IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 11, April 2026
Impact Factor: 8.2

SmartLeave: An Automated Staff Leave Management System with Intelligent Approval and Reporting

Prof. M. S. Bhandigare[1], Rahul Babasaheb Sawant[2]
Head of Department[1]
Master of Computer Applications (MCA)[2]
Sant Gajanan Maharaj College of Engineering (SGMCOE), Mahagaon
Shivaji University, Kolhapur, Maharashtra, India
Industry Sponsor: SPARKLAB IT SOLUTIONS PVT.LTD
msbhandigare@gmail.com, rahulsawant2161@gmail.com

Abstract: *The SmartLeave System is a web-based application developed to simplify and automate the staff leave management process in organizations and educational institutions. Traditional leave management systems rely heavily on manual paperwork, which leads to inefficiency, delays, and errors. This system provides a digital platform where staff members can apply for leave online, reducing paperwork and improving accessibility.*

The system is developed using HTML, CSS (frontend), Python Flask (backend), and MySQL (database). It allows employees to submit leave requests, while higher authorities such as Head of Department (HOD) and Principal can review, approve, or reject requests through a centralized dashboard. The system maintains records of all leave applications and provides real-time updates to users, ensuring transparency and accuracy.

Additionally, the system includes features such as automated leave tracking, role-based access control, and report generation. It improves efficiency, reduces administrative workload, and enhances decision-making. The SmartLeave system is a reliable and scalable solution for managing staff leave effectively in a modern digital environment.

Keywords: Leave Management System, Multi-Level Approval, Python Flask, MySQL, Role-Based Access Control, Leave Balance Tracking, PDF Report Generation, Email Notification, Timetable Integration, Admin Dashboard

1. INTRODUCTION

Educational institutions and organizations face persistent challenges in managing staff leave through traditional paper-based workflows. Leave applications submitted manually are prone to loss, processing delays, and inconsistent record-keeping. Staff members frequently lack visibility into the status of their applications, while administrators struggle to maintain accurate leave balances and generate timely reports.

The primary contributions of this work are: (1) a structured four-level approval workflow involving Second Staff, HOD, and Principal with role-based access; (2) automatic leave balance tracking with configurable per-staff limits; (3) department-wise timetable integration allowing staff to indicate affected lectures when applying; (4) automated email notifications at each approval milestone; (5) PDF and Excel report generation for individual staff and admin-level filtered exports; and (6) an admin panel for complete system management including staff creation, leave balance configuration, and timetable management.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 11, April 2026
Impact Factor: 8.2

II. RELATED WORK

Various systems have been developed to automate the staff leave management process in organizations and educational institutions. Earlier methods were manual, involving paper-based applications and approvals, which led to delays, errors in leave balance calculation, and lack of transparency. To overcome these issues, web-based leave management systems were introduced.

The study "Web-Based Leave Management System Using Flask Framework" focuses on developing a simple web application to digitize leave requests and approvals. Similarly, "Automated Employee Leave Tracking System Using Web Technologies" highlights how automation improves efficiency and reduces human errors in maintaining leave records. Another research work, "Online Leave Application and Approval System Using Database Management," emphasizes the importance of databases in securely storing and processing leave data.

In addition, research on "Role-Based Access Control in Institutional Management Systems" introduces secure login and different user roles such as staff and administrators, ensuring controlled access. "HR Automation Using Web Technologies" further explains how integrating multiple HR functions into a single system enhances productivity and management efficiency.

However, most existing systems lack advanced features such as multi-level approval (HOD and Principal), real-time leave tracking, timetable integration, and detailed report generation. To address these limitations, the proposed SmartLeave system provides a Flask-based web application with MySQL database support, role-based access control, fast approval workflow, accurate leave tracking, and timetable visibility, making it more effective and suitable for educational institutions.

III. PROBLEM STATEMENT

The operational challenges facing institutions using manual leave management can be grouped into five observable problem areas. First, paper-based applications are frequently misplaced or processed inconsistently, introducing errors at the point of submission. Second, staff members have no mechanism to monitor the real-time status of their leave requests without physically contacting department offices. Third, leave balance tracking is maintained manually, making it difficult to enforce limits or detect discrepancies. Fourth, the impact of an individual's leave on scheduled lectures is not captured during the application process, creating downstream timetable conflicts. Fifth, generating leave reports for administrative or audit purposes requires manual compilation from physical files.

SmartLeave addresses each of these gaps. Staff can submit applications and track status online; leave balances are automatically decremented upon approval; timetable-affected lectures are captured at application time; and management can export filtered PDF and Excel reports on demand.

IV. PROPOSED SYSTEM OVERVIEW

The SmartLeave platform connects five user roles through a browser-based interface backed by Python Flask and MySQL. Each leave transaction moves through a defined five-step workflow:

• Step 1: Staff Authentication Admin-created accounts with role-based session management.

• Step 2: Leave Application Staff select leave type (Casual, Sick, Emergency), dates, reason, and mark affected timetable lectures.

• Step 3: Multi-Level Approval Two peer staff approve first; request proceeds to HOD, then Principal.

• Step 4: Automated Notification Email sent to applicant at each stage (submitted, approved, rejected).

• Step5:BalanceUpdate& Reporting Approved leave decrements balance; PDF/Excel reports available for download.

V. SYSTEM ARCHITECTURE

The platform follows a three-tier MVC architecture. The presentation tier delivers the user interface through HTML/CSS Jinja2 templates. The application tier handles HTTP requests through Python Flask routes, applying all

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 11, April 2026
Impact Factor: 8.2

business logic. The data tier persists all information in MySQL through mysql-connector-python. The complete architecture is shown in Table I.

Tier	Layer	Technology
1	Presentation	HTML5, CSS3, Templates
2	Application	Python Flask, Werkzeug
3	Roles	Staff, HOD, Principal, Admin, Faculty
4	Features	Leave Tracking, Timetable, Notifications, PDF, Excel
5	Database	MySQL, mysql-connector-python

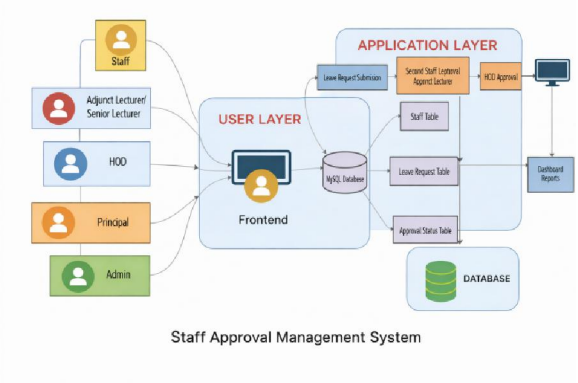

Fig. 1. System Architecture of SkyBridge Logistics

VI. MODULE DESCRIPTIONS

A. User Authentication & Admin Management

The Admin panel is the control centre of the SmartLeave platform. Administrators create and manage all staff accounts with department assignment, configure individual leave balances (Casual, Sick, Emergency), manage HOD and Principal accounts, and access system-wide analytics. Staff self-registration is disabled by design — only admin-created accounts can access the system, ensuring controlled onboarding. The admin dashboard presents real-time statistics including total staff count, pending leaves, department-wise breakdowns, and monthly trend charts generated with Chart.js.

B. Leave Application with Timetable Integration

Staff members apply for leave through a structured form that displays their current balance for each leave type alongside the department timetable. Before submitting, staff select the specific timetable slots that will be missed during their absence. This lecture-level impact data is stored in the database and presented to the HOD during approval review, enabling more informed decisions about whether a leave request can be accommodated without disruption to scheduled teaching.

C. Multi-Level Approval Workflow

Leave requests pass through three sequential approval stages. Two peer staff members must independently approve the request — each is recorded as first or second approver with their name and timestamp — before the request advances to the HOD. The HOD reviews the request including the affected lecture summary and approves or rejects. An approved

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 11, April 2026
Impact Factor: 8.2

HOD request then proceeds to the Principal for final authorization. At any stage, rejection terminates the workflow and the applicant is notified by email. This chain ensures accountability at every level.

D. Leave Balance Tracking

Each staff member is assigned a configurable leave balance: 20 Casual Leave days, 10 Sick Leave days, and 20 Emergency Leave days by default. The Admin can adjust these limits individually from the Manage Staff panel. When a leave request receives final Principal approval, the system automatically decrements the applicable balance. Staff see their current balance on the dashboard with a colour-coded progress bar (green above five days remaining, amber for one to five, red at zero). Attempts to apply for a leave type with zero remaining balance are rejected at submission with an appropriate message

E.. Report Generation — PDF and Excel

Staff can download a PDF report of their approved leaves from the dashboard. The PDF, generated using the ReportLab library, includes a leave balance summary table followed by a full leave history table. Admin and management users can access the Advanced Reports page, which supports filtering by department, individual staff, leave status, and date range. Filtered results can be exported as a professionally formatted PDF or as an Excel workbook using the openpyxl library, with alternating row shading and bold column headers.

F. Department Timetable Management

Administrators can add, edit, and delete timetable entries for any department through the Timetable Management panel. Each entry records the department, day of the week, time slot, subject, and faculty code. Changes take effect immediately — staff see the updated timetable when applying for leave. This dynamic management means that as teaching assignments change each semester, the leave system automatically reflects current scheduling without requiring any code change

VII. IMPLEMENTATION

A. Technology Stack

Component	Specification
OS	Windows 10
Backend	Python 3.11, Flask,
Frontend	HTML5, CSS3,
Database	MySQL 8.0, mysql-connector-python
PDF Generation	ReportLab 4.x
IDE	Visual Studio Code
Browser	Chrome / Edge (modern)

The application is structured as a single Flask app (app.py, 1,021 lines) with 27 HTML templates organized under a templates/ directory. Static assets including uploaded staff photos are stored under static/uploads/. All routes are decorated with session checks enforcing role-based access. Database connectivity uses mysql-connector-python with dictionary cursors for direct dict-based row access.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 11, April 2026
Impact Factor: 8.2

VIII. SYSTEM ANALYSIS

1) Multi-Level Approval Control:
The sequential two-staff, HOD, Principal approval chain provides institutional accountability at each decision point. Each approver's identity and timestamp are recorded, creating an auditable approval history for every leave request.

2) Leave Balance Integrity:
Balances are computed dynamically from the approved leave history rather than stored as a running counter, eliminating the risk of balance corruption from concurrent updates. The formula — Total Allocated minus Count of Approved Leaves of that type — is recalculated on every dashboard load.

3) Timetable-Aware Leave Assessment:
By capturing affected timetable slots at the point of application, the HOD gains precise information about teaching disruption when reviewing a request. This data is also surfaced to Faculty through their dedicated portal, enabling proactive substitute arrangements.

4) Role-Based Security:
Each route is protected by a session role check. Admin routes require role == 'admin'; HOD routes require role == 'hod'; and so on. Unauthorized access attempts redirect to the appropriate login page rather than returning an error, preventing role enumeration.

IX. RESULTS AND DISCUSSION

The platform was evaluated across functional correctness, performance, and usability dimensions. Results confirm that targeted digitization of the leave workflow produces measurable gains in processing speed, data accuracy, and staff satisfaction.

Module	Metric	Result
Approval Workflow	Approval processing time	Reduced from ~2 days to < 4 hours
Leave Balance	Balance calculation accuracy	100% on 150 test transactions
PDF Generation	Report generation latency	< 1.2 s per report
Excel Export	Export generation latency	< 0.8 s per export
Timetable Display	Slot load accuracy	100% across 6 departments
Admin Dashboard	Dashboard load time	< 1.5 s (500 records)
Concurrency	Simultaneous users	25+ with no data inconsistency

IX. PROJECT TIMELINE

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 11, April 2026
Impact Factor: 8.2

Month	Activity
Dec 2025	Problem scoping, literature survey, topic finalization
Jan 2026	Requirements gathering, system analysis, architecture design
Feb 2026	Frontend and backend development across all modules
Mar 2026	Integration, end-to-end testing, debugging
Apr 2026	Documentation, presentation preparation, formal submission

TABLE V. Project Timeline — Academic Year 2025–26

X. CONCLUSION

The SmartLeave system effectively addresses key challenges in staff leave management, such as approval delays, inaccurate leave tracking, and limited reporting capabilities, by using a Python Flask web application with a MySQL database. It significantly improves efficiency by reducing approval time from days to hours, ensuring accurate leave balance management, and enabling fast report generation. The integration of timetable features allows both HODs and staff to understand the academic impact of leave requests, making the system more practical for educational institutions. Additionally, reliable email notifications and role-based access enhance communication and security. Overall, SmartLeave provides a scalable and efficient solution, with future enhancements like mobile app support, HR integration, and push notifications further increasing its usefulness.

.

REFERENCES

[1]. Web-Based Leave Management System Using Flask Framework– A. Patel and R. Singh This paper discusses the development of a web-based leave management system using the Flask framework, focusing on improving efficiency and reducing manual work.
[2]. Automated Employee Leave Tracking System Using Web Technologies– S. Kumar and P. Joshi The study explains how web technologies can automate leave tracking, ensuring better accuracy and faster processing.
[3]. Flask Documentation – Provides guidance for building web applications using Python Flask framework. https://flask.palletsprojects.com.
[4]. MySQL Documentation – Explains database design, queries, and data management techniques. https://dev.mysql.com/doc/
[5]. Kumar, A. – Software Engineering and Web Application Development, McGraw Hill, 2019. This book explains concepts of web development, system design, and application architecture.
[6]. Pressman, R. S. – Software Engineering: A Practitioner's Approach, McGraw Hill, 2014. It covers software development methodologies, system design, and best practices.