

University Workflow Communication and File Tracking System in ERP

Author: **Roja M** rojamrojam667@gmail.com

MCA 4th Sem GM University, Davangere

Guide: **Prof. Krupa Rani S** kruparanis.fcit.sca@gmu.ac.in

Institution: Department of MCA, GM University, Davangere, Karnataka, India

Abstract

Universities depend on a continuous movement of requests, approvals, notices, academic documents, and administrative files. In many higher education institutions, these activities are still handled through paper files, email trails, spreadsheets, and informal messaging. Such practices make it difficult to know where a file is pending, who must act on it, whether a document has been modified, and how long each approval stage takes. Enterprise Resource Planning (ERP) systems have improved admission, examination, finance, and human-resource functions in higher education, but workflow communication and file tracking are often implemented as isolated or weakly integrated modules. This paper proposes a University Workflow Communication and File Tracking System in ERP for a final year MCA project. The proposed system uses React.js, Node.js, Express.js, MongoDB, REST APIs, JWT authentication, role-based access control, Tailwind CSS, and Chart.js. It supports secure login, user and department management, workflow creation, file upload, document routing, approval tracking, notifications, dashboards, audit logs, and reports. The architecture is designed around transparent document movement, controlled authorization, traceable decisions, and measurable administrative performance. A prototype evaluation is presented using functional testing and workload-based response-time observations. The proposed system reduces dependency on manual file movement, improves accountability, and provides a practical foundation for paperless university administration.

I. Introduction

Higher education institutions are not only academic spaces; they are also complex administrative organizations. A university handles student applications, staff service records, examination forms, leave requests, meeting proceedings, circulars, purchase approvals, accreditation documents, scholarship files, research proposals, and internal communications. These documents often move through several offices before reaching completion. A single request may begin at a department, pass through the Head of Department, Dean, Registrar, Vice Chancellor, and finance or examination sections, and finally return to the originator with an approval, rejection, or request for correction.

When these activities are handled manually, the movement of information becomes dependent on physical files, handwritten registers, email forwarding, and personal follow-up. This creates avoidable delays and makes accountability difficult. The sender may not know whether the file has reached the next office. The receiver may not have a complete history of earlier decisions. Administrators may not be able to identify bottlenecks without asking several people. Students and staff may repeatedly visit offices just to learn the status of a request.

ERP adoption in higher education is increasingly studied as part of e-governance, transparency, integration, and administrative modernization. Recent literature notes that ERP systems can improve efficiency and decision-making but face challenges such as user resistance, poor process alignment, inadequate training, and infrastructure limitations [1]-[6]. These

observations are especially relevant for workflow and document movement, because universities are decentralized

and role-heavy organizations. A solution cannot simply digitize paper; it must reflect institutional authority, department boundaries, approval levels, and audit needs.

This paper proposes an ERP-integrated workflow communication and file tracking system for a university

environment. The system is designed as a web application that allows authorized users to create workflows, upload documents, forward files, approve or reject requests, view status, receive notifications, and generate reports. It combines workflow automation with document management and role-based access control. The goal is not to replace the entire university ERP, but to add a practical workflow layer that can integrate with existing ERP modules.

II. Problem Statement

Many universities still use fragmented procedures for document movement and administrative communication. A file may be stored in one office, discussed through email, approved in another system, and tracked in a register. This fragmentation causes five major problems. First, file status is not visible in real time. Second, approvals are delayed because pending responsibility is unclear. Third, document versions may be duplicated or misplaced. Fourth, senior administrators lack consolidated reports on workload and bottlenecks. Fifth, there is limited evidence for audit because decisions are not always timestamped in a central system.

The research problem addressed in this paper is therefore: How can a university ERP system provide secure, transparent, role-aware workflow communication and file tracking so that document movement becomes faster, traceable, and accountable?

III. Objectives

The objectives of the proposed work are:

1. To design an ERP-based workflow communication module for university administration.
2. To implement digital file upload, routing, approval, rejection, and archival functions.
3. To provide real-time file tracking with status, owner, timestamp, and action history.
4. To enforce role-based access control for Admin, Chancellor, Vice Chancellor, Registrar, Dean, HOD, Faculty, Staff, and Student users.
5. To secure API access using JSON Web Token based authentication.
6. To maintain audit logs for important actions such as login, upload, forward, approval, rejection, and download.
7. To provide dashboards and reports for workload, turnaround time, and pending files.
8. To evaluate the prototype through functional testing and basic performance observations.

IV. Literature Review

ERP systems in higher education are widely discussed as tools for integration, cost reduction, administrative transparency, and data-driven decision-making. Sholeh, Samodra, and Widodo [1] conducted a systematic literature review of ERP implementation in higher education and identified benefits such as efficiency and decision support, while also highlighting challenges related to complexity, change resistance, and infrastructure. Subbarao et al. [2] studied user challenges and system refinement in higher-education ERP and emphasized the importance of improving usability after implementation. Yashmita and Srinivas [3] examined ERP as part of e-governance in higher education institutions and reported that ERP can support transparency and paperless administration, although awareness and orientation remain important adoption issues.

Stakeholder alignment is another recurring concern. FattahWeil, Bender, and Gronau [4] studied ERP adoption in German higher education and found that project-team composition, structured change management, training, leadership commitment, and alignment with institutional processes affect adoption. Rizkiana, Ritchi, and Adrianto [5] examined critical success factors for ERP implementation in higher education and emphasized that implementation failures are often linked to organizational rather than purely technical issues. Alhazmi et al. [6] also discussed motivations, challenges, and success factors for ERP in higher education, noting that decentralized governance and institutional readiness influence ERP outcomes.

Document tracking and document management studies provide more specific grounding for this project. Redoblo [7] developed an online multi-function HEIs tracking system covering document management, notifications, user credentials, audit logs, and backup. Castro, Quilon, and Arruejo [8] proposed DocTrack for the University of Northern Philippines, focusing on uploading, tracking, releasing, declining, and archiving documents. Costoiu et al. [9] described an electronic document management information system for a university environment, showing that universities need continuous improvement of document digitization and administrative information systems. Mayasari et al. [10] reviewed web-based document management systems for higher education and connected them to accessibility, workflow efficiency, records control, and accountability.

Other recent work addresses adjacent technologies for trustworthy document handling. A blockchain-based information management system for academic institutions was studied through an international students' workflow case, showing potential benefits in transparency, speed, and information privacy [11]. A blockchain and IPFS-based document management system was proposed for secure document processing and verification [12]. Donner [13] reviewed research data management systems in universities and research institutes, showing that information management requires both technical and organizational solutions. Subiksa, Ariawan, and Peling [14] reported positive usability and service-quality outcomes for a cloud file management system in an academic department context.

For access control and authentication, this paper draws on the NIST RBAC model and JWT standard. Sandhu, Ferraiolo, and Kuhn [15] describe RBAC as a model for assigning permissions through organizational roles, which fits university structures where access depends on authority and responsibility. Jones, Bradley, and Sakimura [16] define JWT as a compact claims format suitable for representing authenticated identity information across web APIs. MongoDB's documentation on document relationships and references also informs the proposed database design for workflow entities and linked documents [17].

Table I. Literature Comparison

Ref.	Focus	Key Contribution	Limitation
[1]	ERP in HEIs	Explains ERP benefits and challenges	Limited workflow design
[2]	ERP User Challenges	Improves user adoption and usability	No file tracking workflow
[3]	ERP & e-Governance	Supports paperless administration	Focuses on perception only
[4]	ERP Adoption	Identifies ERP success factors	No implementation design
[5]	ERP Success Factors	Improves organizational performance	No document routing
[6]	ERP Challenges	Reviews ERP implementation issues	General ERP perspective
[7]	Document Tracking System	Provides document tracking and audit logs	Campus-specific implementation
[8]	University File Workflow	Supports upload and archive tracking	No role-based access
[9]	Electronic Document Management	Digitizes document administration	Uses older technology
[10]	Web Document Management	Improves workflow efficiency	Not a complete ERP solution

V. Research Gap Analysis

The reviewed literature shows that ERP systems and document management systems are both important in higher education, but several gaps remain. First, many ERP studies discuss adoption, success factors, and organizational challenges at a broad level without presenting a concrete workflow communication module. Second, document tracking systems often focus on upload and status visibility but do not fully model complex university role hierarchies. Third, existing studies commonly discuss digital document repositories, while fewer integrate notifications, approval routing, dashboards, and audit logs into one ERP-oriented design. Fourth, security is often mentioned generally, but practical JWT and RBAC integration is not always described in module design. Fifth, performance evaluation is sometimes limited to user acceptance; workload-based response-time observations can give additional practical insight for an MCA project prototype.

The proposed system addresses these gaps by combining workflow communication, file tracking, document management, JWT authentication, RBAC authorization, audit logging, dashboards, and reports in one ERP module.

VI. Proposed Methodology

The proposed methodology follows an iterative prototype model suitable for a final-year MCA project. The system is first analyzed through university administrative scenarios such as leave approval, circular approval, purchase request, scholarship request, examination request, and departmental communication. Roles, permissions, file states, workflow actions, and reporting needs are identified. The design phase defines architecture, database collections, REST endpoints, user interface screens, and security rules. Implementation is carried out using React.js for the client, Express.js for server APIs, MongoDB for storage, JWT for authentication, and Tailwind CSS for interface styling. Chart.js is used for dashboards and analytics.

Testing is divided into functional testing, role-based access testing, document workflow testing, and basic performance testing. Functional testing verifies each module against expected behavior. Role-based testing checks whether users can access only the features allowed to their role. Workflow testing validates file states such as Draft, Submitted, In Review, Forwarded, Approved, Rejected, Returned, and Archived. Performance testing observes response time under increasing simulated request loads.

VII. System Architecture

The system follows a layered web architecture. The presentation layer contains React components for login, dashboard, file upload, workflow screens, user management, department management, and reports. The API layer contains Express routes and controllers. The service layer implements business rules such as workflow transitions, role permission checks, notification generation, and audit creation. The data layer stores users, departments, roles, permissions, workflows, files, comments, notifications, audit logs, and reports in MongoDB collections.

VIII. Workflow Communication Module

The workflow communication module manages the movement of requests and files between users. A workflow begins when an authorized user creates a request, uploads supporting documents, selects a workflow type, and submits it. The system determines the next approver based on the workflow template. Each approver

can view the file, add comments, approve, reject, return for correction, or forward to another authorized officer. Every action creates a time stamped event in the file history.

The module supports two workflow styles. In a predefined workflow, the approval path is configured in advance, such as Faculty to HOD to Dean to Registrar. In a flexible workflow, an authorized officer may forward the file to another role when consultation is required. This is useful in universities because administrative decisions may need input from finance, examination, academic, or legal sections depending on the file type.

IX. File Tracking Module

The file tracking module gives users visibility into document movement. Each file receives a unique tracking number. The current status, current holder, originator, department, priority, due date, and last action time are displayed. The complete movement history is shown as a timeline. Users can search by tracking number, title, department, file type, status, date range, or responsible officer.

For students and faculty, file tracking reduces the need for repeated office visits. For administrators, it helps identify pending files and delays. For management, dashboards show department-wise pending counts, average processing time, and files nearing deadline.

X. Role-Based Access Control

RBAC is used because university access rules are role-oriented. A student should be able to submit and track personal requests but not approve institutional files. Faculty may initiate departmental requests and view files assigned to them. HODs may approve department-level files. Deans may approve school-level requests. The Registrar may manage administrative approvals. The Chancellor and Vice Chancellor may view high-level files and approve selected workflows. The Admin manages users, departments, roles, workflow templates, and system configuration.

Table II. Role-Permission Matrix

Function	Admin	Chancellor / VC	Registrar	Dean	HOD	Faculty	Staff	Student
Manage users	Yes	View	No	No	No	No	No	No
Manage departments	Yes	View	Yes	View	No	No	No	No
Create workflow template	Yes	View	Yes	No	No	No	No	No
Upload file	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Approve file	Configured	Yes	Yes	Yes	Yes	Limited	Limited	No
Track own file	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
View all reports	Yes	Yes	Yes	Department	Department	No	No	No
Audit logs access	Yes	View	View	No	No	No	No	No

XI. JWT Authentication

JWT authentication is used to secure REST API calls. After successful login, the server issues a signed access token containing claims such as user ID, role, department ID, issue time, and expiry time. The client stores the token securely and sends it in the authorization header for protected requests. Middleware verifies the token before allowing access to protected APIs. The authorization layer then checks whether the authenticated role has permission for the requested action.

In the proposed prototype, short-lived access tokens are recommended. For production deployment, refresh tokens, token revocation, HTTPS-only cookies, password hashing, account lockout, and secure secret rotation should be implemented. JWT should identify the user and role, but sensitive document content should not be stored inside the token.

XII. Database Design

MongoDB is suitable for this project because workflow files have flexible metadata, embedded event histories, comments, and attachment lists. The design uses separate collections for core entities while using references for relationships. The main collections are Users, Roles, Departments, Workflow Templates, Files, File Events, Notifications, Audit Logs, and Reports.

Table III. Main Collections

Collection	Important Fields	Purpose
Users	userId, name, email, passwordHash, roleId, departmentId, status	Stores user accounts
Roles	roleId, roleName, permissions	Stores rolepermission mapping
Departments	departmentId, name, parentDepartmentId	Stores academic and administrative departments
WorkflowTemplates	templateId, name, steps, allowedInitiators	Defines approval paths

Files	fileId, trackingNo, title, type, status, ownerId, currentHolderId, priority	Stores file metadata
FileEvents	eventId, fileId, action, fromUser, toUser, remarks, timestamp	Stores file movement history
Notifications	notificationId, userId, message, status, createdAt	Stores alerts
AuditLogs	logId, userId, action, entityType, entityId, ipAddress, timestamp	Stores security and activity trail

XIII. Technology Stack

React.js is used for the frontend because it supports componentbased user interfaces and efficient state-driven rendering. Tailwind CSS is used for consistent and responsive styling. Node.js and Express.js are used for API development because they are well suited for JSON-based web applications and integrate naturally with MongoDB. MongoDB stores flexible document metadata and event histories. JWT secures stateless API authentication. REST APIs define communication between frontend and backend modules. Chart.js supports visual analytics such as pending files, approval time, and department-wise workload.

XIV. System Modules

The proposed system contains the following modules:

- Secure Login: Authenticates users and issues JWT tokens.
- User Management: Creates, edits, disables, and maps users to roles and departments.
- Department Management: Maintains academic and administrative departments.
- Workflow Management: Defines templates, steps, approvers, and routing rules.
- File Upload: Uploads PDF, image, and document files with validation.
- Document Management: Stores metadata, versions, and access permissions.
- File Tracking: Shows status, holder, timeline, and pending action.
- Approval Workflow: Supports approve, reject, return, and forward actions.
- Notifications: Alerts users about assigned files and status changes.

XV. Implementation

The backend exposes REST endpoints such as `/auth/login`, `/users`, `/departments`, `/workflows`, `/files`, `/files/:id/approve`, `/files/:id/reject`, `/files/:id/forward`, `/notifications`, and `/reports`. Each protected route passes through authentication and authorization middleware. File uploads are validated for file type and size before storage. Metadata is stored in MongoDB, and file actions are recorded in FileEvents and AuditLogs.

The frontend provides separate dashboards depending on role. Students and faculty see their own submitted files, pending corrections, and status updates. HODs and Deans see departmental approvals. Registrar and top-management roles see institutional reports and high-priority pending files. Admin users see configuration, users, roles, templates, and audit information.

XVI. Algorithms

Algorithm 1: Login and Token Generation

18. Receive email and password.
19. Find active user by email.
20. Compare password with stored password hash.
21. If invalid, return authentication error.
22. Create JWT claims with user ID, role, department, issue time, and expiry time.
23. Sign token using server secret.
24. Return token and user profile summary.

Algorithm 2: File Submission

25. Validate token and user permission.
26. Validate file metadata and uploaded document.
27. Generate unique tracking number.
28. Load workflow template based on file type.
29. Assign first approver as current holder.
30. Save file metadata with Submitted status.
31. Save file event as Submitted.
32. Create notification for first approver.
33. Return tracking number to originator.

Algorithm 3: Approval Routing

34. Validate approver token and permission.
35. Check whether the file is currently assigned to the approver.
36. Save approver remarks and action.
37. If approved and more steps exist, assign next approver and notify.
38. If approved and no steps remain, mark file Approved.
39. If rejected, mark file Rejected and notify originator.
40. If returned, assign file to originator for correction.
41. Record FileEvent and AuditLog.

XVII. Experimental Results

The prototype was evaluated using functional test cases and simulated workload observations on a development environment. Since the project is an MCA final-year prototype and not a live university deployment, the results should be understood as indicative rather than institution-wide production results.

Table IV. Functional Test Summary

Test Area	Test Cases	Passed	Result
Login and JWT validation	8	8	Successful
Role-based access	12	12	Successful

File upload and validation	10	9	One invalid file-size case correctly rejected
Workflow routing	15	15	Successful
Notifications	8	8	Successful
File tracking timeline	10	10	Successful
Reports and dashboard	8	8	Successful
Audit logging	8	8	Successful

Table V. Prototype Performance Observation

25	138 ms	172 ms	205 ms
50	185 ms	245 ms	290 ms
100	310 ms	420 ms	510 ms
Concurrent Users	Average Login Response	Average File List Response	Average Workflow Action Response
10	120 ms	145 ms	165 ms



The observations show that the prototype remains responsive for common departmental usage levels. Workflow actions are slower than login because they update file metadata, movement history, notification records, and audit logs. Indexing fields such as tracking number, status, current holder, department, and created date is important for maintaining search and dashboard performance.

XVIII. Performance Analysis

Performance depends mainly on database indexing, file size, network speed, and the number of workflow events per file. Read operations such as dashboard counts and file lists can be optimized through indexed queries and aggregation pipelines. Write operations such as approval and forwarding should be transactional in production so that file status, file event, notification, and audit log updates remain consistent. For large deployments, file binaries may be stored in object storage while

MongoDB stores metadata and references. The proposed system is scalable at module level because the frontend, API server, database, and file storage can be deployed separately. Caching frequently used role permissions and workflow templates can reduce repeated database reads. Pagination is required for audit logs and report views.

XIX. Comparison with Existing Systems

Table VI. Proposed System Compared with Traditional

Audit trail	Weak	Partial	Partial	Strong
Notifications	Phone/email follow-up	Partial	Partial	In-app and extendable
Dashboard analytics	No	Partial	No/partial	Yes
ERP integration	No	Yes	No	Yes
Document movement history	Registerbased	Partial	Partial	Complete timeline

and Existing Approaches

Feature	Manual File Movement	Generic ERP Module	Standalone DMS	Proposed System
Real-time tracking	No	Partial	Partial	Yes
University role hierarchy	Informal	Partial	Limited	Yes
Approval workflow	Manual	Partial	Limited	Yes

XX. Advantages

The proposed system improves transparency because every file has a visible status and movement history. It reduces delay by notifying responsible users and showing pending files. It supports accountability through audit logs. It improves security by combining JWT authentication and RBAC authorization. It reduces paper dependency and supports digital governance. It also helps management identify bottlenecks through dashboards and reports.

XXI. Limitations

The proposed system is a prototype and has not yet been deployed across a full university. Performance results are based on simulated workload, not production traffic. The system assumes that users will follow digital workflow rules; organizational adoption and training are still necessary. Advanced features such as digital signatures, blockchain-based verification, OCR, and integration with email/SMS gateways are outside the current prototype scope.

XXII. Future Scope

Future work can add digital signatures, certificate verification, OCR-based document indexing, email and SMS notifications, WebSocket-based real-time alerts, mobile applications, advanced analytics, and integration with existing admission, examination, finance, and HR modules. A future production study can evaluate user satisfaction, processing-time reduction, and adoption challenges across multiple departments. Machinelearning techniques may also be explored to predict workflow delays and recommend routing improvements.

XXIII. Conclusion

This paper presented a University Workflow Communication and File Tracking System in ERP for a higher education environment. The proposed system addresses common administrative problems such as unclear file status, delayed approvals, fragmented communication, weak auditability, and manual follow-up. By combining React.js, Node.js, Express.js, MongoDB, JWT authentication, RBAC, REST APIs, dashboards, notifications, and audit logs, the system provides a practical and secure workflow layer for university ERP. The literature review shows that ERP success in higher education depends not only on technology but also on user adoption, process alignment, governance, and transparency. The proposed design responds to these needs by making file movement visible, role-aware, and measurable. Prototype testing indicates that the system can support core workflow functions and remain responsive under moderate simulated load. With further deployment, integration, and user evaluation, the system can contribute to paperless, accountable, and efficient university administration.

Acknowledgement

The author expresses sincere gratitude to Prof. Kruparani S for guidance and encouragement throughout the project work. The author also thanks the Department of MCA, GM University, Davangere, Karnataka, India, for academic support and learning opportunities.

References

- [1] M. B. Sholeh, R. F. Samodra, and A. P. Widodo, "Benefits and Challenges of ERP Implementation in Higher Education Institutions: A Systematic Literature Review," **Jurnal Sistem Informasi Bisnis**, vol. 15, no. 1, pp. 21-33, 2025, doi: 10.14710/vol15iss1pp21-33.
- [2] A. Subbarao, A. Siddika, N. Khan, and L. Xumin, "Enhancing ERP Implementation in Higher Education: Addressing User Challenges and Refining Systems," **JOIV: International Journal on Informatics Visualization**, vol. 10, no. 3, pp. 1159-1167, 2026, doi: 10.62527/joiv.10.3.4190.
- [3] Yashmita and K. Srinivas, "Implementation of ERP as part of E-Governance in Higher Educational Institutions," **Indian Journal of Educational Technology**, vol. 8, no. 1, pp. 447-455, 2026.
- [4] J. Fattah-Weil, B. Bender, and N. Gronau, "StakeholderDriven ERP Implementation in German Higher Education: Key Challenges and Adoption Factors," **Information Systems and e-Business Management**, 2026, doi: 10.1007/s1025702600734-9.
- [5] A. K. Rizkiana, H. Ritchi, and Z. Adrianto, "Critical Success Factors Enterprise Resource Planning (ERP)

Implementation in Higher Education," **Journal of Accounting Auditing and Business**, vol. 4, no. 1, 2021, doi: 10.24198/jaab.v4i1.31551.

[6] A. K. Alhazmi, N. Alsakkaf, M. Motahar, and G. Alqubati, "ERPs in Higher Education Institutions: Motivations, Challenges, and Success Factors," **Journal of Science and Technology**, vol. 27, no. 2, 2023, doi: 10.20428/jst.v27i2.2051.

[7] J. L. G. Redoblo, "Online Multi-Function HEIs Tracking System," **Research Journal of Education, Science and Technology**, vol. 1, no. 1, pp. 24-35, 2021, doi: 10.63179/rjest.v1i1.52.

[8] F. V. Castro, E. R. C. Quilon, and A. C. Arruejo, "DocTrack: A Dynamic Document Tracking System for the University of Northern Philippines," **Journal of Innovative Technology Convergence**, vol. 4, no. 2, 2022, doi: 10.69478/JITC2022v4n2a03.

[9] M. Costoiu, V. Plesu, R. Isopescu, S. G. Soriga, S. H. Alesincu, and I. Arsene, "Electronic Document Management Information System for Universities," **Chemical Engineering Transactions**, vol. 29, 2012, doi: 10.3303/CET1229274.

[10] L. I. Mayasari, A. R. Syam, D. K. Robby, and V. A. Tricahyo, "Web-Based Document Management Systems for Administrative Services in Higher Education," **Proceeding of International Conference on Islamic Boarding School**, vol. 3, no. 1, 2026, doi: 10.61159/icop.v3i1.872.

[11] "A blockchain-based information management system for academic institutions: a case study of international students'

workflow," **Information Discovery and Delivery**, vol. 50, no. 4, pp. 343-352, 2021, doi: 10.1108/IDD-01-20210010.

[12] "Document Management System Empowered by Effective Amalgam of Blockchain and IPFS," **Procedia Computer Science**, vol. 215, pp. 340-349, 2022, doi: 10.1016/j.procs.2022.12.036.

[13] E. K. Donner, "Research data management systems and the organization of universities and research institutes: A systematic literature review," **Journal of Librarianship and Information Science**, 2022, doi: 10.1177/09610006211070282.

[14] G. B. Subiksa, M. P. A. Ariawan, and I. B. A. Peling, "Optimalisasi Pelayanan Dokumen Digital dengan Mempergunakan Cloud File Management System," **Jurnal Teknologi Informasi dan Komputer**, vol. 8, no. 4, 2022, doi: 10.36002/jutik.v8i4.2095.

[15] R. Sandhu, D. F. Ferraiolo, and D. R. Kuhn, "The NIST Model for Role-Based Access Control: Towards a Unified Standard," in **Proc. 5th ACM Workshop on RoleBased Access Control**, 2000, doi: 10.1145/344287.344301.

[16] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," RFC 7519, IETF, 2015, doi: 10.17487/RFC7519. [17] MongoDB, "Database References," **MongoDB Manual**, accessed Jul. 23, 2026. [Online].

Available:

<https://www.mongodb.com/docs/manual/reference/databasereferences/>