

Digital Hierarchical Approval System for College Administration

Madhavi Dixit, Dhyan H, Chetan Kumar C, Avinash H, Sujay Devadiga

Department of Artificial Intelligence and Machine Learning, T John Institute of Technology, Bengaluru, India

Received: 20 Nov 2025; Accepted: 03 Feb 2025; Date of Publication: 09 Feb 2026

©2026 The Author(s). Published by Infogain Publication. This is an open-access article under the CC BY license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract— In most colleges, student and teacher requisitions for approval of leave letters, bonafide certificates (and others) and permits to hold events are all completed in the manual process with paper. Unfortunately, this results in physical travel between Class Teachers, HODs, and the Principal, where it is often delayed, lost or stressful in times of urgency. Our work seeks to engage an eApproval – Digital Hierarchical Approval System, creating an automated and digital process. It can accommodate role-based logins and allows for users to create requests, see their status and receive recorded, digitally signed approvals. There is the possibility for continuous step-by-step processing, along with real-time notifications and guaranteed digital signatures for the requisitions. It simplifies the process, reduces paperwork and time, and allows administration to be completed more quickly, transparently and reliably.

Keywords— *Digital Hierarchical Approval System, Role- Based Login, Automated Requisition Processing, Real- Time Notifications, Paperless Administration.*

I. INTRODUCTION

In conventional educational establishments, the process of getting consents for academic and administrative requests such as leave letters, project permissions, or certificate applications is mostly manual and very slow. The students are usually required to write the requests by hand, then physically hand them over to the class teacher, and wait for the teacher's signature. After that, the same document is sent in turn to the Head of Department (HOD) and the Principal. The procedure causes the delay that is not necessary and increases the risk of losing, duplicating, and not being able to trace the documents. The digital transformation in higher education has made it imperative to have an efficient system that is structured, paperless, and simplifies the workflow.

The system proposed is called the College E-Approval System. This system will use a web-based platform to automate the multi-level approval but in a secure way. The system will be based on the Flask framework which will present the students with an online digital

environment to submit their requests, and the faculty at every hierarchical level will be able to either approve or reject the requests through electronic means. Role-based access, digital signatures, and PDF generation are all coming together to guarantee that each approval is valid, can be tracked, and is stored for later reference. Not only that, the system will provide real-time tracking of every request thereby increasing the transparency, accountability, and time efficiency within the system. It is through this project that academic institutions are gradually moving towards sustainable, paperless administration, and such practices also play a part in the broader digitalization and e- governance policies of the institution.

II. METHODOLOGY

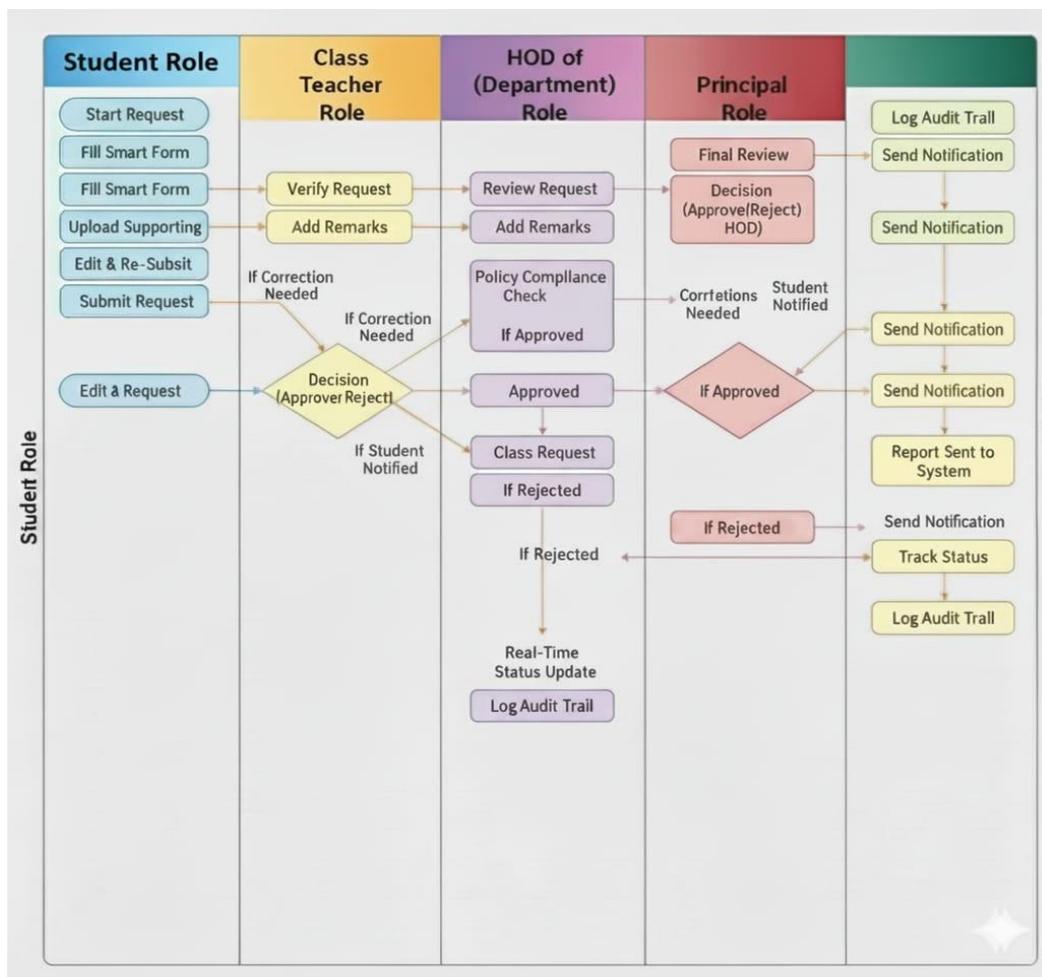
The College E-Approval System was developed by us through a considerate and systematic method which mixed up important software development practices. First of all, we sought to know the difficulty which the students and the faculty were facing with the manual approval systems and so, we gathered their feedback

by means of informal chats and observations. This in turn helped us to figure out which features were absolutely necessary for the new system— such as user logins, personalized dashboards, approval workflows, and automatic generation of PDFs.

The creation of the system was done by using the Flask web framework, and it was divided into three main parts which were done according to the Model-View-Controller (MVC) structure. The Model portion takes care of data, and it uses SQLAlchemy to take care of the user accounts along with the approval requests and their statuses. As for the View part, we built HTML templates for the content that changes depending on who the user is, and it could be either dashboards or forms for new requests. The Controller part is the one that deals with users and handles the actions such as

logging in or approving requests and creating PDFs.

Setting up a step-by-step approval process was one vital improvement that we made. What this means is that a request can only be moved on after the required approvals have been obtained which indicates that every level of authorization ranging from Class Teachers to the Principal will be able to see only those requests which were approved by the previous person in charge. Finally, through the ReportLab library, we included a feature that could generate digitally signed PDFs thereby giving the system the capability of creating official documents which have the details of the request as well as the signatures of the approvers. In summary, our method gives the assurance that the system is productive, flexible, and satisfies the institution’s requirements.



III. SYSTEM DESIGN

The College E-Approval platform is designed to facilitate seamless interactions among its different users, including students, teachers, department heads, and principals. At the backbone, we're using Flask, a web framework that helps in handling backend

processes like routing, session handling, and database interactions through a tool called SQLAlchemy. The whole system uses a three-tier architecture, including the user interface, application logic, and data layers. This structure not only makes it easier to maintain but also keeps things organized.

On the user interface side, HTML, CSS, and Jinja2 templates were used. Each user type has a different interface, depending on the user's role: for example, students will be allowed to make and follow up on their requests, while faculty will be able to approve or reject these requests or even download their final documents in PDF format. The dashboard shows, in summary, each request status, pending approval or finalized, in color.

Application logic controls the flow of the request and permissions depending on the user role. If a student creates a request, the request goes to the database as "Pending Class Teacher." From there, it follows the stages for approval, going through Class Teacher to HOD and then Principal, based on the actions of the respective approvers. The routing system in Flask ensures that every action—approve, reject, view requests, generate PDFs—gets processed in a secured manner with respect to permissions and login status of users. We also use session tokens to keep users logged in and to safeguard against unauthorized access.

For data management, we rely on SQLAlchemy to interact with an SQLite database. There are two significant tables: User and ApprovalRequest. The User table stores information about users, their roles, and paths of their digital signatures, while the ApprovalRequest table includes information such as the request title and description, current status, timestamp, and users involved. This structure enables us to easily fetch and update information. We can add new features, such as email notifications, cloud storage options, or mobile access, without perturbing the existing workflow. This comprehensive design well balances simplicity, functionality, and user-friendliness for all stakeholders.

Entity-Relationship Design

The two entities in this database schema are User, and ApprovalRequest. The User entity has the following attributes: id, username, password (hashed), role, and signature path.

The ApprovalRequest entity has fields for id, title, description, status, created_by, timestamp, and an approval field for each role. There is a one-to-many relationship between the User and the ApprovalRequest entity, so each user can have more than one request; this maintains full user ownership and accountability.

IV. IMPLEMENTATION

The College E-Approval System was developed in several phases that prioritized usability, modularity, and scalability. The application runs on a local Flask development server and combines frontend and backend technologies to create a smooth workflow for users in different roles. The main logic is in Flask routes, which securely handle requests and direct them based on user hierarchy. Each feature—from login authentication to digital PDF generation—was built with a strong focus on reliability, security, and user experience.

4.1 Technology Stack

The proposed system leverages a modern and lightweight technology stack optimized for rapid development and cross-platform compatibility. The frontend layer was developed using HTML5, CSS3, and Jinja2 templates, ensuring responsive and role-specific user interfaces. These templates dynamically adjust the content displayed to each user, depending on their role as a student, class teacher, head of department, or principal.

The backend is built using the Flask framework in Python, responsible for handling routing, session management, and server-side business logic. Flask's simplicity and modularity allowed for quick integration of additional components such as authentication, PDF rendering, and error handling.

For data management, the project employs SQLAlchemy ORM, which provides a Pythonic interface for interacting with relational databases such as SQLite or MySQL. SQLAlchemy abstracts SQL queries into Python classes and methods, improving maintainability and reducing the likelihood of syntax-related errors.

User authentication is implemented using Flask-Login, coupled with Werkzeug Security for password hashing. This ensures that sensitive credentials are securely stored and verified using SHA-256 encryption. The ReportLab library was integrated to handle PDF generation, enabling the system to automatically create official approval documents with embedded digital signatures.

Deployment was initially carried out using Flask's built-in local server for testing and demonstration. However, the application architecture has been designed with future scalability in mind, allowing easy migration to AWS EC2, Google Cloud, or institutional intranet hosting for production-level deployment.

4.2 Functional Flow

The eApproval system demonstrates a multi-tier approval process in a hierarchical way that corresponds to the real-world academic administration setup. The workflow starts with user authentication, where the user logs in with pre-defined credentials associated with his/her role. After authentication, the user is sent to a dashboard that automatically displays only those requests which are relevant to him/her in the approval chain.

The student can present a new request by means of a submission form, including a title and thorough content. After submission, a request is registered in the database with the default status "Pending Class Teacher." Class

Teachers when they log in can view all requests made by students and they are in a pending state, and upon checking, they have the option to either approve or reject them. The approval by the Class Teacher will directly send the request to the HOD and update the status to "Pending HOD."

The same procedure continues at the HOD and Principal levels, with the system automatically updating and routing requests according to the current approval stage. The request once approved by the Principal triggers the automated PDF generation. The PDF document will consist of the request details, applicant's information, approval status, and signatures of the respective authorities embedded digitally.

All actions of approval, rejection, or forwarding are recorded with timestamps, hence providing full visibility and responsibility. This flow function gets rid of all redundancies, cuts down manual work, and provides real-time updates for all concerned parties.

4.3 Security Features

During the implementation, security was an important open issue to safeguard user credentials, opportunities for unauthorized access, and ensure the integrity of the data. Passwords can only be stored in the database after hashing them with SHA-256 encryption, making it unreadable even when the data is leaked. The role-based access control (RBAC) system guarantees that a user can only perform actions allowed for that user role (i.e. a student can submit requests, while teachers, HODs, and principals can review and approve requests according to hierarchical order).

Session management is managed with the use of

Flask-Login, which reserves secure session tokens for any authenticated user thereby restricting privileged routes to logged-in sessions only. Additionally, all forms of input are validated on the server-side to not allow empty or ill-formatted submissions and reduced input injection possibilities. Future versions of the system can even utilize SSL encryption and two-factor authentication (2FA) to ensure deployment security.

V. ALGORITHM AND WORKFLOW

The core of the system is the Dynamic Multi-Level Approval Algorithm, which has been implemented in the Flask backend to manage the request flow among roles. This algorithm alters its functionality automatically according to the interactions of the user and the status of the request, thus making sure that the approval process will take place only when the authorized actions are carried out. When a user logs in, his/her role is recognized by the system and the related records are obtained from the database. In case the user is a student, a new request is created by the system with the following default statuses: Class Teacher = Pending, HOD = Pending, and Principal = Pending. After that, the request is sent to the dashboard of the Class Teacher.

With the approval of the Class Teacher, the request is transferred to the queue of the HOD, where the database status is changed to "Pending HOD." Likewise, when the HOD approves, the status modulates to "Pending Principal." If the Principal's decision is in favor of the request, the final status is altered to "Approved" and at this point, the ReportLab PDF generation module is called. The module generates a digitally signed approval letter and saves it with the utmost confidentiality in the specified folder.

In the case when any approver at any level decides to reject the request, the workflow is interrupted by the system and the record is marked as "Rejected by [Role]." This safeguard is meant to provide a strict and transparent flow of decisions with no hierarchical validation being bypassed.

The algorithm has been structured in such a way that it is capable of scaling—additional levels of approval like "Dean" and "Registrar" can be easily incorporated with minor code changes. This modular system not only allows for the institutions of different scales and hierarchies to embrace the system easily into their administrative structure but it also applies to the case

of institutions that are of various sizes and have different hierarchies.

VI. RESULTS AND DISCUSSION

The College E-Approval System turned out to be a successful venture and was subjected to testing under a simulated college arrangement that was made up of student, class teacher, HOD, and principal accounts. The manual approval process was put against the digital approvals and the results of the comparisons were impressive. For example, requests that used to take one or two days to circulate among the departments were processed and approved within minutes via the digital workflow. The system's automated routing based on roles was such that no request skipped the hierarchical stage, hence the legitimacy of institutional approvals was preserved.

Functional testing, every user role was subjected to a test for the system's usability and correctness. Students were able to create and track requests very easily, while the faculty and administrators enjoyed the clarity of the dashboard interface that summarized all pending and completed approvals with color-coded indicators. The system's PDF generation module was working efficiently by producing digitally signed documents that could be downloaded or printed for official use. Moreover, the application of timestamps and role-based logs formed a complete audit trail for every transaction which in return ensured that the approval process was completely transparent and accountable.

In performance testing, the application that was built on the Flask framework, managed to cope with the concurrent users efficiently in the case of moderate to high load, thus the assumption that the chosen lightweight framework and

database (SQLite) would only be enough for small to mid-scale institutional deployments was confirmed. The modular structure of the system not only facilitated but also made it cheaper to extend functionalities such as email notifications or SMS alerts without major architectural shifts. Pilot users' positive comments underlined the system's simplicity, accessibility, and real-time feedback.

VII. CONCLUSION

The College E-Approval System is a powerful tool that not only removes the obstacle of slow approval from

the processes but also gives an efficient remedy to it — that is, the case of manual approval that takes up a lot of time and effort both for academic and institutional requests. Coating the system with web-based automation, hierarchical workflow management, and secure digital documentation makes the transition to paperless easy for colleges and also, at the same time, the system offers a practical and scalable solution for them.

The built-up structure of a multi-level approval chain, role-based authentication, and PDF-based record generation guarantees the proposed system's transparency, efficiency, and accountability. The testing results and user feedback indicate a remarkable decrease in processing time and administrative burden. The user-friendly interface and modular structure also grant it the flexibility to conform to various school setups without the necessity of extensive customization.

Ultimately, the College E-Approval System is a giant step forward in the digital transformation of the academic administration. It also corresponds with the primary aims of Smart Campus Initiatives and Digital India in the way of facilitating better governance and eliminating environment-polluting practices. With technological advancements like cloud integration, mobile access, and sophisticated analytics, this system can develop into a total online approval and workflow management platform for academic ecosystems. With continued enhancements such as cloud integration, mobile accessibility, and advanced analytics, this system has the potential to evolve into a full-fledged digital approval and workflow management platform for educational ecosystems.

ACKNOWLEDGEMENT

The authors would like to thank the team of six members and their guide for their helpful advice and ongoing support during this research. We appreciate our institution for providing the resources and computing tools we needed. We also want to thank everyone who supported and encouraged us throughout this research journey.

REFERENCES

- [1] Tanuraharja, C. D., Tiara, K. A., Wang, G., & Alianto, H. (2022, November). Applying for E-Signature Approval with TOGAF Framework to Improve Productivity: Case Study SAP Document Management

- System. In *2022 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)* (pp. 77-81). IEEE.
- [2] Saha, A., Ranjan, R., & Sahana, B. C. (2025, March). Advancement of Loan Approval System with Machine Learning Framework. In *2025 IEEE 14th International Conference on Communication Systems and Network Technologies (CSNT)* (pp. 29-33). IEEE.
- [3] Uthayopas, P., & Ketprom, U. (2022, March). Standard and Guideline Development for Digital Government Interoperability. In *2022 International Conference on Digital Government Technology and Innovation (DGTi-CON)* (pp. 30-33). IEEE.
- [4] Pahlawan, M. R., et al. (2024, November). Development of Digital Signature Features Using MVVM Architecture. In *2024 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)* (pp. 876-879). IEEE.
- [5] Widiyanto, W. W., et al. (2022, November). Implementation of RSA Algorithm for Secure Digital Signature Validation. In *2022 International Interdisciplinary Humanitarian Conference for Sustainability (IIHC)* (pp. 599-605). IEEE.