engineering proceedings

MDPI

Proceeding Paper

A Modular Framework for Cloud-Based Educational Content Delivery Systems: Design, Implementation, and Quality Evaluation †

Ritchfildjay L. Mariscal *, Reymark R. Boniza, Diosdado T. Erandio, Jr. and Angelou S. Tupaz

College of Industrial Technology and Teacher Education, Caraga State University, Cabadbaran 8605, Philippines; reymark.boniza@csucc.edu.ph (R.R.B.); diosdado.erandio@csucc.edu.ph (D.T.E.J.); angelou.tupaz@csucc.edu.ph (A.S.T.)

* Correspondence: rlmariscal@carsu.edu.ph; Tel.: +63-9291554677

† Presented at the 8th International Global Conference Series on ICT Integration in Technical Education & Smart Society, Aizuwakamatsu City, Japan, 20–26 January 2026.

Abstract

The increasing demand for scalable digital learning environments has created a need for cloud-based educational content delivery systems that support efficient resource management, platform accessibility, and quality-assured learning experiences. While low-code web development platforms have enabled rapid deployment of educational websites, many implementations remain content-centric and lack systematic architectural design, deployment frameworks, and software quality evaluation mechanisms. This study proposes a modular architecture for cloud-based educational content delivery systems that integrates content management, user access, resource delivery, platform administration, and quality monitoring components within a unified web-based environment. The proposed architecture adopts a structured development framework consisting of requirements analysis, system architecture design, prototype development, deployment configuration, performance testing, and quality evaluation. The framework is designed to support the rapid development of lightweight educational platforms using low-code technologies while maintaining software engineering principles related to reliability, usability, accessibility, compatibility, and performance efficiency. The architecture further incorporates cloud-hosted deployment strategies that facilitate scalable content distribution and cross-platform accessibility for technology-enhanced learning environments. To demonstrate the feasibility of the proposed architecture, a prototype implementation was developed using a low-code web platform and deployed as a cloud-based educational content delivery system. The prototype was evaluated by expert validators using selected software product quality characteristics derived from the ISO/IEC 25010 standard. The evaluation results indicated a high level of technical acceptability across multiple quality dimensions, including performance efficiency, reliability, usability, compatibility, accessibility, and capacity. The findings support the effectiveness of the proposed architecture as a practical framework for developing quality-assured educational delivery platforms. The study contributes a replicable systems architecture and implementation framework for educational content delivery applications. The proposed model provides guidance for the design, deployment, and evaluation of cloud-based learning platforms and offers a foundation for future integration with learning analytics, adaptive content delivery mechanisms, and intelligent educational support systems.

Check for updates

Academic Editors: Debopriyo Roy, George Fragulis and Peter Ilic

Published: 10 August 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Keywords: web-based content delivery system; low-code development; Google Sites; ISO/IEC 25010; educational technology; software quality evaluation

1. Introduction

The growing dependence on browser-based and cloud-hosted systems has reshaped the design and deployment of lightweight digital content delivery platforms. Modern web systems operate through distributed client–server interaction, where browsers request resources and servers return structured responses using web protocols and resources such as HTML documents, scripts, media, and other linked assets [1]. Cloud computing further extends this architecture by enabling convenient, on-demand network access to shared configurable computing resources that can be provisioned and released with minimal management effort [2]. The effectiveness of a cloud-based content delivery system depends not only on the availability of digital materials but also on the quality of its architecture, component organization, access configuration, deployment structure, interface consistency, and maintainability. A technically sound content delivery system requires coherent information architecture, including organization, labeling, navigation, search, and metadata structures that support efficient content retrieval and management [3]. Recent developments in low-code and no-code web development platforms have expanded opportunities for rapid system prototyping and deployment. Low-code development allows applications to be created through visual abstractions, graphical interfaces, reusable components, and minimal manual coding. However, low-code development also introduces software engineering concerns related to customization, third-party integration, database management, automated testing, governance, and maintainability [4]. Therefore, systems developed through low-code tools must still undergo requirements analysis, architecture planning, interface design, deployment configuration, and software quality evaluation. Google Sites represents one example of a low-code, cloud-hosted web development platform that supports browser-based site construction, modular page creation, file embedding, hyperlink navigation, collaborative editing, access control, and web publishing. Google's official documentation indicates that Google Sites allows users to insert files and resources such as Docs, Slides, YouTube videos, calendars, and maps, and to publish changes online [5]. The value of such a platform lies in how these built-in features can be organized into a modular content delivery architecture. Beyond usability, software product quality provides a broader technical basis for evaluating content delivery systems. The ISO/IEC 25010 [6] software product quality model defines quality characteristics relevant to software products and computer systems, including functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability, and portability [7]. These characteristics provide a technical basis for assessing whether a cloud-based content delivery prototype performs its intended functions, delivers resources efficiently, supports user interaction, remains dependable, and allows future modification. Accessibility and interoperability are also critical considerations in cloud-based system design. The World Wide Web Consortium develops standards and guidelines that support accessibility, internationalization, privacy, and security on the Web [8]. The Web Content Accessibility Guidelines organize accessibility around the principles of perceivability, operability, understandability, and robustness [9]. These principles are relevant in ensuring that a content delivery system can provide consistent access across devices, browsers, and user conditions.

https://doi.org/10.3390/engproc2026143058

2. Methodology

This study operationalized a modular framework for cloud-based educational content delivery systems through five major phases: requirements analysis, modular system architecture design, prototype development, implementation testing, and software quality evaluation. These phases guided the transformation of identified system requirements into a functional, cloud-hosted content delivery prototype. The developed system was evaluated using selected software quality characteristics derived from the ISO/IEC 25010 model.

2.1. Research Design

This study employed a design and development research approach anchored on prototype-based system development. It focused on the design, implementation, and quality evaluation of a modular cloud-based educational content delivery system using a low-code web development platform. The study emphasized system architecture, modular organization, cloud-hosted deployment, interface structure, resource integration, and technical quality. The development process followed a five-phase procedure adapted from iterative system development, in which a working prototype is designed, constructed, deployed, tested, evaluated, and refined based on identified functional and technical requirements. The expected output was a functional cloud-based educational content delivery prototype and a replicable modular framework for developing similar lightweight content delivery systems. Google Sites served as the low-code implementation platform because it supports modular page construction, browser-based access, embedded resource integration, hyperlink-based navigation, access configuration, collaborative editing, and cloud-hosted publishing. The educational content used in the prototype functioned only as case content for testing the capability of the system architecture to organize, deliver, and manage structured digital resources. Thus, the primary concern of the methodology was the development and evaluation of the content delivery system as a software-oriented and cloud-hosted platform.

2.2. System Development

The development of the system was carried out through five sequential but interrelated phases.

2.2.1. Requirements Analysis

The requirements analysis phase involved identifying the functional and technical requirements of the proposed content delivery system. The researchers examined the type of digital content to be hosted, expected access environment, modular page structure, navigation requirements, resource integration needs, access configuration, and device and browser compatibility considerations. The analysis also considered the need for a lightweight, scalable, and maintainable system that could operate through standard web browsers without requiring separate software installation. The output of this phase was a set of system requirements that guided the architecture and prototype development. The functional requirements included modular content organization, structured page navigation, embedded multimedia support, downloadable resource integration, hyperlink access, content retrieval, and basic platform administration. The technical requirements included browser accessibility, responsive display, cloud hosting, ease of updating, platform compatibility, interface consistency, link reliability, and maintainability.

2.2.2. Modular System Architecture Design

The modular system architecture design phase focused on the logical and structural organization of the prototype. The researchers designed the homepage, content modules,

resource pages, embedded media sections, interaction or activity pages, reference page, navigation menus, access pathways, and deployment configuration. The architecture followed a modular structure to allow content and system functions to be divided into manageable components that could be accessed, updated, expanded, and maintained independently. It consisted of six major components: user interface layer, content management layer, resource integration layer, navigation and access layer, platform administration layer, and cloud deployment layer. The user interface layer defined the visual layout, page presentation, readability, and interaction structure. The content management layer handled the organization, sequencing, and updating of textual and visual resources. The resource integration layer supported the embedding of files, images, videos, links, and other web-based learning materials. The navigation and access layer provided internal and external hyperlink connections, menu structures, and access pathways among pages and resources. The platform administration layer supported content updating, page editing, access configuration, and system maintenance. The cloud deployment layer enabled browser-based access through a cloud-hosted publishing environment. This phase produced the system blueprint that guided prototype development. The blueprint established the relationship among system components and ensured that the prototype was developed as an integrated modular content delivery system rather than as a collection of independent web pages.

2.2.3. Prototype Development

The prototype development phase involved the actual construction of the system using Google Sites. The researchers created the site structure, configured the homepage, organized the content modules, inserted media resources, established internal navigation links, embedded external resources, configured access settings, and arranged downloadable materials within the platform. The use of a low-code platform allowed the prototype to be constructed through visual editing tools, predefined layout components, drag-and-drop content blocks, embedded Google Workspace resources, hyperlink structures, and built-in cloud publishing features. The prototype did not require server-side programming, database configuration, or external hosting management, making it suitable for rapid development, iterative revision, and lightweight deployment. The prototype was developed as a browser-accessible system composed of modular pages and integrated resources. The sample educational content included text, diagrams, images, embedded videos, hyperlinks, activity files, and reference materials. These materials were used only to demonstrate how the proposed framework could support organized content delivery, resource integration, access configuration, and cloud-based deployment.

2.2.4. Implementation Testing

The implementation testing phase involved the initial deployment and technical checking of the completed prototype. The researchers tested page accessibility, navigation link functionality, embedded media display, file access, device responsiveness, browser compatibility, content visibility, interface consistency, and access permission settings. The testing procedure focused on system behavior and deployment readiness. Technical issues such as broken links, inconsistent formatting, inaccessible files, unclear navigation paths, or restricted resource access were recorded and addressed before expert evaluation. This phase ensured that the prototype was technically functional, cloud-accessible, and ready for formal quality assessment.

2.2.5. System Evaluation

The system evaluation phase determined the technical quality and acceptability of the developed prototype. Expert validators assessed the system using a structured evaluation instrument based on selected software product quality characteristics derived from the

ISO/IEC 25010 model. The evaluation focused on performance efficiency, reliability, usability, compatibility, accessibility, maintainability, and capacity. The validation results were used to determine the strengths, limitations, and areas for improvement of the prototype. The evaluation treated the developed platform as a cloud-based software-oriented content delivery system rather than merely as an educational website.

2.3. Expert Validators

The study involved five purposively selected expert validators who assessed the technical quality and acceptability of the developed prototype. They were selected based on their background in information technology, computer systems, web development, educational technology, electronics, digital content development, or related technical fields. The validators were chosen using the following criteria: (1) relevant academic or professional background in information technology, computer science, web systems, educational technology, electronics, or digital content development; (2) at least three years of experience in software-related work, technology-based system use, web platform evaluation, technical content development, or related areas; (3) familiarity with web-based platforms, content management systems, cloud-hosted environments, or digital resource delivery systems; and (4) willingness to participate in the validation process. Purposive sampling was used because the study required evaluators with specialized knowledge relevant to the technical and functional features of the prototype. Their assessment provided expert-based evidence on the system's architecture, usability, reliability, maintainability, compatibility, accessibility, and deployment readiness.

2.4. Research Instrument

The main instrument used in the study was a structured expert validation questionnaire designed to evaluate the technical quality of the developed system. The instrument was adapted from selected criteria of the ISO/IEC 25010 software product quality model and assessed the prototype based on performance efficiency, reliability, usability, compatibility, accessibility, maintainability, and capacity. Performance efficiency referred to the responsiveness and loading behavior of the prototype when accessed through a browser. Reliability referred to the consistency of system access, functional links, and availability of embedded resources. Usability referred to the clarity of the interface layout, ease of navigation, readability, and user interaction. Compatibility referred to the ability of the system to operate across available devices, browsers, and embedded resource environments. Accessibility referred to the ease with which users could access content, pages, files, and embedded materials under different user and device conditions. Maintainability referred to the ease of modifying, updating, reorganizing, and expanding the system. Capacity referred to the ability of the system to accommodate modular content, embedded resources, and future expansion within the selected cloud-based platform. The questionnaire used a four-point Likert scale to obtain expert judgments on the quality and acceptability of the prototype. The four-point scale was used to encourage clear evaluation by avoiding a neutral midpoint. The scale interpretation was as follows:

Scale	Range	Descriptive Interpretation
4	3.26–4.00	Very Highly Acceptable/Very High Quality
3	2.51–3.25	Highly Acceptable/High Quality
2	1.76–2.50	Moderately Acceptable/Moderate Quality
1	1.00–1.75	Not Acceptable/Needs Major Improvement

The instrument also included an open-ended section where validators could provide comments and recommendations for system refinement.

2.5. Data Gathering Procedure

The data gathering procedure began after the prototype was completed, deployed, and preliminarily checked for technical functionality. The researchers verified the accessibility of pages, links, embedded resources, files, navigation elements, access permissions, and layout structures before expert validation. The expert validators were then provided access to the prototype and the evaluation instrument. They were instructed to examine the system based on its architecture, interface structure, navigation flow, content organization, resource integration, accessibility, compatibility, maintainability, and deployment readiness. The validators independently rated the prototype using the structured questionnaire and provided written comments for improvement. After the evaluation, the researchers compiled the ratings and comments. The quantitative ratings were analyzed using descriptive statistics, while the qualitative comments were reviewed to identify technical issues and possible refinements. The results served as the basis for determining the overall quality and acceptability of the developed system.

2.6. Data Analysis

The data gathered from the expert validation were analyzed using descriptive statistics. The mean was used to determine the average rating of each software quality characteristic, while the grand mean was used to determine the overall technical acceptability of the developed system. The standard deviation was used to describe the consistency of the validators' ratings. Qualitative comments from the expert validators were reviewed and summarized to identify recurring technical observations, interface issues, navigation concerns, access limitations, compatibility concerns, maintainability issues, and recommendations for system improvement. These comments were used to support the interpretation of the quantitative results and to guide prototype refinement. No inferential test was conducted because the study did not aim to determine relationships, differences, or effects. The analysis focused only on system quality and prototype acceptability. The primary objective was to evaluate the developed cloud-based educational content delivery system as a functional, modular, software-oriented prototype.

3. Results and Discussion

The study produced a modular cloud-based educational content delivery system prototype designed to organize, integrate, deploy, and evaluate structured digital learning resources through a lightweight, browser-accessible environment. Although Passive Electronic Devices was used as the sample content domain, the study focused on the system framework rather than on the instructional content itself. The results are presented according to the five major stages of the proposed framework: requirements analysis, modular system architecture design, cloud-based prototype development, deployment and technical testing, and software quality evaluation. As shown in Figure 1, the framework follows a cyclical development and evaluation process in which findings from deployment testing and quality evaluation may be used to revise requirements, improve the architecture, refine the prototype, and support redeployment. Thus, the figure illustrates an iterative improvement cycle for developing and evaluating a modular cloud-based educational content delivery system.

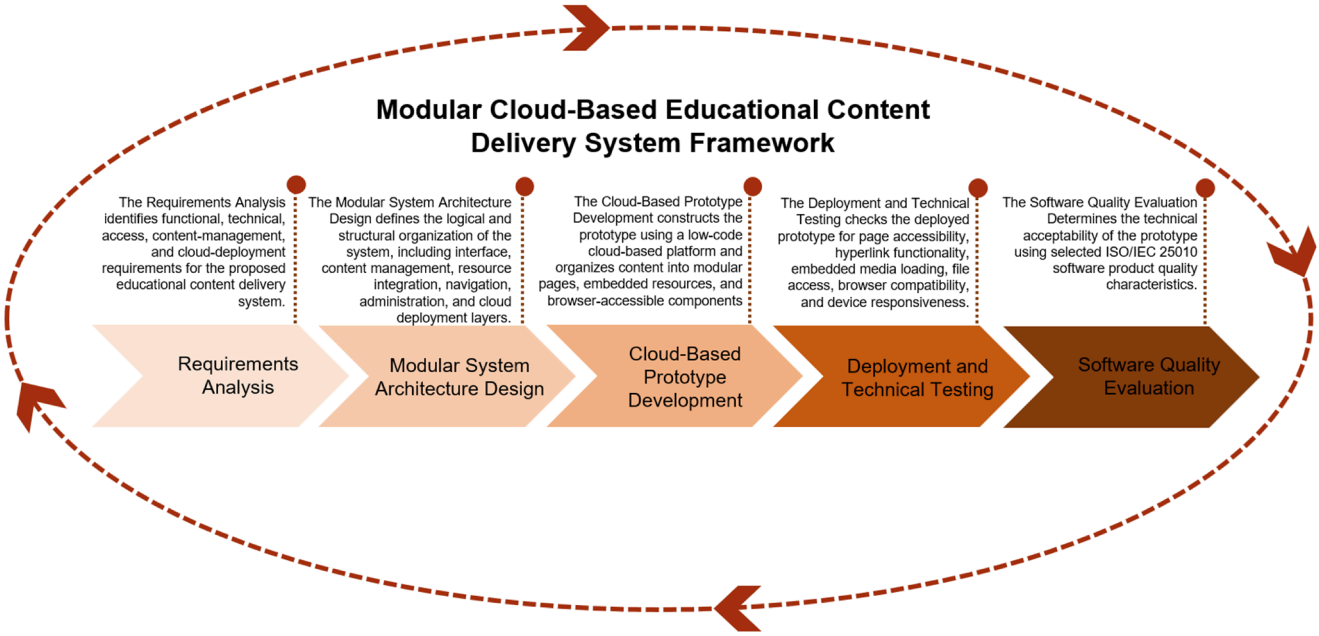


Figure 1. The Modular Cloud-based Content Delivery System Framework.

3.1. Baseline Digital Access Profile and System Design Implications

Table 1 presents the baseline digital access profile that informed the functional and technical requirements of the proposed modular cloud-based educational content delivery system. The data describe the existing access condition prior to the proposed system and should not be interpreted as a performance measure of the prototype. Rather, the results provide evidence of how users commonly accessed and exchanged digital learning materials without a dedicated modular content delivery architecture. The results show that cellphone use recorded the highest number of responses among gadgets, indicating a strongly mobile-oriented access environment. This finding supports the requirement that the proposed system should be browser-accessible, mobile-responsive, lightweight, and readable on small screens. This interpretation is consistent with the mobile-learning literature, which recognizes mobile devices as widely used tools for flexible access to learning resources in higher education [10]. However, mobile access alone does not guarantee effective content delivery because its success also depends on device capability, connectivity, interface design, and user readiness [11,12].

The high frequency of FB Messenger use indicates that digital content access was largely mediated through communication-based platforms. This suggests that users relied on familiar and accessible tools for exchanging learning materials. While messaging platforms may support rapid communication and peer interaction, they are not primarily designed as structured content delivery systems. Prior research on learning management systems and social media indicates that social platforms can support interaction and quick sharing, but they may also create issues related to distraction, fragmented access, technical support, and security when used as substitutes for formal learning environments [13,14]. The low recorded use of Google Sites and general websites further suggests that structured browser-based repositories were underutilized in the existing access environment. This does not imply that web-based platforms are ineffective; rather, it shows that users had not commonly adopted organized online spaces for persistent access, modular navigation, resource integration, and centralized updating. Studies on learning management systems and web-based learning environments emphasize the importance of continuity, reliability, privacy, clear navigation, usable interface design, and coherent access to learning tools and resources [14,15]. On this basis, the proposed system was conceptualized as a design re-

sponse to a mobile-oriented but fragmented access environment. By centralizing resources in a browser-accessible platform, organizing content into modular pages, embedding files and media within one environment, and allowing updates through a cloud-hosted interface, the framework offers a more systematic alternative to scattered message-based file sharing. Its expected contribution, therefore, is not to replace full-scale institutional learning management systems but to demonstrate how a lightweight modular architecture can improve the organization, retrieval, navigation, and maintenance of educational content.

Table 1. Baseline digital access profile and system design implications.

Access Indicator	Evidence from AY 2020–2021 Data	Interpretation of Existing Condition	Design Implication for the Proposed System
Mobile-based access	Cellphone use was the highest among gadgets, with 2128 responses in the 1st semester and 1076 responses in the 2nd semester.	Users commonly accessed digital content through mobile devices.	The system should support mobile-responsive layouts, simplified navigation, readable pages, and lightweight content loading.
Communication-based content sharing	FB Messenger recorded 4978 responses in the 1st semester and 4416 responses in the 2nd semester.	Digital content access was largely mediated through messaging platforms.	The proposed system should provide a more structured alternative to scattered message-based file sharing while retaining ease of access.
Limited use of structured web platforms	Google Sites recorded only 8 responses in the 1st semester and 7 responses in the 2nd semester.	Structured cloud-based content delivery platforms were underutilized.	There is an opportunity to introduce a modular cloud-based system for organized content delivery, resource integration, and centralized access.
Low use of general websites	Website use was reported at 36 responses in the 1st semester and 17 responses in the 2nd semester.	Existing content access was not strongly supported by persistent web-based repositories.	The system should function as a stable browser-accessible repository with persistent links and organized page hierarchy.
Moderate laptop and computer use	Laptop use decreased from 386 to 186 responses, while computer use decreased from 135 to 57 responses.	Desktop-based access was present but less dominant than mobile-based access.	The system should remain compatible across both desktop and mobile environments.
Platform fragmentation	Users relied on different devices and platforms for access.	Content may be distributed across multiple channels, files, and message threads.	A centralized cloud-hosted content delivery system can improve organization, retrieval, updating, and maintainability.

Note. The table interprets the institutional technology-resource data as baseline access conditions prior to the proposed modular cloud-based content delivery system. The values do not measure the performance of the proposed system but provide justification for its functional and technical requirements.

3.2. Layered Modular Architecture of the Cloud-Based Content Delivery System

In response to the baseline access condition, the proposed system was organized into five architectural layers: user interface, content organization, resource integration, navigation, and deployment. These layers separate the major functions of the system so that page presentation, content arrangement, resource embedding, access pathways, and cloud publishing can be managed systematically. This structure is consistent with information architecture principles, which emphasize the organization, labeling, navigation, and structuring of information in web-based environments [16,17]. The user interface

layer defined the visual structure of the prototype, including the page layout, headings, content blocks, embedded materials, and overall page presentation. Its purpose was to provide a consistent interface across pages and reduce the difficulty of locating learning resources. The website usability literature supports the importance of layout, readability, and interface organization in improving user interaction with web-based systems [17,18]. The content organization layer arranged the sample technical content into modular sections, including overview, resistors, capacitors, inductors, activities, references, and related resources. Passive Electronic Devices was used only as case content to test the proposed architecture. The main value of this layer is that it allows learning resources to be grouped, updated, expanded, and retrieved by module instead of being managed as disconnected files. This supports software modularity principles, where well-structured components can improve maintainability and reduce complexity [19,20]. However, modularity requires clear naming conventions, documentation, and update procedures; otherwise, poorly maintained modules may still create management difficulties [20]. The resource integration layer allowed the prototype to embed or link images, videos, Google Drive files, Google Docs, Google Forms, hyperlinks, and other digital resources within a single platform. This layer supports more seamless access by consolidating related materials in one browser-accessible environment. Cloud-based e-learning studies support this approach by noting that cloud environments can improve access flexibility, resource sharing, scalability, and reduced infrastructure requirements [20,21]. However, these benefits depend on proper management of permissions, file ownership, embedded links, and platform availability. The navigation layer connected the homepage, content pages, activity pages, and reference materials through internal and external links. This layer was necessary to support efficient retrieval through persistent menus, page hierarchy, and organized access pathways. Information architecture and website-quality studies similarly emphasize the role of clear navigation in helping users locate and use information effectively [16,17]. The deployment layer used the cloud-hosted publishing environment of Google Sites. This enabled browser-based access without local installation, independent server configuration, or separate hosting management. While this layer reduces technical barriers to deployment, the system remains dependent on internet connectivity, Google account permissions, and the Google ecosystem. Therefore, the architecture is best understood as a lightweight content delivery framework rather than a full replacement for database-driven platforms or enterprise-level learning management systems that require advanced authentication, analytics, tracking, and migration control.

3.3. Prototype Implementation

The prototype, as shown in Figure 2, was implemented using Google Sites as the low-code cloud-based development platform. The implementation process involved creating the site structure, configuring the homepage, organizing modular pages, embedding resources, inserting downloadable files, connecting hyperlinks, and publishing the system for browser-based access. This process demonstrated how a low-code platform could be configured into a lightweight educational content delivery system without requiring extensive manual coding, database management, or independent server deployment. The low-code development literature supports this interpretation because low-code platforms are generally designed to accelerate application development through visual interfaces, reusable components, and reduced programming requirements [22]. However, low-code development should not be viewed as free from technical constraints. Studies report challenges related to customization, platform adoption, database management, third-party integration, and automated testing [23,24]. The homepage served as the main entry point and central navigation hub of the prototype. It introduced the content domain, provided

access to major sections, and established a consistent interface structure through a banner, overview section, navigation menu, and access points to modular content pages. From a system-design perspective, the homepage was intended to reduce the number of steps required to locate resources and to provide a unified entry point for users. This is consistent with website usability and information architecture studies, which emphasize that clear layout, navigation, labeling, and information organization influence how efficiently users locate and use web-based resources [17,18]. The modular pages contained structured technical content, images, diagrams, embedded videos, hyperlinks, and activity files. The more significant result is not the subject matter itself but the demonstration that content can be modularized, embedded, linked, and published through a unified cloud-hosted environment. The implementation also showed the practical value of integrating Google Workspace resources within the same platform. Documents, forms, videos, links, and downloadable files could be embedded or connected within the Google Sites environment.

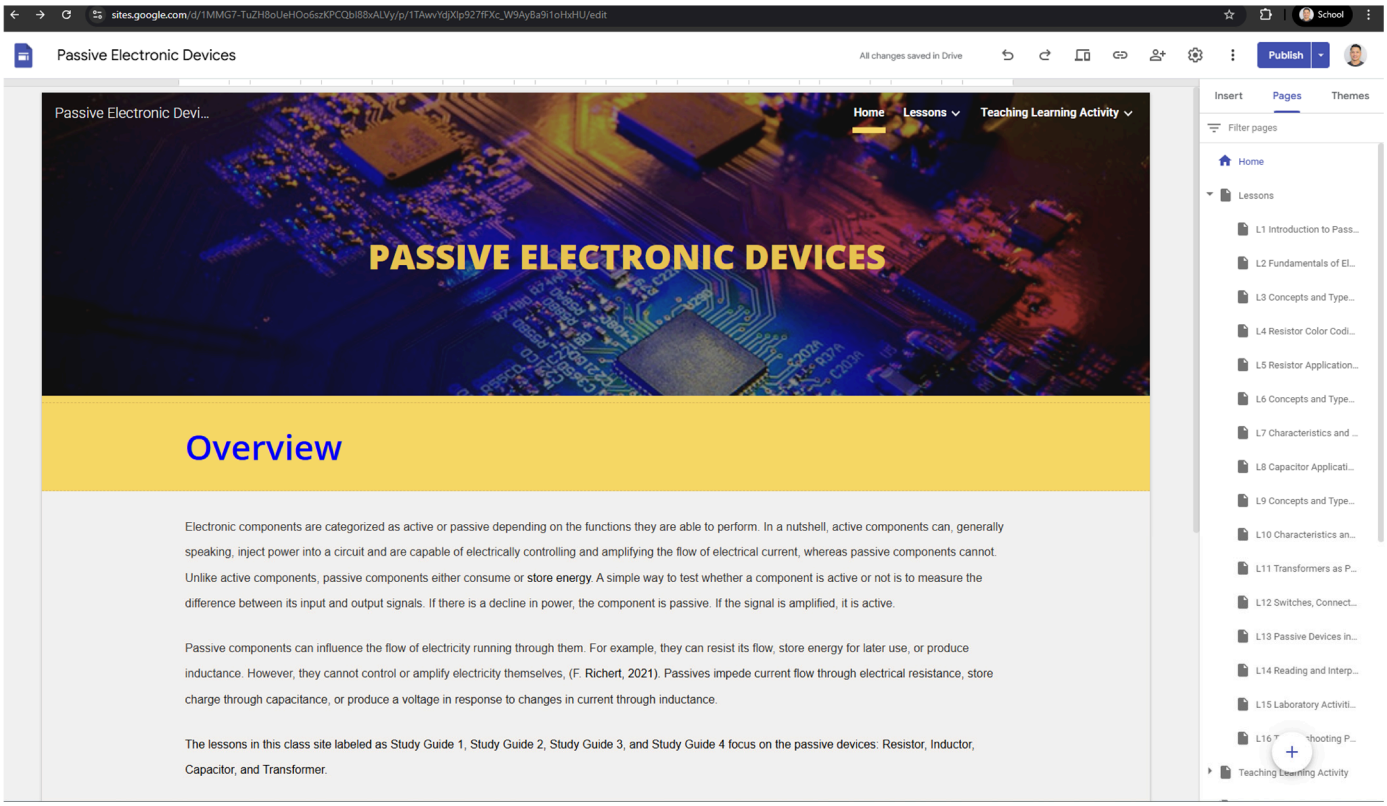


Figure 2. Homepage interface of the Google Sites-based modular cloud-based educational content delivery prototype.

The findings may therefore be interpreted as evidence of implementation feasibility for a lightweight prototype. The system demonstrates how a modular, cloud-hosted structure can support content organization and browser-based retrieval. However, it remains dependent on internet connectivity, platform availability, file permissions, and the Google ecosystem. These conditions limit the extent to which the prototype can be generalized as a replacement for full learning management systems, database-driven repositories, or enterprise-level content management systems.

3.4. Deployment and Technical Testing

The deployment and technical testing phase examined whether the proposed modular cloud-based content delivery prototype could function as a browser-accessible system after publication. The prototype was tested in terms of page accessibility, hyperlink functionality,

embedded media loading, file access, interface consistency, browser-based viewing, and device responsiveness. These checks were necessary because a content delivery system must operate reliably after publication and not merely contain digital resources. Website-quality and usability studies emphasize that functional access, navigation clarity, interface consistency, and user experience are important criteria in evaluating web-based systems [25]. Similarly, ISO/IEC 25010-based software quality evaluation highlights reliability, usability, compatibility, performance efficiency, and maintainability as relevant characteristics for assessing software-oriented systems [25,26]. The deployment process confirmed that the prototype could be published through Google Sites and accessed through a standard browser without local installation, independent server configuration, or separate hosting management. This result supports the technical feasibility of using a low-code cloud-based platform for lightweight educational content delivery. However, successful publication should not be interpreted as evidence of enterprise-level robustness. The deployment result only confirms that the prototype can operate under the tested conditions and that its basic access, navigation, embedded resources, and page structures functioned as intended. The technical testing also showed the need to consider platform-related constraints. Since the system depends on Google Sites and linked Google Workspace resources, its reliability is affected by internet connectivity, account permissions, file ownership, sharing settings, and platform availability. Therefore, broader implementation would require stronger protocols for access control, backup, link maintenance, version management, and content updating.

3.5. Expert Evaluation Using ISO/IEC 25010 Software Product Quality Characteristics

Table 2 presents the expert-based quality evaluation of the proposed modular cloud-based educational content delivery prototype using selected ISO/IEC 25010 software product quality characteristics. Unlike the baseline technology-resource data, which describe the existing access condition, the results in Table 2 represent the technical acceptability of the developed prototype after it was configured as a cloud-hosted, browser-accessible content delivery system. The overall mean of 3.50, interpreted as "Always," indicates that the prototype demonstrated a high level of technical acceptability as a lightweight content delivery platform. The strongest quality areas were performance efficiency and reliability, both with mean scores of 3.67. These results suggest that the prototype was perceived as responsive, stable, and capable of supporting access to structured pages, embedded resources, and linked materials. This interpretation is consistent with the software quality literature, which positions performance efficiency and reliability as essential indicators of system behavior and operational dependability under ISO/IEC 25010-based evaluation [25].

Usability obtained a mean of 3.56, indicating that the prototype's interface, navigation structure, readability, and access pathways were generally acceptable. This result is important because a content delivery system must not only store materials but also enable users to locate and retrieve them efficiently. Website-quality and information architecture studies support this interpretation, emphasizing that navigation, labeling, layout, and interface organization influence how users access and use web-based information [26]. Compatibility also received a favorable rating, with a mean of 3.50. This suggests that the prototype was perceived to operate adequately across available browsers, devices, and embedded resource environments. This result is relevant to the mobile-oriented and platform-diverse access condition identified in the requirements analysis. A cloud-based content delivery system must therefore support cross-device access and resource interoperability. Cloud-based e-learning studies similarly identify accessibility, resource sharing, scalability, and reduced infrastructure requirements as advantages of cloud-supported educational systems [20,21]. However, this interpretation should be limited to technical feasibility because the study did not conduct large-scale user testing across varied devices, network conditions, or insti-

tutional deployment settings. The lower-rated areas were functional suitability, security, maintainability, and portability, with mean scores ranging from 3.36 to 3.44. These results do not indicate that the prototype failed; rather, they identify areas where the proposed system requires refinement before broader implementation. Functional suitability may be strengthened by improving content completeness, link precision, and alignment between system functions and user requirements. Security requires attention to authentication, permission settings, access control, and identity verification. Maintainability depends on content-update procedures, file organization, naming conventions, version control, and documentation. Portability requires clearer strategies for backup, migration, and platform transfer. These limitations are consistent with the low-code development literature, which notes that low-code platforms can accelerate implementation but may introduce challenges related to customization, platform dependence, database control, third-party integration, and automated testing [2,24].

Table 2. System-oriented interpretation of ISO/IEC 25010 expert evaluation results.

System Concern	Related ISO/IEC 25010 Characteristic	Mean	Interpretation	Design Meaning
Content delivery responsiveness	Performance Efficiency	3.67	Very Highly Acceptable	The prototype was perceived as responsive enough for browser-based access and embedded resource loading.
Continuous content access	Reliability	3.67	Very Highly Acceptable	The prototype supported stable access to pages, links, and resources.
Interface and navigation clarity	Usability	3.56	Very Highly Acceptable	The system interface and navigation structure were generally clear and manageable.
Cross-platform access	Compatibility	3.50	Very Highly Acceptable	The prototype could operate across available devices, browsers, and embedded resource environments.
Cloud-based transferability	Portability	3.44	Very Highly Acceptable	The system was accessible through browsers but still needs stronger migration and backup readiness.
System updating and revision	Maintainability	3.43	Very Highly Acceptable	The prototype can be modified but requires clearer content-update protocols and version management.
Access control and identity assurance	Security	3.40	Very Highly Acceptable	The prototype needs stronger authentication and access-control mechanisms for wider institutional use.
Completeness of intended functions	Functional Suitability	3.36	Very Highly Acceptable	The prototype met basic functional requirements but needs refinement in content completeness and system-function alignment.
Overall technical acceptability	All characteristics	3.50	Very Highly Acceptable	The prototype is technically acceptable as a modular cloud-based content delivery system.

4. Conclusions

This study designed, implemented, and evaluated a modular framework for cloud-based educational content delivery systems, with emphasis on requirements analysis, system architecture, low-code prototype development, deployment readiness, and software quality evaluation. The baseline access data showed that digital content access was largely

mobile-based and communication-platform-centered, particularly through cellphones and FB Messenger, while structured web-based repositories were minimally used. These findings were interpreted not as outcomes of the proposed system but as evidence of the existing access condition without a dedicated modular content delivery architecture. In response to this condition, the proposed framework was organized into five major components: requirements analysis, modular system architecture design, cloud-based prototype implementation, deployment and technical testing, and software quality evaluation. Using Google Sites as a low-code development platform, the prototype demonstrated that educational resources could be organized into modular pages, integrated with multimedia and downloadable files, connected through persistent navigation, and deployed through a browser-accessible cloud environment. Expert evaluation using selected ISO/IEC 25010 software product quality characteristics indicated that the prototype was technically acceptable as a lightweight cloud-based content delivery system, with an overall mean of 3.50. The strongest areas were performance efficiency and reliability, followed by usability and compatibility, suggesting that the prototype was perceived as responsive, stable, navigable, and suitable for basic browser-based content delivery. Lower-rated areas, including functional suitability, security, maintainability, and portability, indicate the need for further refinement in content completeness, authentication, access control, update procedures, version management, backup, and migration readiness. The study implies that a modular cloud-based framework can support more systematic digital resource management by centralizing access, organizing content into modules, integrating resources within one platform, and enabling quality-assured deployment. However, the study was limited to prototype development and expert validation. It did not measure learning outcomes, learner satisfaction, long-term engagement, or institutional-scale performance. Future studies should conduct user-based usability testing, accessibility auditing, security assessment, performance monitoring, cross-platform comparison, and broader implementation across different content domains, devices, browsers, and network conditions.

Author Contributions: Conceptualization, R.R.B., D.T.E.J., A.S.T. and R.L.M.; methodology, R.R.B., D.T.E.J., A.S.T. and R.L.M.; software, R.R.B., D.T.E.J., A.S.T. and R.L.M.; validation, R.R.B., D.T.E.J., A.S.T. and R.L.M.; formal analysis, R.R.B., D.T.E.J., A.S.T. and R.L.M.; investigation, R.R.B., D.T.E.J., A.S.T. and R.L.M.; resources, R.R.B., D.T.E.J., A.S.T. and R.L.M.; data curation, R.R.B., D.T.E.J., A.S.T. and R.L.M.; writing—original draft preparation, R.R.B., D.T.E.J., A.S.T. and R.L.M.; writing—review and editing, R.R.B., D.T.E.J., A.S.T. and R.L.M.; visualization, R.R.B., D.T.E.J., A.S.T. and R.L.M.; supervision, R.R.B. and R.L.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. The authors wishes to extend gratitude to Caraga State University for the funding support.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author by contacting rlmariscal@carsu.edu.ph.

Acknowledgments: The authors wish to extend their gratitude to Caraga State University for support extended to its faculty and researchers.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FB	Facebook
LMS	Learning Management System
ISO/IEC	International Organization for Standardization/International Electrotechnical Commission
AY	Academic Year
ICT	Information and Communication Technology
URL	Uniform Resource Locator

References

1. How the Web Works-Learn Web Development | MDN. MDN Web Docs. Available online: https://developer.mozilla.org/en-US/docs/Learn_web_development/Getting_started/Web_standards/How_the_web_works (accessed on 18 June 2026).
2. MAl Alamin, A.; Malakar, S.; Uddin, G.; Afroz, S.; Bin Haider, T.; Iqbal, A. An Empirical Study of Developer Discussions on Low-Code Software Development Challenges. In Proceedings of the 2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR), Madrid, Spain, 17–19 May 2021; pp. 46–57. [CrossRef]
3. Publish & Share Your Site-Sites Help. Available online: https://support.google.com/sites/answer/6372880 (accessed on 18 June 2026).
4. 10 Usability Heuristics for User Interface Design. Nielsen Norman Group. Available online: https://www.nngroup.com/articles/ten-usability-heuristics/ (accessed on 18 June 2026).
5. Cai, C.-H.; Sun, J.; Dobbie, G. Measuring the Quality of B Abstract Machines with ISO/IEC 25010. *arXiv* **2020**, arXiv:2003.02619. [CrossRef]
6. "W3C," W3C. Available online: https://www.w3.org/ (accessed on 18 June 2026).
7. Boyalakuntla, K.; Venigalla, A.S.M.; Chimalakonda, S. WAccess—A Web Accessibility Tool based on WCAG 2.2, 2.1 and 2.0 Guidelines. *arXiv* **2021**, arXiv:2107.06799. [CrossRef]
8. Web Content Accessibility Guidelines (WCAG) 2.2. Available online: https://www.w3.org/TR/WCAG22/ (accessed on 27 June 2026).
9. Crompton, H.; Burke, D. The use of mobile learning in higher education: A systematic review. *Comput. Educ.* **2018**, *123*, 53–64. [CrossRef]
10. Naveed, Q.N.; Choudhary, H.; Ahmad, N.; Alqahtani, J.; Qahmash, A.I. Mobile Learning in Higher Education: A Systematic Literature Review. *Sustainability* **2023**, *15*, 13566. [CrossRef]
11. Alturki, U.; Aldraiweesh, A. Students' Perceptions of the Actual Use of Mobile Learning during COVID-19 Pandemic in Higher Education. *Sustainability* **2022**, *14*, 1125. [CrossRef]
12. Sophonhiranrak, S. Features, barriers, and influencing factors of mobile learning in higher education: A systematic review. *Heliyon* **2021**, *7*, e06696. [CrossRef] [PubMed]
13. Turnbull, D.; Chugh, R.; Luck, J. Learning Management Systems and Social Media: A Case for Their Integration in Higher Education Institutions. In *Research in Learning Technology*; Taylor & Francis Open: London, UK, 2023; Volume 31. Available online: https://eric.ed.gov/?id=EJ1398263 (accessed on 27 June 2026).
14. Zanjani, N. The important elements of LMS design that affect user engagement with e-learning tools within LMSs in the higher education sector. *Australas. J. Educ. Technol.* **2017**, *33*, 19–31. [CrossRef]
15. An Overview of the Common Elements of Learning Management System Policies in Higher Education Institutions | TechTrends | Springer Nature Link. Available online: https://link.springer.com/article/10.1007/s11528-022-00752-7 (accessed on 27 June 2026).
16. Guizani, M. A Decade of Information Architecture in HCI: A Systematic Literature Review. *arXiv* **2022**, arXiv:2202.13412. [CrossRef]
17. e-Repositori UPF: Website Quality: An Analysis of Scientific Production. Available online: https://repositori.upf.edu/items/2206452c-9d57-4f4b-891e-afc940d148ac?locale=en (accessed on 27 June 2026).
18. Munim, K.M.; Islam, I.; Milton, M.R.; Ara, L.A.; Al Faisal, F.; Islam, M.N. Exploring the impact of design technique on usability: A case study on designing the eHealth websites using card sorting and interactive dialogue model. *Eng. Rep.* **2024**, *6*, e12738. [CrossRef]
19. Naveen; Singh, R.; Rathee, A.; Naveen, N. Improving Software Modularity Using Software Remodularization: Challenges and Opportunities. *ITM Web Conf.* **2023**, *54*, 01008. [CrossRef]
20. Mapping Knowledge Area Analysis in E-Learning Systems Based on Cloud Computing. Available online: https://www.mdpi.com/2079-9292/12/1/62 (accessed on 27 June 2026).

21. E-learning in the Cloud Computing Environment: Features, Architecture, Challenges and Solutions | International Journal of Engineering Pedagogy (iJEP). Available online: https://online-journals.org/index.php/i-jep/article/view/47109 (accessed on 27 June 2026).
22. Modelling in Low-Code Development: A Multi-Vocal Systematic Review | Software and Systems Modeling | Springer Nature Link. Available online: https://link.springer.com/article/10.1007/s10270-021-00964-0 (accessed on 27 June 2026).
23. Luo, Y.; Liang, P.; Wang, C.; Shahin, M.; Zhan, J. Characteristics and Challenges of Low-Code Development: The Practitioners' Perspective. In Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), Bari, Italy, 11–15 October 2021; pp. 1–11. [CrossRef]
24. *ISO/IEC 25010:2011*; ISO. Available online: https://www.iso.org/standard/35733.html (accessed on 27 June 2026).
25. Gullikson, S.; Blades, R.; Bragdon, M.; McKibbon, S.; Sparling, M.; Toms, E.G. The impact of information architecture on academic web site usability. *Electron. Libr.* **1999**, *17*, 293–304. [CrossRef]
26. Dekkers, T.; Melles, M.; Vehmeijer, S.B.W.; de Ridder, H. Effects of Information Architecture on the Effectiveness and User Experience of Web-Based Patient Education in Middle-Aged and Older Adults: Online Randomized Experiment. *J. Med. Internet Res.* **2021**, *23*, e15846. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.