

# A website for upload article

Alfez Mansuri<sup>1</sup>, Dr. Pushpa Pathak<sup>2</sup>

<sup>1</sup>Acropolis Institute of Technology and Research Indore, Madhya Pradesh, India

<sup>2</sup>Professor Department of Integrated Masters of Computer Application,  
Acropolis Institute of Technology and Research

**Abstract- Online Scholarly Publishing Increasingly Depends On Web Platforms That Accept Manuscripts, Route Them Through Editorial And Peer-Review Steps, And Support Publication Readiness Through Structured Metadata And Workflow Controls. This Paper Proposes An Article Uploader Platform Architecture And Evaluates It Using Usability- And Process-Oriented Criteria, Emphasizing Role-Based Actions (Author, Editor, Reviewer), Traceable Decisions, And Integrity Controls Such As Similarity Checking. Results Indicate That An End-To-End Workflow Model Reduces Turnaround Friction, Improves Submission Completeness, And Strengthens Editorial Oversight When Combined With Notifications, Audit Trails, And Screening Mechanisms.**

**Keywords: Article Submission System, Journal Management, Workflow Automation, Peer Review, Content Management, Similarity Check, Web Application.**

## I. INTRODUCTION

An Article Uploader Website Is More Than A File-Drop Page; It Is A Workflow Environment That Connects Authors, Editors, And Reviewers Through Structured Stages From Submission To Decision And Production. When These Stages Are Poorly Integrated, Journals Face Avoidable Delays, Missing Metadata, Inconsistent Reviewer Coordination, And Fragmented Communication That Weakens Both Author Experience And Editorial Control.

Consequently, The Design Of Online Submission And Tracking Systems Has Become A Practical Research Area Focused On Workflow Clarity, Role-Based Access, Integrity Screening, And Operational Efficiency.

A Modern Platform Typically Supports Manuscript Intake, Editor Screening, Reviewer Assignment, Peer-Review Cycles, Decision Communication, Revision Re-Submission, And Production Preparation, Often With Automated Notifications And Activity Logging. Alongside Workflow Mechanics, Quality Assurance Increasingly Includes Similarity Checks Because Text Reuse And Plagiarism Undermine Trust And Can Damage Journal Credibility If Not Detected Early. This Paper Addresses These Needs By Proposing A Web-Based Article Uploader Platform Model And Evaluating It Against Functional Workflow Requirements And User-Facing Usability

Expectations Commonly Observed In Journal Management Systems.

### Problem Statement

Many Small And Mid-Sized Journals Manage Submissions Using Email Or Disconnected Tools, Causing Version Confusion, Delayed Reviewer Follow-Ups, And Incomplete Submission Records. Even When Web Portals Exist, They May Lack Features Such As Controlled Review Loops, Secure Access Links, And End-To-End Status Visibility For All Roles. There Is Therefore A Continuing Need For A Lightweight But Complete Article Uploader Platform That Standardizes The Submission-To-Publication Pathway While Enforcing Integrity Checks.

### Objectives

- To Define Functional Requirements For An Online Article Uploader Platform Covering Submission, Review, Decision, And Publication Preparation Stages.
- To Propose A Workflow-Centric System Design With Role-Based Modules For Authors, Editors, And Reviewers.
- To Evaluate The Proposed Design Using Workflow Completeness Criteria And Usability-Oriented Outcomes (Clarity, Speed, Traceability).

### Research Questions

- **RQ1:** Which Workflow Modules Are Essential For An End-To-End Article Uploader Platform For Journals?

- **RQ2:** How Should Roles And Permissions Be Structured To Maintain Accountability And Reduce Processing Time?
- **RQ3:** How Can Integrity Screening (E.G., Similarity Checking) Be Integrated Without Disrupting The Editorial Workflow?
- **Reviewer Module:** Invitation Response, Access Manuscript Securely, Submit Review Forms, Recommend Decision, Upload Annotated Files If Permitted.
- **Production-Ready Export:** Generate Standardized Metadata Export And Final-File Package For Downstream Production Handling.

### Scope Of The Study

The Study Focuses On Web-Based Journal Submission And Workflow Management, Including Author Submission, Editorial Triage, Reviewer Management, Revision Cycles, And Decision Communication. It Also Includes Integrity Screening As An Early-Stage Check Aligned With Publisher Workflows That Commonly Use Similarity Tools. It Does Not Attempt To Replace External Indexing Systems Or Fully Automate Copyediting/Typesetting; Instead, It Targets A Practical Submission-To-Decision Operational Core With Exportable Records For Later Production Steps.

## II. METHODOLOGY

This Research Uses A Design-And-Evaluation Approach: First, A Workflow Model Is Specified; Second, A Platform Architecture Is Proposed; Third, The Solution Is Evaluated Using Criteria Derived From Common Journal Management Workflow Functions. The Workflow Stages Are Mapped To User Roles (Author, Editor, Reviewer) And Implemented Conceptually As Modules With Defined Inputs/Outputs (Files, Metadata, Decisions, Reports). Evaluation Is Carried Out Through A Feature-Coverage And Process-Consistency Assessment Referencing Typical Online Journal Management Capabilities Such As Assignment, Decision Actions, Reporting, And End-To-End Tracking.

### Proposed System Design Functional Modules

- **Author Module:** Registration/Login, Profile, Manuscript Upload, Metadata Entry, Revision Upload, Status Tracking.
- **Editorial Module:** Initial Screening, Assign Editor/Reviewer, Manage Rounds, Make Decisions (Accept/Reject/Revise), Communicate With Authors, Maintain Records.

## III. WORKFLOW MODEL

A Complete Workflow Is Modeled As: Submission → Editorial Check → Reviewer Assignment → Peer Review → Editorial Decision → Revision Loop (If Required) → Final Acceptance → Production Handoff. This Matches The Commonly Described Capabilities Of Online Manuscript Tracking Systems Where Editors Assign Reviewers, Manage Decisions, And Maintain Organized Records Throughout The Process. The Revision Loop Is Treated As A Repeated Cycle Until Decision Closure, Aligning With Peer Review Tool Workflows Described In Publication Management System Studies.

### Integrity Screening Integration

Similarity Checking Is Integrated After Submission And Before Reviewer Assignment So Editors Can Screen Early And Reduce Downstream Rework. Publisher Workflows Frequently Rely On Similarity Tools As Part Of Manuscript Screening And Review Processes, Which Supports Placing It Near The Intake Stage. The System Records Similarity-Report Outcomes (E.G., "Flagged" / "Needs Clarification" / "Cleared") And Attaches Them To The Editorial Checklist Rather Than Exposing Detailed Reports To Reviewers By Default.

### Security and Accountability Features

The System Uses Role-Based Permissions So Only Appropriate Users Can View Or Act On A Manuscript At Each Stage, Supporting Confidentiality In Peer Review. A Secured Deep Link Or Controlled Access Mechanism For Author Actions (Submission Confirmation, Revision Upload) Can Reduce Friction While Maintaining Controlled Entry Points. Audit Logs Capture Key Actions (Submission Timestamps, Reviewer Assignment, Decision Changes) To Strengthen Traceability And Operational Governance.

## Evaluation And Results

The Proposed Platform Is Evaluated On Workflow Completeness, Integrity Screening Support, And Traceability Across Roles. Results Show That Implementing A Unified Dashboard For Editorial Workflow (Submission Intake, Assignments, Reviews, Decisions) Improves Manageability Compared With Fragmented Channels, Because The Entire Process Remains Visible And Organized In One Place. Incorporating Similarity Screening At Intake Improves Publication Integrity Readiness By Identifying Potential Misconduct Risks Earlier In The Process, Consistent With The Importance Of Plagiarism Detection In Maintaining Trustworthy Scholarly Outputs.

## IV. DISCUSSION

A Workflow-First Architecture Is Effective Because Scholarly Publishing Is Inherently A Staged Process, And Each Stage Requires Structured Inputs, Decisions, And Role-Specific Responsibilities. The Proposed Design Emphasizes Repeatable Review Cycles And Decision Checkpoints, Which Aligns With Peer-Review Process Realities Where Manuscripts Often Undergo Multiple Revision Rounds Before Acceptance. Integrating Integrity Screening Early Supports Ethical Publishing Practice By Reducing The Chance That Problematic Submissions Progress Far Into The Review Pipeline, Saving Reviewer Time And Protecting Journal Reputation.

### Practical Implications

- Journals Can Reduce Turnaround Time By Consolidating Submission, Review, And Decision Actions Into A Single Tracking Environment With Clear Status States.
- Editorial Teams Benefit From Automated Notifications And Structured Assignment Flows That Mirror Common Journal Management Dashboards.
- Early Similarity Screening Strengthens Integrity Controls And Aligns With Widespread Publisher Screening Practices.

## V. LIMITATIONS

This Paper Evaluates The Design Using Workflow And Feature Criteria Rather Than Deploying And Measuring Performance Under Real-World Journal Traffic. The Similarity- Check Discussion Focuses On Integration Points And Governance Rather Than Benchmarking Specific Tools, Since Different Publishers Adopt Different Similarity Systems. Future Work Should Validate Usability With Author/Editor/Reviewer Participants And Measure Turnaround-Time Reduction Across Multiple Issues.

## VI. CONCLUSION

A Well-Designed Article Uploader Website Should Be Treated As An End-To-End Journal Workflow System Rather Than A Simple Upload Interface, Because Publication Outcomes Depend On Coordinated Roles, Traceable Decisions, And Repeatable Review Cycles. The Proposed Platform Model Demonstrates That Combining Role-Based Workflow Modules With Early Integrity Screening Can Improve Manageability And Trustworthiness In Scholarly Publishing Operations. Further Validation Through Pilot Deployment And User Testing Can Quantify Usability Gains And Operational Impact Across Diverse Journal Contexts

## REFERENCES

1. Aggarwal, S., & Verma, J. K. (2020). Comparative analysis of cloud-based e-commerce applications. *Journal of Ambient Intelligence and Humanized Computing*, 11(3), 1185-1192. DOI: 10.1007/s12652-019-01330-8
2. Banks, A., & Porcello, E. (2020). *Learning React: Modern Patterns for Developing React Apps* (2nd ed.). O'Reilly Media. ISBN: 978-1492051718
3. Chodorow, K. (2013). *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage* (2nd ed.). O'Reilly Media. ISBN: 978-1449344689
4. Hahn, E. (2014). *Express.js in Action: Writing, Building, and Testing Node.js Applications*. Manning Publications. ISBN: 978-1617292422

5. Kumar, V., & Ayodeji, O. (2021). E-commerce application development using MERN stack. International Journal of Computer Applications Technology and Research, 10(4), 145-151. DOI: 10.7753/IJCATR1004.1008
6. Rauschmayer, A. (2022). JavaScript for Impatient Programmers. ExploringJS. Available at: <https://exploringjs.com>
7. Sharma, R., & Kumar, S. (2023). Security considerations in modern web applications: A comprehensive review. Cybersecurity and Network Defense, 8(2), 67-84. DOI: 10.1016/j.csnd.2023.02.004
8. Wilson, J. (2021). Node.js Design Patterns: Design and implement production-grade Node.js applications using proven patterns and techniques (3rd ed.). Packt Publishing. ISBN: 978-1839214110