IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Academic Affairs Management System (Aams)

[1]Anirudh .Y, [2] Sri Chaitanya .D, [3] Srujan .G, [4] Rajeswari . T

[1, 2, 3,4] Student

[1] Computer Science and Engineering Department,

Hyderabad Institute of Technology and Management, Hyderabad, Telangana, India.

Abstract: Academic institutions manage a wide range of official documents such as permission requests, reports, letters, and circulars as part of their daily administrative activities. Traditional manual methods of handling these documents are often time-consuming, inefficient, and difficult to track. This paper presents a **web-based Academic Affairs Management System (AAMS)** designed to streamline academic document management through a centralized digital platform. The system supports structured submission, review, approval, and secure storage of academic documents, thereby reducing paperwork and improving administrative efficiency. By organizing workflows in a systematic manner, the proposed system enhances coordination among departments and ensures better record management. The Academic Affairs Management System provides a practical and efficient solution for improving academic administration in higher educational institutions.

Index Terms - Academic Affairs Management System, Academic Administration, Document Management System, Digital Workflow, Paperless System.

I. INTRODUCTION

Educational institutions are required to manage a large number of academic and administrative documents on a regular basis. These documents include permission requests, internal reports, official letters, circulars, and approval forms that move across multiple departments and levels of authority. In many institutions, such activities are still handled using traditional paper-based methods. While these methods have been followed for years, they often result in delays, increased workload, lack of transparency, and difficulties in tracking document status.

Manual document handling also increases the risk of document loss, duplication, and miscommunication between departments. As institutions grow in size and complexity, the need for an organized and efficient system to manage academic affairs becomes increasingly important. The absence of a centralized platform makes it difficult to maintain records, monitor approvals, and ensure accountability.

To overcome these challenges, a digital approach to academic administration is essential. The Academic Affairs Management System (AAMS) is proposed as a web-based solution that enables systematic handling of academic documents through a centralized platform. The system supports digital submission, review, approval, and storage of documents, reducing dependency on paper-based workflows. By streamlining academic processes and improving coordination among departments, AAMS enhances administrative efficiency and supports transparent governance within educational institutions. The system is designed to provide a structured, reliable, and user-friendly environment for managing academic affairs effectively.

The Major Project version of the IAOMS is a further development of these Mini Project to which it has now been added on the computer. It has the same type of excellent facilities and features but has now introduced additional advanced features and intelligent automation taken a step further all computerised facilities have been installed which fully take care of the original pattern.

II. Literature Survey

The rapid growth of information technology has encouraged educational institutions to adopt digital systems for managing academic and administrative activities. Several studies have been conducted on document management systems, academic administration platforms, and workflow automation to improve institutional efficiency. This section reviews relevant literature related to digital academic management systems.

Various researchers have highlighted the limitations of traditional paper-based academic administration. Manual handling of documents often leads to delays, lack of transparency, difficulty in record maintenance, and increased administrative burden. Studies emphasize that paper-based systems are inefficient when dealing with large volumes of documents and multi-level approval processes.

Research on **Document Management Systems (DMS)** suggests that digital platforms significantly improve document organization, accessibility, and retrieval. Such systems provide centralized storage and controlled access, reducing the risk of document loss and duplication. Several authors have reported that digital document workflows help institutions maintain accurate records and ensure better coordination among departments.

Studies related to **Academic Management Systems** focus on automating institutional processes such as student records, staff administration, and internal communication. These systems have been found to enhance operational efficiency by reducing manual effort and improving data accuracy. Researchers also note that centralized systems support better monitoring and reporting of academic activities.

Workflow-based systems for approval management have also been discussed in the literature. These systems allow documents to move systematically through predefined approval stages, enabling better tracking and accountability. Research findings indicate that workflow automation reduces processing time and improves decision-making efficiency in organizational environments.

Some studies further highlight the importance of **role-based access control** in academic systems to ensure security and data confidentiality. By assigning specific roles and permissions, institutions can protect sensitive information and maintain control over document access and approvals.

Overall, the literature indicates that digital academic administration systems offer significant advantages over traditional methods. However, many existing systems focus on limited functionalities or are complex to implement. The proposed **Academic Affairs Management System (AAMS)** aims to address these gaps by providing a simple, structured, and efficient platform for managing academic documents and approval workflows in higher educational institutions.

III. Existing System

1. ERP-Based Academic Management Systems

- **SAP Student Lifecycle Management (SLcM)**
  - Full-featured ERP for higher education
  - Handles document workflows, approvals, and reporting
  - Supports role-based access for staff and administrators

- **Oracle PeopleSoft Campus Solutions**
  - Centralized platform for student and administrative management
  - Includes document tracking, notifications, and dashboards

2. Open-Source Academic Systems

- **Fedena**
  - School and college management system
  - Document submission, approval workflows, notifications
  - Role-based access for Employees, HODs, Principal
- **OpenEduCat**
  - Open-source ERP for educational institutions
  - Manages student records, document approvals, and reporting
  - Supports multiple departments and workflows
- **EduSec**
  - Open-source higher education ERP
  - Academic document management and workflow approval
  - Secure record keeping and multi-role support

3. Learning Management Systems (LMS) with Document Features

- **Moodle**
  - Supports assignment submissions and approvals
  - Can integrate plugins for workflow automation
  - Real-time notifications via email and in-app
- **Blackboard**
  - Course management with document submission and review
  - Role-based access and notifications for staff and students

4. Cloud/Collaboration Platforms

- **Microsoft SharePoint for Education**
  - Centralized document storage and workflow automation
  - Version control, approvals, and notifications
  - Integration with institutional login systems
- **Google Workspace for Education**
  - Document creation, sharing, and approval workflows
  - Real-time collaboration and notifications
  - Can integrate AI-based tools (e.g., Google Gemini) for summaries.

IV. Proposed system

Key Features of the Proposed System

1. Centralized Document Management

- All academic documents, including permission requests, reports, letters, and circulars, are managed from a single platform.
- Reduces paperwork, duplication, and loss of records.

2. Role-Based Access Control (RBAC)

- Secure authentication using Supabase Auth with Google Sign-In restricted to @hitam.org and HITAM ID/password login.
- Access permissions tailored for Employees, HODs, Principal, and Admin.

3. Real-Time Workflow & Notifications

- Document status updates, approvals, and rejections are synchronized in real-time using Supabase Real-time Database.
- Instant notifications via WebSocket API and AI-enhanced email alerts.

4. Secure Cloud Storage Integration

- Documents are stored and managed securely on Google Drive.
- Ensures easy retrieval, archival, and audit tracking.

5. AI-Assisted Document Summarization

- Gemini AI generates concise summaries of documents for faster review.
- Reduces review time for HODs and Principal while highlighting key information.

6. Responsive User Interface

- React + Tailwind CSS dashboards with a White & Green theme for clarity and usability.
- Accessible on desktops, tablets, and mobile devices for flexible usage.

7. Approval & Tracking System

- Multi-stage workflow: Employee → HOD → Principal.
- Tracks the complete history of approvals, rejections, and modifications.

8. AI-Enhanced Communication

- Automated email notifications and in-app alerts are enhanced with AI-generated templates for professionalism and clarity.

9. Efficient Coordination & Transparency

- Provides a clear view of all ongoing document workflows across departments.
- Ensures accountability and reduces administrative delays.

10. Customizable & Scalable

- The system can be easily customized to meet additional institutional requirements.
- Scalable for handling growing academic document loads across multiple departments.

V. System Architecture

5.1 Functional Requirements

- Users can register and log in using Google Sign-In or HITAM ID/password.
- Role-based access for Employees, HODs, Principal, and Admin.
- Employees can upload documents (reports, letters, circulars, permission requests).
- HODs and Principal can review, approve, or reject documents.
- Track document status in real-time (pending, approved, rejected).
- Send notifications via in-app alerts and email.
- AI summarizes documents for faster review.
- Generate reports of submissions, approvals, and rejections.
- Search and filter documents by type, date, department, or status.
- Responsive dashboards for desktops, tablets, and mobile devices.

5.2 Non-Functional Requirements

- System responds quickly (within 2–3 seconds).
- Real-time updates of document status.
- Reliable with 99% uptime and backup features.
- Secure authentication and data encryption.
- Easy-to-use interface and accessible on multiple devices.
- Modular design for easy maintenance and future upgrades.
- Integration with Google Drive, Supabase Database, AI summarization, and email services.
- Audit logs and monitoring for all activities.

5.3 User Requirements

Employees:

- Upload and submit documents.
- Track status of documents.
- Receive notifications for approvals or rejections.
- Search past submissions.

HODs:

- Review and approve/reject employee documents.
- Receive notifications for pending documents.
- View AI summaries for quick decision-making.
- Monitor departmental workflows.

Principal:

- Final approval for documents.
- View status of all documents in the institution.
- Generate reports for workflow and departmental performance.
- Receive AI-enhanced notifications.

Admin:

- Manage users, roles, and permissions.
- Monitor system performance and workflow activity.
- Ensure backups and integration with storage and AI tools.

All Users:

- Easy-to-use, responsive, and secure interface.
- Reliable system with minimal downtime.
- Access to help or support for guidance.

5.4 Hardware and Software Requirements

5.4.1 Hardware Requirements

- Server: Processor Intel Core i5 or AMD Ryzen 5, 8 GB RAM (16 GB recommended), 500 GB storage (1 TB recommended), stable internet connection, and backup storage.
- Client Devices: Processor Intel Core i3 or equivalent, 4 GB RAM (8 GB recommended), 100 GB free storage, stable internet, and access via desktop, laptop, tablet, or mobile device.

5.4.2 Software Requirements

- Server: Operating System Windows Server 2016 or Linux (Ubuntu 20.04 or higher), backend using Node.js with TypeScript, Supabase Realtime Database, Google Drive API, Google Gemini API for AI summarization, SendGrid/Gmail API for email, and web server like Nginx or Apache.
- Client: Operating System Windows 10 or higher / macOS / Android / iOS, web browser (Chrome, Firefox, Edge, Safari), frontend using React.js with Tailwind CSS.
- Tools & Utilities: Git for version control, IDE such as VS Code or WebStorm, Node Package Manager (npm) for dependencies.

5.5 Frontend Implementation

Frontend Technologies

- React.js:
  - A modern JavaScript library for building interactive, component-based user interfaces.
  - Supports reusable components and efficient rendering.
- Tailwind CSS:
  - Utility-first CSS framework for responsive, clean, and customizable UI design.
  - Ensures a consistent White & Green theme throughout the platform.
- HTML5 & CSS3: Core technologies for structuring and styling web pages.
- JavaScript / TypeScript: For client-side logic, event handling, and integration with backend APIs.
- Axios / Fetch API: For communicating with backend services and fetching data in real-time.

- React Router: For navigation and handling multiple pages/views within the web application.

5.5.1 Key Frontend Features

Responsive Dashboard:

- Role-specific dashboards for Employees, HODs, Principal, and Admin.
- Displays pending tasks, notifications, and document workflow summaries.

Real-Time Notifications:

- In-app alerts for document submission, approvals, rejections, and modifications.
- Visual indicators for pending actions.

Document Management Interface:

- Upload, view, and download academic documents.
- Display document status (pending, approved, rejected) clearly.

Forms and Input Validation:

- User-friendly forms for submission, approval, and comments.
- Automatic validation for required fields, file type, and size.

Search and Filter:

- Search documents by type, date, department, or status.
- Filter options to quickly locate specific records.

Navigation and User Experience (UX):

- Intuitive menu layout and smooth page transitions.
- Quick access to all functionalities from the main dashboard.

Integration with Backend and APIs:

- Real-time data from **Supabase Realtime Database**.
- Secure file upload/download via **Google Drive API**.
- Notifications and AI summaries integrated seamlessly.

Accessibility and Compatibility:

- Works across desktops, tablets, and mobile devices.
- Compliant with basic accessibility standards for readability and usability.

5.6 Backend Implementation

Backend Technologies:

- **Node.js:**
  - JavaScript runtime for building fast, scalable, and event-driven server-side applications.
  - Handles multiple requests concurrently, ensuring smooth performance.
- **TypeScript:**
  - Superset of JavaScript that adds static typing for more secure and maintainable code.
  - Reduces runtime errors and improves code quality.
- **Supabase (Realtime Database):**
  - Cloud-hosted PostgreSQL database with real-time data synchronization.
  - Stores document metadata, workflow status, and user activity logs.
- **Google Drive API:**
  - Secure storage and management of academic documents.
  - Enables uploading, downloading, and version tracking of files.
- **WebSocket API:** Provides real-time notifications for document submission, approval, and workflow updates.
- **SendGrid / Gmail API:** For sending automated email alerts with AI-enhanced templates.
- **Gemini AI:** AI service for document summarization and workflow assistance.

5.6.1 Key Features of Backend

1. **User Authentication and Role-Based Access Control (RBAC):**
   - Secure login via Google Sign-In (@hitam.org) or HITAM ID/password.
   - Enforces access permissions for Employees, HODs, Principal, and Admin.
2. **Document Workflow Management:**
   - Supports multi-stage approval: Employee → HOD → Principal.
   - Tracks document status in real-time and updates dashboards instantly.
3. **Secure Document Storage:**
   - Stores files in **Google Drive** with proper access control.
   - Maintains version history for auditing and recovery purposes.
4. **Real-Time Updates and Notifications:**
   - Uses **Supabase Realtime Database** and **WebSocket API** for instant updates.
   - Sends automated email alerts with AI-enhanced summaries for better decision-making.
5. **AI-Enhanced Processing:**
   - **Gemini AI** generates summaries of uploaded documents to assist HODs and Principal.
   - Reduces review time and highlights key points for faster approvals.
6. **Reporting and Analytics:**
   - Generates workflow reports, document status summaries, and departmental activity analytics.
   - Data can be exported in PDF or Excel for audits and record keeping.
7. **Scalability and Performance:**
   - Handles multiple concurrent users and high-volume document submissions efficiently.
   - Modular backend architecture allows future feature expansion.
8. **Security and Reliability:**
   - HTTPS communication and secure API access.
   - Maintains logs for all document actions and user activities.

5.6.2 Database Architecture

The database architecture of AAMS is designed to provide a **secure, scalable, and real-time data management system**, ensuring efficient tracking of academic documents, workflows, notifications, and user actions. It supports multiple roles, ensures data integrity, and integrates seamlessly with AI and cloud storage services for higher education administration.

5.7 Deployment Implementation

Setup: Install backend and frontend frameworks; configure database and APIs.
Frontend Development: Build dashboards, forms, and navigation; connect to backend APIs.
Backend Development: Create APIs for authentication, document handling, workflows, and notifications.
Testing: Conduct unit testing, integration testing, and user testing.
Deployment: Host frontend and backend on cloud servers; connect with database and APIs.
Maintenance: Monitor system performance, backup data, and update features regularly.

Proposed Improvements

Integrate **advanced AI** for document classification and action suggestions.
Enhance **reporting and analytics** with charts and workflow insights.
Introduce **granular role-based access control** and sub-roles.
Integrate with **existing institutional systems** like ERP or student management.

Add **SMS or push notifications** along with email alerts.
Enable **offline mode** for drafting and syncing documents.
Improve **document versioning and history tracking** for auditing.
Implement **two-factor authentication (2FA)** and stronger encryption.
Enhance **user experience** with simplified navigation and tutorials.

VI. RESULTS

1. Enables digital submission and management of reports, letters, circulars, and permission requests.
2. Employees, HODs, and Principal can track document status in real-time.
3. Automated notifications reduce delays and improve workflow efficiency.
4. AI-assisted summaries help faster review and decision-making.
5. Multi-level approval workflow is streamlined and transparent.
6. Role-based access ensures secure handling of documents.
7. Cloud deployment allows access from desktops, tablets, and mobile devices.
8. Document metadata and version history are securely stored.
9. Search and filter options enable quick document retrieval.
10. Reporting and analytics provide insights into departmental performance.
11. Reduces paperwork and administrative workload.
12. Improves coordination and communication among departments.
13. Enhances accountability and transparency in approvals.
14. Demonstrates efficiency over traditional manual systems.

Overall, AAMS provides a user-friendly, secure, and scalable solution for efficient academic administration.

REFERENCES

1. Laudon, K. C., & Laudon, J. P. (2020). *Management Information Systems: Managing the Digital Firm*.16th Edition. Pearson.
2. Sommerville, I. (2016). *Software Engineering*. 10th Edition. Pearson.
3. Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach*. 9th Edition. McGraw-Hill Education.
4. Pautasso, C., Zimmermann, O., & Leymann, F. (2008). *RESTful Web Services vs. "Big" Web Services: Making the Right Architectural Decision*. Proceedings of the 17th International Conference on World Wide Web.
5. Google Developers. (2025). *Google Drive API Documentation*. https://developers.google.com/drive
6. Supabase Docs. (2025). *Realtime Database and Authentication*. https://supabase.com/docs
7. Gemini AI. (2025). *AI-Powered Text Summarization and NLP Tools*. https://developers.google.com/gemini
8. SendGrid. (2025). *Email API Documentation*. https://sendgrid.com/docs
9. Crockford, D. (2008). *JavaScript: The Good Parts*. O'Reilly Media.
10. React Official Documentation. (2025). https://reactjs.org/docs/getting-started.html
11. Tailwind CSS Documentation. (2025). https://tailwindcss.com/docs
12. Node.js Documentation. (2025). https://nodejs.org/en/docs