

# **Markdown as a Writing and Production Environment for Academic Theses**

*A Criteria-Based Evaluation of mdedit.io Compared with Word and LaTeX*

**Max Example**

Master's thesis · Example University · 2026-05-10

## Abstract

Academic writing today operates between three dominant logics: the word-processing logic of Microsoft Word, the typesetting logic of LaTeX, and the structure-oriented logic of Markdown-based workflows. This thesis examines under which conditions a browser-based Markdown editor, using mdedit.io as the case, can serve as the primary production environment for academic theses. The starting point is the observation that the suitability of writing tools in academic practice is often judged through habit, disciplinary culture, or tool myths, while an explicit evaluation model for structural transparency, source work, export stability, and supervision compatibility is often missing.

Methodologically, the thesis follows a criteria-guided artifact analysis with a design-oriented evaluation logic. Based on literature on academic writing, Markdown, Pandoc, and design-science evaluation approaches, five evaluation dimensions are derived: structural transparency, source and citation reliability, layout and export stability, review and supervision compatibility, and governance and operational viability (Booth et al. 2016; Swales and Feak 2012; Hevner et al. 2004; Wieringa 2014). These dimensions are first applied to Word, LaTeX, and Markdown-based workflows (Abschnitt 3.5) and then used to evaluate mdedit.io through official documentation, institutional requirements, and a reproducible reference artifact (Abschnitt 5).

The results show that Markdown, when combined with specification, metadata, and converters, can function as a robust intermediate format for long-form academic documents (Gruber 2004; CommonMark Contributors 2026; MacFarlane 2025). At the same time, Markdown gains special relevance in the context of generative AI as a structurally stable working and exchange format: headings, lists, tables, code blocks, and citation markers remain readable for humans while also staying addressable for models. For mdedit.io this yields a strong fit in the structuring and drafting phase, in the navigation of large documents through outline and tree view, in AI-supported document work, in early layout control, and in controlled PDF and DOCX export paths (Hertel 2026c, 2026b). It is also relevant that the scientific export path integrates current mdedit extensions for bibliography handling, citeproc, and reference sections (Hertel 2026d). Limits remain in note-style footnotes, strongly standardized special templates, and semantically rich reference systems. The thesis therefore does not claim an empirical proof of user effectiveness, but it does provide a criteria-based decision model and a testable reference artifact for evaluating thesis-ready Markdown workflows.

## Keywords

Markdown, academic thesis, mdedit.io, design science, LaTeX, Microsoft Word, Pandoc, AI applications, writing tools, export workflow

# Inhaltsverzeichnis

---

List of Figures .....	5
List of Tables .....	5
1. Introduction .....	6
1.1 Research question .....	6
1.2 Working theses .....	7
1.3 Methodological approach .....	7
1.4 Structure of the thesis .....	7
2. Markdown: concept, origin, and development .....	8
2.1 What Markdown is and why it emerged .....	8
2.2 Why Markdown did not remain at its origin .....	8
2.3 Markdown in the wider ecosystem: Pandoc as a bridge .....	8
2.4 Markdown as a working format in AI applications .....	8
3. Options for producing academic theses .....	10
3.1 Special requirements of academic documents .....	10
3.2 Microsoft Word .....	10
3.3 LaTeX .....	10
3.4 Markdown-based workflows .....	11
3.5 Comparison of Word, LaTeX, and Markdown .....	11
4. mdedit.io as the case under investigation .....	12
4.1 Markdown as an academic intermediate format .....	12
4.2 Functional profile of the case .....	12
4.3 Preliminary placement within the evaluation grid .....	12
4.4 Anticipated limits of the case .....	13
5. Criteria-based evaluation of mdedit.io .....	14
5.1 Evaluation logic and the investigation artifact .....	14
5.2 Structural transparency and the writing process .....	14
5.3 Source apparatus and scholarly integrity .....	16
5.4 Layout, typesetting, and export stability .....	17
5.5 Review, supervision, and collaboration capability .....	17
5.6 Exclusion criteria and edge cases .....	17
5.7 Stress test against demanding university profiles .....	18

5.8 Design implications for institutional profiles .....	19
5.9 Reasons not to choose mdedit.io as the tool for one’s own master’s thesis .....	20
6. Governance, rights, and operational viability .....	22
6.1 Licensing and rights .....	22
6.2 Access model and practical usability .....	22
6.3 Availability and operation .....	22
7. Discussion .....	23
8. Conclusion .....	24
References .....	25
Appendix A. Reproducible reference artifact for layout and feature testing .....	26
A.1 Figure specimen .....	26
A.2 Scientific table .....	27
A.3 Compact table .....	27
A.4 Mathematics, page-footnote, and endnote specimen .....	27
A.5 Multi-column specimen .....	27
A.6 Unicode, cross-reference, and listing specimen .....	28
A.7 Figure and table cross-references (SPR-04) .....	29

## List of Figures

<b>Abb. 1:</b> Neutral image artifact for caption, list, and print tests .....	26
<b>Abb. 2:</b> Export path from Markdown to PDF and DOCX through the mdedit converter .....	29

## List of Tables

<b>Tab. 1:</b> Frequently used AI-supported chat systems and their typical working formats. ....	9
<b>Tab. 2:</b> Comparison of central tool logics for academic theses. ....	11
<b>Tab. 3:</b> Evaluation dimensions of the analysis. ....	14
<b>Tab. 4:</b> Summary evaluation of mdedit.io within the requirement grid. ....	18
<b>Tab. 5:</b> Criteria set for the final quality check of an academic thesis in mdedit.io. ....	27
<b>Tab. 6:</b> Typical tasks before submission. ....	27
<b>Tab. 7:</b> Tool comparison for the academic export path (cross-reference test). ....	29

# 1. Introduction

Academic theses are no longer merely text products, but organized writing and revision processes. A master's thesis emerges over weeks or months in an interplay of material collection, structuring, drafting, feedback, language revision, formal control, and final export. The quality of the chosen tool does not determine the academic quality of the argument itself, but it does determine friction, visibility of structure, and the stability of the production process. That is why the question of suitable writing environments is not a side issue but part of academic work organization.

In practice, Microsoft Word and LaTeX still dominate. Word is institutionally widespread, familiar in review processes, and firmly embedded in university procedures through template cultures. LaTeX, in turn, remains the reference point in mathematics, computer science, and science-adjacent disciplines for typographically demanding and formally regulated documents (LaTeX Project 2026). Markdown-based workflows, by contrast, are often still perceived as informal, technical, or blog-like. That perception is too narrow, because modern Markdown ecosystems, through specifications such as CommonMark, converters such as Pandoc, and editors built on top of them, deliver far more than pure note-taking or web-authoring scenarios (CommonMark Contributors 2026; MacFarlane 2025).

The rise of generative AI changes the assessment of Markdown yet again. Many AI applications do not operate at the level of visible page formatting, but on linear, segmentable text. This is where Markdown is strong: headings, lists, tables, code blocks, and metadata remain explicit, copyable, and transformable in chat contexts. For academic writing environments this matters because AI support then works not on screenshots or local visual formatting, but on the documented structural core of the document.

Against this backdrop, the thesis investigates which role Markdown, and mdedit.io in particular, can play for academic writing. The focus is not the claim that one tool can replace all competing production logics. The question is rather for which requirement profiles of academic theses a browser-based Markdown workflow is methodologically defensible, technically stable, and institutionally compatible. In addition, the thesis carries a reproducible reference artifact that makes the layout and export elements relevant to long-form academic documents practically testable.

## 1.1 Research question

The guiding research question is: Under which conditions can a browser-based Markdown editor, using mdedit.io as the case, serve as the primary production environment for academic theses, and under which formal boundary conditions do Word or LaTeX remain preferable?

Three sub-questions structure the answer:

1. Which requirements are constitutive for academic theses and therefore decisive for tool choice?
2. Which production logics distinguish Word, LaTeX, and Markdown-based workflows in relation to these requirements?
3. At what point does mdedit.io prove to be a viable primary workflow, and where do exclusion criteria or institutional special cases begin?

## 1.2 Working theses

The thesis is based on four working theses:

1. Markdown is no longer merely a simple web syntax, but a serious intermediate format for structured long documents, provided that specification, metadata, and export tools work together. The same property also makes it a highly compatible working format for AI-supported editorial and transformation processes (Gruber 2004; CommonMark Contributors 2026; MacFarlane 2025).
2. The suitability of a tool for master's theses is determined less by isolated typography than by structural transparency, source integrity, review capability, and reproducible export paths.
3. mdedit.io is especially strong where long-form structure, early layout control, controlled multi-export, document-near AI support, and browser-based collaboration are central (Hertel 2026c).
4. mdedit.io remains limited where semantically rich reference systems, note-style footnotes, or rigid university templates are mandatory (Hertel 2026e, 2026f).

## 1.3 Methodological approach

Methodologically, the thesis follows a criteria-guided artifact analysis in the sense of design-oriented research (Hevner et al. 2004; Wieringa 2014). In the first step, a catalogue of requirements for academic theses is reconstructed from the literature on academic writing and from institutional citation guidance (Booth et al. 2016; Swales and Feak 2012; University of Reading Library 2026). In the second step, the three dominant tool logics, Word, LaTeX, and Markdown, are compared against these requirements. In the third step, mdedit.io is examined as a case through official product documentation, science-adjacent project concepts, and the reference artifact carried within this thesis (Hertel 2026c, 2026e, 2026f).

The evaluation is conducted along five dimensions: structural transparency, source and citation reliability, layout and export stability, review and supervision compatibility, and governance and operational viability. In this logic, a tool counts as thesis-ready if it can carry the production chain of a thesis across these dimensions without methodologically critical breaks. The thesis does not claim generalization from user counts or experiment data, but an analytically traceable suitability assessment.

## 1.4 Structure of the thesis

Chapter 2 reconstructs the origin and development of Markdown as an academically relevant intermediate format and adds its special role in AI applications. Chapter 3 derives the evaluation framework for academic writing tools from the literature and institutional requirements and positions Word, LaTeX, and Markdown comparatively. Chapter 4 defines mdedit.io as the case under investigation. Chapter 5 performs the criteria-based evaluation and reflects the findings against three demanding university profiles. Chapter 6 addresses governance, legal, and operational issues. Chapter 7 discusses contribution and limits, and Chapter 8 answers the research question. The appendix serves as a reproducible reference artifact for layout- and export-critical document elements.

## **2. Markdown: concept, origin, and development**

### **2.1 What Markdown is and why it emerged**

Markdown was introduced in 2004 by John Gruber as a text-to-HTML tool. The core idea is that a document should already be readable as raw text and only in a second step be transformed into structurally valid HTML (Gruber 2004). This double logic remains decisive: Markdown is both a syntax and a conversion idea. Its original purpose was not academic publishing, but readable writing for the web. That is precisely where one of its enduring strengths lies. Markdown separates content, structure, and later presentation more clearly than classic WYSIWYG word processing.

The normative point lies in the goal of readability. While HTML tags can visually overload raw text, a Markdown document should remain understandable as structured text even without rendering (Gruber 2004). This property makes Markdown interesting for academic processes because in early stages scholarly writing depends less on perfect formatting than on an intelligible outline.

### **2.2 Why Markdown did not remain at its origin**

The openness of the original Markdown description was both success and problem. Because the syntax was not originally specified unambiguously, numerous implementations developed with diverging behavior. CommonMark explicitly describes this historical situation as a problem of divergent parsers and missing test suites (CommonMark Contributors 2026). For academic documents, this divergence is critical because reproducibility and portability are central requirements.

CommonMark therefore pushed Markdown toward a more formal, testable, and interoperable specification (CommonMark Contributors 2026). The development of Markdown can thus be read as a shift from a pragmatic web-writing practice to a standardizable format for structured documents. That step is central to academic work because it creates the basis for reliable converters and stable rendering paths.

### **2.3 Markdown in the wider ecosystem: Pandoc as a bridge**

Pandoc marks the second decisive stage of development. While Gruber's Markdown primarily targeted HTML, Pandoc turns Markdown into a universal intermediate format. By its own description, Pandoc converts between numerous markup, document, and office formats, including Markdown, DOCX, LaTeX, and several PDF paths (MacFarlane 2025). Pandoc also supports metadata, tables, footnotes, mathematics, and automatic citations and bibliographies (MacFarlane 2025).

For academic theses, this is highly significant. Only the combination of specified syntax, metadata, and converter enables a workflow in which raw text, scholarly structure, and output media are systematically coupled. This does not make Markdown automatically the best tool for every thesis. But it does make it a robust alternative for authors who value structure and portability more than permanent visible formatting.

### **2.4 Markdown as a working format in AI applications**

Generative AI applications do not fundamentally operate on page geometry, but on tokenized text, semantic segments, and explicit structure markers. That is precisely why Markdown has a special role in AI applications: headings, lists, tables, quotation blocks, code fences, and YAML-like metadata remain readable for humans and comparatively well addressable for models. Whereas visual office formatting is

often implicit or tied to binary formats, Markdown represents structure in a linear, copyable, and promptable form.

For AI-supported writing environments this is more than a convenience issue. A model can not only respond stylistically to a Markdown document, but can also move sections, extend tables, consolidate lists, adjust frontmatter, or preserve citation markers unchanged. Markdown therefore functions as display format, transformation format, and exchange format at once between authorship, chat interface, agent logic, and downstream converters.

**Tab. 1:** Frequently used AI-supported chat systems and their typical working formats.

<b>System</b>	<b>Primary interaction</b>	<b>Typical working formats</b>	<b>Role of Markdown</b>
ChatGP T	Dialog-based chat with file and image context	Free text, Markdown, PDF, office files, tables, images, code	Markdown works as a robust intermediate format for structured prompts, lists, tables, and format-stable answers
Claude	Document-centered chat with long context	Free text, Markdown, PDF, DOCX, images, code	Markdown supports long sections, revisions, and clear section references
Gemini	Search- and workspace-near chat	Free text, Markdown, PDF, workspace and office documents, tables, images	Markdown eases switching between chat, document, and export without hidden layout logic
Perplexi ty	Search-oriented research chat	Free text, Markdown, web sources, PDF, images	Markdown is useful for structured syntheses, lists, and citation-near notes
mdedit.i o AI chat	Document-bound editor chat	Markdown documents, frontmatter, citation markers, layout commands	Markdown is the primary format of the working document here, not just a side format

In the academic context this difference is significant. Anyone working with AI on a Markdown core is not merely editing text surfaces, but explicit document structure. That lowers the risk that revisions unintentionally damage semantic outline, citation markers, or export paths. For long-form academic writing, Markdown is therefore not only a lean writing format, but increasingly the more suitable interface format for AI-supported editorial work.

## 3. Options for producing academic theses

### 3.1 Special requirements of academic documents

Academic theses differ from everyday documents through a series of specific requirements. These include at minimum a stable chapter hierarchy, consistent citation, clean bibliographies, reproducible integration of tables and figures, controllable export paths, feedback loops with supervisors, and formal compatibility with university requirements. Added to this are practical requirements such as version stability, reviewability, data frugality, and long-term readability. In the relevant literature on academic writing, this combination of knowledge logic, structuring, citation discipline, and editorial consistency has long been described as the core of scholarly text production (Booth et al. 2016; Swales and Feak 2012).

These requirements make clear that no tool is sufficient merely because it offers attractive typography or fast handling. What matters is how well a tool organizes the interplay of writing, structuring, commenting, exporting, and formal control. In addition, citation and reference systems vary by discipline: APA, Chicago, MHRA, OSCOLA, and Vancouver are not interchangeable matters of taste, but culturally distinct regimes with their own expectations regarding in-text citation, footnotes, bibliographies, and source apparatus (University of Reading Library 2026; American Psychological Association 2019; Paver, Nelson, and Davies 2024; Faculty of Law, University of Oxford 2026; Imperial College London Library Services 2026).

From these requirements the thesis derives a five-dimensional evaluation grid: (1) structural transparency, (2) source and citation reliability, (3) layout and export stability, (4) review and supervision compatibility, and (5) governance and operational viability. This grid serves in what follows as the comparison and evaluation instrument for the three tool logics and for the mdedit.io case.

### 3.2 Microsoft Word

Microsoft Word is widely established in universities. A major advantage lies in its institutional compatibility: many supervisors work with Word comments, templates, and revision modes. In particular, the review area with Track Changes, multiple display modes, reviewer-based filtering, and Reviewing Pane is highly developed for traditional supervision and correction processes (Microsoft Support 2026).

Its weakness, however, often lies in the close coupling of content and immediate formatting. Although Word offers styles and professional layout functions, in practice long documents frequently become unstable through local formatting corrections, inconsistent styles, and hard-to-trace breaks. In heavily revised theses this often leads to format drift, that is, a gradual loss of structural discipline.

### 3.3 LaTeX

LaTeX does not understand itself as word processing, but as a document-preparation system for high-quality typography, especially in academic and technical contexts (LaTeX Project 2026). The system is designed for large documents, cross-references, tables, formulas, bibliographies, and indexes, and relieves authors from many immediate design decisions (LaTeX Project 2026).

This is also LaTeX's classical strength: a clean separation of content and typesetting, strong mathematical and typographic competence, and high stability for large documents. The downside is the higher learning curve. Anyone who does not already master LaTeX invests significant time in toolchains, error diagnosis,

package logic, and template systems. For many students, LaTeX therefore becomes attractive mainly when disciplinary culture, prior knowledge, or formal complexity justify that investment.

### 3.4 Markdown-based workflows

Markdown-based workflows occupy the space between Word and LaTeX. They offer less immediate layout control than Word and less typographic depth than LaTeX, but can combine structure, raw-text readability, and portability particularly well. In combination with Pandoc, metadata, citations, bibliographies, and DOCX and PDF exports become available as well (MacFarlane 2025).

Markdown is therefore not a substitute for every disciplinary culture and every university requirement. Its special strength lies where writing itself, visibility of structure, and controlled multi-export matter more than pixel-precise formatting during the draft phase. In the context of AI-supported writing, one more point matters: Markdown carries its structure explicitly and therefore lends itself better to transformation, revision, and downstream processing than purely visually oriented formats.

### 3.5 Comparison of Word, LaTeX, and Markdown

The tool classes differ less by simple ranking than by their production logic. Word is review- and institution-friendly, LaTeX is strong in typesetting and formal control, and Markdown is structure- and portability-oriented.

**Tab. 2:** Comparison of central tool logics for academic theses.

<b>Criterion</b>	<b>Word</b>	<b>LaTeX</b>	<b>Markdown-based workflow</b>
Entry barrier	Low to medium	Medium to high	Low to medium
Review with supervisors	Very strong	Often indirect or PDF-based	Strong if the editor supports sharing and comments
Typographic control	Medium to high	Very high	Medium
Raw-text readability	Low	Medium	High
Portability	Medium	High	High
Citation maturity	High	Very high	High with Pandoc citeproc
Risk of local format drift	High	Low	Low
Fit for spontaneous revision	High	Medium	High

No overall winner follows from this comparison. Suitability depends on the profile of the project. The requirements of academic documents were derived in Abschnitt 3.1; the consequences for mdedit.io are developed in detail in Abschnitt 5. Where a university enforces rigid Word templates, Word remains dominant in practice. Where mathematical typesetting quality and complex reference systems are central, LaTeX remains strong. But where structural transparency, editorial lightness, portability, web review, and early export control are central, Markdown gains substantial attractiveness.

## **4. mdedit.io as the case under investigation**

### **4.1 Markdown as an academic intermediate format**

Markdown combines three advantages that are especially relevant for long-form academic texts. First, it enforces a structure-oriented writing style. Second, raw text remains permanently readable and portable. Third, it can be converted into established target formats through tools such as Pandoc (Gruber 2004; MacFarlane 2025). These properties reduce the risk that a project sinks into local formatting corrections before the argument itself is stable.

Markdown is especially plausible when writing is understood as thinking in structures. The question then is no longer which font a subheading currently has, but whether the section has a clear role in the argument at all. For academic texts this is not a side issue, but often the difference between editorial activity and scholarly coherence. The same structure orientation also explains its growing importance in AI applications: if models operate on explicit text segments rather than hidden layout semantics, Markdown becomes the more efficient interface format.

### **4.2 Functional profile of the case**

According to project documentation, mdedit.io is a browser-based Markdown editor for long-form writing, structured documents, and exportable drafts. The application combines live preview, document-based outline navigation, collaboration, AI assistance, and export to Markdown, DOCX, and PDF in a self-hosted web application (Hertel 2026c). The help interface explicitly describes the right-side work area as a combination of preview, tree view, and AI chat, and also names a layout editor and print-near paged preview as distinct modes (Hertel 2026b). For the present investigation, the key point is that mdedit.io addresses not just text editing, but the coupling of writing process, document structure, and output format.

The case is therefore suitable as an object of study because several requirements central to these are touched at once: visibility of chapter structure, early preview of page appearance, browser-based access without local desktop installation, and the ability to move one Markdown core into different submission and review formats (Hertel 2026c). A further point matters as well: the AI chat works not on a foreign binary-formatted document, but on explicit Markdown source. mdedit.io therefore becomes a case in which Markdown is not just export basis, but the immediate editing medium for both human and AI-supported editorial work.

### **4.3 Preliminary placement within the evaluation grid**

From a theoretical perspective, mdedit.io is especially relevant where academic work is not understood primarily as layout production, but as structured document development. Outline reduces the distance between structure and raw text, preview shortens the path to print-near control, and multi-export makes the same document core usable for different institutional situations (Hertel 2026c).

This positions mdedit.io between the review- and institution-near logic of Word and the more typesetting- and formal-control-oriented logic of LaTeX. That intermediate position is especially interesting for these because many projects require neither pure office routine nor fully developed TeX workflows, but they do require a stable and transparent production environment.

## 4.4 Anticipated limits of the case

mdedit.io's own scientific concept documentation also names clear limits. The main bottleneck lies less in page layout than in semantic document functions such as YAML frontmatter, labels, numbering, cross-references, and robust citation paths (Hertel 2026e). Likewise, true page-bound footnotes for note-style citation regimes are not promised as secure within the current HTML and paged stack (Hertel 2026e).

The factual conclusion is therefore this: mdedit.io is a serious case for long-form academic writing, but not a universally complete system. Its suitability rises where authors consciously work with structure, frontmatter, citeproc export, and layout rules. The concrete edge cases are named in Abschnitt 5.6 and Abschnitt 4.4. It declines where absolute compatibility with very specific template systems or LaTeX-centered footnote styles is required.

## 5. Criteria-based evaluation of mdedit.io

### 5.1 Evaluation logic and the investigation artifact

The evaluation does not ask whether mdedit.io replicates every single feature of Word or LaTeX. What matters is whether the tool can carry the scholarly production chain of a thesis without methodologically critical breaks. In this thesis, breaks count as critical when they cause structural loss, inconsistent citation, uncontrollable exports, supervision frictions, or institutionally problematic operation.

Tab. 3: Evaluation dimensions of the analysis.

Dimension	Guiding question	Observable indicators
Structural transparency	Does the argument remain visible and stable during the working process?	Outline, raw-text readability, ease of restructuring
Source apparatus	Can citations be processed in a data-based, consistent, and style-adaptive way?	Bibliography path, citeproc, reference section, style switching
Layout and export	Can print-near outputs be controlled early and reproducibly?	Page view, PDF, DOCX, lists, caption behavior
Review and supervision	Is the workflow compatible with feedback and submission cultures?	Sharing, commentability, export into review formats
Governance and operations	Is the tool legally and technically viable for extended academic use?	License, self-hosting, portability, dependencies

The analysis relies not only on feature descriptions, but also on the reference artifact documented in the appendix. This artifact intentionally contains figures, tables, mathematics, footnotes, columns, and page breaks because precisely these elements typically expose the export path's failure points in academic theses.

### 5.2 Structural transparency and the writing process

On the criterion of structural transparency, mdedit.io reaches a high degree of suitability. The combination of plain-text basis, chapter hierarchy, and outline navigation makes it easier to move argument segments early, detect gaps, and revise drafts without formatting ballast (Hertel 2026c). Compared with Word, this reduces the temptation of local formatting corrections; compared with LaTeX, it lowers initial toolchain complexity.

For master's theses this point is not trivial, because theses are typically restructured several times. A tool that keeps outline stability under revision therefore has real methodological value. In this dimension, mdedit.io is not merely usable; for many writing situations it is analytically superior to the classic WYSIWYG paradigm.

#### 5.2.1 Navigation in large documents and tree view

For extensive academic documents, however, a linear editor view is not enough. mdedit.io addresses this through a heading-based outline and a tree view that, according to the product documentation, is based exclusively on heading levels H1 to H6 (Hertel 2026c). The help system describes the tree view not as

decorative extra, but as a full work surface for preview, tree, and AI chat (Hertel 2026b). In long theory chapters, methods sections, and appendices, this makes it visible whether chapter weight, subchapter structure, and argumentative hierarchy remain sound.

Methodologically, that is more than comfort. A master's thesis often fails not because of missing sentences, but because of weak macro-structure. A tree view that instantly shows section depth, order, and emphasis shortens the path between outline decision and text revision. Its limit is equally visible: if tree view is exclusively heading-based, it does not model semantic reference objects such as figures, tables, listings, or bibliography nodes with their own logic. It remains a strong structure tool, but not a full semantic navigation system.

### 5.2.2 AI integration for content editing

mdeit.io integrates AI not only as an external prompt channel, but as an embedded working instrument for document editing. The README speaks of AI-assisted editing, and the help system of an integrated AI panel (Hertel 2026c, 2026b). The API documentation also shows that the editing path is not limited to plain chat answers, but includes structured actions such as `REPLACE`, `INSERT`, `APPEND`, `PREPEND`, and `ADVICE` (Hertel 2026a). For work on a master's thesis this matters above all where introductions must be tightened, transitions rephrased, summaries condensed, or alternative outlines produced quickly.

The methodologically decisive point lies deeper: in mdeit.io the AI works on a canonical Markdown core. This means heading hierarchy, lists, tables, citation markers, and frontmatter remain not only visible but directly editable. In contrast to visually dominated document formats, the AI does not first have to reconstruct structure from implicit formatting. For academic texts this matters because revision thereby stays closer to semantic document state than to mere surface reformulation.

The suitability of this integration remains conditional. For academically legitimate use, AI is suitable for language revision, condensation, producing interim headings, reordering argument chains, and making implicit structural assumptions explicit. It is not suitable as a source of empirical claims, not for unsupported literature assertions, and not for simulating disciplinary evidence. The sensible role of AI in mdeit.io is therefore editorial and heuristic, not epistemic.

### 5.2.3 AI integration for layout work

The AI integration in mdeit.io is not limited to running text according to the current intent model. In the server, layout-related requests such as `layout`, `margin`, `typography`, `columns`, `header`, `footer`, and `table of contents` are explicitly recognized as edit intents (Hertel 2026d). Together with the layout editor, print-near preview presets, and the paged workflow, this forms an integrated editing path in which layout rules can be set manually but also prepared through AI-supported formulation or adjustment requests (Hertel 2026b).

This coupling is especially interesting for master's theses because layout questions rarely occur in isolation. If, for example, figure density increases in a results chapter, columns, caption styles, page breaks, or table-of-contents depth often need to change as well. The relevant limitation is that this AI integration offers no guarantee of typographic correctness. It can accelerate layout work, but it does not replace visual inspection in page view or formal final control against institutional requirements.

## 5.3 Source apparatus and scholarly integrity

In the area of source apparatus, mdedit.io is thesis-ready only if citations are treated consistently as data-based objects rather than as manually formatted text. The present reference file already operationalizes this logic through YAML frontmatter, `citation-source: embedded`, a document-bound `mdedit-bibliography` block, linked citations, and an explicit reference section. In combination with Pandoc and citeproc, this is viable for author-date and numeric workflows (MacFarlane 2025; Hertel 2026e, 2026d).

### 5.3.1 Integration of bibliographies and citation standards

Especially in light of current mdedit extensions, it must be stated that the scholarly source path now goes beyond classic plain-text Markdown. In scientific documents, the server evaluates YAML fields such as `citation-source`, `cs1`, and `reference-section-title`, and also recognizes `link-citations`, `link-bibliography`, and `nocite` as relevant metadata for export (Hertel 2026d). The normative local source path is an `mdedit-bibliography` block with embedded CSL JSON, while export itself is built through Pandoc with `--citeproc` (MacFarlane 2025; Hertel 2026d). File-based `bibliography:` paths are still recognized in current server logic, but only for clear migration errors and no longer accepted as a valid export workflow (Hertel 2026d, 2026f).

For the present reference file it is especially important that it now uses this embedded path itself. The document contains the library as a local snapshot in the source and therefore tests not only citation syntax, but the entire document-bound source mode. The current mdedit marker `#refs` also remains intentionally present. This marker is not intended as visible final output in the source itself, but as part of the current scientific interface: in the export path it is normalized server-side into a real references section with anchors (Hertel 2026d). That is exactly why `#refs` here is not editorial negligence, but an end-to-end test of the current bibliography integration.

Import and export perspectives must be distinguished from this. Product planning foresees adapters for BibTeX, RIS, and CSL JSON on the local document core so that existing reference collections can be imported into the embedded library and exported from it again (Hertel 2026f). The integrated mdedit format is therefore not conceived as an isolation from existing standards, but as a reproducible document core to which standard imports and exports attach.

### 5.3.2 Options for connecting external online catalogs

Planning for scientific documents distinguishes the document-bound source mode `embedded` as well as the future modes `zotero` and `hybrid` (Hertel 2026f). This leads to an important architectural decision for theses: external online catalogs should support search, selection, and metadata enrichment, but should not become the only normative export source of the submitted thesis.

The first external connector is planned as a read-only Zotero path. Sources should be searchable there and then copied into the local library or into document-bound snapshots without the export or sharing path depending at runtime on a live Zotero connection (Hertel 2026f). In parallel, OpenAlex is planned as an open lookup for title, DOI, or author searches; here as well, hits should be transferred into the local library, while OpenAlex itself does not remain the normative export source of the document (Hertel 2026f).

For thesis-ready workflows, exactly this snapshot logic is central. Anyone integrating external online catalogs should treat them as an upstream research and import layer, not as the only storage location of the submission-relevant bibliography. The workflow becomes reproducible only when the source state relevant

for submission remains fixed as an embedded mdedit library or as a controlled hybrid snapshot in the document context (Hertel 2026f, 2026e).

The limit lies where note-style footnotes, complex cross-references, or highly specific institutional logic are productively required. For such regimes, the current certainty of the HTML and paged stack is, according to the project's own sources, not yet sufficient (Hertel 2026e). The judgment therefore remains deliberately conditional: highly compatible for data-based standard workflows, limited for legal or template-heavy special cases.

## **5.4 Layout, typesetting, and export stability**

On the criterion of layout and export, mdedit.io occupies a strong but not universal position. Scientific preset, page view, and PDF and DOCX export permit early control of breaks, tables, figures, and indexes (Hertel 2026c). For the present reference file no local `layout` block is deliberately set; this tests how viable the global scientific preset path already is.

The methodological benefit of this decision is that no institution-specific special case is concealed. If the document remains stable under the global Scientific preset, this speaks for robust baseline suitability. If special cases can only be solved through local layout overrides, they must be identified as profile-bound additional requirements and must not be mistaken for general thesis readiness.

## **5.5 Review, supervision, and collaboration capability**

For a real master's thesis, technical typesetting capability alone is not enough; compatibility with supervision matters just as much. Here Word remains strong through Track Changes, commentable DOCX workflows, and institutional routine (Microsoft Support 2026). mdedit.io instead relies on browser-based sharing, direct availability, and multi-export (Hertel 2026c). That is plausible for digital supervision scenarios, but not fully equivalent to established Word markup routines.

The evaluation therefore does not say that mdedit.io is generally better than Word, but that it is viable under different supervision conditions. As soon as supervisors require DOCX comment workflows, the export path must become part of the methodological work plan. Under that condition mdedit.io remains compatible; without it, breaks may arise that are social-organizational rather than technical.

## **5.6 Exclusion criteria and edge cases**

The analysis yields clear edge cases in which mdedit.io should currently not be recommended as the primary environment: first, strictly prescribed Word templates with fragile formatting semantics; second, legal note-style citation regimes; and third, projects that require semantically rich cross-references and automatic reference objects as a hard requirement (Hertel 2026e, 2026f).

Conversely, mdedit.io is particularly suitable for text-centered, argumentatively structured, and export-oriented projects in the humanities, social sciences, economics, and parts of computer science or life

sciences, as long as the formal target architecture is compatible with author-date or numeric citation and a controlled PDF or DOCX path.

**Tab. 4:** Summary evaluation of *mdedit.io* within the requirement grid.

<i>Evaluation dimension</i>	<i>Assessment</i>	<i>Rationale</i>
<i>Structural transparency</i>	<i>high</i>	<i>Outline, raw-text readability, and low formatting ballast stabilize the writing phase</i>
<i>Source apparatus</i>	<i>medium to high</i>	<i>strong for data-based standard workflows, limited for note-style and special logics</i>
<i>Layout and export</i>	<i>high</i>	<i>early page control and multi-export, but special cases remain profile-dependent</i>
<i>Review and supervision</i>	<i>medium</i>	<i>strong through sharing and browser access, limited against institutional Word routines</i>
<i>Governance and operations</i>	<i>high</i>	<i>self-hosting, license clarity, and portable document basis support institutional viability</i>

## 5.7 Stress test against demanding university profiles

To avoid claiming baseline suitability only in generic terms, *mdedit.io* is mirrored against three demanding university profiles. These profiles stand as examples for legal, computer-science, and life-science boundary conditions and therefore mark different stress points for the system.

### 5.7.1 University of Oxford, Law

The Oxford Law profile is the academically strictest candidate for legal writing. OSCOLA is explicitly designed for precise citation of case law, legislation, and legal secondary sources and is published by Oxford Law Faculty (Faculty of Law, University of Oxford 2026). Even the official quick reference shows that this regime differs significantly from classic author-date systems: cases, statutes, commentaries, articles, and reports follow distinct and highly formalized patterns (Faculty of Law, University of Oxford 2026).

For *mdedit.io* this implies the following: the Scientific preset is suitable for baseline typography, margin calmness, and table layout, but a genuine Oxford Law profile would additionally require note-style footnotes, legal short forms, clean source abbreviations, and a legal-specific export path. That is why Oxford Law does not mark a generic success scenario, but a deliberate exclusion or development profile.

### 5.7.2 ETH Zurich, Computer Science

The ETH profile is especially interesting for computer science because the official D-INFK memo defines the master's thesis as independent scholarly research or constructive development work with a written report and oral presentation (Department of Computer Science, ETH Zurich 2025). The memo also names clear framework conditions such as a 28-week full-time duration and mandatory electronic submission of a PDF including a signed declaration of originality by 23:59 on the end date (Department of Computer Science, ETH Zurich 2025).

For *mdedit.io*, this profile is comparatively compatible. It requires less exotic citation logic than Oxford Law, but does require a stable scholarly production path with title page, abstract, declaration, table of contents, figure and table logic, code appendices, and reliable PDF submission. As a stress test, ETH therefore represents a realistic high-demand profile that is fundamentally achievable with a disciplined Markdown workflow.

### 5.7.3 Imperial College London, Medicine and Life Sciences

The Imperial profile represents a natural- and life-science context with numeric citation. The official Vancouver guidance describes a numbered citation system in which references appear in the text numerically and the reference list at the end is ordered numerically; uncited but consulted sources may optionally appear in a separate alphabetical bibliography (Imperial College London Library Services 2026). Longer quotations are also expected to appear as their own indented paragraph (Imperial College London Library Services 2026).

Technically, this is the easiest profile to operationalize for mdedit.io. Vancouver can be built quite well on top of a Scientific preset through citeproc and CSL. An Imperial-near profile would therefore not be a special experiment, but a realistic target state for numeric, export-oriented, and figure-sensitive theses.

## 5.8 Design implications for institutional profiles

The stress test shows that university profiles differ less in actual page geometry than in citation, mandatory sections, validation rules, and export requirements. From this follows a concrete design implication for mdedit.io: institutional thesis profiles should not primarily be modeled as hard-wired layout special cases, but as profile packages that build on the Scientific preset (Hertel 2026f).

A robust model would have five layers:

1. `base=scientific`: the global Scientific preset remains the typographic base layer for all academic profiles.
2. Citation package: CSL, citation mode, bibliography rules, and optionally note-style export logic are bundled per profile.
3. Metadata package: title page fields, declaration of originality, abstract languages, appendix rules, and mandatory sections are delivered as profile defaults.
4. Validation package: a profile checks whether, for example, OSCOLA footnotes, Vancouver numbering, ETH declaration, or faculty-specific section ordering are fulfilled.
5. Distribution package: profiles are delivered as downloadable manifests or small packages so that universities, chairs, or communities can version and share them.

*One possible manifest could look like this conceptually:*

```
id: ethz-dinfk-masterthesis
base: scientific
citationStyle: author-date
bibliographyMode: reference-list
requiredSections:
  - title-page
  - abstract
  - declaration-of-originality
  - references
validators:
  - declaration-required
  - figures-and-tables-numbered
  - bibliography-present
exports:
  pdf: required
  docx: optional
```

The decisive point is this: not every profile needs its own document layout. Many institutions differ more in citation, frontmatter, mandatory sections, and validation than in actual page geometry. That is exactly why the present reference file deliberately has no local layout block and is suitable as a base scenario for a later institutional profile system.

## 5.9 Reasons not to choose mdedit.io as the tool for one's own master's thesis

Even after a benevolent criteria-based evaluation, a long list of legitimate reasons remains for not choosing mdedit.io as the primary tool for one's own master's thesis. The most important counterarguments are these:

1. The university prescribes a rigid Word template that is binding not only formally, but technically.
2. Supervision relies on `Track Changes`, margin comments, and reviewer-based DOCX routines, so downstream export is not merely convenient but central to the process (Microsoft Support 2026).
3. The citation regime requires true note-style footnotes, legal short forms, or OSCOLA-like references that are not yet sufficiently secured in the current mdedit stack (Faculty of Law, University of Oxford 2026; Hertel 2026e).
4. The thesis needs semantically rich cross-references between figures, tables, formulas, listings, and appendices that should not be reduced to manual references (Hertel 2026e).
5. The field requires deep typographic control, specific LaTeX packages, or mathematical typesetting quality beyond a Markdown-Pandoc path (LaTeX Project 2026).
6. The chair's academic infrastructure is already stably based on Overleaf, LaTeX classes, or institutionally maintained templates, so a tool switch would create more risk than insight.
7. The project is heavily code-, listing-, or table-driven and therefore calls for a semantically rich, mature publishing system rather than a structure-oriented intermediate system.
8. The author wants maximum visible formatting and immediate WYSIWYG control throughout the whole writing process rather than a Markdown-centered production logic.

9. The supervision or examination environment is organizationally conservative and views unusual export paths skeptically even when the result is formally correct.
10. No reliable self-hosting or operating environment is available, so privacy, availability, or long-term stability cannot be institutionally controlled (Hertel 2026c).
11. The working process should deliberately avoid AI proximity, browser artifacts, and editorial multi-system logic.
12. The project is extremely risk-sensitive regarding export drift or preview discrepancies and therefore wants to rely exclusively on maximally established, long-tested thesis workflows.
13. One's own writing practice is already deeply integrated into Zotero-Word or TeX bibliography workflows, so mdedit.io would offer no real productivity gain.
14. The stabilization effort around the scientific special case would cost more time than the thesis itself gains in insight.

No general verdict against mdedit.io follows from this list. It shows, however, that the decision for or against a tool of choice must always be read as a fit decision between disciplinary culture, examination logic, supervision practice, and individual working style.

## **6. Governance, rights, and operational viability**

### **6.1 Licensing and rights**

According to the project documentation, the source code of mdedit.io is licensed under Apache License 2.0; separately marked brand assets and trademark elements are excluded (Hertel 2026c). This is relevant for academic use because it creates transparency about the legal basis and means the application does not have to be treated as a proprietary black box. For universities, research groups, or individuals who value understandable and self-operable tools, this is a substantial advantage.

### **6.2 Access model and practical usability**

The usability of mdedit.io derives above all from four factors: browser-based access, no mandatory account creation, private documents through a session model, and an interface strongly oriented toward the writing process (Hertel 2026c). This is especially relevant for students because entry is possible without local desktop installation and the working context remains easier to move across devices.

### **6.3 Availability and operation**

mdedit.io is documented as a self-hosted web application. The README describes both a Docker-based standard path and a direct Node.js start for local development (Hertel 2026c). Browser dependencies are bundled locally as well, so the active runtime path does not depend on external browser CDNs (Hertel 2026c). This availability logic matters for data-sensitive or institutionally controlled environments because operation, updates, and data storage do not necessarily have to be outsourced into external SaaS infrastructures.

## 7. Discussion

The thesis answers the research question in a differentiated way. A browser-based Markdown editor can serve as the primary production environment for academic theses when the requirement profile of the thesis is characterized by structure-oriented writing, data-based citation, controlled multi-export, and digitally compatible supervision. Under these conditions, mdedit.io is not merely a convenient writing tool, but a viable scholarly artifact.

The scholarly contribution of this thesis lies less in claiming universal superiority than in offering an explicit evaluation model. The usual debate over Word, LaTeX, and Markdown is often conducted normatively or identitarianly; the present study translates it into testable criteria and shows that the suitability of a tool depends on the coupling between document type, citation regime, supervision practice, and export architecture. From this perspective, mdedit.io appears not as a full replacement, but as a robust solution for a clearly delimited segment of academic theses.

The results nevertheless remain limited. No user study, no time comparison of real writing processes, and no broad template benchmark with institutional templates were conducted. The findings must therefore be read as an analytical suitability assessment, not as an empirical proof of effectiveness. That is exactly where the next research step follows: controlled case studies, export parity tests, and institutional profile packages based on real submission requirements.

## 8. Conclusion

The research question can be answered as follows: mdedit.io can serve as the primary production environment for academic theses when the project requires structure-oriented, data-based, and export-controlled authoring and no mandatory note-style or template special cases are present. Under these conditions, a browser-based Markdown workflow is methodologically defensible, technically robust, and institutionally compatible.

Markdown therefore does not prove to be a blanket replacement for Word or LaTeX, but rather an independent production logic for long-form academic texts (Gruber 2004; CommonMark Contributors 2026; MacFarlane 2025). Word remains strong where institutional review routines dominate; LaTeX remains superior where typographic and reference-related depth control is decisive. mdedit.io occupies the space in between: strong in structural transparency, heading-based tree navigation, AI-supported document work, frontmatter- and citeproc-based bibliography integration, early layout control, and controlled multi-export, but limited in heavily standardized edge cases (Hertel 2026c, 2026b, 2026d, 2026e).

The value of the thesis therefore lies in a reasoned decision aid. It shows not only that mdedit.io is viable for certain theses, but also under which conditions this is not the case. Precisely this conditional answer is more valuable for academic practice than a simple yes or no.

# References

- American Psychological Association. 2019. *Publication Manual of the American Psychological Association*. 7th ed. American Psychological Association.
- Booth, Wayne C., Gregory G. Colomb, Joseph M. Williams, Joseph Bizup, and William T. Fitzgerald. 2016. *The Craft of Research*. 4th ed. Chicago: University of Chicago Press.
- CommonMark Contributors. 2026. “CommonMark.”<https://commonmark.org/>.
- Department of Computer Science, ETH Zurich. 2025. “Memo: Master’s Theses in Computer Science.”
- Faculty of Law, University of Oxford. 2026. “The Oxford University Standard for Citation of Legal Authorities (OSCOLA), 5th Edition.”<https://www.law.ox.ac.uk/oscola>.
- Gruber, John. 2004. “Markdown.”<https://daringfireball.net/projects/markdown/>.
- Hertel, Matthias. 2026a. “Mdedit.io AI Chat API Documentation.”
- . 2026b. “Mdedit.io Help Page.”
- . 2026c. “Mdedit.io README.”
- . 2026d. “Mdedit.io Scientific Citation and Export Implementation.”
- . 2026e. “Scientific Documents and Citations Concept.”
- . 2026f. “Scientific Documents Implementation Plan.”
- Hevner, Alan R., Salvatore T. March, Jinsoo Park, and Sudha Ram. 2004. “Design Science in Information Systems Research.” *MIS Quarterly* 28 (1): 75–105.
- Imperial College London Library Services. 2026. “Vancouver Style.”<https://www.imperial.ac.uk/admin-services/library/learning-support/reference-management/vancouver-style/>.
- LaTeX Project. 2026. “An Introduction to LaTeX.”<https://www.latex-project.org/about/>.
- MacFarlane, John. 2025. “Pandoc.”<https://pandoc.org/>.
- Microsoft Support. 2026. “Track Changes in Word.”<https://support.microsoft.com/en-us/office/track-changes-in-word-197ba630-0f5f-4a8e-9a77-3712475e806a>.
- Paver, Chloe, Graham Nelson, and Simon F. Davies, eds. 2024. *MHRA Style Guide*. 4th ed. Modern Humanities Research Association.
- Pears, Richard, and Graham Shields. 2025. *Cite Them Right: The Essential Referencing Guide*. 13th ed.
- Swales, John M., and Christine B. Feak. 2012. *Academic Writing for Graduate Students: Essential Tasks and Skills*. 3rd ed. Ann Arbor: University of Michigan Press.
- University of Reading Library. 2026. “Different Styles and Systems of Referencing.”<https://libguides.reading.ac.uk/citing-references/referencingstyles>.
- Wieringa, Roel J. 2014. *Design Science Methodology for Information Systems and Software Engineering*. Berlin: Springer.

## Appendix A. Reproducible reference artifact for layout and feature testing

The following appendix is not a decorative extra, but part of the research design. It bundles those elements that are typically error-prone in the scholarly export path and thereby serves as a reproducible test artifact of the thesis.

### A.1 Figure specimen



**Abb. 1:** *Neutral image artifact for caption, list, and print tests*

## A.2 Scientific table

Tab. 5: Criteria set for the final quality check of an academic thesis in mdedit.io.

Criterion	Expectation	Purpose
Chapter structure	Complete and logical	Check argumentative guidance
Citation	Consistent and data-based	Protect source integrity
Export	PDF and DOCX controlled	Secure submission and review paths
Layout	Pages, tables, and images stable	Check formal robustness

## A.3 Compact table

Tab. 6: Typical tasks before submission.

Task	Status
Sharpen research question	open
Check bibliography	open
Run final PDF inspection	open
Export DOCX for supervision	optional

## A.4 Mathematics, page-footnote, and endnote specimen

The formula  $E = mc^2$  serves as an inline test. The following block works as a display-math test:

$$\int_0^{\infty} e^{-x^2} dx = \frac{\sqrt{\pi}}{2}$$

In addition, a page footnote remains directly in the running text as a typesetting and export specimen.

A separate document-end note also remains so the export path covers both note modes in the same reference document.<sup>2</sup>

## A.5 Multi-column specimen

Scientific main text rarely needs columns, but appendices and special pages can benefit from a two-column layout. The reference file therefore contains a real columns block here with a deliberate column change.

The left column checks whether longer running text in an academic document is carried into a two-column layout without spacing, paragraph separation, or line breaking collapsing. At the same time, the text remains factual enough to serve as a realistic appendix or handout specimen.

The right column checks the same path after an explicit column break. This makes it possible to distinguish in preview and PDF whether automatic distribution and manually set column breaks both remain stable. For thesis workflows this matters especially in appendices, lists, or compressed supplementary information.

1. This page footnote checks whether the note stays attached to the sentence and lands cleanly at the bottom of the page in paged preview and PDF.

## A.6 Unicode, cross-reference, and listing specimen

The new semantic cross-references connect sections directly in the text. As described in Abschnitt 1.3, the analysis follows a criteria-guided artifact evaluation. The research question from Abschnitt 1.1 was operationalized in Abschnitt 5.1; the limits of the case are documented in Abschnitt 4.4 and Abschnitt 5.6. This `[@sec: . . . ]` syntax is a feature of SPR-03: the editor resolves these references in preview to linked section numbers, and Pandoc preserves them in export through the `header_attributes` reader.

The English typography specimen intentionally includes real Unicode as well: naive, co-operate, “curly quotation marks”, a dash -, and § 1 para. 2. This checks whether copyability, searchability, and character rendering remain stable during export.

*Listing 1: Example profile configuration for an institutional thesis package.*

```
id: example-masterthesis
base: scientific
citationStyle: author-date
requiredSections:
  - abstract
  - references
```

An intentionally long URL additionally serves as a line-breaking test in running text:

```
https://example.org/a/very/long/path/with/many/segments/and-parameters?
alpha=123&beta=456&gamma=789
```

## A.7 Figure and table cross-references (SPR-04)

This section checks the new semantic object IDs and cross-references for figures and tables. The architecture overview in Abbildung 2 shows the export path; the tool comparison is documented in Tabelle 7.



*Abb. 2: Export path from Markdown to PDF and DOCX through the mdedit converter*

**Tab. 7:** Tool comparison for the academic export path (cross-reference test).

<b>Tool</b>	<b>Path</b>	<b>Function</b>
Pandoc	Markdown -> DOCX	Document structure and citation
Chromium	Paged HTML -> PDF	Print-near page geometry
citeproc	BibTeX -> bibliography	Source resolution under CSL

1. This document-end note deliberately stays in the endnotes block at the end of the document and checks that page footnotes and endnotes stay separated.

