
Design and Implementation of a Web Based E-Archive System

Virginia G. Figueroa, EdD¹, Shawn A. Dacles, EdD², Lavena Gee A. Boluso³, Roden B. Pelegrino Jr.⁴

^{1,2,3,4}Capiz State University, Philippines

ABSTRACT: This study presents the design and implementation of a web based E-Archive System intended to organize, review, publish, and retrieve student research outputs within an academic institution. The project addressed fragmented storage, inconsistent review workflows, and limited access to completed research documents by providing separate interfaces for students, faculty members, administrators, and public readers. A developmental research approach was used. The existing PHP and MySQL application was examined through source-code inventory, database-schema review, workflow tracing, interface revision, and PHP 5.6 compatibility testing. The resulting system supports account registration and approval, group assignment, research proposal submission, chapter-level document management, faculty announcements and comments, administrative approval and publication, public search, and configurable branding. The reviewed codebase contains 80 application PHP files outside bundled editor samples and 15 tables in the supplied database schema. Syntax verification confirmed that the revised application files parse under PHP 5.6.32. The interface was also consolidated around administrator-controlled title, logo, primary, banner, and text colors so public, student, faculty, and administrator pages share a consistent identity. Technical inspection nevertheless identified limitations in legacy request handling, unrestricted research-document uploads, hard-coded local URLs, and incomplete database constraints. Because the available evidence consisted of source files and a schema rather than participant data, findings are limited to implemented functionality, code-level observations, and repeatable compatibility checks; they do not represent measured user satisfaction. The study concludes that the system establishes a workable role-based research archive and workflow foundation, but production deployment requires systematic security hardening, controlled file preservation, database migration, backup, and formal usability evaluation.

KEYWORDS: academic archive, digital repository, document management, research workflow, web application

INTRODUCTION

Universities produce research proposals, chapter drafts, final manuscripts, and supporting files that must remain available beyond a single class or academic year. When these records are stored in personal devices, email threads, or disconnected office folders, the institution has difficulty determining which copy is current, whether a paper has been reviewed, and who may retrieve it. An institutional repository addresses part of this problem by collecting the intellectual output of an organization and making it discoverable under an explicit stewardship arrangement. Lynch (2003) describes such repositories as infrastructure for managing and disseminating institutional scholarship, while Li and Banach (2011) emphasize that repository operation must also include organizational commitment to long-term preservation.

Repository value depends on more than the presence of a file-upload page. Research records move through stages and participants: students prepare a proposal, faculty members provide direction, administrators verify accounts and approve records, and readers search published outputs. Technology adoption studies further show that awareness, information needs, usability, and perceived value influence whether university scholars use institutional repositories (Rifai, 2023). A system designed around the repository alone, without a clear review workflow or understandable interfaces, may therefore preserve files but fail to support routine academic work.

The E-Archive System examined in this study was developed as a browser-based application for the Bachelor of Science in Computer Science. It uses PHP, MySQL, HTML, Cascading Style Sheets, and JavaScript and separates public, student, faculty, and administrator interfaces. Its data model records users, teachers, groups, selected faculty-group relationships, research records, chapters, authorship, messages, courses, sections, schools, and departments. The system is intended to preserve the relationship between a document and the people, group, academic batch, and approval state associated with it.

Design and Implementation of a Web Based E-Archive System

This paper documents the application as implemented and revised. It focuses on the architecture, role-based workflow, research-document lifecycle, configurable appearance, compatibility with the existing PHP runtime, and limitations that affect production readiness. The analysis does not claim measured user satisfaction, retrieval-time improvement, or institutional adoption because no completed survey, controlled comparison, or production usage dataset was supplied. Instead, the Results and Discussion section reports observable system capabilities and repeatable technical checks.

OBJECTIVES OF THE STUDY

The study aimed to describe and assess the implemented E-Archive System as a role-based platform for managing academic research. Specifically, it sought to identify the functions available to public readers, students, faculty members, and administrators; trace how a research record progresses from proposal to publication; examine the database and application structure that supports these functions; improve visual consistency through administrator-controlled branding; verify that revised PHP files remain compatible with PHP 5.6.32; and identify technical controls required before the application is exposed to production users.

MATERIALS AND METHODS

Research Design

A developmental and descriptive system-evaluation design was used. The software artifact was treated as the principal source of evidence. This approach was appropriate because the immediate purpose was to document what the application implements, improve its interface and configuration behavior, and determine whether the revised files remain executable under the installed runtime. No human participants were recruited and no personal records were analyzed.

System Materials

The reviewed materials consisted of the project source files, the supplied archive_system SQL dump, uploaded sample documents already present in the project directory, and the local XAMPP environment. The installed command-line runtime reported PHP 5.6.32. Source inventory excluded bundled CKEditor samples when counting application PHP files. The system was organized into administrator, student, faculty, public-main, controller, and shared-include directories.

Evaluation Procedure

Evaluation proceeded in five stages. First, the directory structure and entry points were inventoried to identify role interfaces and controller boundaries. Second, SQL schema definitions and query usage were reviewed to map entities and workflow states. Third, login, registration, research submission, chapter management, account approval, publication, search, comment, and appearance-setting flows were traced from form submission to database action and redirect. Fourth, the shared interface was revised so that title, logo, primary color, banner color, and public text color are retrieved from one settings record and used across administrator, public, student, and faculty pages. Fifth, PHP lint checks were run against the modified files using the installed PHP executable.

Evaluation Criteria

The system was evaluated against four criteria derived from its intended use: functional coverage of the research workflow, role separation, interface consistency, and technical readiness. Functional coverage was established when a corresponding page, controller action, and data entity could be traced. Role separation was assessed from session checks and role-specific interfaces. Interface consistency was assessed from shared settings and local style dependencies. Technical readiness included syntax compatibility, input handling, password treatment, file-upload controls, portability, and database integrity. The review used direct evidence from the code and schema; it did not substitute code inspection for penetration testing, accessibility conformance testing, load testing, or user acceptance testing.

RESULTS AND DISCUSSION

Application Structure

The inventory identified 80 application PHP files outside bundled editor samples and 15 tables in the supplied schema. The public main area provides a landing page, search, proposed-research listings, ongoing-research listings, published research, research details, and chapter views. The student area provides registration, login, profile display, group membership, research creation and editing, chapter creation and editing, pending-chapter review, document links, and group messaging. The faculty area provides registration, login, group selection, announcement creation, group views, archived-research review, and comments. The administrator area provides registration approval, student and faculty lists, research-state management, publication actions, profile access, and system appearance settings.

Design and Implementation of a Web Based E-Archive System

Table 1. Role-Based Functions of the E-Archive System

Role	Primary responsibility	Observed system functions
Public reader	Discover institutional research	Browse proposal, ongoing, and published lists; search records; read research and chapter details; open available documents.
Student	Create and maintain group research	Register; access group membership; submit and edit research; add chapters and authorship; view approval states; receive announcements; use group messaging.
Faculty	Guide assigned research groups	Register; select groups; publish announcements; view group members and archived research; add comments for reviewed work.
Administrator	Govern accounts, workflow, and presentation	Approve accounts; manage research states; publish records; review users; configure title, logo, primary color, banner color, and text color.

Research Workflow

The data model represents research as a stateful record rather than an isolated file. A student group is represented through group and selected-group records that associate a leader, academic batch, and faculty member. Research records store title, description, abstract, file location, authoring member, selected group, creation and approval dates, and a status code. Chapter records store chapter number, title, abstract, file location, date, and status, while a separate authorship table connects members to chapters. This separation supports multiple contributors without repeating chapter data.

Observed research states distinguish proposed, ongoing, published, and disapproved records. Administrator actions move a proposal into ongoing work and later publication, recording approval or publication dates. Public listing pages filter by these states, which allows readers to distinguish work under development from completed work. This workflow is useful for academic oversight because access and visibility can follow review decisions rather than the mere presence of an uploaded file.

Shared Appearance Configuration

A shared site-settings record was introduced to reduce visual differences among the four interfaces. The administrator can set the site title, logo, primary theme color, banner color, and public text color. Shared helper functions provide defaults when a legacy database has not yet been migrated. The public site, administrator dashboard, student portal, faculty portal, login pages, and registration pages read the same settings. Local style sheets replace core remote-template dependencies for the revised layouts, reducing the risk that an external demonstration site changes or becomes unavailable.

The resulting interfaces use compact headers, visually distinct sidebars, responsive login and registration cards, consistent input heights, aligned labels, controlled form gutters, and mobile stacking. This does not by itself establish formal usability or accessibility conformance. It does, however, create a stable interface baseline that can be evaluated with students, faculty members, administrators, and public readers. Repository adoption research indicates that user awareness and information-seeking needs explain only part of actual use (Rifai, 2023), so future interface evaluation should examine both ease of use and the perceived value of the archive in daily academic work.

Compatibility and Technical Verification

The revised files were checked with the installed PHP 5.6.32 command-line interpreter. The checked authentication, shared configuration, administrator settings, administrator actions, portal headers, footers, login pages, and registration pages produced no PHP syntax errors. The implementation intentionally avoided language features introduced after PHP 5.6. Password verification uses functions available in that version. Successful login of an account that still contains a legacy clear-text password replaces that value with a password hash, allowing gradual migration without immediately invalidating existing accounts.

Security and Preservation Limitations

Technical inspection identified limitations that prevent the current application from being treated as a preservation system without further work. Several legacy controllers still construct SQL statements from request values instead of using prepared statements. Research-document upload handlers retain client filenames and do not consistently enforce an allowlist, size limit, generated storage name, or non-public storage location. Some records store complete localhost URLs, which couples content to one deployment path. The schema defines primary keys but few uniqueness constraints, foreign keys, or cascading rules. The SQL dump also differs from tables referenced by parts of the application, including announcements and comments, which means a clean installation may require additional migration scripts.

These limitations are significant because preservation requires more than retaining bytes. Guallar et al. (2015) found that file-format controls in repositories may not align with declared preservation policies or available technical tools. Li and Banach (2011)

Design and Implementation of a Web Based E-Archive System

similarly reported that repository preservation depends on policies, resources, and sustained organizational commitment. For this E-Archive System, database backup, file-integrity checks, format identification, redundant storage, restoration testing, retention rules, and documented responsibility should accompany code-level security controls.

CONCLUSION AND RECOMMENDATION

The E-Archive System provides a coherent foundation for organizing student research through role-specific interfaces and a visible progression from proposal to publication. The implemented modules connect users, groups, faculty guidance, research metadata, chapters, authorship, comments, announcements, and public discovery. Shared appearance settings now allow the institution to maintain one identity across public and authenticated pages, and the revised interface remains syntactically compatible with PHP 5.6.32. These results answer the study objectives at the implementation level: the system covers the central academic workflow, separates principal responsibilities, and can be configured without editing every page.

Before production deployment, the institution should complete a database migration that includes all referenced tables and constraints; convert every query that accepts request data to a prepared statement; add authorization checks that verify ownership and role for each action; apply Cross-Site Request Forgery protection to state-changing requests; and escape or sanitize stored content according to its intended presentation. Research uploads should be stored outside the public web root under generated names, checked against file-type and size policies, scanned where practical, and served through an authorization-aware download endpoint.

Operational recommendations include automated database and file backups, periodic restoration tests, audit logging, HTTPS, secure session-cookie settings, account-recovery procedures, and a documented retention and preservation policy. The project should also migrate from PHP 5.6 after short-term compatibility requirements end because that runtime no longer receives current security support. Finally, a formal evaluation should involve representative students, faculty members, administrators, and public readers. Completion rate, error rate, search success, task time, accessibility issues, and perceived usefulness should be measured before claims about efficiency or usability are made.

REFERENCES

- 1) Bashir, S., Gul, S., Bashir, S., Nisa, N. T., & Ganaie, S. A. (2022). Evolution of institutional repositories: Managing institutional research output to remove the gap of academic elitism. *Journal of Librarianship and Information Science*, 54(3), 518-531.
- 2) Cullen, R., & Chawner, B. (2011). Institutional repositories, open access, and scholarly communication: A study of conflicting paradigms. *The Journal of Academic Librarianship*, 37(6), 460-470.
<https://doi.org/10.1016/j.acalib.2011.07.002>
- 3) Guallar, J., Abadal, E., & Codina, L. (2015). An analysis of file format control in institutional repositories. *Library Hi Tech*, 33(2), 162-174.
<https://doi.org/10.1108/LHT-10-2014-0098>
- 4) Li, Y., & Banach, M. (2011). Institutional repositories and digital preservation: Assessing current practices at research libraries. *D-Lib Magazine*, 17(5/6).
<https://doi.org/10.1045/may2011-yuanli>
- 5) Lynch, C. A. (2003). Institutional repositories: Essential infrastructure for scholarship in the digital age. *portal: Libraries and the Academy*, 3(2), 327-336.
<https://doi.org/10.1353/pla.2003.0039>
- 6) Mgonzo, W. J., & Yonah, Z. O. (2014). Design and development of a web based digital repository for scholarly communication. *International Journal of Knowledge Content Development & Technology*, 4(2), 97-108.
- 7) Musisi, J. B., & Mutwiri, C. (2024). Assessing digital preservation strategies implemented for the institutional repository in Kaimosi Friends University Library. *Regional Journal of Information and Knowledge Management*, 9(1), 1-10.
<https://doi.org/10.70759/n90p5j38>
- 8) Rifai, A. (2023). Technology adoption among university scholars: Institutional repositories case study. *Khizanah Al-Hikmah: Jurnal Ilmu Perpustakaan, Informasi, dan Kearsipan*, 11(2), 221-230.
<https://doi.org/10.24252/kah.v11i2a7>