Psychology

Minimising Cognitive Load on the Shell API Portal to Maximise Usability

CA

DL913 - Psychology, Information Architecture and Visual Design

Institute of Art, Design and Technology

20 February 2022

Agnieszka Przygocka

N00173057

Table of Contents

1. Introduction

APIs are at the centre of software development nowadays. They enable developers to share their code and reuse code that others build (Jin et al. 2018). The API reference documentation is a starting point for developers. It explains how to use APIs and drives adoption.

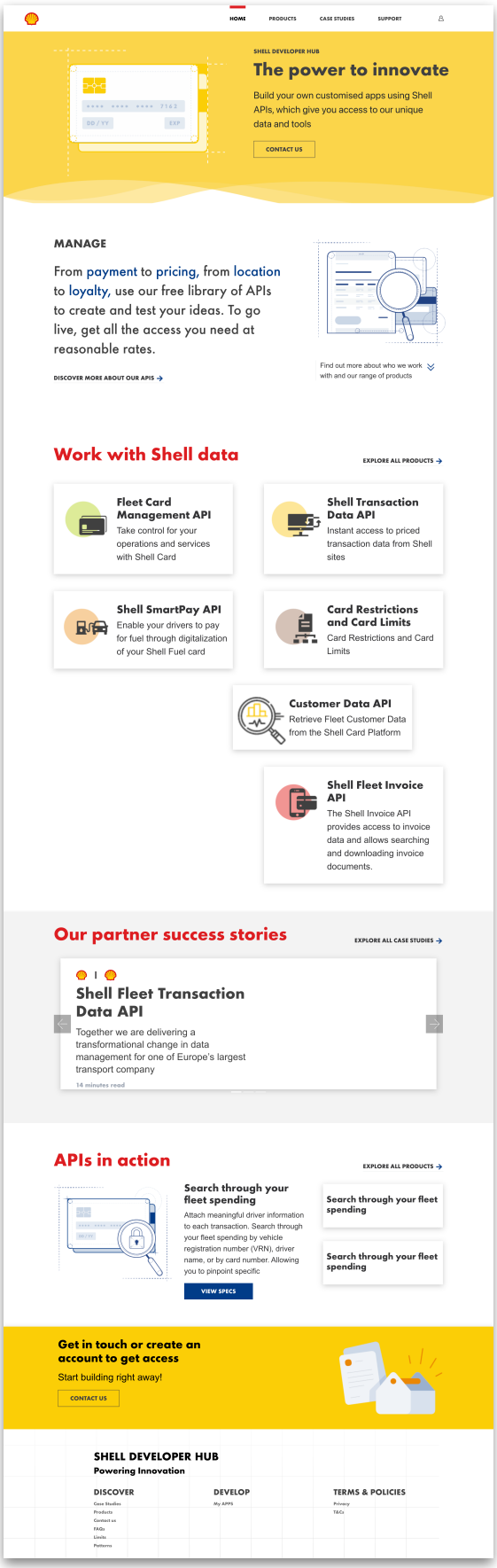
According to the 10th Nielsen's usability heuristic, "*Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.*" (Joyce, A. 2020)

Shell API Portal helps developers build custom applications utilising Shell APIs. Shell APIs allow accessing tools and data, like customer, invoice or transaction data. Each Shell API is documented in three tabs: Overview, Specification and Interaction, containing a large volume of technical information.

Our goal is to reduce cognitive load, promote self-service and improve access to the API information on Shell API Portal so the developers can faster deploy their applications.

First, we will define who our audience is, then outline cognitive psychological theories, evaluate the existing Portal, and propose a redesign.

Shell API Portal link: *https://developer.shell.com/*



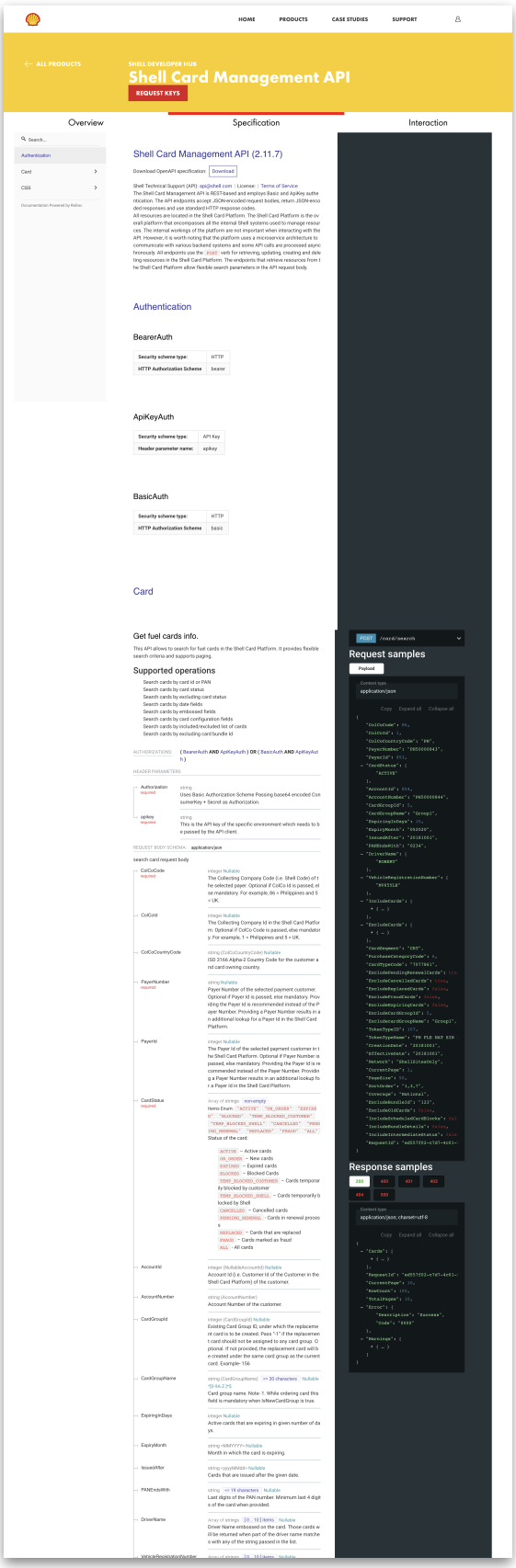


Fig.1.1 Shell API Portal in Feb 2022

2. The Audience of Shell API Portal

We assume that the audience of Shell Developer Portal will include: external API consumers, internal API consumers, internal API producers, content creators, and infrastructure providers.

In this project, we are focusing on External API Consumers. To better understand their needs and goals, we created a proto-persona.

Fig.2.1 Proto-persona - External API Consumer

Meng et al. (2018), in the paper "*How Developers Use API Documentation: An Observation Study*", investigate how developers use API documentation and what type of problems they face. He states that developers often face issues when learning new APIs due to the content and structure of the documentation. It usually doesn't match the expectations and workflows of the users. The most frequent response of the study participants was that the content was not organised to allow efficient access.

Meng et al. (2018) point out that there are two approaches when using API Docs - opportunistic and systematic. Opportunistic developers work more intuitively, they don't follow and double-check the documentation, and they don't mind making errors. They need a general overview of the API and direct access to documentation, and they search for specific information.

As a result of the study, Meng et al. (2018) suggest these key points when designing API documentation:

To enable efficient access to relevant content:

- Structure content according to API functionality (e.g. Locations, Mobility, Payments, etc.) rather than the content domain of the API (e.g. Samples, Concepts, etc.
- Present conceptual information integrated with related tasks or show usage scenarios where knowledge of these concepts is needed.
- Provide clear navigation and a powerful search function.

To facilitate initial entry into the API:

- Provide clean and working code examples demonstrating the intended use of the API (starting point for developers).
- Provide relevant background knowledge.
- Connect concepts to code by using appropriate code examples.

To support different development strategies (opportunistic and systematic):

- Enable selective access to code, and distinguish code examples from the text.
- Signal text-to-code connections (like colour-coding).
- Provide important information redundantly (e.g. comments in the code).
- Enable fast use of the API for both opportunistic and systematic developers.

3. Cognitive Psychology Theories

Cognitive psychology focuses on how people think. With cognitive psychology, we try to understand the processes in the brain, like the perception of the environment, attention to information, problem-solving, and use of language or decision-making.

Cognitive psychology can contribute to UX design quality and help create a better user experience. UX design and psychology have a common goal - to understand patterns of human cognition, the user's needs, and the decision-making process.

According to Baddeley's theory (Baddeley, 2005), two kinds of memory are "short-term" and "long-term". "Long-term" memory or working memory is responsible for decision making and behaviour, and "short-term" memory maintains information without manipulation, problem-solving and direct interaction (Smith, 1999).

Cognitive Load Theory

Cognitive load is the processing effort of the brain to make sense of the information. To hold that information for a short time, we use working memory. Our processing capacity is limited and vulnerable to overload, especially when learning complicated new concepts. It requires an extra effort from the learner to process the information (Vinney, 2020).

Cognitive load is affected by the number of units interacting with each other in the working memory. The more information the person takes in, the more likely it will not be retained. To improve learning, we need to manage cognitive load efficiently.

We can perform a visual and verbal task simultaneously, but it is more challenging to perform two visual functions simultaneously as they interfere. (Cognitive Psychology in UX - How to Use Your Short Term Memory, 2018)

According to Miller (1956), “seven is the magic number”- the amount we can remember. Miller’s Law states that our short-term (working) memory can hold simultaneously seven elements +/- 2.

Cognitive Load and API Documentation

As Hogan (2021) describes in her article, learning and doing are primary goals of the users of technical documentation. To accomplish their goals, they need to read the documentation. Reading is a cognitive behaviour in which perception, recognition, encoding, storing, and retrieving information from memory, and complex forms of reasoning and problem-solving (Hogan, 2021). Good documentation should support the cognitive processing of technical knowledge. Hogan

(2021) recommends "consider how much information or technical content is just enough for the user to receive, learn, and take action".

According to Malhotra (2021), the time to find an API function in the documentation depends on working memory. It might be difficult for developers to find the correct function without knowing its name, but they might know a partial type signature. Developers should have easy access to that information in the reference documentation.

Crichton (2020), in his white paper "Documentation Generation as Information Visualization", recommends two ways to support easy access to information about functions - scanning and searching. Scanning is just looking for the relevant content on the page; searching can be done with a Ctrl+F shortcut or via a search engine. Crichton also says that reference documentations often have too much content to scan the information, and they do not provide interfaces to search.

According to Crichton et al. (2021), programmers can hold seven variables in their working memory before making significant recall errors. Regarding program tracing, it is a highly intensive working memory task and "it is hard to remember a lot of state," and different tracing strategies showed a similar number of memory errors.

Minimising Cognitive Load

"We can't change the actual processing power of our users. What we can do is get to know their limits, and minimise their processing efforts." (Margot, 2019)

We cannot completely eliminate the cognitive load, but we can minimise it. Minimising cognitive load helps with learning and remembering things. There are three types of loads, which have different characteristics when the brain processes information.

1. Intrinsic Cognitive Load - is associated with the nature of the subject that the learner finds challenging and represents the difficulty or complexity of a certain task. It depends on the learner's prior knowledge, aptitude and capacity for learning.
Psychologist Jean Piaget came up with ideas of schemas, which are ways of organising information and knowledge (Boogaard, n.d.) - cognitive learning theory. When we are learning, our brain takes in information and processes it. Then we reorganise that information, find new explanations and alter old ones in our memory. That influences future learning.
"*only be altered by changing the nature of what is learned or by the act of learning itself.*" Sweller (1988). We need to manage the intrinsic load.

2. Extrinsic Cognitive Load - is generated by the way information is presented and by the unneeded information that interferes with the learning process. Unnecessarily confusing and

complicated learning methods cause that load, for example switching between resources.
We need to optimise extrinsic load.

3. Germane Cognitive Load - refers to the work we need to put in to construct a long-lasting store of knowledge. It involves developing patterns to organise information, and it contributes to learning. We need to promote germane load.

According to Sweller (1988), we can improve learning by reducing cognitive load in instructional design. Sweller studied the mental paths that aid in the understanding of messages. Understanding is influenced by the way content is presented to the viewer.
Working memory capacity is anywhere from two to nine items, but we can use "chunking" to increase the number of items. Chunking is combining information, and it reduces the cognitive load.
We might want to consider grouping some of the content in the documentation to help the user. We might also break up text with meaningful subheadings to help users scan the content. White space between content helps users memorise the information (Hogan, 2021).

The Principle of Least Effort

The principle of least effort was first discovered in 1894 by French philosopher Guillaume Ferrero.

"It is human nature to want the greatest outcome at the least amount of work."

People to get something done will do the least amount of work. Users want the information to be provided as effectively and efficiently as possible.

Gestalt Principle of Proximity

Don Norman (2013) discusses the importance of perceptual groupings, which originate from the Gestalt principles. Gestalt Principle of Proximity states that elements placed next to each other are seen as groups. According to Woodman et al. (2003), items are memorised together in visual memory if organised according to the Proximity Principle.
Yablonski (2015), in his Medium article "Design Principles for Reducing Cognitive Load", recommends eliminating cognitive load by always displaying choices as a group.

Colour Psychology

Bad colour combinations can affect the user experience (Dalangin, M. A. 2020). We all have colour preferences which vary depending on age, gender and culture. Hanson A. (2018) explains how we

perceive colour based on the physics of light. Colour psychology suggests that different colours evoke different moods and may affect our behaviour (Kendra Cherry, 2019).

In her article (2019), Kendra Cherry focuses on the psychology of yellow colour. It is a most visible colour and, for that reason, is attention-grabbing. Used in a small amount like on CTA buttons can be used to draw notice. The downside of using yellow is that it is difficult to read. It is the most fatiguing colour for the eye. It can lead to eyestrain and, in extreme causes vision loss. Yellow can make us feel frustrated and even angry.

4. Evaluation, Recommendations and Redesign

We identified many issues on the existing portal, generating cognitive load. For example, the documentation structure lacked a consistent system of navigation aids and required switching between pages and tabs to find information about APIs.

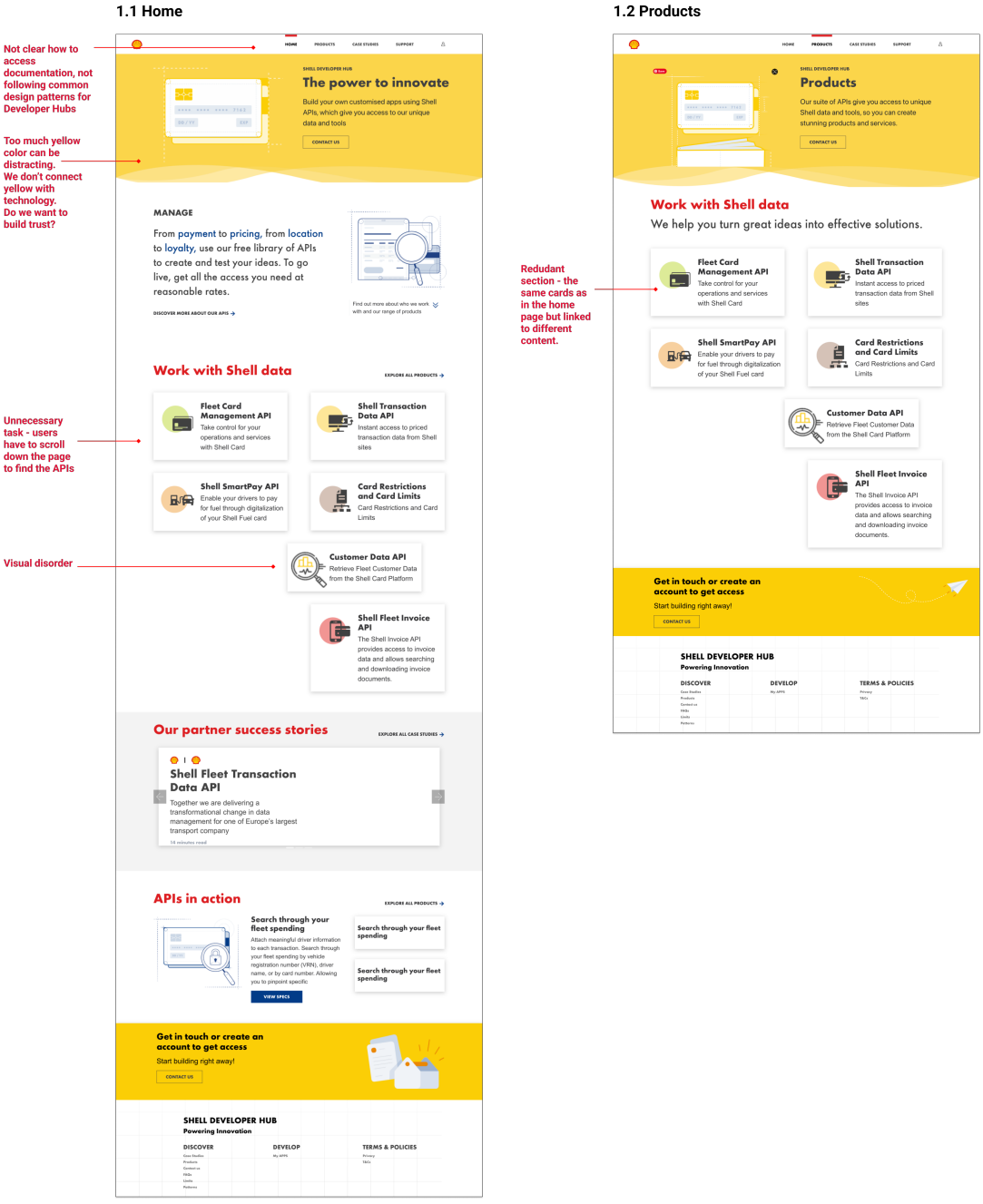


Fig.4.1 Existing Shell API Portal Home and Products Page

Figma Link: https://www.figma.com/file/je6NtxaDOV9F9jTtofJLd3/Shell-API-Portal?node-id=0%3A1

We redesign Shell API Portal with psychological principles to minimise cognitive load. The redesign takes into consideration the needs of the opportunistic and systematic users.

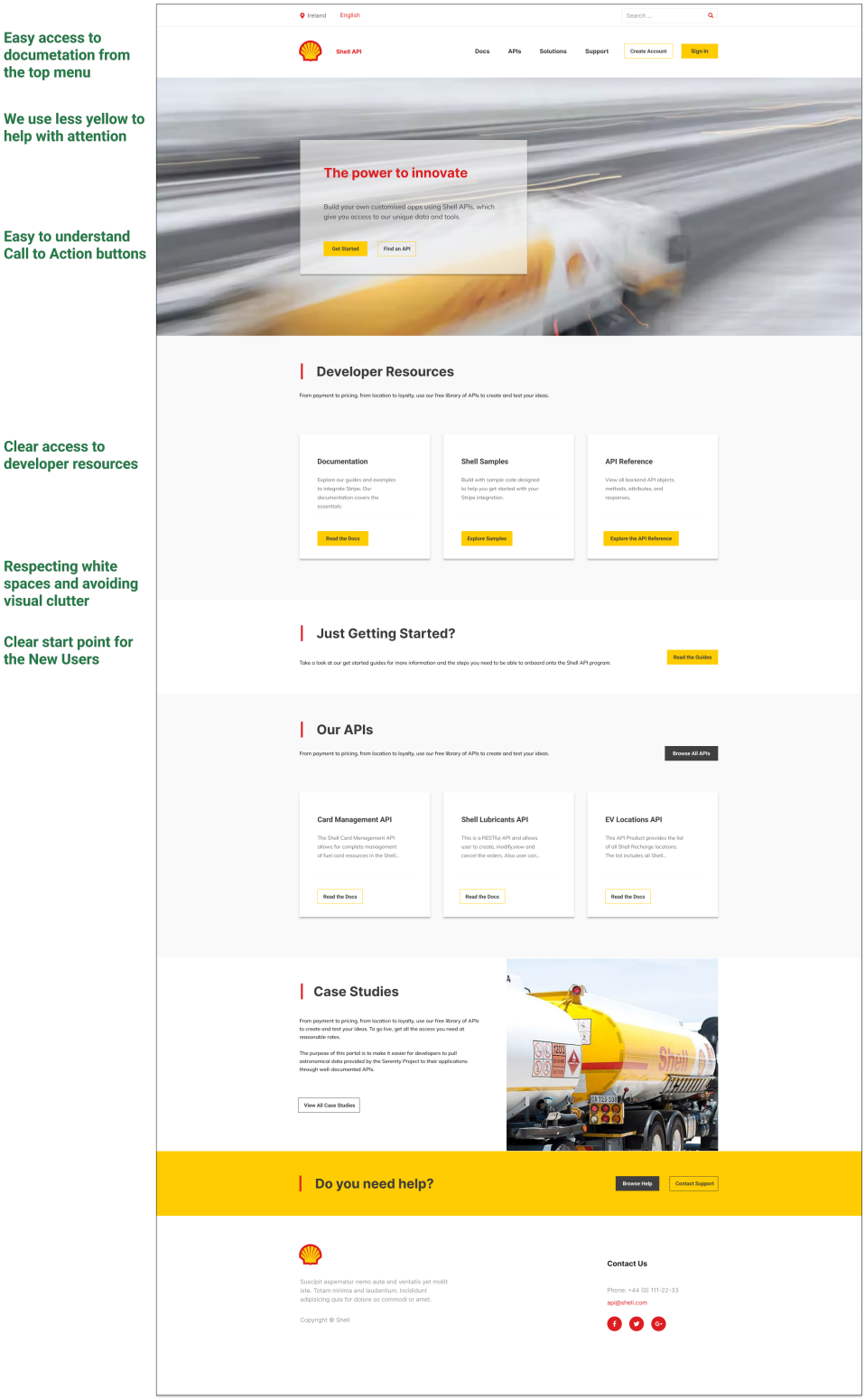


Fig.4.2 Proposed Shell API Portal Home Page

Figma Link: https://www.figma.com/file/je6NtxaDOV9F9jTtofJLd3/Shell-API-Portal?node-id=0%3A1

Know the mental model of your users

The existing website doesn't use common documentation patterns and is scattered over multiple pages and tabs. Each API has an individual page with three tabs. In one survey, developers said they "*had difficulty navigating through the myriad pages in an API document to find information,*" with one respondent explaining:

"*Fragmented documentation I find really difficult to use, where you have to have 10s of clicks through links to find the information you need, and page after page to read.*"(Johnson, 2022).

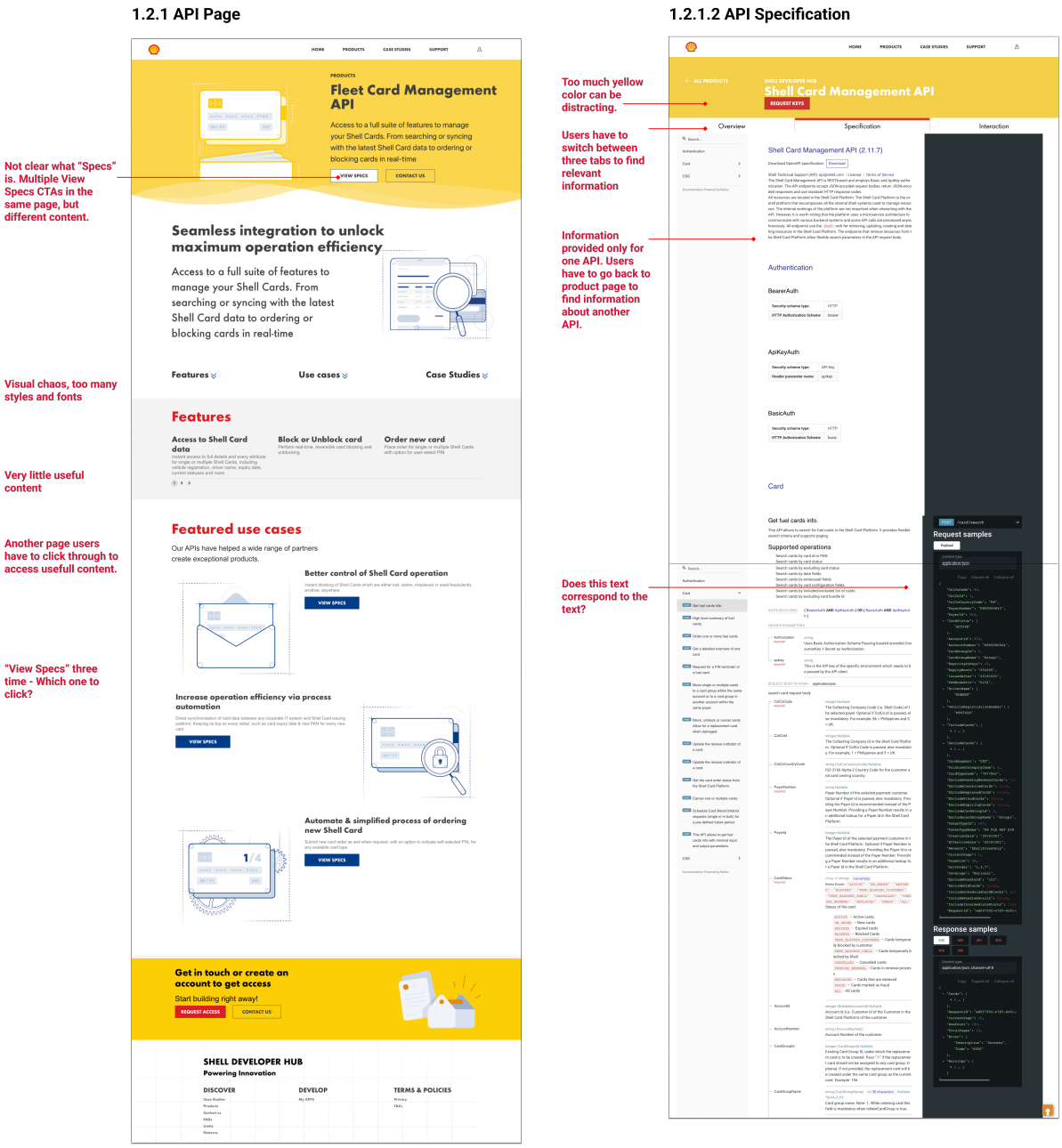


Fig.4.3 Existing Shell API Portal API Page and API Specification Page

Figma: https://www.figma.com/file/je6NtxaDOV9F9jTtofJLd3/Shell-API-Portal?node-id=0%3A1

To reduce cognitive load, we use our design patterns familiar to developers (Johnson, 2022). Commonly, API reference sites provide one integrated website to present all the information. Code examples with syntax highlighting are another typical pattern.

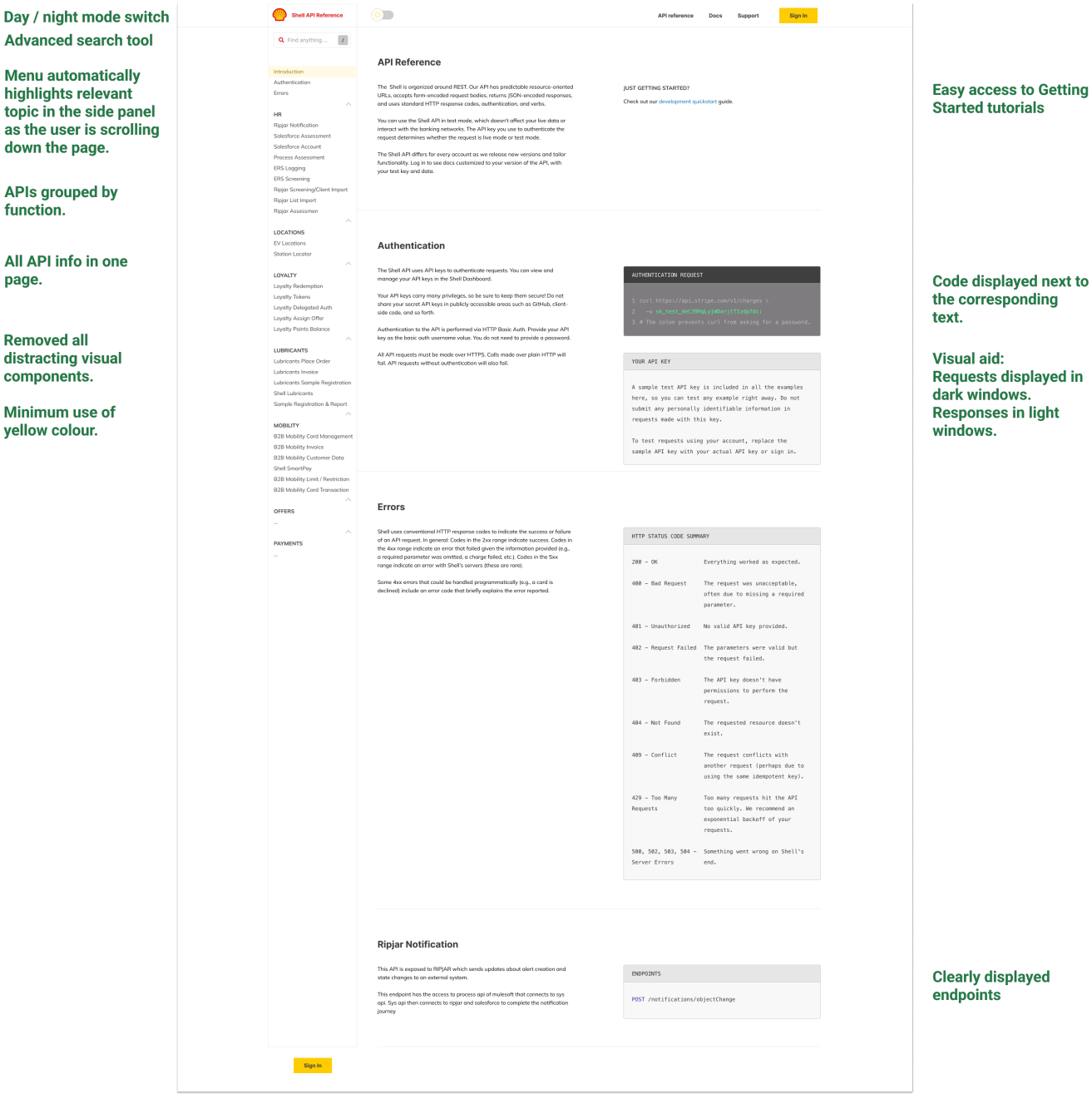


Fig.4.4 Proposed Shell API Reference Page

Figma: https://www.figma.com/file/je6NtxaDOV9F9jTtofJLd3/Shell-API-Portal?node-id=0%3A1

Enable clear and efficient access to information

Without straightforward navigation, developers cannot understand what is essential and where they are. Every new element of the UI increases its complexity. We can simplify how users understand the UI by using sufficient contrast and size, visible elements, and a clear hierarchy of importance.

We changed the website's information architecture and decided to show information about all the APIs on one page. Users don't have to leave the page to find information about individual APIs.

As a result, we had to redesign the side menu. The new side navigation enlists all the APIs grouped by functionality - Locations, Mobility, Payments, etc. (Figure 4.4).

We removed tabs from the API specification page to simplify access to information.

The existing search tool doesn't offer advanced options. We proposed an advanced search widget to enable users to find any information quickly.

Guide users by facilitating initial entry into the API

We provide code concepts and API endpoints just next to the corresponding text in the additional column. To improve visual clarity, the requests are shown in dark grey code blocks and responses in light grey (Figure 4.4).

Support different development strategies - opportunistic and systematic.

Both groups of developers - opportunistic and systematic can find information on the new portal quickly. They can easily distinguish code examples from the text. Opportunistic users can scroll the page or use the search by switching the sites.

Keep it simple / Avoid visual clutter / Make it consistent

Design should be simple. We removed all the unnecessary pages, repeated content and components from the Portal.

We grouped APIs that are similar in functionality or content that is similar. Consistent shapes, colours, or white space automatically manipulate the brain to relate things as a unit.

In addition, we introduced consistent typography to improve readability and help visually scan documentation.

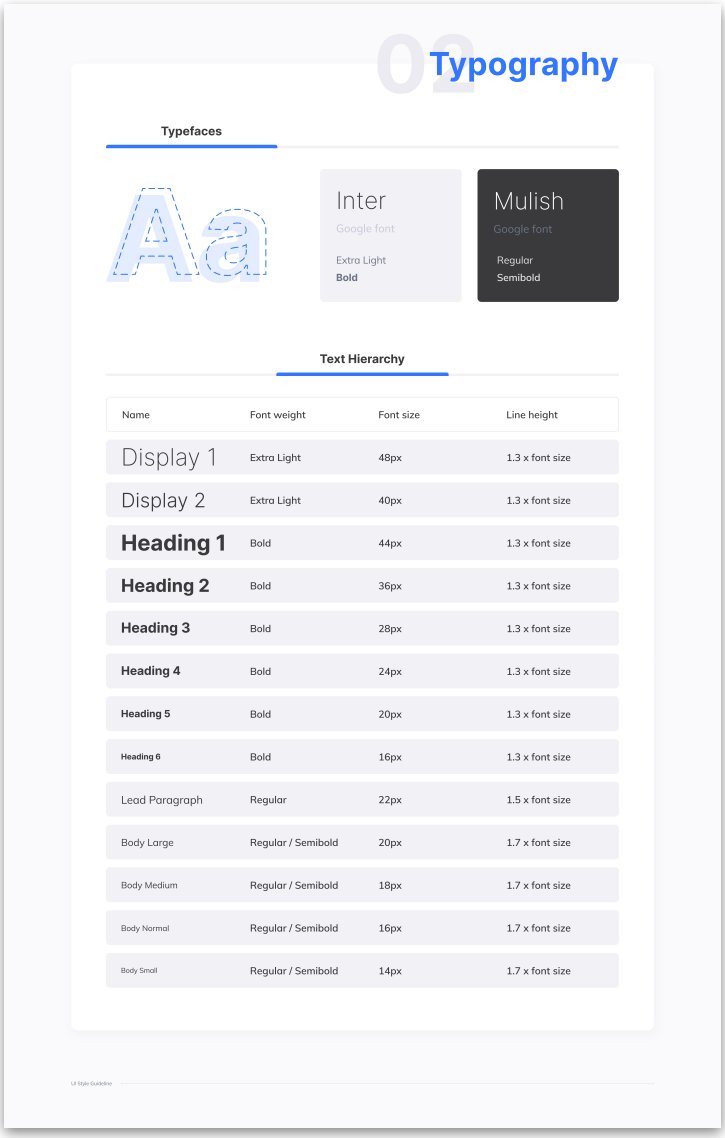


Fig.4.5 Proposed Typography

Colour and Contrast / Psychology of Yellow

We proposed a colour system complying with WCAG (W3C, 2016) accessibility standards.

We significantly reduced the use of yellow throughout the Portal. The goal was to improve attention and make it easier to read.

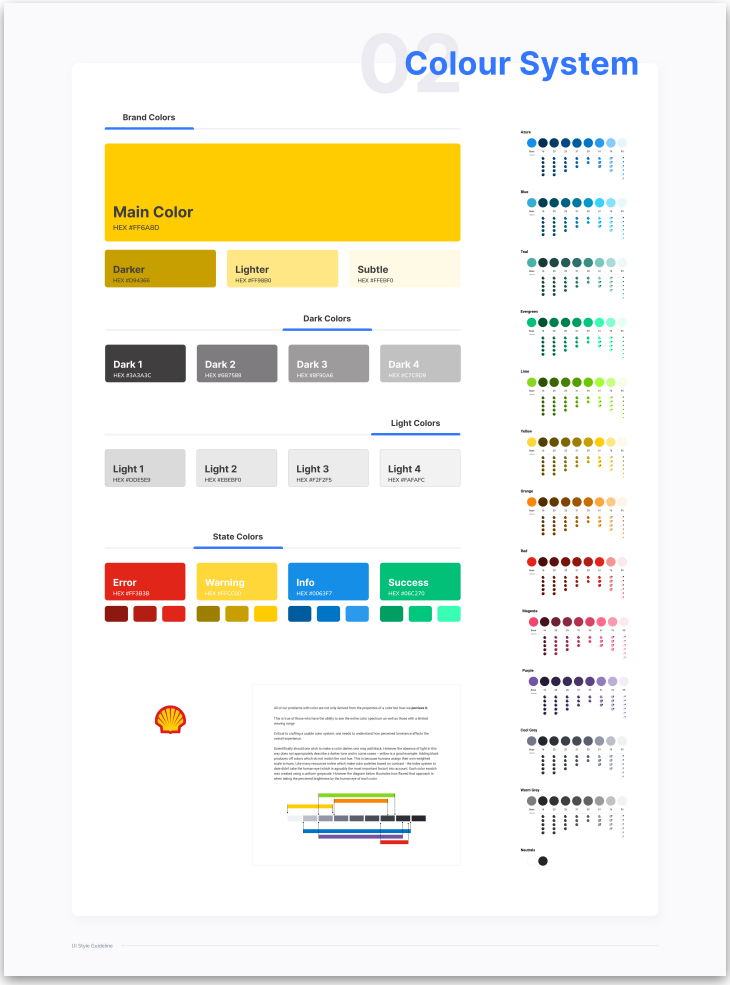


Fig.3.1 Proposed Colour System

Design File - Figma: *https://www.figma.com/file/je6NtxaDOV9F9jTtofJLd3/Shell-API-Portal?node-id=0%3A1*

Prototype: *https://www.figma.com/proto/je6NtxaDOV9F9jTtofJLd3/Shell-API-Portal?page-id=475%3A1022&node-id=475%3A1023&viewport=327%2C549%2C0.25&scaling=min-zoom&starting-point-node-id=475%3A1023*

References

Baddeley, A. (2005). Human memory : theory and practice. Psychology Press.

Birren, F. (2013). Color psychology and color therapy : a factual study of the influence of color on human life. Literary Licensing.

Boogaard, K. (n.d.). Your Guide to Learning Theories (and Why They're Important). GoSkills.com. https://www.goskills.com/Resources/Learning-theories

Cognitive load in programming: why the brain is the bottleneck | Alberto Fernández-Capel. (n.d.). Afcapel.com. from https://afcapel.com/articles/2019/01/18/cognitive-load-in-programming.html

Cognitive Psychology in UX - How to Use Your Short Term Memory. (2018, February 6). UX Studio. https://uxstudioteam.com/ux-blog/cognitive-psychology-ux/

Crichton, W. (2020). Documentation Generation as Information Visualization. http://reports-archive.adm.cs.cmu.edu/anon/isr2020/CMU-ISR-20-115E.pdf

Crichton, W., Agrawala, M., & Hanrahan, P. (2021). The Role of Working Memory in Program Tracing. Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. https://doi.org/10.1145/3411764.3445257

Dalangin, M. A. (2020). What is Color Psychology In UX design? | Userpeek.com. Userpeek.com. https://userpeek.com/blog/what-is-color-psychology-in-ux/

Gruenbaum, P. (2012). Using Color in API Documentation. ProgrammableWeb. https://www.programmableweb.com/news/using-color-api-documentation/how-to/2012/09/28

Guest Author (2019, February 24). Cognitive Load Theory: Definition, Types, And Applications For Learning [Guest Post]. Cognition Today. https://cognitiontoday.com/cognitive-load-theory-definition-types-and-applications-for-learning-guest-post/

Hallock, J. (2003) Colour Assignment. http://www.joehallock.com/?page_id=1281

Hanson, A. (2018) (The Physics and Psychology of Colour - with Andrew Hanson. (n.d.). The Royal Institution., https://youtu.be/af78RPi6ayE

Hogan, D. (2021, December 8). How to Write Technical Documentation. MadCap Software. https://www.madcapsoftware.com/blog/how-to-write-technical-documentation-using-madcap-flare/

Jin, B., Sahni, S., & Shevat, A. (2018). Designing web apis building apis that developers love. O'Reilly Media, Inc, USA.

Johnson, T. Design patterns with API doc sites. Documenting APIs. https://idratherbewriting.com/learnapidoc/pubapis_design_patterns.html

Joyce, A. (2020, December 13). Help and Documentation: The 10th Usability Heuristic. Nielsen Norman Group. https://www.nngroup.com/articles/help-and-documentation/

Kendra Cherry. (2019). The Impact of the Color Yellow on Your Mood. Verywell Mind. https://www.verywellmind.com/the-color-psychology-of-yellow-2795823

Krug, S. (2014). Don't make me think! Web Usability: Das intuitive Web. Wachtendonk Mitp.

Malhotra, A. T. (2021). Improving API Reference Documentation: A Cognitivist Approach to Design and Evaluation. Dash.harvard.edu. https://dash.harvard.edu/handle/1/37368552

Margot, A. (2019, April 11). Cognitive psychology in UX: Minimising the cognitive load. Medium. https://medium.com/design-signals/cognitive-psychology-in-ux-minimising-the-cognitive-load-d97ad8e3115b

Meng, M., Steinhardt, S., & Schubert, A. (2019). How developers use API documentation. Communication Design Quarterly, 7(2), 40–49. https://doi.org/10.1145/3358931.3358937 https://www.researchgate.net/publication/330741779_How_Developers_Use_API_Documentation_An_Observation_Study

Norman, D. A. (2013). The Design of Everyday Things. Mit Press. (Original work published 1988)

Smith, E. (1999). Storage and Executive Processes in the Frontal Lobes. Science. https://www.academia.edu/5630581/Storage_and_Executive_Processes_in_the_Frontal_Lobes

Stack Overflow. (2021, May). Stack Overflow Developer Survey 2021. Stack Overflow. https://insights.stackoverflow.com/survey/2021

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. Cognitive Science. https://www.academia.edu/1953251/Cognitive_load_during_problem_solving_Effects_on_learning

Vinney, C. (2020, March 21). Information Processing Theory: Definition and Examples. ThoughtCo. https://www.thoughtco.com/information-processing-theory-definition-and-examples-4797966

W3C. (2016). Understanding Success Criterion 1.4.3 | Understanding WCAG 2.0. W3.org. https://www.w3.org/TR/UNDERSTANDING-WCAG20/visual-audio-contrast-contrast.html

Weinschenk, S. M. (2011). 100 Things Every Designer Needs To Know About People. New Riders.

Whitenton, K. (2013, December 22). Minimize Cognitive Load to Maximize Usability. Nielsen Norman Group. https://www.nngroup.com/articles/minimize-cognitive-load/

Woodman, G. F., Vecera, S. P., & Luck, S. J. (2003). Perceptual organization influences visual working memory. Psychonomic Bulletin & Review, 10(1), 80–87. https://doi.org/10.3758/bf03196470

Yablonski, J. (2019, April 13). Design Principles for Reducing Cognitive Load. Medium. https://blog.prototypr.io/design-principles-for-reducing-cognitive-load-84e82ca61abd

Yablonski, J. (2020). Laws of UX : design principles for persuasive and ethical products. O'reilly Media, Inc, Usa.

Yocco, V. S. (2016). Design for the mind : seven psychological principles of persuasive design. Manning.