Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

Smart E-Permit App

Ravinder Mogili
Professor and Head
Dept. of Computer Science and Engineering
Jyothishmathi Institute of Technology and Science
(JNTUH)
Karimnagar, Telangana, India

Pranitha Polsani
Associate professor
Dept. of Computer Science and Engineering
Jyothishmathi Institute of Technology and Science
(JNTUH)
Karimnagar, Telanagana, India

Ramya Reddaboina
UG Student
Dept. of Computer Science and Engineering
Jyothishmathi Institute of Technology and Science
(JNTUH)
Karimnagar,Telanagana,India

Srihitha Macherla
UG Student
Dept. of Computer Science and Engineering
Jyothishmathi Institute of Technology and Science
(JNTUH) Karimnagar, Telanagana, India

Varshitha Porandla
UG Student
Dept. of Computer Science and Engineering
Jyothishmathi Institute of Technology and Science
(JNTUH)
Karimnagar, Telanagana, India

Alekhya Ellala
UG Student
Dept. of Computer Science and Engineering
Jyothishmathi Institute of Technology and Science
(JNTUH) Karimnagar, Telanagana, India

ABSTRACT - The Smart Leave Request Management System is a mobile application developed using Flutter and Firebase to automate the leave approval process in educational institutions. Students can submit leave requests with selected dates, leave type, and custom timings. The request is first reviewed by the Incharge and then forwarded to the HOD for final approval. The system uses Firebase Authentication for secure login and Firestore for real-time data storage and status tracking. This application reduces paperwork, improves transparency, speeds up approvals, and provides an efficient digital solution for managing student leave requests.

KEYWORDS - Smart Leave Management System, Mobile Application, Flutter, Firebase, Cloud Firestore, Firebase Authentication, Role-Based Access Control, Digital Leave Approval, Real-Time Database, Educational Institution Management System.

INTRODUCTION

The Smart Leave Request Management System is a mobile application developed to simplify and digitize the student leave approval process in educational institutions. Instead of using manual paperwork, students can apply for leave through the app by selecting the date, leave type, and reason. The request is reviewed by the Incharge and then forwarded to the HOD for final approval. The system uses Flutter for the frontend and Firebase for authentication and database management. This application improves efficiency, reduces delays, and ensures transparency in the leave approval workflow.

LITERATURE SURVEY

- Alade, Samuel Mayowa et al. (2022) presented a Web Based Leave Management System which emphasizes efficiency in leave tracking and history maintenance
- Abubakar Adamu (2020) in Employee Leave Management System demonstrated the automation of leave request and approval in institutional settings
- Praveen. T, Vetrivel H.M, Arumugasamy B(2020)built an Online Leave Management System for colleges allowing students to apply leave, check status, and cancel their leave .
- Mohammed & Rahman (2021) proposed a mobilebased leave system with SMS notifications, but did not support real-time push notifications or advanced role-based dashboards.
- Li & Wang (2021) implemented a blockchain-based secure leave system, improving data integrity but lacking mobile usability and instant notifications.
- Srinivas & Rao (2022) developed a college leave system with conditional approvals, but without proper multi-level approval and push notification support.

Drawback: Several papers highlight automation and cloud support, but many lack features like multi - level approval (e. g., class advisor → HOD → principal), role based dashboards, and real - time notifications tailored for educational institutions

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

PROBLEM STATEMENT

In many colleges, the permission approval process is manual, paper-based, and time-consuming. Students must physically submit applications, and faculty members face difficulty tracking multiple requests, leading to delays, miscommunication, and lack of transparency.

Existing digital solutions often lack academic-specific workflows, multi-level approvals, real-time notifications, and proper record management. Therefore, there is a need for a mobile-first, role-based Smart E-Permit System that streamlines leave requests, enables instant approvals, ensures transparency, and maintains secure digital records.

OBJECTIVE

- To digitize and automate the traditional paper-based permission process in colleges.
- To implement a structured multi-level approval workflow (Student → Incharge → HOD → Admin).
- To provide role-based dashboards for students, faculty, HOD, and admin.
- To enable real-time notifications for request submission, approval, and rejection.
- To maintain secure digital records of all leave requests for tracking and analysis.
- To support document upload for valid leave verification.
- To improve transparency, efficiency, and accountability within the institution.

EXISTING AND PROPOSED SYSTEM

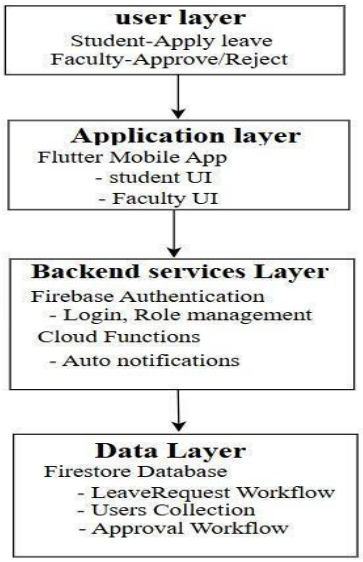
1. Existing System

- The permission process is paper-based and handled manually.
- Students must physically submit applications and collect signatures from faculty members.
- Approval process is slow and depends on the availability of staff.
- Prone to errors, misplacement of documents, and lack of proper record maintenance.
- No centralized database to track past or pending requests.
- Students have no real-time updates about their request status.
- Generic online forms (if used) lack structured multi-level approval and role-based control.
- Limited transparency and accountability in the workflow.

2. Proposed System

- A fully digital, mobile-first Smart E-Permit System with secure login.
- Role-based access for Student, Incharge, HOD, and Admin. • Students can submit leave requests online with document upload support.
- Multi-level approval workflow (Student → Incharge → HOD → Admin).
- Real-time notifications for submission, approval, and rejection.
- Centralized database for secure storage and tracking of all records.
- Dashboard analytics for admin to monitor request statistics and trends.
- Faster processing, improved transparency, and reduced paperwork.

SYSTEM ARCHITECTURE



PROPOSE METHOD

The proposed system is a mobile-based Leave Request Management System developed using Flutter and Firebase. The system replaces manual leave applications with a digital workflow.

In this method

1. The student logs into the application using secure authentication.
2. The student submits a leave request by selecting the date, leave type, timing, and reason.
3. The request is stored in the Firestore database with a default status of "Pending".
4. The request is forwarded to the Incharge for review.
5. If approved, it is escalated to the HOD for final approval.
6. The final status (Approved/Rejected) is updated in the database and reflected in the student dashboard.

This method ensures a transparent, role-based, and real-time

IMPLEMENTATION

The Smart E-Permit System is implemented using modern mobile and cloud technologies to ensure scalability, security, and real-time performance.

Frontend Implementation

The frontend of the application is developed using:

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

- Flutter Framework – For building a cross-platform mobile application with a single codebase.
- Dart Programming Language – Used for implementing business logic and UI design.
- Material Design 3 UI Components – For creating a modern, responsive, and user-friendly interface.
- Form Validation – Ensures correct input of leave details such as reason, dates, and document upload.
- Role-Based UI Screens – Separate dashboards and functionalities for: •Student
•Incharge
•HOD
- **Navigation & Routing** – Redirects users based on their roles after login.
•**StateManagement (Provider/Riverpod)** – Manages user authentication and leave request data efficiently.
- **File Upload** Support – Allows students to upload supporting documents (PDF/JPG/PNG).
- **Notification UI –** Displays real-time updates for approvals and rejections.

Backend Implementation

The backend services are implemented using Firebase cloud services:
- **Firebase Authentication** – Provides secure login and user identity management. • **Cloud Firestore Database** – Stores:
•User details (role, department, email)
•Leave request information
•Approval status and remarks
•Notification records
- **Role-Based Access Control** – Ensures only authorized users can perform specific actions.
- **Real-Time Database Updates** – Enables instant status updates and notifications.
- **Cloud Storage** – Stores uploaded leave documents securely.
- **Firebase Security Rules** – Protects data from unauthorized access.

System Integration
- The frontend communicates with Firebase services using secure APIs.
- Real-time listeners update the UI instantly when approval status changes.
- All leave workflows follow a structured multi-level approval process.
- Data is securely stored in the cloud, ensuring reliability and availability.

ALGORITHMS

Step 1: Start
Step 2: User opens the application **Step 3:**Login using credentials
• Authenticate using Firebase Authentication
• If invalid → Show error and stop
• If valid → Identify user role (Student / Incharge / HOD)
Step 4: IF role = Student
a. Enter leave details (date, type, reason)
b. Validate input fields
c. Store leave request in Firestore
d. Set status = "Pending"
e. Notify Incharge **Step 5:** IF role = Incharge
a. Retrieve pending requests
b. Review request
c. Approve or Reject
d. If Approved → Forward to HOD
e. Update status in database
f. Notify Student **Step 6:** IF role = HOD
a. Retrieve Incharge-approved requests
b. Review request
c. Approve or Reject
d. Update final status (Approved / Rejected)
e. Notify Student
Step 7: Student checks final status in application
Step 8: End

RESULT ANALYSIS

The Smart E-Permit System was successfully implemented and tested using Flutter for the frontend and Firebase for backend services. The results obtained from the developed system demonstrate that the application works efficiently and fulfills the objectives of automating the leave approval process.

1. Login Authentication Result

The login module was tested using Firebase Authentication. Users were able to log in securely using their registered credentials. Based on the authenticated user role (Student, Incharge, or HOD), the system automatically redirected users to their respective dashboards. Invalid login attempts were handled properly with error messages.

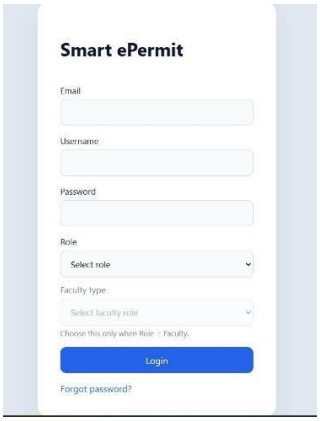


Figure 1: Login Screen

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

2. Student Leave Request Submission

Students were able to successfully submit leave requests by entering required details such as leave date, leave type, timing, and reason. The form validation ensured that incomplete or incorrect data could not be submitted. After submission, the request was stored in the Cloud Firestore database with the default status set to **"Pending."**

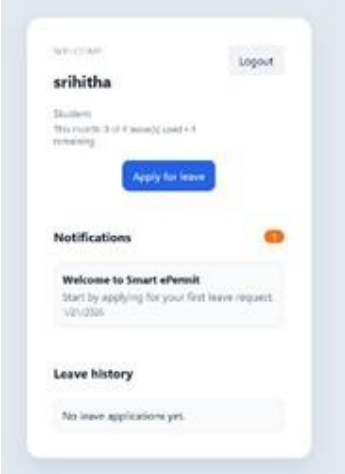


Figure 2: Student Dashboard

3. Leave Request Storage in Database

All submitted leave requests were stored securely in the Firestore database. The database maintained records including:

- Student details
- Leave type
- Leave date and timing
- Leave reason
- Approval status

This centralized storage allowed easy retrieval and tracking of leave requests.

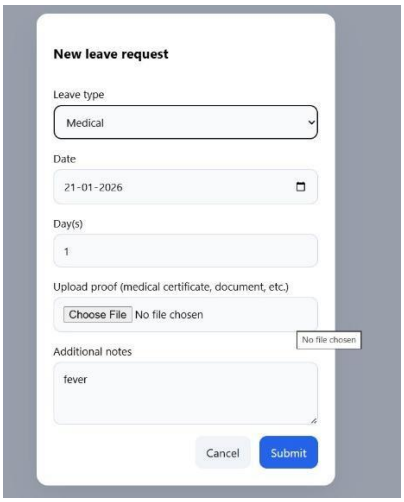


Figure 3: Leave Request Form

4. Incharge Approval Result

The Incharge dashboard displayed all pending leave requests submitted by students. The Incharge was able to review each request and either approve or reject it. If approved, the request was automatically forwarded to the HOD for final approval. The system successfully updated the request status in real time.

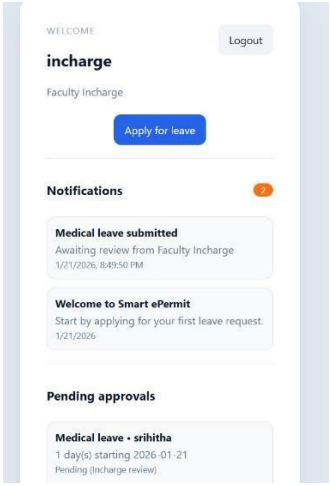


Figure 4: Incharge Approval Screen

5. HOD Final Approval

The HOD dashboard displayed all leave requests that were approved by the Incharge. The HOD could review the request and provide final approval or rejection. Once the HOD made a decision, the final status was updated in the database and reflected immediately in the student dashboard.

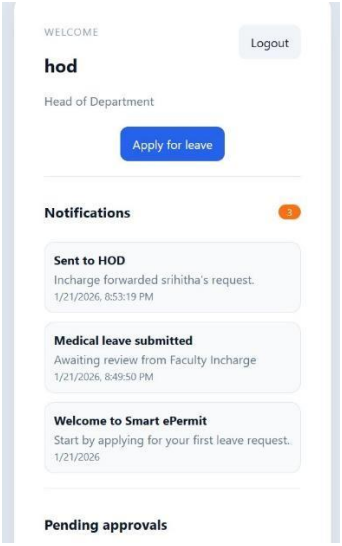


Figure 5: HOD Approval Screen

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

6. Real-Time Status Updates

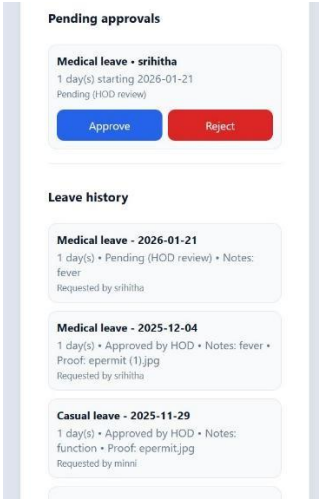


Figure 6: Leave Status Result Screen

One of the major outcomes of the system is the realtime update feature. Whenever the Incharge or HOD approved or rejected a request, the status was instantly updated in the student interface. This eliminated delays and improved transparency in the leave approval workflow.

7. User Interface Result

The application interface was developed using Flutter with Material Design components. The UI was responsive, user-friendly, and easy to navigate. Separate dashboards for Students, Incharge, and HOD ensured role-based access and simplified system interaction.

8. Overall System Performance

Testing results indicate that the system performs reliably with quick response times and accurate data handling. The application successfully reduced manual paperwork and streamlined the leave approval process. The digital workflow improved efficiency, minimized communication gaps, and maintained secure records of leave requests.

CONCLUSION

The Smart Leave Request Management System successfully digitizes and streamlines the traditional leave approval process in educational institutions. By using Flutter for the frontend and Firebase for backend services, the system ensures secure authentication, real-time data updates, and role-based access control.

The application reduces paperwork, minimizes delays, improves transparency, and enhances communication between students, Incharge faculty, and the HOD. Testing results confirm that the system is reliable, efficient, and userfriendly.

Overall, the project achieves its objectives and provides a practical, scalable solution for managing leave requests digitally in a structured and secure manner.

REDFERENCES

[1] Google, “Flutter – UI toolkit for building natively compiled applications,” *Flutter Documentation*. [Online]. Available: https://flutter.dev

[2] Google, “Dart Programming Language,” *Dart Documentation*. [Online]. Available: https://dart.dev

[3] Google, “Firebase – Backend-as-a-Service Platform,” *Firebase Documentation*. [Online]. Available: https://firebase.google.com

[4] Google, “Cloud Firestore Documentation,” *Firebase Documentation*. [Online]. Available: https://firebase.google.com/docs/firestore

[5] Google, “Firebase Authentication Documentation,” *Firebase Documentation*. [Online]. Available: https://firebase.google.com/docs/auth

[6] I. Sommerville, *Software Engineering*, 10th ed. Boston, MA, USA: Pearson Education, 2016.

[7] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner’s Approach*, 8th ed. New York, NY, USA: McGraw-Hill Education, 2015.