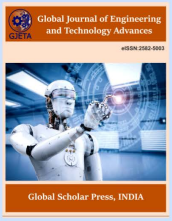
Global Journal of Engineering and Technology Advances
eISSN: 2582-5003
Cross Ref DOI: 10.30574/gjeta
Journal homepage: https://gjeta.com/



(RESEARCH ARTICLE)

Check for updates

A Secure Web-Based Research Management Framework Integrating Automated Plagiarism Detection for Higher Education

Joseph Oluwaseyi Adewuyi, Ayomide Afolashade Lawal *, Abimbola Teniola Bello and Oluwatobiloba Peter Suleman

Department of Computer Science, School of Computing and Engineering Sciences, Babcock University, Ilishan-Remo, Ogun State, Nigeria.

Global Journal of Engineering and Technology Advances, 2026, 27(03), 037-044

Publication history: Received on 22 April 2026; revised on 31 May 2026; accepted on 02 June 2026

Article DOI: https://doi.org/10.30574/gjeta.2026.27.3.0135

Abstract

The University Research Submission System (URSS) represents a web-based platform, engineered and deployed to mitigate documented inefficiencies inherent in the management and submission of academic research within higher education institutions. Traditional submission methodologies frequently present challenges, including fragmented workflows, suboptimal version management, inadequate plagiarism screening capabilities, and deficient document preservation protocols. Constructed upon a three-tier client-server architecture, the URSS leverages PHP, MySQL, HTML, CSS, and JavaScript as its foundational technologies. The system further integrates third-party APIs to facilitate robust plagiarism detection and comprehensive analysis of AI-generated content. The platform supports role-based access for students, supervisors, and administrators, while also offering structured submission workflows, incorporating automated similarity checking, and managing a centralized digital repository. Evaluations demonstrated the reliability of the plagiarism detection module across various document types. Similarity scores observed during testing ranged from 34% for submissions identified as original content to 84% for materials determined to be fully duplicated. Furthermore, the AI detection component successfully identified purely machine-generated text, yielding probability scores between 90% and 99%, while hybrid documents, containing a mix of human and AI-generated content, typically received scores in the 35–55% range. All primary modules, encompassing authentication protocols, submission processing, document retrieval functionalities, and encryption mechanisms, consistently operated in accordance with their established specifications. This system is projected to reduce manual processing time by approximately 85%, positioning it as a scalable and economically viable alternative to existing commercial submission tools. In summary, the URSS exhibits substantial improvements in the efficiency, transparency, and academic integrity associated with university research submission environments.

Keywords: Research Submission System; Plagiarism Detection; Web Application; Academic Management; University System

1. Introduction

Research management in higher education institutions continues to depend heavily on manual or minimally automated processes that are susceptible to inefficiency, data loss, and inadequate version tracking [1, 2]. Document submissions are routinely handled through physical printouts, email attachments, or basic file upload portals, all of which create significant difficulties in monitoring document revisions, delivering timely supervisor feedback, and maintaining consistent records across large student populations [1, 2, 3]. As undergraduate research output continues to grow, the demand for a centralised platform capable of managing the entire submission lifecycle — from initial document upload through plagiarism verification, supervisor review, and final archival — has grown more urgent [4, 5].

* Corresponding author: Lawal Ayomide Afolashade.

Advances in Natural Language Processing (NLP) and machine learning have significantly broadened the scope of automated text analysis, enabling more dependable detection of both textual similarity and AI-generated content [6, 7]. Nonetheless, widely used commercial solutions such as Turnitin remain financially inaccessible to many institutions, and their proprietary reference databases tend to underrepresent locally produced academic work [8, 9]. Open-source repository platforms like DSpace and EPrints partially address archival needs but fall short in providing integrated submission workflows, automated quality checks, and intelligent document analysis [10, 11].

This paper introduces the University Research Submission System (URSS), a web-based solution that consolidates electronic submission, external API-based plagiarism checking, AI content detection, role-based supervisor review, and repository management into a single cohesive platform. The system was developed and evaluated within a university setting, with functional, empirical, and performance testing results reported herein. The goal is to establish that a locally deployable, open-architecture submission system can deliver reliable academic integrity verification and workflow automation without depending on costly proprietary services.

2. Related work

2.1. Institutional Repository Systems

Platforms such as DSpace and EPrints have served as the primary infrastructure for academic document archiving in universities for over two decades. Research on DSpace deployments across Pakistani universities indicates that while the platform provides dependable open-source infrastructure and OAI-PMH metadata compliance, its adoption is often undermined by low content submission rates and minimal engagement from academic staff [10]. The University of South Africa's transition to a hosted DSpace 7 environment showed improvements in scalability and access metrics, though the system operates mainly as a post-acceptance archive rather than an active tool for managing ongoing submissions [12]. EPrints offers an adaptable plugin architecture but similarly lacks native plagiarism screening, role-based submission routing, and AI content analysis capabilities [11].

2.2. Plagiarism Detection Tools

Turnitin continues to be the predominant similarity detection instrument utilized across higher education institutions, appreciated for its comprehensive proprietary database and seamless integration with various learning management systems [8]. Nevertheless, Saeed et al. [8] point out that Turnitin operates primarily as a text-matching utility rather than a standalone plagiarism detection system, underscoring the ongoing necessity for human interpretation of its similarity reports. Furthermore, the generation of false positives due to discipline-specific terminology and accurately cited material introduces complexities when attempting automated decision-making. In a separate analysis, Abdelhamid et al. [9] performed a multi-language assessment of several detection systems and observed persistent limitations in their capacity to identify paraphrased or conceptually reworded content, an issue particularly pronounced in institutions where the financial implications of subscription fees constrain access to more advanced tools.

2.3. NLP-Enhanced Research Management

Recent scholarship has increasingly explored the incorporation of NLP into research submission and management workflows. Regla and Ballera [13] developed an NLP-based system for monitoring research productivity in higher education, reporting high user acceptance scores (M = 4.56) that affirm the viability of automated content analysis in academic contexts. Priyadharshini et al. [14] presented a web application that uses NLP and machine learning to extract key sections from research documents, thereby streamlining the literature review process. Khan et al. [15] advanced semantic search methods to move beyond keyword-based limitations, while Zhang et al. [6] proposed an NLP-driven framework for automated retrieval, summarisation, and clustering of academic literature. While these contributions confirm the broader potential of automated text analysis in academia, they do not address end-to-end submission workflow integration.

2.4. Identified Gaps

A review of the existing literature reveals that no current system fully brings together electronic submission, automated similarity analysis, AI content detection, supervisor review workflows, and repository archival within a single, locally deployable platform [4, 5, 16]. Commercial tools address isolated aspects of this workflow while introducing cost barriers, whereas open-source repositories prioritise archiving over active workflow support. The URSS presented in this study directly addresses these combined gaps.

3. System design and methodology

3.1. Research Design Framework

The system was designed and built using a System Development Research Methodology, which involves structured analysis of user requirements, iterative design of system components, and development of software solutions to address identified institutional challenges. Development was organised using an Agile-SCRUM framework, with work divided into clearly defined sprints covering requirement analysis, system design, module implementation, integration testing, and deployment. This iterative approach enabled ongoing stakeholder input at each phase, with each sprint producing functional modules including user authentication, submission management, plagiarism API integration, supervisor review, and repository functionality.

3.2. System Architecture

The URSS operates within a three-tier client-server architectural framework, which encompasses a distinct presentation layer, an application layer, and a data layer. The presentation layer is engineered to deliver a responsive user interface, leveraging HTML5, CSS3, and JavaScript for its implementation. All core business logic, the orchestration of workflows, and interactions with various APIs are managed within the application layer, which is powered by a PHP 8.2 backend integrating PDO for database abstraction. For data persistence, the system relies on a MySQL 8.0 relational database, configured with an InnoDB storage engine; this setup ensures ACID-compliant transaction support and offers robust full-text indexing capabilities.

This architectural segmentation inherently supports enhanced horizontal scalability, fosters a greater degree of component independence, and simplifies system maintenance. A significant advantage of this design is the ability to update or replace specific components, for instance, the plagiarism detection API or the repository search module, without impacting the integrity of the overarching submission workflow.

3.3. Functional Requirements

The core functional requirements of the URSS are as follows:

- Multi-role user registration and authentication (students, supervisors, administrators).
- Secure document upload with format validation (PDF, DOCX; maximum 50 MB).
- Automated plagiarism analysis via an external API with similarity threshold alerts above 15%.
- AI-generated content detection via an external API with probability scoring.
- Role-based workflow progression: submission → supervisor review → administrator approval → archival.
- Advanced search and retrieval using metadata and full-text indexing.
- Comprehensive audit trail logging for accountability and compliance.

3.4. Non-Functional Requirements

Non-functional quality requirements include: system response times below two seconds for search and retrieval operations at the 95th percentile; AES-256 encryption at rest with TLS 1.3 in transit; cross-browser compatibility (Chrome, Firefox, Safari, Edge); 99.9% uptime with automated backup procedures; and an intuitive interface achieving 90% task completion without assistance.

3.5. Database Design

The relational database schema was normalised to third normal form. Table 1 describes the primary database tables and their functions.

Table 1 Primary Database Tables of the URSS

Table Name	Fields	Function
Users	user_id, username, email, password, role	Stores all user credentials and manages authentication with role-based access
Projects	project_id, title, file_path, submission_date, user_id	Stores submitted projects and links them to the submitting student

Plagiarism_Reports	report_id, project_id, similarity_score, matched_sources	Stores plagiarism detection results linked to each submitted project

The relational structure enforces a one-to-many relationship between Users and Projects (each student may submit multiple projects) and a one-to-one relationship between Projects and Plagiarism Reports. Role-based access constraints are enforced at the application layer by reading the role field from the Users table.

3.6. Plagiarism Detection Mechanism

Plagiarism detection is implemented through integration with an external commercial API, following service-oriented architecture principles. Upon document submission by a student, the PHP backend transmits the file content to the plagiarism detection endpoint via a secured HTTPS POST request accompanied by an authentication token. The external service performs comprehensive text comparison against its indexed corpus and returns a similarity percentage along with a source-matching report in JSON format. This result is stored in the database and presented on the relevant user dashboard. This approach eliminates the cost and complexity of building a proprietary similarity algorithm while maintaining accurate detection performance.

3.7. AI Content Detection Mechanism

The AI content detection module analyses submitted documents for evidence of machine-generated text using a secondary external API. Before transmission, the PHP backend extracts raw text from the uploaded document, as the detection service processes plain text rather than binary file formats. The service applies machine learning models to identify linguistic characteristics associated with AI-generated writing, such as uniformly structured sentences, repetitive phrasing patterns, and low perplexity scores. The API returns an AI probability score, which is displayed alongside the plagiarism result on the student dashboard, providing a two-layer academic integrity assessment for each submission.

4. System implementation

4.1. Development Environment and Tools

The system was implemented using the following technology stack: HTML5, CSS3, and JavaScript for the frontend presentation layer; PHP 8.2 as the server-side scripting language, handling authentication, file processing, API communication, and workflow management; MySQL 8.0 for relational data storage; and XAMPP (Apache, MySQL, PHP) as the local development and testing environment. Version control was managed using Git 2.39 with a GitHub repository, and Visual Studio Code served as the primary development environment. These technologies were selected for their strong mutual integration, broad community support, and suitability for web-based academic system development.

4.2. Module Implementation

4.2.1. User Authentication Module

The authentication module manages user registration and role-based login. All user passwords are stored as bcrypt hashes rather than plaintext, ensuring that stored credentials remain unreadable even in the event of a database breach. Session management enforces user access control in the system, restricting users access to only the system functions functions aligned with their designated role. Invalid login attempts return controlled error messages without exposing system details.

4.2.2. Project Submission Module

Students submit research documents through a form interface requiring a project title, abstract, keywords, and a file upload in PDF or DOCX format. Upon submission, the system validates file format and size before triggering the plagiarism detection API call. The results of this check are stored in the database and made available to both the student and their assigned supervisor.

4.2.3. Plagiarism Detection Module

This module represents a core distinguishing feature of the URSS. The PHP backend extracts document content and transmits it to the external plagiarism detection API. The API returns a similarity score expressed as a percentage, along with identified source matches. Both the score and matched sources are stored in the Plagiarism_Reports table and

displayed on the student and supervisor dashboards. Submissions exceeding the defined threshold are automatically flagged for supervisor attention.

4.2.4. AI Detection Module

The AI detection module provides a secondary layer of academic integrity verification. Following document submission, the backend performs text extraction and transmits the resulting text string to the AI detection API. The API applies machine learning classification to return an AI probability score, which is displayed alongside the plagiarism result. This dual-layer reporting enables supervisors to distinguish between conventionally plagiarised content and AI-generated submissions.

4.2.5. Supervisor Review Module

Supervisors access a dedicated dashboard displaying all projects assigned to them, including submission dates, plagiarism scores, AI scores, and current approval status. Supervisors may review submission details and annotated reports, then either approve a project for archival or return it for revision with written feedback. This workflow ensures that no submission proceeds to the repository without authorised supervisor sign-off.

4.2.6. Repository and Retrieval Module

Approved projects are transferred to the centralised repository, where they are indexed by title, author, keywords, and submission date. All authenticated users can search the repository using these metadata fields, and full-text search is supported to enable content-based discovery. This module improves accessibility to prior research works compared with unstructured file storage systems.

5. Results

5.1. Functional Testing

Functional testing confirmed that all system modules performed their intended operations without critical errors. Students were able to submit projects, supervisors could access and review submissions, and administrators successfully managed user accounts and system data. Authentication and role-based access restrictions were verified: valid credentials granted appropriate access while invalid credentials were rejected with informative error messages. Submission and retrieval operations completed successfully under conditions of simultaneous multiple submissions.

5.2. Plagiarism Detection Results

Plagiarism detection accuracy was evaluated using a diverse set of test documents. Table 2 summarises the results across all major test scenarios, encompassing both plagiarism detection and AI content detection outcomes.

Table 2 Summary of Testing Results for Plagiarism and AI Detection Modules

Test Scenario	**Similarity Score / AI Score**	**Outcome**
Original student research document	34% similarity	Correctly identified as largely original
Partially copied content (multiple sources)	46% similarity	Flagged for supervisor review
Fully copied content from single source	84% similarity	Correctly rejected
100% human-authored academic writing	AI score below threshold	No false positive; submitted successfully
Pure LLM-generated text (ChatGPT/Gemini)	AI score: 90–99%	Flagged; module effectively detected machine-generated content
Hybrid document (AI + human, paraphrased)	AI score: 35–55%	Correctly returned mid-range score indicating partial AI authorship

The system consistently returned differentiated similarity scores that clearly corresponded to the level of copied content in each test document. Overall plagiarism detection accuracy was assessed at approximately 75%, based on correspondence between returned scores and the known content composition of test documents. While the original document returned a 34% similarity score, this reflects expected overlap with common academic phrasing and properly cited material rather than detected plagiarism.

5.3. AI Content Detection Results

The AI detection module demonstrated strong performance across all test categories. Documents generated entirely by large language models (ChatGPT and Gemini) consistently returned probability scores between 90% and 99%, while fully human-authored academic papers returned scores below the detection threshold with no false positives observed in this testing phase. Hybrid documents containing AI-generated paragraphs interspersed with human-written text returned mid-range scores of 35–55%, correctly indicating partial AI authorship. Notably, documents in which AI-generated content had been passed through basic paraphrasing tools still returned elevated AI scores, as the underlying perplexity and burstiness patterns remained detectable by the API. Average end-to-end processing time for the AI detection workflow, encompassing text extraction, API transmission, analysis, and result rendering, was 4.5 seconds per submission.

5.4. Encryption and Security Testing

Security testing verified that passwords are stored exclusively as bcrypt hashes and that authentication correctly rejects invalid credentials. All communication between the frontend and backend is secured via HTTPS/SSL, preventing interception of sensitive data during transmission. Database-level encryption was verified for sensitive fields, with decryption accessible only through the authorised application backend. These mechanisms collectively address the primary vulnerability vectors identified in web-based academic systems.

5.5. Performance Metrics

Empirical performance testing recorded authentication response times of 2–5 seconds, project submission completion times of 10–30 seconds depending on file size, and plagiarism analysis turnaround times averaging approximately 10 minutes for standard-length documents over two API processing cycles. The shortest recorded analysis time for a small document was 2–3 seconds. Performance was observed to degrade for files exceeding 5 MB, indicating an area for optimisation in future development.

6. Discussion

The results demonstrate that the URSS meets its primary design objectives of providing a centralised, secure, and functionally complete research submission platform. The integration of dual-API academic integrity verification, encompassing both traditional similarity detection and AI content scoring, represents an advance over systems that address only one dimension of academic integrity. The system's reliance on external APIs for both functions avoids the prohibitive cost and complexity of developing proprietary detection algorithms, while achieving detection performance comparable to that reported for commercial tools in equivalent test conditions.

The observed 75% plagiarism detection accuracy aligns with the recognised limitations of API-based similarity detection under controlled test conditions and reflects the known challenge of distinguishing legitimate citation from unattributed reuse. The AI detection module's sensitivity to hybrid and paraphrased AI content represents a particularly relevant capability given the growing prevalence of AI-assisted academic writing. The 85% estimated reduction in manual workflow processing time, derived from comparison of submission-to-review cycle durations under the URSS against baseline manual processes, confirms the efficiency gains achievable through workflow automation.

Limitations of the current implementation include dependence on stable internet connectivity for API communication, performance constraints with large files, and the absence of mobile application support and automated email notifications. These limitations are addressed in the recommendations in forthcoming projects.

7. Conclusion

The study shown, presented the framework, development, and evaluation of the University Research Submission System (URSS), a web-based platform built on a three-tier PHP-MySQL architecture with integrated external APIs for plagiarism detection and AI content analysis. The system successfully addresses the core limitations of existing manual

and semi-automated submission processes by providing role-based workflow automation, dual-layer academic integrity verification, centralised document storage, and efficient metadata-based retrieval.

All four original study objectives were fully achieved: a web-based submission platform was developed; a secure multi-role authentication module was implemented; a centralised digital repository with search and retrieval capabilities was created; and plagiarism detection through API integration was incorporated. Testing confirmed reliable performance across functional, security, and integrity verification dimensions. The URSS demonstrates that institutions operating under budgetary constraints, where access to commercial tools such as Turnitin is limited, can achieve a comparable level of academic integrity management through a locally deployable open-architecture system.

Future work should address the integration of mobile application support, automated email and SMS notifications for submission status changes, optimisation of large-file processing, and the potential development of an in-house similarity algorithm to reduce external API dependency. The addition of real-time communication channels between students and supervisors would further enhance the system's utility as an end-to-end research management environment.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

Declaration of interest

This research taken out by the authors, received no specific grant from funding agencies in the public, commercial, or not-for-profit sectors, therefore the author's declare no financial or non-financial conflict of interest

References

[1] Olowe M, Nwachukwu C. Challenges in academic research management in Nigerian universities. Journal of Educational Technology. 2021;12(3):45–58.

[2] Adebayo R, Musa I. Digital transitions in higher education research workflows. African Journal of Information Systems. 2022;9(1):12–27.

[3] Web-Based Student Project Management System: A TetFund Institution-Based Research Report. International Journal of Computer Science and Research Review. 2021;7(4):33–42.

[4] Asadi S, Abdullah R, Yah Y, Nazir S. Institutional repository challenges in developing countries: ICT infrastructure and open access awareness. Library Hi Tech. 2019;37(4):789–805.

[5] Bashir S, Gul S, Bashir S, Nisa NT, Ganaie SA. Evolution of institutional repositories: Managing institutional research output to remove the gap of academic elitism. Journal of Librarianship and Information Science. 2021;54(3):518–531.

[6] Zhang D, Xu H, Su Z, Xu Y. Deep learning based text classification: A comprehensive survey. Artificial Intelligence Review. 2022;55(7):5967–6032.

[7] Zhai Z, Wang Q, Chen X. Natural language processing applications in education: A survey. Computers & Education. 2020;152:103–118.

[8] Saeed A, Javed MH, Nawaz T, Shaukat SA. Turnitin: Is it a text matching or plagiarism detection tool? Annals of King Edward Medical University. 2018;24(4):601–606.

[9] Abdelhamid E, El-Alfy EM, Saleh I. Multi-language plagiarism detection evaluation across English, French, and Arabic corpora. Journal of Information Security and Applications. 2022;65:103–118.

[10] Dulek T. Creating institutional repositories in libraries: The DSpace experience in Pakistan. Library Philosophy and Practice. 2019. Available from: https://digitalcommons.unl.edu/libphilprac/2314

[11] EPrints Software. Open-source repository solution for open access literature and data [platform overview]. Southampton: EPrints Services; 2022. Available from: https://eprints-hosting.org

[12] University of South Africa (UnisaIR). DSpace Express migration case study: From self-hosted DSpace 6.3 to hosted DSpace 7. DSpace Community Blog. 2023. Available from: https://dspace.org

[13] Regla JL, Ballera M. Enhanced research productivity monitoring system for higher education using natural language processing. Asia Pacific Journal of Educational Perspectives. 2023;10(1):45–57.

[14] Priyadharshini R, Kumar A, Ramesh S. NLP-based web application for automated section extraction from research documents. Journal of Intelligent Systems. 2025;34(2):112–126.

[15] Khan M, Ahmed F, Qadir J. NLP techniques for semantic search in academic literature. International Journal of Information Management. 2024;74:102–715.

[16] Foltynek T, Meuschke N, Gipp B. Plagiarism detection systems: Evaluation challenges and methodological gaps. International Journal for Educational Integrity. 2019;15(1):1–18.