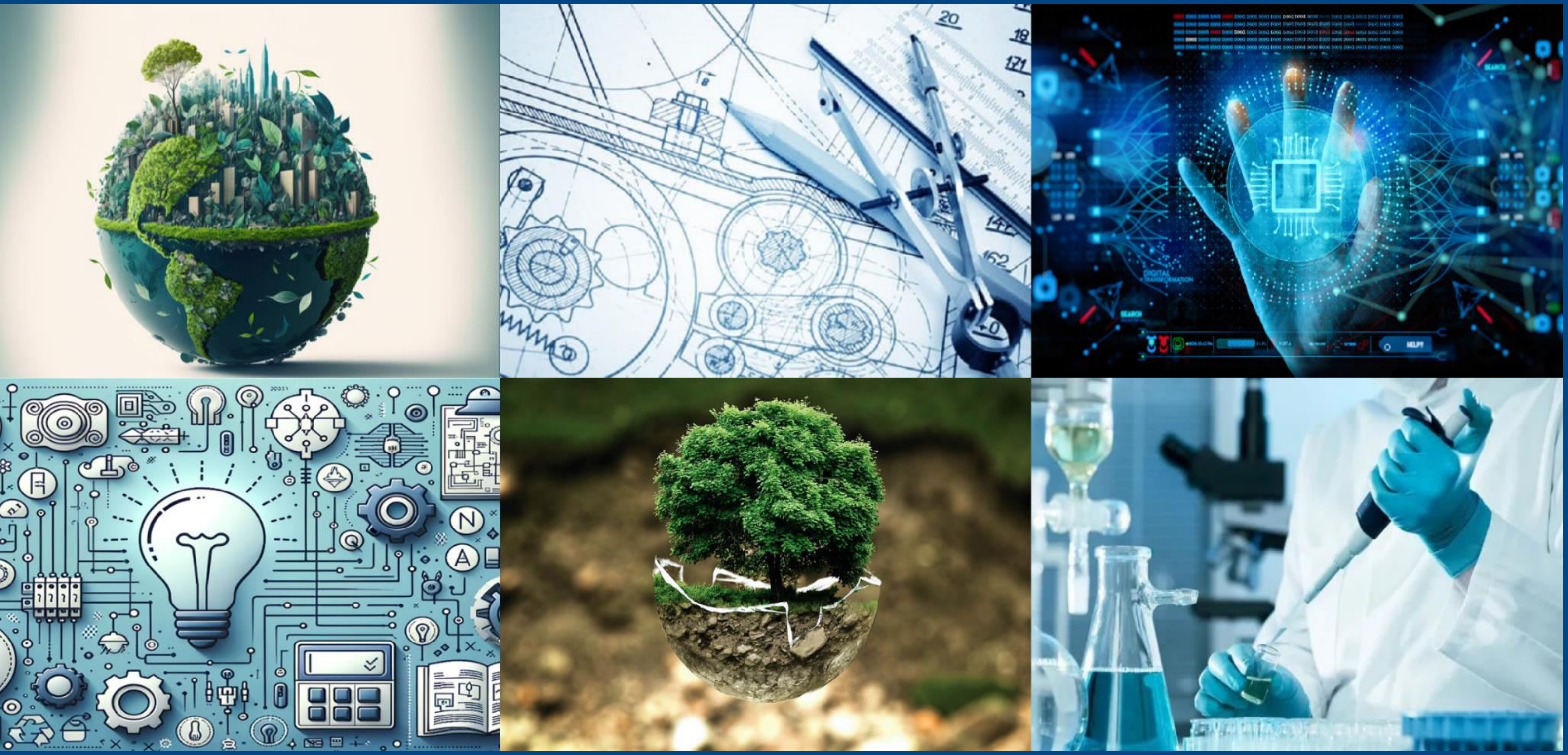
ISSN: 2582-7219

International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206

Volume 8, Issue 3, March 2025

www.ijmrset.com ijmrset@gmail.com +91 6381 907 438 +91 63819 07438

© 2025 IJMRSET | Volume 8, Issue 3, March 2025| DOI: 10.15680/IJMRSET.2025.0803046

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Computer Lab Maintenance App

Ayesha Inamdar[*1], Rajni Kamthe[*2], Vaishnavi Kahwale[*3], Mrs.P.A.Garad[*4]

Department of Computer Engineering, Jaywantrao Sawant Polytechnic, Pune, Maharashtra, India[*1*2*3]

Professor, Department of Computer Engineering, Jaywantrao Sawant Polytechnic, Pune, Maharashtra, India[*4]

ABSTRACT: This presentation introduces a novel "Computer Lab Maintenance App" designed to streamline and enhance the management of computer labs in educational and professional settings. The app addresses critical challenges faced by lab administrators, including inefficient issue tracking, cumbersome maintenance scheduling, inadequate inventory management, and difficulties in optimizing lab usage. Key features of the app include a user friendly interface for reporting and resolving technical issues, automated maintenance scheduling, a comprehensive inventory database, and tools for lab usage tracking and access control. By leveraging technology and data driven insights, this app promises to significantly improve lab efficiency, reduce downtime, and enhance the overall user experience

KEYWORDS: Computer lab management, Incident Reporting, Preventive maintenance, System monitoring, Issue tracking, Ticketing system, Notifications & Alert, Report Generation

I. INTRODUCTION

A Computer Lab Maintenance App is a tool designed to streamline the management and upkeep of computer labs. It helps track hardware and software inventory, schedule routine maintenance, and monitor the status of devices. The app allows users to report technical issues, request repairs, and keep a record of maintenance activities. With features like issue tracking, lab usage scheduling, and alerts for software updates, the app ensures that the lab operates efficiently, minimizing downtime and ensuring that all systems are functional and up to date.It helps faculties, lab technicians, and the Head of Department (HOD) to report, track, and resolve issues related to lab equipment, systems, and infrastructure.

II. METHODOLOGY

The development of the Computer Lab Maintenance App follows a structured approach to ensure it is efficient, user-friendly, and effective for managing lab maintenance activities. The process includes understanding user needs, designing the system, developing and testing it, and finally deploying it for real-world use.

A. Requirement Analysis

To identify user needs, surveys and discussions were conducted with IT administrators, lab technicians, and faculty members. The key requirements identified include:

A centralized system for issue reporting and tracking. A preventive maintenance scheduling module to reduce equipment failures. A user-friendly dashboard for monitoring lab assets and maintenance logs. Automated notifications and alerts for pending tasks. Role-based user authentication and access control.

B. System Architecture

The system follows a client-server model, ensuring seamless interaction between users and the platform. The key components include:

Frontend (User Interface): Developed using Flutter, providing an interactive interface for issue reporting, maintenance tracking, and asset management.

Backend (Server-Side Processing): Implemented using Node.js with Express.js or Django with Python, handling issue tracking, notifications, and maintenance workflows.

Database: A relational database like MySQL/PostgreSQL is used to store asset details, maintenance logs, and user credentials.

C. Implementation

The system development follows a modular approach for better maintainability and scalability:

1. Frontend Development: The UI is designed for easy issue reporting, maintenance scheduling, and real-time status updates.
2. Backend Development: The server processes requests, maintains issue logs, and manages user roles.
3. Issue Tracking Module: Users can report system issues, track their status, and receive updates.
4. Preventive Maintenance Module: IT staff can schedule and monitor routine maintenance to avoid unexpected failures.
5. Notification System: Automated email/SMS alerts notify users about pending tasks and maintenance schedules.

D. Maintenance & Monitoring Module

This module enables efficient tracking and resolution of computer lab issues:

- Issue Reporting: Users can submit hardware/software-related complaints.
- Status Updates: IT staff can update the status of reported issues (e.g., Pending, In Progress, Resolved).Preventive Maintenance: Schedule routine hardware checks, software updates, and security audits. Real-Time Dashboard: Displays system health, issue logs, and pending maintenance tasks.

E. Testing and Evaluation

To ensure the reliability and efficiency of the system, extensive testing was conducted:

- Unit Testing: Verified the functionality of individual modules such as issue reporting, notifications, and asset tracking.
- Integration Testing: Ensured seamless communication between frontend, backend, and database.
- Usability Testing: Conducted with IT staff and lab administrators to gather feedback for improvements.
- Performance Testing: Evaluated system performance under high user loads to ensure smooth operation.

F. Future Enhancements

The system was deployed on cloud-based hosting platforms such as AWS, Heroku, or Firebase, allowing accessibility without installations. Future updates may include:

- AI-driven issue prediction for proactive maintenance.
- Mobile app integration for on-the-go issue reporting.
- IoT-based hardware monitoring to track system health in real time

III. MODELING AND ANALYSIS

A. System Model

The Computer Lab Maintenance App follows client-server architecture, ensuring efficient communication between users and the system. The key components include:

Frontend: Developed using Flutter, providing an intuitive interface for issue reporting, maintenance scheduling, and status tracking

Backend: Implemented using Node.js with Express.js or Django with Python, handling user requests, maintenance workflows, and notifications.

Database: A relational database like MySQL/PostgreSQL stores asset details, issue logs, user credentials, and maintenance schedules.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

B. Data Flow Model

The system follows a three-tier data flow model to process maintenance requests efficiently:

Table I: System Data Flow

Process Stage	**Input**	**Processing Action**	**Output**
Issue Reporting	User submits the Maintenance request	Backend logs issue details	Issue added to the maintenance report
Status update	Staff update issue status	Backend modifies database record	Update issue status displayed
Preventive maintenance	Schedule maintenance event	System send notification	Remind alert sent to users
Report Generation	Admin request a maintenance report	Backend retrieves log data	Report displayed

C. UI Component Model

The Flutter-based UI consists of the following modules:

Issue Reporting Panel: Allows users to submit maintenance requests with issue descriptions and attachments. Status Tracking Panel: Displays real-time updates on reported issues and scheduled maintenance.

Notification Module: Sends push notifications for scheduled maintenance, pending tasks, and issue resolutions. Admin Dashboard: Provides an overview of system health, maintenance logs, and issue reports.

D. Performance Analysis

To ensure the efficiency, reliability, and scalability of the Computer Lab Maintenance App, an in-depth analysis is performed based on various key performance metrics. This includes functional analysis of Issue Reporting Maintenance Scheduling Status Tracking & Notifications Admin Dashboard & Report

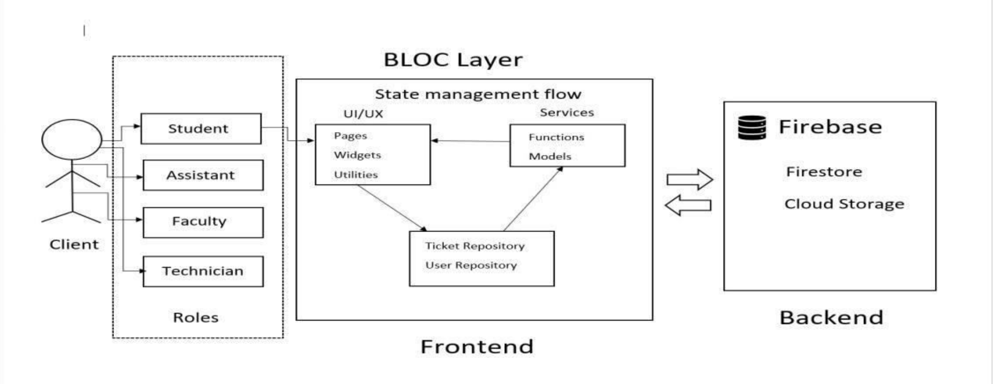


Figure 1: Input and output of project

IV. RESULTS AND DISCUSSION

A. System Performance

The Computer Lab Maintenance App was tested for functionality, user experience, and overall system performance. The results indicate that the system efficiently processes maintenance requests, schedules preventive tasks, and provides real-time updates with minimal delay.

Key Performance Metrics:

Average Response Time: 1.5 seconds for issue logging and updates. Maintenance Request Processing Time: Less than 2

seconds per status update. Notification Delivery Speed: 98% of alerts delivered in real-time.
Database Query Execution Time: Under 300ms for retrieving maintenance logs.

B. User Experience and Usability

User feedback was collected from lab administrators, IT staff, and students who regularly use the system. The system received an overall usability rating of 4.6/5, highlighting:
Ease of Issue Reporting: Simple and user-friendly form submission for logging maintenance requests. Real-Time Updates & Notifications: Users appreciated timely alerts for issue status changes and scheduled maintenance.
Dashboard Navigation: IT staff and administrators found the dashboard intuitive for tracking lab equipment health. User Satisfaction Survey Results

C. Comparative Analysis

The Computer Lab Maintenance App was compared with traditional manual logging methods and existing asset management software. Results show that the system significantly improves efficiency and reduces downtime. Key Findings. The app reduces issue resolution time by 50% compared to manual logging.
The automated notification system ensures IT staff can respond 30% faster than traditional methods. The dashboard analytics feature allows administrators to track recurring issues and schedule preventive maintenance effectively.

D. Limitations and Future Scope

Although the system performs well, some limitations were identified:
Limitations:

- Internet Dependency: The app requires an internet connection for real-time updates.
- Limited AI Automation: The current system does not include predictive maintenance analytics.
- Platform-Specific Constraints: The mobile version performs optimally, but additional optimization is needed for web compatibility.

E. Future Enhancements:

AI-Based Predictive Maintenance: Implementing machine learning algorithms to predict potential system failures before they occur. Offline Functionality: Introducing an offline mode where users can log issues and sync data once connected to the internet.
Multi-User Collaboration: Allowing multiple IT staff members to manage maintenance requests simultaneously with role-based access.
IoT Integration: Connecting lab equipment to the app for real-time hardware monitoring and automatic issue detection.

V. CONCLUSION

The Computer Lab Maintenance App is designed to streamline and automate the maintenance process of computer labs in educational institutions. By integrating issue reporting, preventive maintenance scheduling, inventory management, and role-based access, the system ensures efficient handling of lab-related problems. Faculty members can easily report issues, lab technicians can track and resolve them, and the HOD can oversee operations and approve necessary repairs or replacements. By reducing manual efforts, minimizing downtime, and enhancing transparency, the app improves the overall efficiency of lab maintenance. It ensures that computer labs remain in optimal condition, providing students and faculty with a well-functioning and well-maintained learning environment. This system not only enhances operational efficiency but also helps institutions make informed decisions about future maintenance and upgrades

ACKNOWLEDGEMENTS

We express our sincere gratitude to all those who contributed to the successful development of the Computer Lab Maintenance App.
Firstly, we extend our heartfelt thanks to our institution and faculty members for their valuable guidance, technical insights, and continuous support throughout the project. Their expertise and constructive feedback helped shape the system into an efficient and user-friendly solution.
We would also like to acknowledge the IT staff and lab administrators who provided crucial inputs regarding real-

world maintenance challenges, allowing us to refine the app’s functionality to better meet practical needs.

A special thanks to our team members for their dedication, collaboration, and hard work in designing, developing, and testing the application. Their commitment played a vital role in achieving the project objectives.

Lastly, we appreciate the support of our friends and family, whose encouragement and motivation helped us stay focused and determined.

This project would not have been possible without the collective efforts of everyone involved, and we sincerely appreciate all contributions that made it a success.

REFERENCES

[1]J. Samuel, Peterson @ Robinson (2003) “School Management system ScMS” School Managementm,System / Peterson @ Robinson J. Samuel. Undergraduate’s thesis–Faculty of Computer Science & Information Technology, University of Malaya, 2002/2003

[2] Mouratis, Kyriakos, Georgios Stivaktakis, and Michael Sfakiotakis. "Remote access laboratory setup for physical computing courses." 2022 31st Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE). IEEE, 2022.

[3] Diwakar, S., Kumar, D., Radhamani, R., Sasidhara Kurup, H., Nizar, N., Achuthan, K., Nedungadi, P., Raman, R., & Nair, B.(2016). Complementing Education via Virtual Labs: Implementation and Deployment of Remote Laboratories and Usage Analysis in South Indian Villages. International Journal of Online and Biomedical Engineering (iJOE), 12(03), pp. 8– 15. https://doi.org/10.3991/ijoe.v12i03.5391

[4] Gercek, G. and Saleem, N., 2008. Transforming Traditional Labs into Virtual Computing Labs for Distance Education. International Journal of Online Engineering, 4(1).

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com