing Results on real-world data (Precision, Recall, and F1 Score)

2. Support Vector Machine

Similar to Bernoulli Naïve Bayes models, the SVM models trained before version 9 all have serious overfitting as well. The recall score was above 0.95 for all the models, with a low precision score all the time before enriching the dataset, resulting in the poor capability of classifying normal content into its own class. From this, we could see the SVM models worked well to detect actual dark patterns as well but would very likely wrongly regard the normal content as dark patterns when the models were trained with data before enriching.

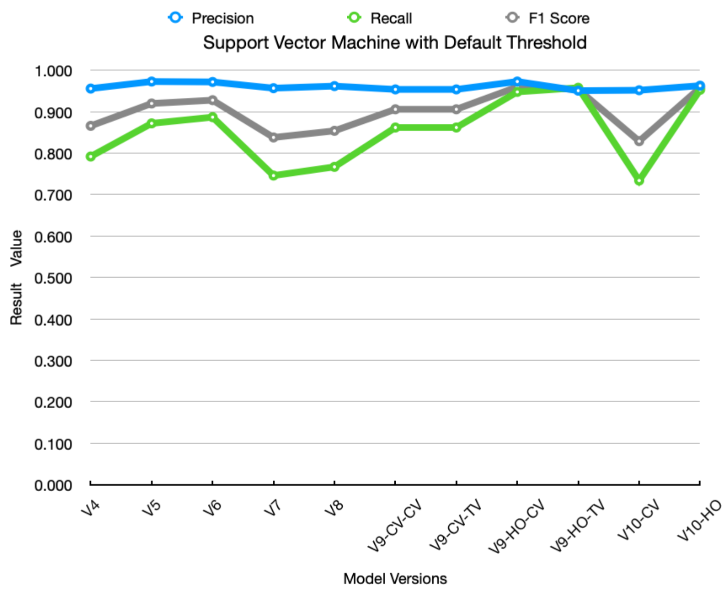


Figure 80 - Validation Results during training (Precision, Recall, and F1 Score)

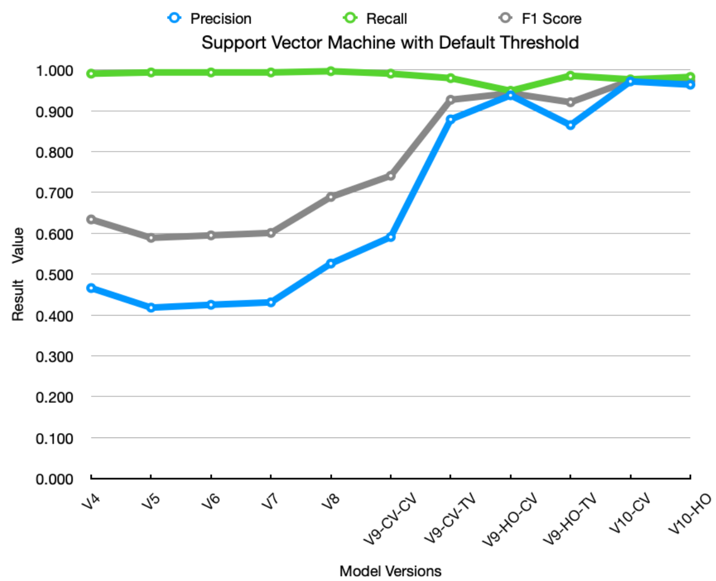


Figure 81 - Testing Results on real-world data (Precision, Recall, and F1 Score)

3. Random Forest

Random Forest models are much better at handling overfitting problems, even before the dataset was enriched (before V9), the F1 Score of the models was still above 0.8. Since the Random Forest classifiers are good at dealing with text classification with high dimensional noisy data, they performed relatively much better even before the noise in the dataset was manually removed. But some degree of overfitting can still be observed for the models trained before enriching the dataset.

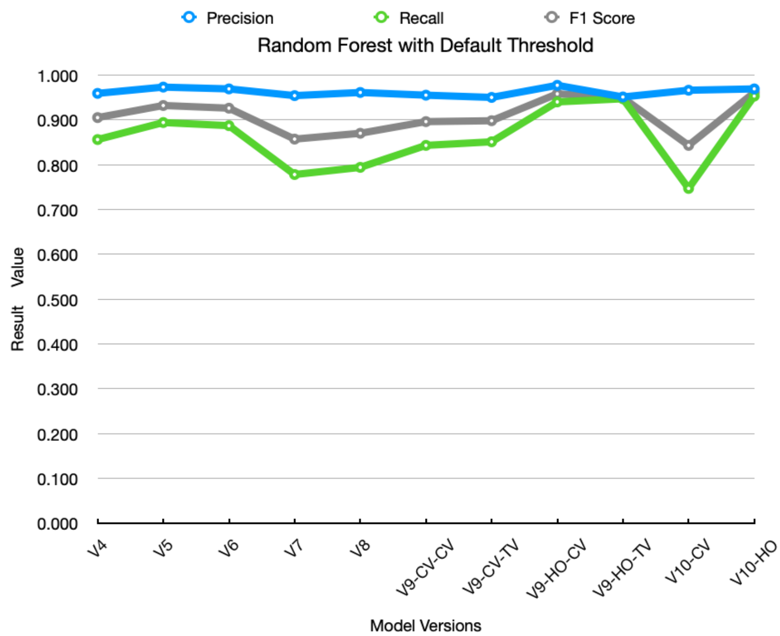


Figure 82 - Validation Results during training (Precision, Recall, and F1 Score)

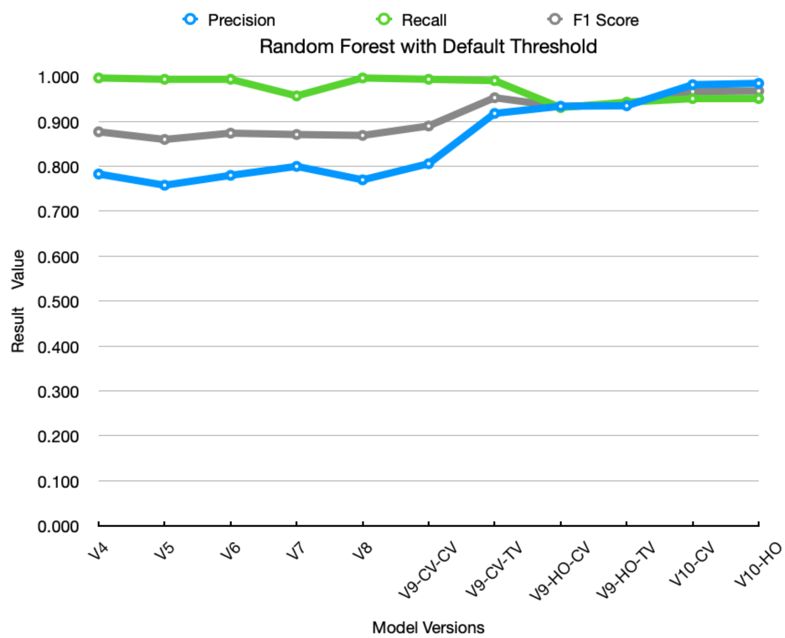


Figure 83 - Testing Results on real-world data (Precision, Recall, and F1 Score)

4. CNN

CNN models were trained on V3 (not listed here because it was totally underfitted) and V10-HO. The five versions trained on the V10-HO dataset are shown in Table 31, with the first 3 models were trained with word embedding trained on the fly, while the other 2 models were trained with pre-trained word embedding GloVe:

Table 31 - Versions of the CNN Models for the 5 Dark Pattern Types Detection

Versions	Descriptions and Modifications
EmbOTF-V1: number of filters = 64; kernel size = 2; dense layer units = 32;	1. This 6-layer model used 64 filters, kernel size to be 2, and 32 units in a dense layer, with a maxpooling layer and a dropout layer to reduce overfitting. The Embedding layer used the word embedding trained on the fly with length to be 20 and dimension to be 20. 2. Early stopping was applied for training to reduce overfitting, in total 3 epochs of training were conducted.
EmbOTF-V2: number of filters = 32; kernel size = 2; dense layer units = 32;	1. This 6-layer model used 32 filtered, kernel size to be 2, and 32 units in a dense layer, with a maxpooling layer and a dropout layer to reduce overfitting. The Embedding layer used the word embedding trained on the fly with length to be 20 and dimension to be 20. 2. Early stopping was applied for training to reduce overfitting, in total 3 epochs of training were conducted.
EmbOTF-V3: number of filters = 64; kernel size = 1; dense layer units = 32;	1. This 6-layer model used 64 filtered, kernel size to be 1, and 32 units in a dense layer, with a maxpooling layer and

	a dropout layer to reduce overfitting. The Embedding layer used the word embedding trained on the fly with length to be 20 and dimension to be 20. 2. Early stopping was applied for training to reduce overfitting, in total 4 epochs of training were conducted.
PreEmb-Cased: number of filters = 64; kernel size = 2; dense layer units = 32;	1. This 7-layer model used 64 filtered, kernel size to be 2, and 32 units in a dense layer, with 2 maxpooling layers and a dropout layer to reduce overfitting. The Embedding layer used the pre-trained word embedding GloVe with length to be 20 and dimension to be 300. 2. Early stopping was applied for training to reduce overfitting, in total 5 epochs of training were conducted.
PreEmb-Uncased: number of filters = 64; kernel size = 2; dense layer units = 32;	1. Transformed the text in the dataset into lowercase for training. 2. This 7-layer model used 64 filtered, kernel size to be 2, and 32 units in a dense layer, with 2 maxpooling layers and a dropout layer to reduce overfitting. The Embedding layer used the pre-trained word embedding GloVe with length to be 20 and dimension to be 300. 3. Early stopping was applied for training to reduce overfitting, in total 5 epochs of training were conducted.

CNN models trained with word embedding trained on the fly worked much better than the models trained using pre-trained embedding GloVe. For the models trained with customised word embedding (embedding on the fly), the F1 score were all above 0.9, and the precision and recall were all above 0.9. At this stage, the performance of the CNN models trained with customized word embedding was similar to the Random Forest models trained after enriching the dataset.

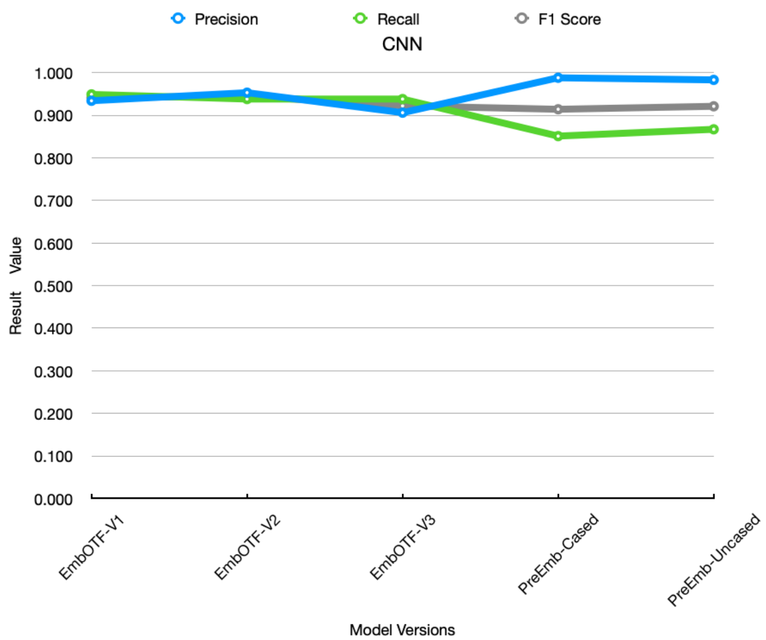


Figure 84 - Validation Results during training (Precision, Recall, and F1 Score)

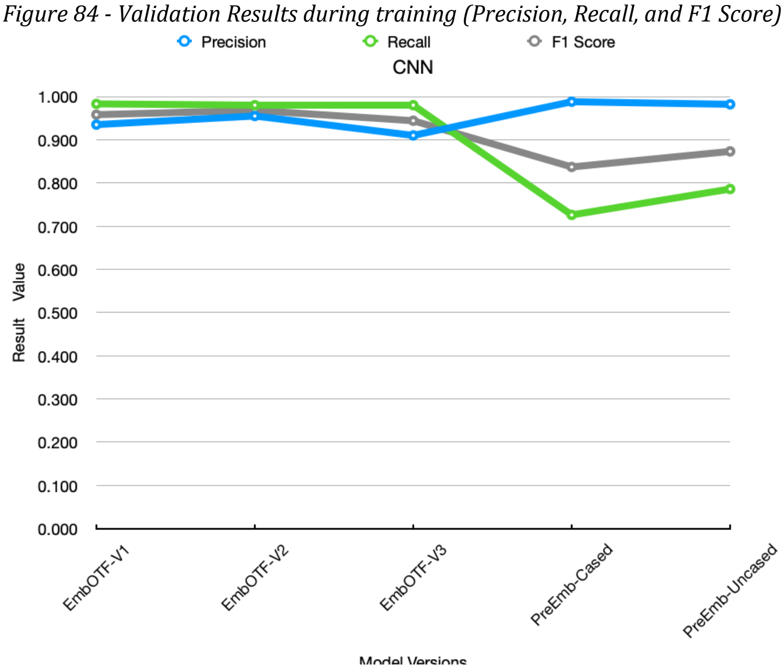


Figure 85 - Testing Results on real-world data (Precision, Recall, and F1 Score)

5.1.2.2 Confirmshaming Detection Model

5.1.2.2.1 Model Versions

There are in total 10 versions of the models trained to detect Confirmshaming dark pattern, they are shown in Table 32.

Table 32 - Versions of Confirmshaming Detection Models

Version	Descriptions and Modifications
V1: Original Dataset — Case sensitive, using 80% of the dataset for training and 20% of the dataset for testing.	
V2: Original Dataset — Case sensitive, using the whole dataset for training and testing with 5-fold cross-validation.	
V3: Enriched Dataset — Case sensitive, using 80% of the dataset for training and 20% of the dataset for testing.	• Enriched the dataset in V2 with the Confirmshaming examples on the dark pattern website.
V4: Enriched Dataset — Case sensitive, using the whole dataset for training and testing with 5-fold cross-validation.	
V5: Enriched Dataset — Lowercase input features, using 80% of the dataset for training and 20% of the dataset for testing.	• Transformed the text in the V3 dataset into lowercase for training. • Found out that this new version of models performed better than the case-sensitive models. • Therefore, from this time on, the models would be trained with lowercase text.
V6: Enriched Dataset — Lowercase input features, using the whole dataset for training and testing.	
V7: Balanced Training Dataset (Duplicate Minority Class)— Lowercase input features, using 80% of the dataset for training and 20% of the dataset for testing.	• Found the problem of severe underfitting in the Bernoulli Naive Bayes models with the previous imbalance data. • Balanced the dataset by randomly duplicating minority class instances in the training dataset V6. • Trained the models with the

	balanced training dataset.
V8: Balanced Training Dataset (SMOTE)— Lowercase input features, using 80% of the dataset for training and 20% of the dataset for testing.	• Balanced the dataset by using SMOTE (synthetic minority class oversampling technique) in the training dataset V6. • Trained the models with the balanced training dataset.
Original Dataset Distribution: 135: 6361 (Confirmshaming Dark Pattern: Not Confirmshaming Dark Pattern = around 1:47) **Enriched Dataset Distribution**: 193: 7994 (Confirmshaming Dark Pattern: Not Confirmshaming Dark Pattern = around 1:41)	

5.1.2.2.2 Models Evaluation Results

1. Bernoulli Naive Bayes

Bernoulli Naive Bayes models appeared to be the worst models. Before the training data was balanced, Bernoulli Naive Bayes models failed to learn from the actual shape of the data and went into underfitting. The reason was that from the perspective of accuracy, the model could achieve 98% accuracy by labelling everything negative in the training dataset since the training dataset was highly imbalanced, so the models were learning that the easiest way was to classify everything into one class – the negative class. After oversampling, the number of positive and negative instances was equal in the training data, so it forced the models to work much harder to figure out the true distribution of the data, as throwing everything into one category wouldn't work anymore under that condition. However, we could still observe severe overfitting from the evaluation results, as the duplicating results didn't add extra information to the model for training.

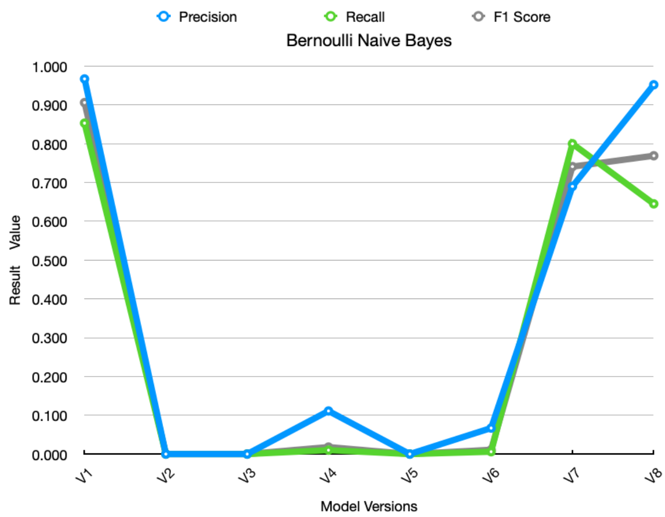


Figure 86 - Validation Results during training (Precision, Recall, and F1 Score)

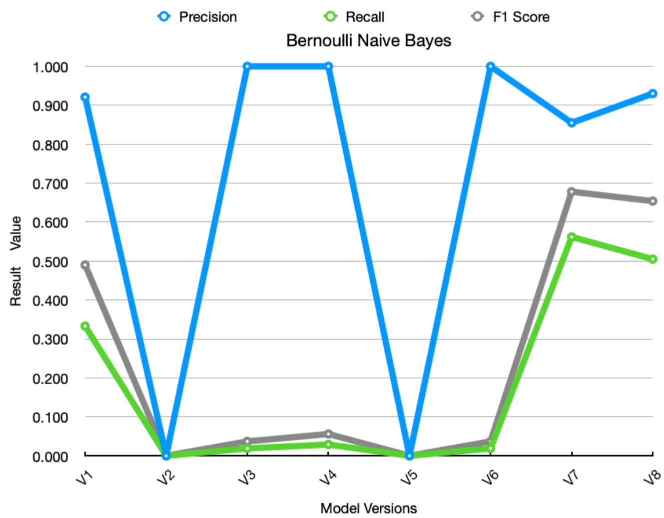


Figure 87 - Testing Results on real-world data (Precision, Recall, and F1 Score)

2. Support Vector Machine

For SVM models, looking at the evaluation results, we can see that besides the V1 and V2 were overfitting, the rest of the models were performing well, with an F1 Score over 0.9 for all of them. From the validation and the evaluation results we could also see that the lucky split happened for V3 and V5 which used hold-out training and validation strategy, so as a result, the V4 and V6 used the whole dataset for training and validation perform the best in real-world data. Balancing the training dataset didn't make much difference in the model performance during the training and evaluation stage for SVM models.

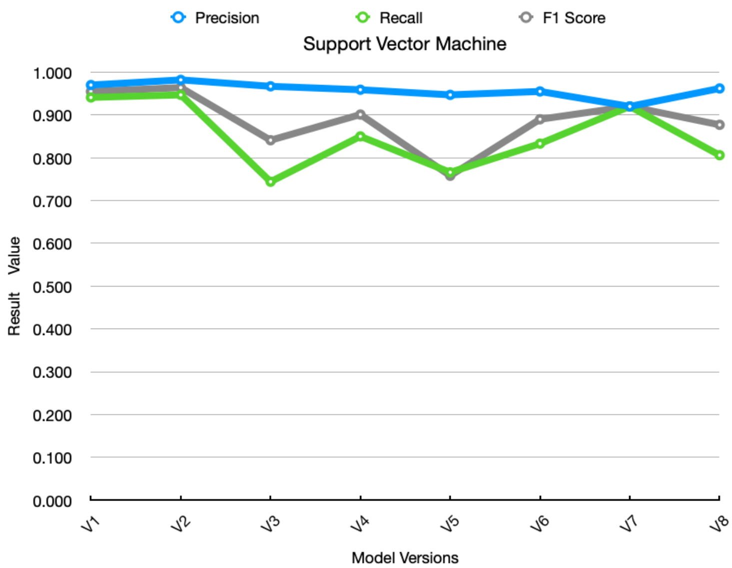


Figure 88 - Validation Results during training (Precision, Recall, and F1 Score)

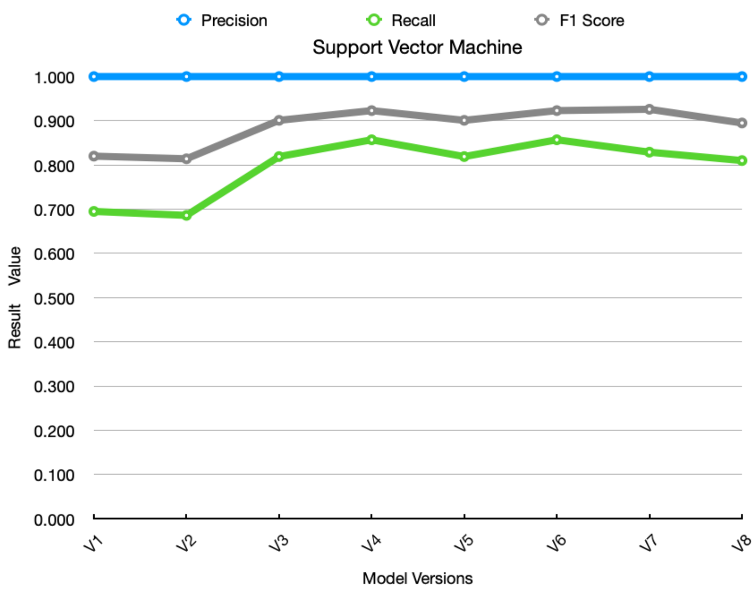


Figure 89 - Testing Results on real-world data (Precision, Recall, and F1 Score)

3. Logistic Regression

For Logistic Regression models, models V1 and V2 are shown to be overfitting. The models trained after enriching the dataset witnessed a drop in recall, indicating around a 15% drop in the capability of detecting Confirmshaming Dark Pattern. From the evaluation results, we can see those models are not overfitting, but the recall is still kind of low. Balancing the training dataset did make a huge difference in the model performance during the training and evaluation stage. On the other hand, randomly duplicating the minority class in the training data was a better choice than using SMOTE for balancing training data in this case.

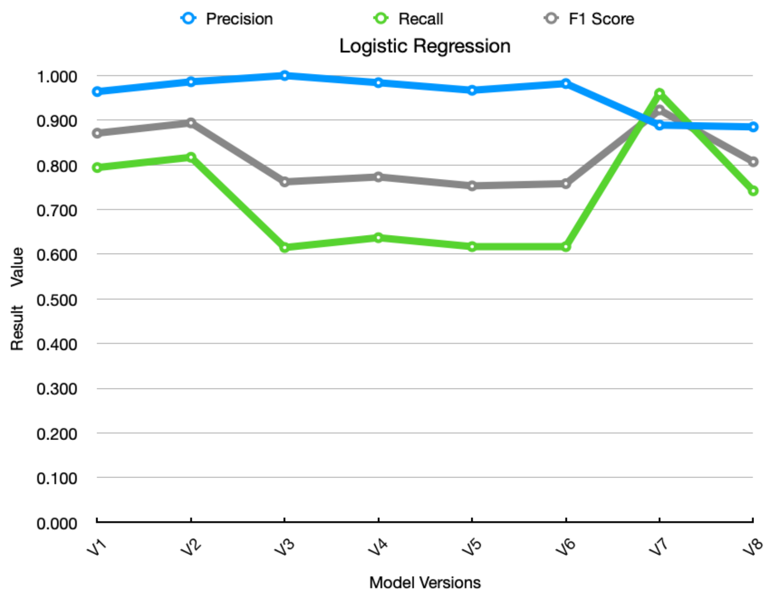


Figure 90 - Validation Results during training (Precision, Recall, and F1 Score)

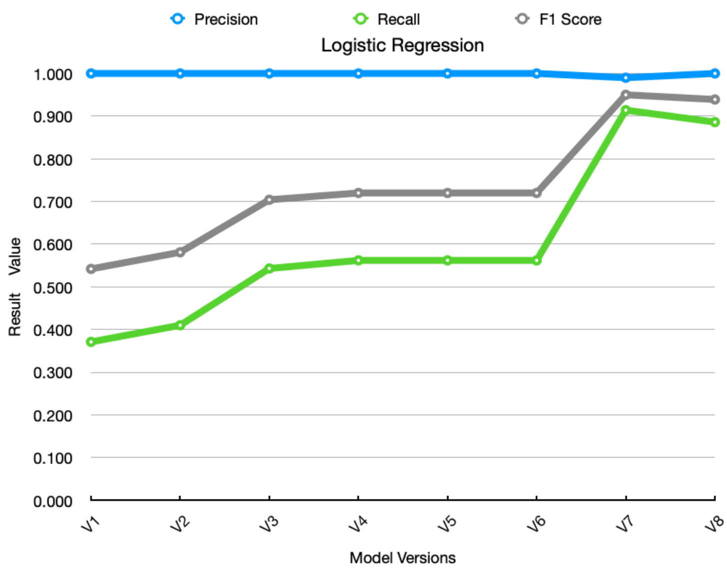


Figure 91 - Testing Results on real-world data (Precision, Recall, and F1 Score)

4. Random Forest

For Random Forest, the models performed similarly to the SVM models, and the V6 model was chosen to be the winner for deployment in the end as it had the highest F1 score, which was 0.929, among all the models before balancing the training dataset for all the models training. And, for this model version, the input features are not case sensitive, so when doing data processing, the text needs to be transformed into lowercase before conducting vectorization. and sent to the model for prediction.

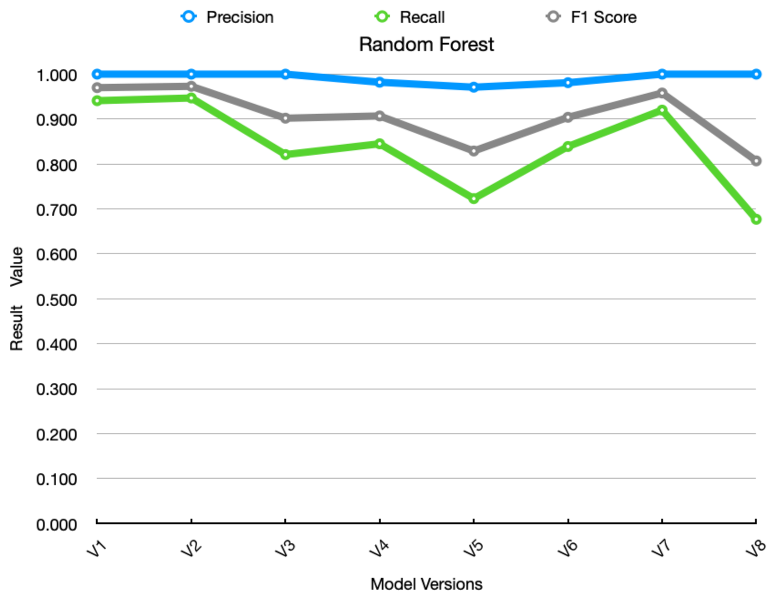


Figure 92 - Validation Results during training (Precision, Recall, and F1 Score)

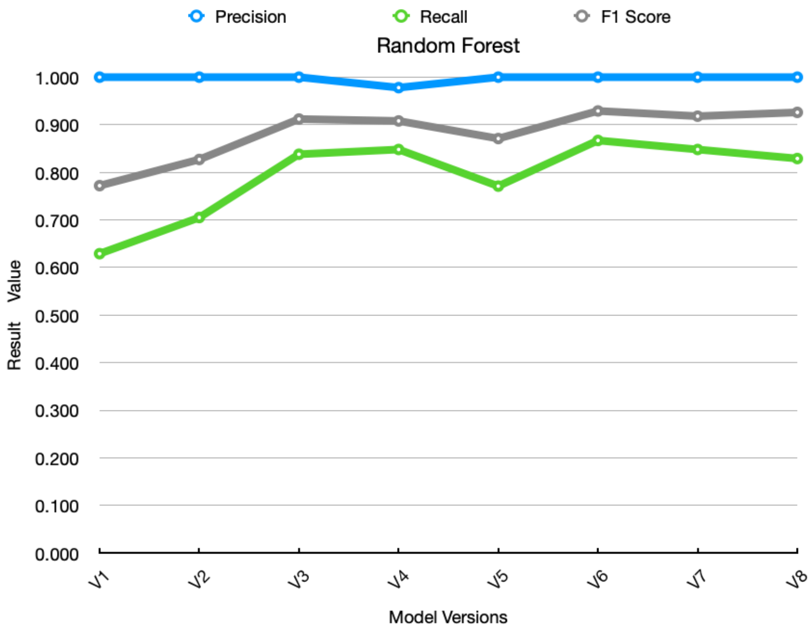


Figure 93 - Testing Results on real-world data (Precision, Recall, and F1 Score)

5.2 The 5 Dark Pattern Types Classification Model Evaluation

5.2.1 Evaluation Metrics

The dark pattern types of classifiers are trained and validated on the dataset made from a subset of the dataset in the research from Mathur et al. (2019) and an additional part of manually enriched data, with a train and test split as 0.6/0.4, using the evaluation parameter "accuracy".

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Figure 94 - Calculation of Accuracy

5.2.2 Models Comparison

5.2.2.1 Model Versions

There are 3 versions in total for the 5 Dark Pattern Types classification models as shown in the table below, where the first version was trained with the original subset data, and the other two versions were trained with the enriched data.

Table 33 - Versions of the 5 Dark Pattern Types Classification Models

Versions	Descriptions and Modifications
V1: Subset Dataset — CountVectorizer as text vectorization, using 60% of the dataset for training and 40% of the dataset for testing.	• Found the problem of overfitting, can't generalize well since the pattern type dataset wasn't enriched with the pattern type detection dataset.
V2: Enriched Dataset — CountVectorizer as text vectorization, using 60% of the dataset for training and 40% of the dataset for testing.	• Enriched the 5-pattern type dataset according to the enriched instances in the 5 dark pattern type detection datasets. • Used CountVectorizer as the way to convert the input text into a frequency representation.
V3: Enriched Dataset — TfidfVectorizer as text vectorization, using 60% of the dataset for training and 40% of the dataset for testing.	• Used TfidfVectorizer as the way to convert the input text into 'inverse document frequency' representation, in this way, not only focus on the frequency of the words but also the importance of the words

5.2.2.2 Model Evaluation Results

Random Forest models were the 'most stable' models during the training and validation, reaching an accuracy of 0.967 with the model trained on the enriched dataset. After enriching the dataset, the accuracy of all the models dropped, among which the Random Forest dropped the least. When enriching the dataset, some noises were added to the dataset on purpose, to improve the performance of the models in the real world, hoping to generalize better in the real world by training with a certain degree of noise in the data.

The 5 Dark Pattern Types Classification Models

Model_Versions	Multinomial Naive Bayes	Random Forest	Logistic Regression
V1	0.965	0.969	0.974
V2	0.943	0.967	0.965
V3	0.957	0.967	0.960

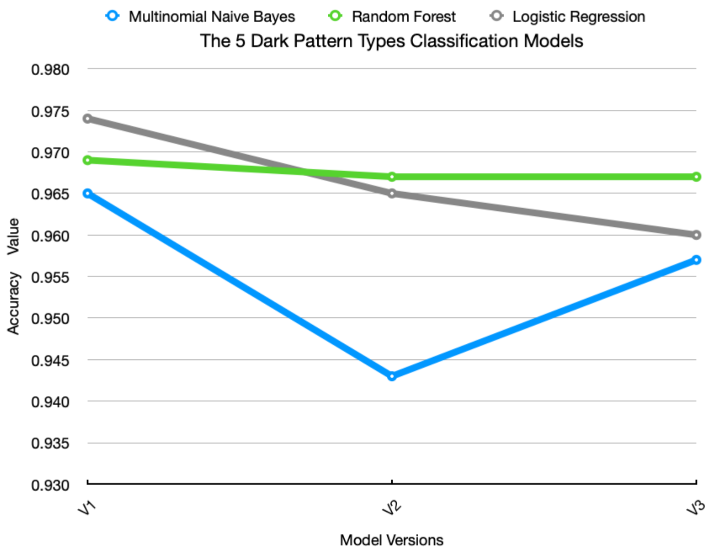


Figure 95 - Testing Results after training (Accuracy)

6. Conclusion

6.1 Project Management Strategy

During the beginning of the development of the system, technologies such as Microsoft Teams and Confluence were used to manage the project. Shortly after, it was realized that this approach is inefficient, so we then switched to an Agile approach, specifically Scrum Methodology. The primary form of project management used is ZenHub and GitHub.

An issue or task that needs to be done for the system is added into the project backlog. The weekly management process is as follows:

- Friday - After class, we have a 1 -2 hour in-person meeting. The sprint that was just completed is reviewed, such as burndown charts, and notes are taken on where it needs to be improved, if an issue is still not closed it gets added to the new sprint. In this meeting, a scrum master is assigned and creates a new sprint in ZenHub. Issues from the project backlog are added into the sprint backlog and assigned to the appropriate assignees. Each issue is also assigned an estimated number of points, the scale for these points is 1 to 1 with the number of days it will take to complete the task. For example, 1 point means it may take one day.

- Saturday to Wednesday - Each day at 2 pm there is a stand-up meeting that varies in time. In this stand-up meeting, each person explains what they have worked on the day before and if they have completed a task and need it to be reviewed or if any help is needed this is when the team will get together and discuss possible workarounds.

- Thursday - In our daily stand-up at 2 pm we discuss the same way as the rest of the week but at the end of the stand-up, we do our retrospective for the current sprint. Each member will add 1-2 lines on what made them mad, sad, or glad this sprint. The sprint is then reviewed, for the number of tasks completed and a draft demo is made for Friday's demo to the class and lecturers. During each sprint, the issues are moved on the Scrum board, by the assignees, to “In progress”, “Review Q/A” and “Closed”, depending on how the task is going. For a task to be considered closed, it must be reviewed in the daily stand-up. Every group member must complete pull requests to GitHub each week, these pull requests are reviewed by other members, and comments are left on the requests to acknowledge the work or to ask questions. Once reviewed, the pull request is then merged into the main branch on GitHub. Each pull request should have a connection issue from the current sprint.

6.2 Biggest Challenges

6.2.1 Frontend

When developing the front end, there were a couple of challenges that had to be overcome to develop the system. The first main challenge was how the extension was going to be able to run on multiple tabs at once. Originally when the user clicked the detection button and the data on the dark patterns was sent back to the extension, this data would then be shown across all the extension popups that were open on any tab, with any new detection on a tab changing the data in every other tab. To overcome this, the chrome storage API was used to now store the data for a specific tab, in chrome storage, using the tab id of the tab that made the request. Now instead of the data being sent straight to the UI, it is put into chrome storage first and then when the UI is opened, it uses its tab id to look in the storage for data that belongs to that tab. This allowed the extension to be run across multiple tabs, each having all different data being displayed in the UI.

Another challenge for the front end was anytime that content was being injected onto a webpage through a content script, the content could often inherit styles and formatting from the web page itself. This sometimes caused the UI to look different on different web pages. This was a major issue at the beginning of the development of the extension, as the UI was being injected onto the page. However, when the UI got moved to the popup of the extension, this was no longer an issue for the main UI. It was still an issue for the popup messages that get displayed during certain events. To overcome this, the extension had to be tested across many different sites and through these common stylings and inheritances were found that caused the problems, which then could be overwritten in the content scripts CSS files in the extension.

6.2.2 Backend

Whilst developing the back end, many challenges were faced, some that were not able to be overcome given the time frame and complexity. Here is a list of all the challenges faced while developing the back end:

- AWS DynamoDB - In order to limit the cost of the system, the read/write capacity of the DynamoDB database was set to 1. This did not become a problem until our datasets got bigger and bigger. Whilst initially adding the dataset to the database, some data was not added as the rate exceeded our read/write capacity. This problem was overcome by succumbing to AWS's costs and increasing our read/write capacity to 5 and this problem never occurred since.

- AWS ElasticBeanStalk - This was the biggest challenge by far, at the beginning of development, it was decided that AWS ElasticBeanStalk would be used as the service to host the Node and Python system as the documentation was full of information and easy to follow, and the cost was not too extreme. The system was hosted on a T2.micro system, which is the smallest and least powerful option that AWS provides. This only became an issue when OCR was added, OCR heavily depends on hardware, and since the system was hosted on the least hardware intensive option it would fail and crash a lot. When it crashed, the only way to find out the reason why was to use SSH and most of the time the error would not be clear or would be completely irrelevant. When the instance would crash, it made it impossible for all the systems to work or be tested, the only way to test them at that point was to do it locally. In order to overcome the challenge, OCR had to change, so it was optimized to use less hardware in a quicker time.
- AWS S3 - All the updated datasets get saved into S3 as CSV files. These files are secured and cannot be accessed without the correct credentials. This was an issue because, when the Python side wants to auto train the training model it must read from the CSV file but it cannot as the credentials are not public. To overcome this challenge, the S3 bucket and all files that would be added to it would become public so anyone can access them.

6.2.3 Machine Learning Models

The biggest challenges for building machine learning models include dealing with noises and handling overfitting and underfitting problems during the training process.

The noises in the dataset for training were found after the models were trained and applied to the real-world data. By analyzing the wrong prediction text strings from the results, the noises were recognized and then manually removed from the training dataset. Removing the noises also improved reducing the overfitting during the model training.

Overfitting happened at the beginning of the project, and lasted for a few weeks, before the correct direction was found to improve it, which was enriching the training dataset with thousands of examples of normal content on the website. The model generalized much better after training on the enriched dataset. At the same time, not all the noises were removed in the training dataset, because a certain degree of the noises in the dataset could help the models to generalize better, in other words, more adaptable to unseen data.

The underfitting happened when the dataset was highly imbalanced for the models to train, especially when training the Confirmshaming detection model, where the ratio of the Confirmshaming dark pattern instances and the normal content was 1:47. The Bernoulli Naive Bayes models were fully underfitting all the time, as they could already achieve an accuracy of 98% by labeling everything into one class during training. To balance the dataset for reducing underfitting, two different methods were used, one was randomly duplicating the minority class in the training dataset, and another one was using SMOTE technique for balancing the training dataset with synthetic data. The Bernoulli Naive Bayes started learning properly for the real shape of the data, but overfitting still happened during training, as it could be seen the model didn't generalize well on the real-world data. Therefore, duplicating the instances in the dataset could end up overfitting and the duplicating instances didn't add extra information for training. Using SMOTE on the word frequency vectorizer didn't have any real meaning for training, as the synthetic data couldn't reflect relational textual meaning after reversing the word vector back into the text. Therefore, although balancing the training data using duplicates and synthetic data could improve the underfitting during training, the models were not deployed for use.

6.2.4 Optical Character Recognition (OCR)

In the first version of the extension with OCR, it was required to detect the images for Disguised Ads and five dark patterns in texture detection. Disguised Ads is later removed because Google has already addressed this dark pattern, by marking them as downloadable image advertisements for users' attention.

The early process of OCR is downloading images as jpg format files, loading image files, preprocessing images, detecting texture, and removing the image files. It took about 40 seconds to detect images on one website. To decrease its time, base64 is used to encode and decode the downloaded images data. It did decrease the time by a lot; however, the accuracy of detection became lower as well. Base64 seems to ignore some image data when encoding and decoding. After many attempts, the balanced way is using OpenCV to read the downloaded image data directly. With filtering repeated images, the time has been reduced within 5 to 25 seconds for most shopping websites' homepages.

There are also some issues when the auto training was added into the project. Once the auto training is starting, the OCR service barely does not work. For the website which has amounts of images on it, the extension will report an error to tell user to detect it again. For the dark pattern example website, it sometimes will not detect the image with dark pattern. The cause of this problem may be running out of memory.

6.2.5 Definition of Dark Pattern

There is not a clear boundary between the marketing strategy and dark patterns for this project to detect. For example, when someone puts one item into a basket, there may be a pop-up window showing “people who bought this item also got another one”. Some users believe this is a dark pattern, but the other users think it is just a normal marketing strategy. Therefore, it is necessary to set a clear boundary for the dark patterns and market strategy for further detection.

6.3 Future Improvements

6.3.1 Dark Pattern Detection

In Table 34, it shows a list of the other dark patterns that, with more time and research, could have been detected and how to go about doing so.

Table 34 - Potential Dark Patterns that could be detected.

Pattern Type	***Description***	***Detection***
Sneak into Basket	Additional items are added into the user's checkout basket when purchasing a product, usually caused by pre-checked buttons or checkboxes on a previous page.	(1) Develop a web crawler to follow the process of checking out (2) Compare items on the checkout page and product pages. **Challenge:** (1) Checkout process is different for different websites, hard to apply the web crawler and comparison technique to all the websites.
Hidden Cost	Unexpected charges may appear in the user’s checkout basket, usually happening at the last step of the checkout	(1) Develop a web crawler to follow the process of checking out

	process. (e.g., delivery charges)	(2) Compare prices and items on the checkout page and product pages. **Challenge:** (1) Checkout process is different for different websites, hard to apply the web crawler and comparison technique to all the websites. (2) Tax should not be regarded as a hidden cost.
Friend Spam	The product asks users' permission for their social media or email, to spam all their connections.	(1) Analyze the HTML if the site asked for email or social media permissions. (2) Analyze the requirement for sending the email.
Trick Questions	Use complicated sentence structure to confuse users, attempting users to make wrong choices, often with checkboxes or buttons. For example, using the double negative sentence. (e.g., "If you do not wish to be contacted via email, please ensure that the box is not checked.")	(1) Gather the text on the checkout boxes, buttons, and links. (2) Use Natural Language Processing for Double Negation Detection.
Fake Review	The reviews can be found to be copied from other websites, where there may be different customer names but exact same review content.	(1) Develop a web scraper for gathering reviews on various shopping websites frequently. (2) Save the gathered reviews in a dataset. (3) Compare the review with the ones

		saved in the dataset, check the level of similarity and coverage.
Misdirection	The design purposely attracts users' attention to one thing, to make them ignore or difficult to notice another.	(1) Get the HTML and the CSS style sheets of the webpage. (2) Check if there is any font or image pixel size that is highly different from the content around it. **Challenge:** (1) Significant variations exist regarding how the patterns are shown across different websites.

In addition, to shrink the time for image detection, it is suggested to try more in the hardware and software way. For hardware, putting OCR on multithread processing or graphic card like CUDA will decrease the detection time, however, it will also cost more money for the server service. For the software, filtering some small size's images should be useful as well.

6.3.2 Website

After detecting more types of dark patterns, the sample page is supposed to be enriched with extra examples of dark patterns for testing the chrome extension.

In addition, it is necessary to further develop a webpage for enhancing user interactive level. It will include a quiz and grading system, so users can play with dark patterns examples in the quiz on the website, which can attract more users and educate the users during the quiz.

7. References and Key Resources

Brignull, H. (2010). Dark Patterns. Darkpatterns.org. Retrieved 20 December 2021, from https://www.darkpatterns.org/.

Browser Market Share Worldwide. (n.d.). Statcounter. Retrieved October 25, 2021, from https://gs.statcounter.com.

Chi, M. (2021, July 23). How to Send Data Between Chrome Extension Scripts. Medium. https://javascript.plainenglish.io/how-to-send-data-between-chrome-extension-scripts-1182ce67b659

Chromik, M., Eiband, M., Völkel, S. T., & Buschek, D. (2019, March). Dark Patterns of Explainability, Transparency, and User Control for Intelligent Systems. In IUI workshops (Vol. 2327).

Co-discovery Learning. (n.d.). Understanding Chi. Retrieved November 17, 2021, from http://hci.ilikecake.ie/eval_codiscoverylearning.htm

Curley, A., O'Sullivan, D., Gordon, D., Tierney, B., & Stavrakakis, I. (2021). "Give light, and the darkness will disappear of itself": The Design of a Framework for the Detection of Web-Based Dark Patterns. Retrieved 13 December 2021, from https://arrow.tudublin.ie/cgi/viewcontent.cgi?article=1002&context=ascnetcon.

Dalrymple, B. (2018). Cognitive Walkthroughs. Medium. Retrieved 17 November 2021, from https://medium.com/user-research/cognitive-walkthroughs-b84c4f0a14d4.

Dewi, P. W. S., Dantes, G. R., & Indrawan, G. (2020). User experience evaluation of e-report application using cognitive walkthrough (cw), heuristic evaluation (he) and userexperience questionnaire (ueq). Journal of Physics: Conference Series, 1516, 012024. https://doi.org/10.1088/1742-6596/1516/1/012024

Easy Checks – A First Review of Web Accessibility. (n.d.). W3C. Retrieved November 15, 2021, from https://www.w3.org/WAI/test-evaluate/preliminary/

Getting started. (n.d.). Chrome Developers. Retrieved October 26, 2021, from https://developer.chrome.com/docs/extensions/mv3/getstarted/

Hes, B. (2017). Data relevance – Crucial in Asset Performance Management. Stork. Retrieved 17 November 2021, from https://www.stork.com/en/about-us/blog/data-relevance-crucial-in-asset-performance-management.

IshΔn. (2019, February 3). What and Why of Usability Evaluation. Medium.https://blog.prototypr.io/what-and-why-of-usability-evaluation-46bf4b6dee07

Kuar, A. (2018). Accessibility guidelines for UX Designers. Medium. Retrieved 17 November 2021, from https://uxdesign.cc/accessibility-guidelines-for-a-ux-designer-c3ba775539be.

Lacey,C. and Caudwell,C., 2019. Cuteness as a 'dark pattern' in home robots. In Proceedings of the 14th ACM/IEEE International Conference on Human-Robot Interaction. IEEE Press, 374–381

Maier, M., & Harr, R. (2020). Dark Design Patterns: An End-User Perspective. Human Technology, 16(2), 170–199. https://doi.org/10.17011/ht/urn.202008245641

Mathur, A., Acar, G., Friedman, M. J., Lucherini, E., Mayer, J., Chetty, M., & Narayanan, A. (2019). Dark Patterns at Scale. Proceedings of the ACM on Human-Computer Interaction, 3(CSCW), 1–32. https://doi.org/10.1145/3359183

Mathur, A., Kshirsagar, M., & Mayer, J. (2021). What Makes a Dark Pattern. . . Dark? Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, 2–3. https://doi.org/10.1145/3411764.3445610

Mcleod, S. (n.d.). Likert Scale Definition, Examples and Analysis | Simply Psychology. Simply Psychology. Retrieved November 15, 2021, from https://www.simplypsychology.org/likert-scale.html

McNealy, J. (n.d.). What are dark patterns? An online media expert explains. The Conversation. Retrieved October 20, 2021, from http://theconversation.com/what-are-dark-patterns-an-online-media-expert-explains-165362

NI Business Info. (n.d.). Consistency in web design | nibusinessinfo.co.uk. Retrieved November 15, 2021, from https://www.nibusinessinfo.co.uk/content/consistency-web-design

Parrilli, D. M., &Hernández-Ramírez, R. (2020). Re-Designing Dark Patterns to Improve Privacy. 2020 IEEE International Symposium on Technology and Society (ISTAS), 2020, pp. 253-254, https://doi.org/10.1109/ISTAS50296.2020.9462197

Siting, Z., Wenxing, H., Ning, Z., & Fan, Y. (2012). Job recommender systems: A survey. 2012 7th International Conference on Computer Science & Education (ICCSE), 2–5. https://doi.org/10.1109/iccse.2012.6295216

Sauro, J., PhD. (2013, July 30). Rating the Severity of Usability Problems – MeasuringU. Measuring U. Retrieved November 15, 2021, from https://measuringu.com/rating-severity/

Woodford, C. (2021). Optical character recognition (OCR). explainthatstuff. Retrieved 19 December 2021, from https://www.explainthatstuff.com/how-ocr-works.html.

8. Appendix

1. Dark Pattern Survey
   https://www.surveymonkey.com/r/XGXLMXC

2. AB Test Survey
   Survey 1:
   https://docs.google.com/forms/d/e/1FAIpQLSeixQCIZsumVlAk7wJzBkjshm-P8ynHCwZAOcW8alv4eNh_jA/viewform?usp=sf_link

   Survey 2:
   https://www.wjx.cn/vj/PEMdGjb.aspx

3. Questionnaire after Cognitive Walkthrough
   Questionnaire for Version 3:
   https://www.surveymonkey.com/r/F6MKQWF

   Questionnaire for Version 4:
   https://docs.google.com/forms/d/e/1FAIpQLSeH3KwozqdBrn_iFb7BJHh8d_v1gtqpCKunxICG_JzewC4tYg/viewform?usp=sf_link

   Questionnaire for Version 5:
   https://docs.google.com/forms/d/e/1FAIpQLScWm6DMAet3rUhy0CMTkqXq0foRvveCJeaaGFDDueHQ92jTxA/viewform?usp=sf_link