A NEXT-GENERATION WEB-BASED SYSTEM FOR MANAGING AND EDITING COMPLEX DIGITAL DOCUMENTS

P. SAI PRANATHI [1]| Dr. S SRI LAKSHMI PARVATHI[2]|

[1]PG SCHOLAR, KKR&KSR Institute of Technology & Sciences, AP, India

[2]PROFESSOR, KKR&KSR Institute of Technology & Sciences, AP, India

ABSTRACT: This study details the processes and procedures that went into creating the web-based Ultra Text Editor, a tool for handling and editing complicated documents with granular control over text. The Ultra Text Editor is a state-of-the-art online app that uses cutting-edge web technologies like React and the react-dnd package. It has a streamlined user interface, powerful text editing tools, and rapid navigation across content hierarchies. Powerful document management systems that can handle complex content structures at the paragraph, phrase, and word levels are in high demand, and this one satisfies that need. This paper provides a comprehensive analysis of this editor's Navigation Panel and Hierarchical Panel, two of its key components.

Key words: structured authoring, text editor, drag-and-drop, React, rich text, HCI, usability.

1. INTRODUCTION

A text editor's principal function is to facilitate the editing of plain text inside a computer. You could see this content in several places, such source code, log files, documentation, and more. The operating system often comes with a basic editor, such as "Windows Notepad," and users may use it for free. Enhanced with cutting-edge features, Ultra Text is a classic text editor with a twist. Ultra Text Editor was born out of a desire for a versatile tool that strikes a good mix between ease of use and robust functionality. Editors with a lot of features might be intimidating to first timers, while those with less features tend to be shallow. Ultra Text Editor intends to unite these two groups by providing an easy-to-use interface that both novices and advanced users will love. Designed to be user-friendly and efficient, it includes essential features including a navigation panel, a hierarchy panel, split editing, and drag-and-drop support. Many occupations rely on text editing, including those of authors, programmers, researchers, and educators. More complex editing tools are required to keep up with the ever-growing volume of digital information. Present editors have their uses, but they aren't always user-friendly, flexible, or up-to-date. By

combining cutting-edge technology with an intuitive UI, Ultra Text Editor is able to circumvent these problems while simultaneously increasing productivity and inspiration.

2. LITERATURE SURVEY

It is amazing to see how text editors have evolved. In the early 1940s, punched cards were used for editing and storing plain text [1]. The development of "The Jacquard Loom" by Joseph-Marie Jacquard in 1801 gave rise to this idea. Text editors have been essential tools for programmers ever since computers first came into being. Making and editing plain text files has always been their primary function.

The Unix programs Ed, developed by Ken Thompson in 1969, and Vi, published by Bill Joy in 1976, are two classic examples [2]. These editors relied on the command line interface and made text editing possible with the use of shortcut keys and commands. Even though they were simple, the tools that developers and system administrators used were very efficient. Better, easier-to-understand text editors were available as graphical user interfaces gained popularity. For instance, there is TextEdit, which debuted for Mac OS X in 2001, and Notepad, which debuted with Windows in 1985[1].

By providing easy-to-memorize functionality like cut, copy, paste, and search, these editors made text editing accessible to non-programmers. The need for more sophisticated text editors arose in response to user demands. Among them were capabilities for integrating with version control systems, code completion, and syntax highlighting. Syntax highlighting made the code easier to understand by drawing attention to certain sections. By recommending appropriate code alternatives, code completion helped cut down on mistakes and saved time. Developers were able to streamline their workflow using version control integration, which enabled them to manage project changes immediately inside the editor.

Some examples of popular current editors include Atom[11], Visual Studio Code[10], and Sublime Text [10].

Since its 2008 release, Sublime Text has gained a reputation for being both fast and flexible, thanks to its extensive plugin system. With its extensive set of features including an integrated terminal, debugging tools, and an extensive library of extensions Visual Studio Code, which was introduced by Microsoft in 2015, has swiftly become a favorite among developers. Atom, created by GitHub in 2014, is a very customisable and multi-language programming environment.

Since the beginning of word processing, rich text editing has been an essential component. When it comes to text editors, users have come to anticipate certain features, such as the ability to style text according to user preferences and the smooth handling of complicated documents with several levels of headings and subheadings. Documents saved in HTML or XML often have their underlying structure simplified by these editors. Users may format and change text without understanding or interacting with the underlying code thanks to this abstraction. This allows users to concentrate on writing while the technical formatting takes place automatically. the third [4].

3. METHODOLOGY

The Ultra Text Editor's main user interface areas are: To make sure Ultra Text Editor was both technically solid and easy to use, the development process included system design, implementation, and user testing. The three main modules that made up the system are the Main Editor, which offers a rich-text interface using React-Quill; the Navigation Panel, which allows visibility and reordering of sections at the section level by dynamically extracting document headings (H1-H6); and the Hierarchical Panel, which represents text as a nested structure of paragraphs, sentences, phrases, and words for fine-grained drag-and-drop manipulation. Thanks to its modular design, features may run independently while yet being in sync with one another thanks to shared application state.

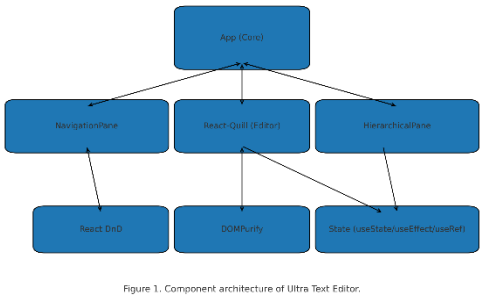

Fig 1: System Architecture

Secure content sanitization was accomplished using DOMPurify, and drag-and-drop operations were made possible using the React framework with React DnD. In order to provide real-time interaction across panels, React hooks were used to handle dynamic state changes. While the Hierarchical Panel used tokenization to project material into several levels of granularity, the Navigation Panel depended on sanitized HTML parsing.

The efficacy and usability were assessed by a formative user research that included six individuals. All the way from sections to paragraphs to sentences to phrases and words, participants were given organized tasks to accomplish. Observation, post-task surveys, and Likert-scale evaluations were used to gather data. Qualitative analysis was

performed on the findings in order to determine their strengths, areas for improvement in terms of usability, and potential areas for future growth.

4. DESIGN AND CONSTRUCTION

Because of its effectiveness in handling dynamic user interfaces, component-based structure, and modularity, the React framework was chosen to create the Ultra Text Editor. Essential formatting capabilities like bold, italics, lists, and headers are provided by React-Quill, a wrapper over the Quill rich-text editor, which is integrated at its core by the editor. The Navigation Panel and the Hierarchical Panel were created to provide capabilities beyond traditional editors.

To make use of the Navigation Panel, the editor's HTML content is parsed and cleaned up using DOM Purify. Then, structural components like headers (H1-H6) are extracted. To facilitate the creation of a table of contents in real-time, they are saved in an array that also contains IDs, tags, and locations. Thanks to React DnD, users can simply drag and drop pieces to rearrange or remove them; any changes made in the editor will be updated immediately.

Fig 2: Navigation Panel

The Hierarchical Panel tokenizes content into progressively smaller units: paragraphs, sentences, phrases, and words. Each unit is displayed in a hierarchical tree-like structure that users can manipulate directly via drag-and-drop.

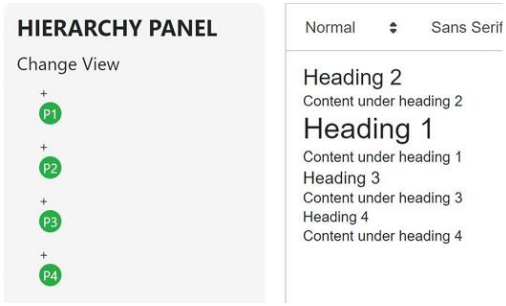

Fig 3: Hierarchical Panel

React hooks such as use State, use Effect, and use Ref were employed to manage editor state, synchronize updates, and maintain smooth interactions. This architecture ensures that every edit remains consistent across components, enabling robust, fine-grained text manipulation.

5 RESULTS AND DISCUSSION

Ultra Text Editor showed great promise for granular editing and excellent usability at the section level in the examination.

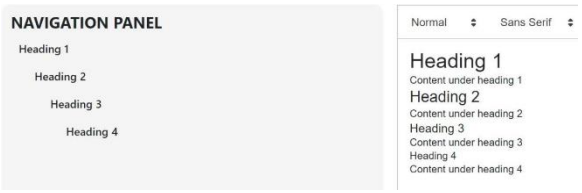

Fig 4: Navigation Panel

Figure 4 shows that the Navigation Panel automatically created a table of contents, which made it easy for participants to rearrange and remove parts. With praise for its simplicity and effectiveness in handling lengthy papers, all six participants gave this feature a good rating.

Fig 5: Hierarchical Panel

Figure 6, which depicts the Hierarchical Panel, added another layer of intricacy. Editing at the phrase and word levels was seen as mentally taxing, in contrast to rearranging paragraphs and sentences, which participants found useful for improving structure. Users were sometimes overwhelmed by the thick hierarchical view, which brought attention to the need for gradual disclosure and collapsible layers. In spite of these obstacles, participants still saw granular editing's unique potential for revision assignments.

5. CONCLUSION

An Ultra Text Editor with sophisticated features that improve the user experience was created; it is well-structured and very efficient. To begin with, when it comes to giving you more control over text editing, you can obtain roughly half of what you need from the Navigation Panel, the rich text editor, and Split View combined. However, the Hierarchical Panel fills in the blanks, allowing users full command as they draft papers with ease. This program takes text editing to a whole new level with all the extra tools it offers. According to user feedback from testing, the newly added capabilities are well-received, and eventually developed and implemented on a broad scale, this project will revolutionize text editing.

REFERENCES

1. “The Evolution of Text Editors and Word Processors”. https://www.linkedin.com/pulse/evolution-text-editors-word-processors-miguel-rivas-perez-ilwbc/

2. https://en.wikipedia.org/wiki/Text_editor

3. Craig A. Finseth, “The Craft of Text Editing: Emacs for the Modern World”, 1991.

4. David W. Embley, George Nagy, “Behavioral Aspects of Text Editors”.

5. A. Srivastava, S. Bhardwaj and S. Saraswat, "SCRUM model for agile methodology," 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 2017.

6. A. R. Chaudhari and S. D. Joshi, "Study of effect of Agile software development Methodology on Software Development Process," 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2021.

7. “Built-in React Hooks – React,” react.dev. https://react.dev/reference/react/hooks.

8. “react-quill,” npm, Aug. 03, 2022. https://www.npmjs.com/package/react-quill.

9. “Sublime Text”, Sublime, https://www.sublimetext.com/.

10. “VS Code”, Microsoft, https://code.visualstudio.com/docs.

11. “Atom”, GitHub, https://github.blog/news-insights/product-news/sunsetting-atom/.