Analysis and Proposed Solution to The Lode Content Management System

Liam Houston

ACSHF, Michigan Technological University

HF 5850: Human Factors

Dr Betsy Lehman

December 11, 2025

Analysis and Proposed Solution to The Lode Content Management System

The Lode at Michigan Tech is a student-run and operated student news organization that creates weekly print editions for its readers across campus. The Lode has developed its own production system to meet its weekly content deliverables and deadlines. Currently this production pipeline utilizes a content management system (CMS) built from tools that students at The Lode have readily, and freely available to them. There are several shortcomings where these tools cannot adequately meet the specific needs of production, or cannot address the constraints placed on the CMS by certain components. Several work arounds have been developed for these shortcomings that are almost always at the expense of a human actor within the system, and introduce unnecessary potential for human errors. This document covers the issues identified through a preliminary analysis of the CMS, proposing a novel solution to current concerns.

Background Context

The problem space surrounding The Lode is unique and requires some explanation of background knowledge on the environment surrounding The Lode for a clear understanding of both the problems and solutions presented in this paper. The Lode is a student-run organization at Michigan Tech, funded through the Undergraduate Student Government (USG) as a Special Budgetary Group (SBG). The structure of The Lode consists of a core production and management board composed of executive staff.

The Lode has a partnership with the Experience in Journalism course offered through the Humanities department, which is how it sources the majority of its written and photographic content. Students that take the course gain journalism experience by producing article content that is submitted to both The Lode and the course Canvas.

Front End Analysis of The Lode CMS

To fully understand the current issues with the system, a small front end analysis was conducted. The system was investigated through short, informal interviews with high level personnel in The Lode,

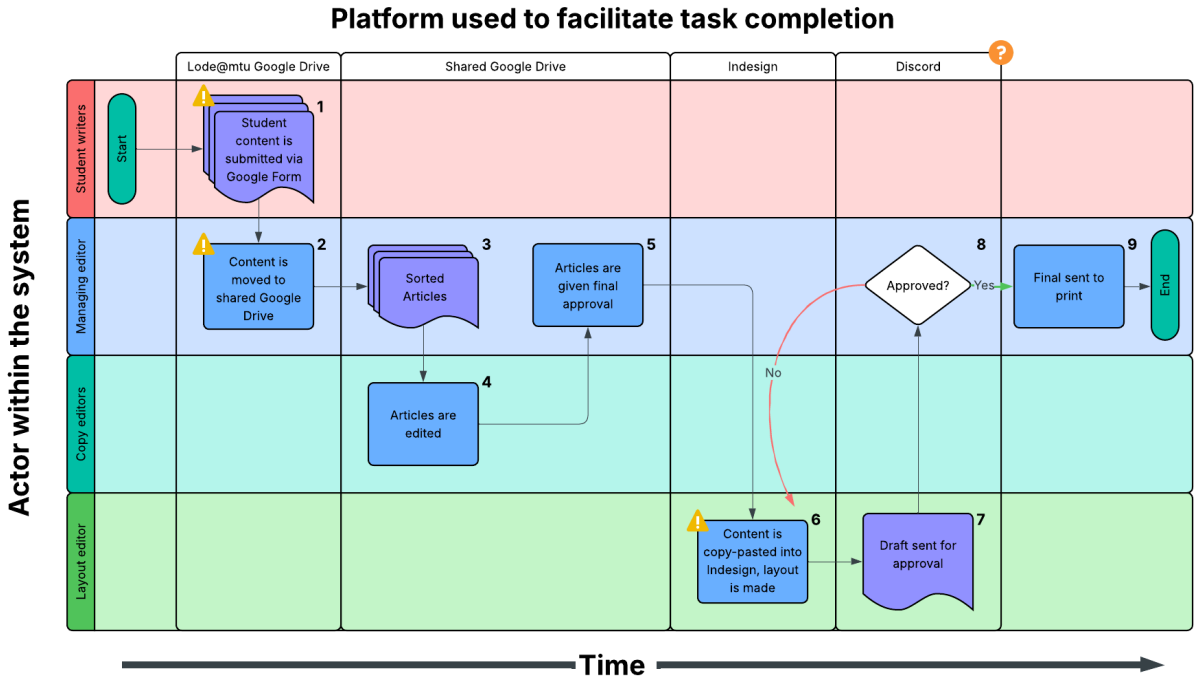
who have experience interacting with the system under different roles. These interviews were used to map out the current system and identify existing issues faced by different system actors. The understanding of the system was also informed by the human factors practitioner and author of this paper, as they are a situated expert on the system—being the sole personnel in a critical role within the system.

Task Flow Mapping

One of the goals of the analysis of the current Lode CMS was to understand the task flow through the system, the artifacts involved at each point of the process, and who was performing each task. Figure one displays a swimlane diagram of the current system, divided by key system actors that directly interact with the system and the components that make up the Lode CMS.

Figure 1: Task Flow Diagram

Swimlane diagram depicting key actors, task flow within the CMS, and highlighted points of concern.



The four major actors identified were student writers, the managing editor, copy editors, and the layout editor. There are other system actors and stakeholders tied to the system, these four were isolated due to their direct interaction with CMS components and the common stress points associated with them.

The components that make up the CMS were identified as a Google Form hosted through The Lode's official Gmail account, The shared Google Drive used by the organization for central storage and access, Adobe Indesign, and The Lode Discord.

An important time constraint of the system not communicated in Figure one is The Lode's tight schedule. The Lode's weekly print schedule demands quick turnarounds and strict deadlines for all content. Articles and photo content is due on Friday nights at midnight, all production is completed in a single production meeting on Sundays.

Personas

The key actors identified in the mapping of system structure were used to guide the creation of four personas that reflect the testimony of real users in those roles. These personas have been used to highlight these users' functions, how they interact with the system, and key points of friction that they can experience.

Persona 1: Joseph Hadel—Student Writer

Joseph Hadel (20) is an undergraduate student enrolled in the Experience in Journalism course. Joseph's main goal is to complete article assignments—accuratly submitting them on time every Friday night. A major concern for Joseph is the dual submission requirement for his articles. Articles are due for a course grade in Canvas and to The Lode's Submission form on the same day. Managing these redundant deadlines results in an increased cognitive load, resulting in the occasional error of omission, where Joseph makes late submissions or misses deadlines entirely.

A second concern for Joseph is the poor design of The Lode's submission form. Joseph was thrust into reporting week one of the course, and has only used the submission form a handful of times. The fields of the form are not well described outside of simple labels, and there are no guardrails to prevent user error. This results in the occasional commission error where information is entered into the system wrong.

Persona 2: Emma Beck—Managing Editor

Emma Beck (22) is the Managing Editor for The Lode. In this position she is responsible for oversight of the entire CMS and the final approval of all content before it continues through the production pipeline.

Emma's first frustration with the current system involves taking submitted content and sorting it from its submission location to the Shared Drive that editors will access for copy editing. This process arises from the inability for Google Forms to send file submissions to a shared drive. The process of transferring articles—although not necessarily time consuming—is completed manually. This introduces the potential for human error, especially when content is submitted incorrectly and requires Emma to fix incorrect submissions.

Emma's subsequent interactions with the system are at critical content approval steps. The process for tracking changes and approval of articles is done through adding characters to the file name of articles. This method has poor system status visibility, and often results in status being communicated incorrectly.

Persona 3: Matt Spiegel—Copyeditor

Matt Spiegel (20) is a copy editor in The Lode. Every week he attends the Sunday production meetings where he goes through submitted articles and suggests edits to be later approved by Emma.

The only concern Matt has is the same status visibility issue as Emma. To communicate that he has edited an article, Matt has to add a lower case ‘e’ to that article’s filename. This action is sometimes omitted—likely due to the high attention focus required by editing several papers in one session.

Persona 4: Jeremy Green—Layout Editor

Jeremy Green (24) is the layout editor for The Lode. He is responsible for taking final articles and incorporating them into the layout of the paper which will be sent to printers and distributed later in the week. Jeremy works between two system components; Google Drive and Adobe Indesign.

Jeremy’s most pressing concern with the current system is the incompatibility between Google Drive and Indesign. In order to transfer content from the drive into Indesign, he has to manually copy and paste each article from Google Docs to a text box in the final layout. This process involves a high cognitive load due to the frequent task switching between programs. Errors are often made due to the speed demands of this process and the reliance on memory to keep track of what content has, and has not been transferred over. This issue is further compounded by the fact that advertisement and photo content are also transferred over in a similar time frame, adding more task switching.

Other Stakeholders

Outside of the identified personas, there are other stakeholders of the CMS that have been considered. These stakeholders have not been defined through personas due to either their removed interaction with the CMS, or the fact that the front end analysis determined that they do not experience pain points to be addressed in this analysis. The vested interest in the Lode CMS and the impact the proposed solution will have on these stakeholders' interaction with the system is still important, requiring their consideration.

The first major stakeholder is the instructor of the Experience in Journalism course. While this individual does not directly interact with the CMS, they are a recipient of the work done by students of

the course. If the dual submission requirement of student writers is to be addressed, how this impacts the instructor's access to articles for grading needs to be accounted for.

The second stakeholder in the CMS is the Business Manager of The Lode. The Business Manager acts as both the treasurer and head advertisement manager for The Lode. Their only interaction with the system is the submission of ad content through the backend of the CMS. Any change to the CMS made will need to account for—or streamline—their need to access the internal file structure of the system.

The third and last stakeholder to consider is the media editor. The media editor has a similar level of access to the CMS as the Business Manager. The media editor needs access to the file management system used by the proposed solution to ensure that they can still access and edit submitted photo content.

Heuristic Evaluation

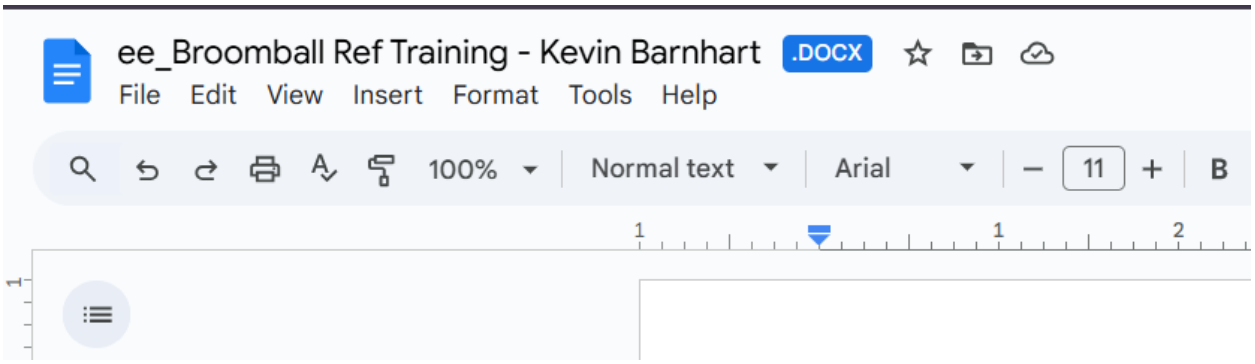
To corroborate the existence of user pain points, and diagnose the specific issues behind issues, A heuristic evaluation of the system was conducted. This evaluation utilized the 10 heuristic categories established by Nielsen (2024). Nielsen's heuristics were not used to evaluate the entire design of the current CMS. Heuristics were evaluated in the major friction points identified in the mapping of the system and the interviewing of current users. This practice was intended to diagnose the cause of user error within the design, so that it could be specifically addressed in the design of the solution. To this end, only four heuristics were relevant and are covered in this section.

Visibility of System Status

The current system does a poor job of communicating adequate visibility of system status. This is most relevant to the article approval process managed by the Managing editor and copyeditors. As shown in Figure 2, the system relies on changing filenames—adding a lowercase 'e' character—to indicate document status. This is a weak signifier that fails to support adequate situation awareness. The system forces users to rely on active perception to successfully perform the desired action. This design

flaw is directly reflected in the experience of Matt, who—under the high attention demands of a production meeting—can commit errors of omission by failing to properly update status characters. Because of this, Emma is often given an inaccurate state of the system, forcing her to expend additional attention to verify which articles are actually ready for production. Due to Emma using the same file-renaming convention—exept with a capital 'E' charcter—to denote an article has been approved, the same inaccurate system reporting is experienced by Jeremy.

Figure 2: Example of file-renaming convention

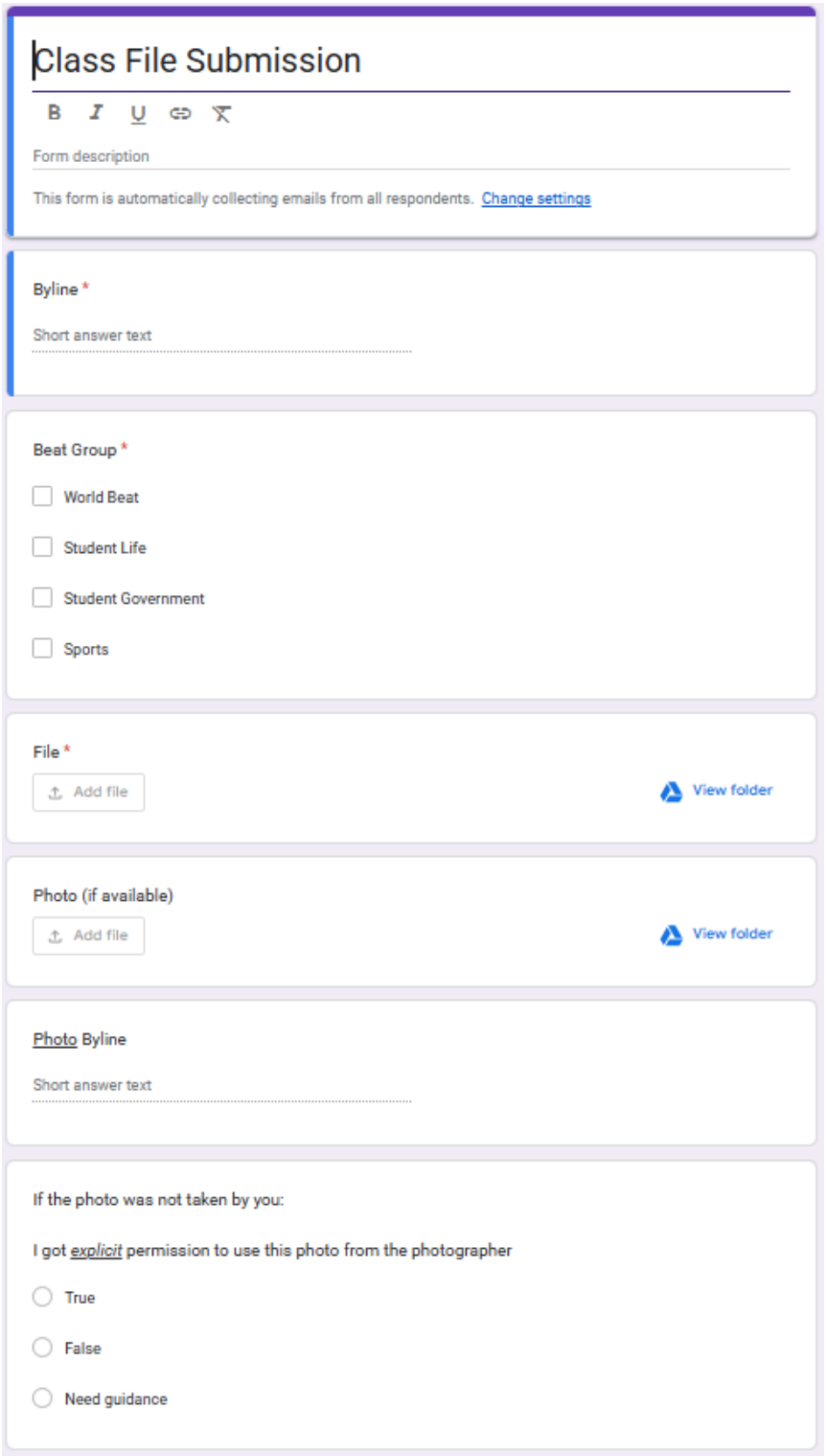


Error Prevention

The current CMS does a very poor job at preventing errors. This heuristic is particularly important to the content submission interface of the system. Almost all student writers are novice users with limited exposure to the system like Joseph. The lack of guardrails and directions within the submission interface—shown in Figure 3—allows for significant errors of commission. Specifically, the incorrect field entries that require future correcting.

Furthermore, the system relies on manual file management by the Managing editor to transfer between Google Forms and the Shared Drive. This manual transfer process introduces a high probability of human error, as the system does not automate sorting. By relying on human vigilance rather than system constraints, the workflow facilitates potential user errors like those Joseph and Emma struggle to manage.

Figure 3: Submission Google Form



Recognition Rather Than Recall

The current workflow imposes a mental load on the layout editor, violating the principle of recognition rather than recall. The incompatibility between Google Drive and the Indesign layout software forces the layout editor to engage in rapid, repetitive task switching. The layout editor must manually track which content has been transferred and which remains pending. Because the system provides no visual checklist or “transferred” state, Jeremy is forced to rely on working memory to track progress. This heavy reliance on internal memory, compounded by the speed demands of production, increases the required resource load for operation.

This requirement for recall is also present in the interactions of other actors mentioned earlier. Due to the lack of adequate signifiers for system reporting, the Managing editor and copy editors are forced to rely on recall rather than recognition.

Help Users Diagnose and Recover From Errors

Finally, the system fails to help users recover from errors in crucial steps. specifically regarding the student submission process. The need for students' articles to be submitted accurately is partially constrained by the time added to the Managing editor’s workload when fixes need to be made, and the inability to reach out to student writers to fix incorrect—or missing—content submissions once production has started. The “occasional error of omission” noted in Joseph’s persona suggests users may believe the submission task is complete, but the system state does not reflect this. Without clear, plain-language feedback confirming a successful submission or highlighting missing fields, the user cannot diagnose their own mistakes. The importance for the system to allow users to correct their mistakes is also important for the CMS to have the possibility of sorting articles automatically.

Solution Requirements

The outcome of the heuristic design is a clear understanding of specific requirements for improving the known pain points within the system. The proposed solution must be able to meet these

requirements to adequately address all concerns. The following list outlines the specific requirement the proposed solution has been designed to resolve:

- **Automate Status Visibility:** The system must replace file-renaming conventions with visual status indicators. This removes the burden of "active signaling" from all editors, and provides better status reporting and situation awareness to users.
- **Implement Input Validation:** The submission form interface must utilize functions that prevent incomplete or malformed data entry. This will reduce commission errors for novice users.
- **Simplify Workflow:** The solution must reduce the taxing content transfer between the Google Drive and InDesign. By minimizing task-switching and manual copy-pasting, the system will reduce the cognitive load currently placed on the layout editor.
- **Provide Feedback Loops:** The system must provide explicit confirmation upon successful submission or update pushed to the system, and provide warnings to help diagnose failure. This will ensure users can evaluate the outcome of their actions without external assistance.

Proposed Solution

The solution proposed to solve all present problems with The Lode's CMS is a series of plugin applications and custom software solutions that add desired functionality to the current system. This solution has been determined to be the best solution given identified constraints and other considerations. The solution involves two parts; a custom file submission process achieved through a series of plugins on The Lode's WordPress website, and a custom Google Workspace plugin that adds desired content and system status tracking features.

Wordpress submission solution

The submission process solution will take advantage of two free commercial plugins and a custom LTI integration into Canvas. The first plugin is called Integrate Google Drive. This plugin allows for seamless integration between WordPress and Google Drive. There is a pro package available with this

plugin, but it is not required for this solution. The other plugin is the WordPress Forms. These plugins will likely require some custom coding to ensure they interact with each other correctly. Once set up, WPForms allows for a custom submission webpage to be created with custom guardrails and warning messages to prevent errors in submission. The submissions can then be stored directly in The Lode's Google Drive through the use of the Integrate Google Drive plugin. To solve the dual submission problem, a custom Canvas LTI integration would allow content to also be directly sent to Canvas. A custom solution is required for Canvas integration as a commercial option for achieving this feature does not readily exist.

This plugin solution adequately addresses many of the issues present in the original system. Content would be submitted in one, easy to access location. The behavior and safeguards around the submission process would also be robust under this method as WPForms allows for a large degree of customizability. This proposition also automates content sorting into Google Drive, eliminating the potential for human error present in the current system.

The use of WordPress to house this submission process solution also comes with an additional benefit. The analysis of the Lode CMS did not originally cover the fact that on the day papers are distributed, all articles are manually transferred from the Google Drive to The Lode website for digital publication. Utilizing the well-established Lode website for this solution eliminates the need to transfer content to the site as it will already be submitted there.

Custom Google Add-on Solution

In order to solve the remaining issues with system visibility and layout transfer, a custom Google Workplace add-on. The reason for developing a custom plugin rather than using commercially available plugins is that available plugins do not resolve all issues in a single package. In the interest of having this solution be easy to operate by editors using their personal machines, a custom code solution became

more desirable. This plugin would utilize the Google API to add custom functionality and UI cards to the existing Google Drive and Google Docs UI.

The first addition of this plugin would be the addition of a system tracking checklist that would exist as a widget alongside the file navigator in Google Drive. This checklist would automatically show the status of articles within the pipeline and provide clear and upfront signaling of all changes. Alongside this checklist, a UI card would be added to the Google Docs UI that would present the option for editors to mark an article as edited. A popup card would also be used to clearly call out to editors asking if they intend to mark an article as edited whenever they try to navigate back to the file directory. This feature design prevents system updating errors and tracking visibility across the entire system.

The second addition this add-on would make is a more robust paragraph styles function within the Google Docs UI. This would allow for the creation of specific text styles that could be applied to text within Google Docs and be used to tag text content for easier transfer into Indesign. This part of the add-on would require the addition of XML—or similar Indesign compatible markdown language—as an option for exporting files. These features would greatly reduce the cognitive load placed on the layout editor, as properly tagged content could be directly loaded into Indesign through the Google Drive desktop app, rather than being copied and pasted over by hand.

Solution Justification

There were many factors considered in the evaluation of potential solutions to the current CMS issues. The driving rationale behind the proposed solution was the recognition that members of The Lode—especially student writers—are quickly replaced as students move through four year degree cycles. Any solution that was created would greatly benefit from taking advantage of the parts of the current CMS that students know well. Replacing Google Drive with an entirely new file management system was considered, but such a change would make it hard for current members to easily make the switch. It would also require a larger amount of training to achieve proficiency, as all new students would

be entirely foreign to the system. It is for this and other similar reasons that a solution that adapted to what users already know was prioritized.

Potential Shortcomings

This solution is not without its potential drawbacks. The execution of this solution would require an experienced development team that could create the moderately complex Google Workspace add-on presented. The submission integration with Canvas would also require a capable coder to get it working. These custom code solutions would also require very robust documentation and troubleshooting guides. Training materials would also be necessary for at least the Managing editor and Experience in Journalism instructor, as their positions would need to be trained on the backend and administration of both parts of the solution.

Upkeeping the solution also proves a potential downside. The current system relies on the inter compatibility of Google's products, but encounters issues where compatibility does not exist. The proposed solution adds compatibility concerns, as the Canvas LTI integration and Google Workspace add-on are prone to breaking due to the other components they interact with updating and becoming incompatible.

Evaluating the Solution

The success of the proposed solution in addressing the issues of the current CMS would require two things. Whether or not the new system is able to eliminate the pain points associated with automating processes is a major indicator of project success. The reduction of error rates in the submission process and in the article tracking process would also be crucial indicators of success. Finally, the qualitative data on perceived improvement of users that switch to the new system would be valuable to evaluating the success of the solution.

Conclusion

This project set out to analyze the operational friction within The Lode's content management system and propose a novel solution based on human factor principles. By relying on manual file-renaming conventions and disjointed software ecosystems, the current CMS places an unsustainable demand on the working memory and vigilance of student staff. As evidenced by the friction experienced by student writers and the Managing editor, the system currently prioritizes software constraints over human limitations.

The proposed solution—a hybrid integration of commercial WordPress plugins and a custom Google Workspace add-on—directly addresses these limitations without discarding the tools the students already know. By automating the gulf of execution between submission and storage, and by replacing manual status tracking with visual signifiers, the redesign satisfies the critical requirements of system visibility and error prevention. Furthermore, the strategic decision to build upon the existing Google ecosystem rather than replacing it ensures that the solution respects the high-turnover nature of a student organization, minimizing the learning curve for future cohorts.

Ultimately, this intervention shifts control from the user to the system. By eliminating the potential for errors of commission and automating tasks, the new CMS should result in an improved error rate during weekly production. The result is a production environment that is not only more efficient but more resilient to human error, allowing the organization to focus on its primary mission of student journalism.

References

Best wordpress forms plugin: Free online form builder. WPForms. (n.d.). https://wpforms.com/

Canvas LMS - REST API and extensions documentation. Overview of an LTI Launch - Canvas LMS REST API Documentation. (n.d.). https://documentation.instructure.com/doc/api/file.lti_launch_overview.html

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors*, 37(1), 32-64.

Google. (n.d.). Google drive API overview | google for developers. https://developers.google.com/workspace/drive/api/guides/about-sdk

Nielsen, J. (2024, February 20). *10 usability heuristics for user interface design*. Nielsen Norman Group. https://www.nngroup.com/articles/ten-usability-heuristics/

Prince. (2025, November 28). *File manager for google drive – integrate Google Drive*. WordPress.org. https://wordpress.org/plugins/integrate-google-drive/

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2), 257-285.