SmartCampus: A Web-Based Issue Reporting and Maintenance Management System for Campus Environments.

Vandhana[1], Varshini[2], Shree Dhaksha[3]

[1] Adhiyamaan College of Engineering, Computer Science and Engineering, Hosur, Tamilnadu, India

[2] Adhiyamaan College of Engineering, Computer Science and Engineering, Hosur, Tamilnadu, India

[3] Adhiyamaan College of Engineering, Computer Science and Engineering, Hosur, Tamilnadu, India

rockyvandhana17@gmail.com[1], varshiniparthiban21@gmail.com[2], shreedhakshadjeee@gmail.com[3]

Abstract

Efficient campus maintenance is essential for providing a safe, accessible, and well-organized educational environment. However, campus issues such as damaged infrastructure, electrical faults, water leakage, sanitation problems, and malfunctioning facilities may be reported through informal communication channels, making it difficult to track their progress and ensure timely resolution. This paper presents SmartCampus, a web-based Campus Issue Reporting and Maintenance Management System designed to centralize the reporting, assignment, monitoring, and resolution of campus maintenance issues. The proposed system provides role-based access for students, faculty members, administrators, and maintenance personnel.Users can submit complaints with relevant details, categorize issues, and monitor their progress, while authorized administrators can review reports, assign maintenance tasks, and track resolution activities. Maintenance personnel can access assigned tasks and update their status as work progresses.

The system is designed using React, TypeScript, Node.js, Express.js, MongoDB, and Mongoose, with authentication and role-based authorization supporting controlled access to system functionalities. The architecture also supports evidence uploads and status notifications where configured. By maintaining a centralized record of reported issues and their resolution history, SmartCampus aims to improve communication, accountability, and visibility in campus maintenance operations. The paper describes the system architecture, development methodology, principal modules, security considerations, and testing approach. Quantitative performance improvements are not claimed without measured experimental results.

Keywords: SmartCampus, Campus Maintenance, Issue Reporting, Complaint Management, Web Application, MongoDB.

1. Introduction

Educational institutions require an effective system to manage maintenance issues and respond to problems reported by students and faculty. Issues such as damaged classroom equipment, electrical faults, water leakage, sanitation problems, and infrastructure maintenance can affect the daily functioning of a campus. When complaints are handled through informal communication, phone calls, or manual records, tracking their progress and coordinating the responsible personnel can become difficult.

SmartCampus – Issue Reporting and Maintenance Management System is a web-based application developed to organize campus complaints through a centralized platform. It enables users to submit issues with descriptions, categories, location details, and supporting images. The system also provides complaint-tracking capabilities so users can monitor the status of submitted issues.

The application supports administrative workflows for reviewing complaints, assigning maintenance responsibilities, and monitoring progress. It also includes an emergency-reporting concept intended to support the reporting of serious campus incidents to authorized personnel. The availability of external notification services depends on the corresponding integration and configuration.SmartCampus is developed using a modern web technology stack, with React and TypeScript for the user interface, Node.js and Express for backend operations, and MongoDB for data storage. Socket.IO supports real-time communication where implemented. The modular design separates the user interface, application logic, and data-management components.

The main objective of this project is to provide a structured approach to campus issue reporting and maintenance management. By organizing complaint records and their progress in one application, SmartCampus aims to improve visibility, accountability, and coordination between campus users, administrators, and maintenance personnel.

2. Literature Review

2.1 Traditional Complaint Management in Educational Institutions

Traditional campus maintenance management commonly relies on direct communication, telephone calls, written complaints, or manual registers. Although these approaches allow users to report problems, maintaining a centralized record of complaints and monitoring their progress can be challenging. Information may be distributed across different departments, making it difficult to identify pending issues and determine whether corrective action has been completed.

2.2 Digital Complaint Management Systems

Web-based complaint management systems provide structured mechanisms for submitting, recording, reviewing, and tracking complaints. Previous studies on campus and student complaint management have explored the use of centralized applications to organize complaint information and improve administrative monitoring [1], [2]. Research on maintenance management also highlights the importance of organizing maintenance-related information and improving complaint-handling processes [9], [10]. However, the effectiveness of a digital system depends on appropriate access controls, accurate records, and consistent status updates.

2.3 Real-Time Communication and Issue Tracking

Real-time communication technologies can help applications update information without requiring users to repeatedly refresh the interface. In a campus maintenance system, these capabilities can support the visibility of complaint-status changes and administrative updates. Socket.IO is used in SmartCampus to support real-time communication where configured and implemented. Reliable status management remains important for ensuring that users receive meaningful and consistent information.

2.4 Need for an Integrated Campus Maintenance Platform

A campus maintenance application must support more than complaint submission. It should also organize issue categories, record relevant locations, allow supporting images where required, and provide mechanisms for reviewing and tracking reported problems. Administrative functions are important for coordinating maintenance responsibilities and monitoring the progress of unresolved complaints.

3. EXISTING SYSTEM AND ITS LIMITATIONS

3.1 Traditional Complaint Management

Traditional complaint management relies on manual registers, phone calls, or direct communication. These methods can make complaint tracking difficult and may result in incomplete records, delayed updates, and limited transparency.

3.2 Digital Complaint Management Systems

Web-based complaint management systems provide a structured way to record issues and monitor their status. Centralized records help users follow submitted complaints and enable administrators to review pending maintenance activities.

4. TECHNOLOGY INTEGRATION

4.1 Web Application Technologies

SmartCampus uses React and TypeScript for its frontend and Node.js with Express for backend operations. MongoDB provides persistent data storage, while Vite supports frontend development and building.

4.2 Real-Time Communication

Socket.IO supports real-time communication where implemented. This capability can help reflect relevant complaint updates in the application and improve visibility into the maintenance workflow.

5. DIGITAL ISSUE TRACKING

5.1 Complaint Tracking

The complaint-tracking module allows users to monitor the recorded status of submitted issues. It helps organize complaint information and provides visibility into the progress of maintenance activities.

5.2 Administrative Management

The administrative module supports complaint review and maintenance coordination. Centralized records can help authorized personnel identify pending issues and monitor updates.

6. CAMPUS MAINTENANCE WORKFLOW

6.1 Complaint Submission and Tracking

Users can submit campus maintenance complaints by providing the issue description, category, location, and supporting images where applicable. The system stores complaint information and allows users to monitor the recorded status of their submissions.

6.2 Administrative Review and Maintenance Coordination

Administrators can review submitted complaints and monitor their progress through the available management functions. Centralized complaint records support the identification of pending issues and help coordinate maintenance activities among the responsible personnel.

3. Problem Statement

Campus maintenance issues such as electrical faults, water leakage, damaged equipment, and sanitation problems require an organized reporting process. Traditional complaint management methods can make it difficult to maintain records, track complaint status, and coordinate maintenance personnel.

SmartCampus addresses these challenges through a centralized web application that supports complaint submission, issue tracking, and administrative management. The system aims to improve the organization of campus maintenance records and communication between users and responsible authorities.

4. Methodology

1. Requirement Analysis

1.1 Functional Requirements

The system allows users to submit complaints, provide issue categories and locations, upload supporting images, and track complaint status. Administrators can review complaints and manage maintenance activities.

1.2 Non-Functional Requirements

- Usability: Provide a simple and user-friendly interface.
- Maintainability: Organize the application into manageable components.
- Data Integrity: Maintain consistent complaint records.
- Security: Control access to protected functions using implemented access mechanisms.

2. System Design

2.1 Technology Stack Selection

- Frontend: React and TypeScript provide the user interface.
- Backend: Node.js and Express process application requests.
- Database: MongoDB stores application records using Mongoose.
- Build Tool: Vite supports frontend development.
- Real-Time Communication: Socket.IO supports real-time communication where implemented.

3. Development Process

3.1 Frontend Development

The frontend is developed using React and TypeScript to provide interfaces for complaint submission, issue tracking, and administrative activities.

3.2 Backend Development

Node.js and Express handle application requests, process complaint information, and coordinate database operations.

3.3 Database Integration

MongoDB maintains complaint and application records, while Mongoose supports database models and interactions.

4. Testing and Validation

4.1 Functional Testing

Functional testing checks complaint submission, user access, issue tracking, and administrative operations.

4.2 Integration Testing

Integration testing verifies communication between the frontend, backend, and database.

4.3 Workflow Testing

Workflow testing examines complaint submission, administrative review, and status updates through the available application functions.

5. Proposed System

SmartCampus provides a centralized platform for reporting campus issues and managing maintenance complaints. It combines complaint registration, tracking, and administrative functions within a single web application.

5.1 System Overview

Users can submit complaints with descriptions, categories, location details, and supporting images. The system stores complaint information and provides access to the available tracking and management functions.

5.2 Key Features

Complaint Submission: Users can report campus maintenance issues by entering the required details and attaching supporting images where applicable.

Issue Tracking: Users can monitor the recorded status of their submitted complaints.

Administrative Management: Authorized personnel can review complaints and coordinate maintenance activities.

Emergency Reporting: The system includes an emergency-reporting workflow intended for serious campus incidents. External notification delivery depends on successful service integration and testing.

6. Technical Architecture

SmartCampus uses a web-based architecture that connects the frontend, backend, and database to manage campus maintenance complaints.

React and TypeScript: Used to develop the frontend interface and interactive components.

Node.js and Express: Handle backend requests and complaint-management operations. **MongoDB and**

Mongoose: Store and manage application data and complaint records. **Socket.IO:** Supports real-time

communication where implemented.

Multer: Handles supported image and file uploads.

7. Security and Privacy Considerations

To protect user information and maintain application security, SmartCampus considers the following aspects:

Data Protection: Complaint details and uploaded images should be handled securely to prevent unauthorized access.

Access Control: Access to administrative functions and complaint records should be restricted according to user roles and implemented permissions.

Input Validation: User-submitted information should be validated to reduce invalid data and prevent malicious input.

File Upload Security: Uploaded images should be checked against permitted file types and size limits before processing.

Database Security: Database access credentials should be protected using environment variables, and database permissions should be configured appropriately.

Privacy: Personal information contained in complaints should only be accessible to authorized users for legitimate campus maintenance purposes.

8. Result

The implementation of the SmartCampus Issue Reporting and Maintenance Management System provides a centralized approach to managing campus maintenance complaints. The application enables users to report issues by entering descriptions, selecting appropriate categories, providing location details, and uploading supporting images. By organizing complaint information within a single web-based platform, the system provides a structured alternative to scattered manual reporting methods. Its complaint-tracking functionality helps users monitor the recorded status of their submissions, while administrative features support the review and management of reported issues. The application is designed to improve the accessibility and organization of maintenance information and to support coordination between campus users and responsible personnel.

The system also supports administrative maintenance workflows by providing access to complaint records and their recorded progress. Administrators can review submitted issues, identify complaints requiring attention, and coordinate maintenance activities through the available management functions. Maintaining centralized records can help reduce confusion caused by information being distributed across multiple communication channels. The availability of complaint details and status information also provides a foundation for reviewing recurring maintenance concerns and identifying areas that may require further attention. However, the actual improvement in response time and complaint-resolution efficiency depends on the accuracy of status updates, the actions of maintenance personnel, and the consistent use of the system.

Furthermore, SmartCampus demonstrates the integration of modern web technologies in developing a campus maintenance management application. React and TypeScript are used to implement the frontend, while Node.js and Express handle backend operations. MongoDB and Mongoose support persistent storage and management of application records, and Socket.IO supports real-time communication where implemented. The application's modular structure separates interface components, backend processing, and database operations, making the system easier to maintain and extend. Functional testing across different user roles and complaint-management workflows provides a basis for evaluating the behaviour of implemented features. Additional testing with documented results and real-world usage data would help establish the system's reliability, usability, and practical effectiveness in campus environments.

9. Conclusion

SmartCampus is a web-based Issue Reporting and Maintenance Management System developed to organize campus maintenance complaints through a centralized platform. The application supports structured complaint submission, issue tracking, and administrative management, providing a common interface for recording and reviewing reported problems. By maintaining complaint information in a centralized database, the system helps organize issue records and makes their recorded status accessible through the available application functions.The system integrates React and TypeScript for the frontend, Node.js and Express for backend processing, and MongoDB with Mongoose for data storage. Socket.IO supports real-time communication where implemented. The modular architecture separates user-interface components, application logic, and data management, providing a foundation for maintaining and extending the system. Functional testing across different user roles and workflows can help verify the behaviour of the implemented features.

In conclusion, SmartCampus provides a structured technological approach to campus issue reporting and maintenance management. Future enhancements may include verified external notification services, advanced complaint analytics, detailed maintenance reports, and mobile accessibility. Further evaluation using documented test results and real-world usage data is required to measure the system's effectiveness and its impact on campus maintenance operations.

10. References

[1] F. A. Alifat, A. B. Olorunlomerue, A. O. Oluwatobi, J. I. Ugwu, and T. R. Amosa, “Web-based Campus Complaint Management System (WCCMS),” International Journal of Computer Trends and Technology, vol. 70, no. 10, pp. 28–36, 2022, doi: 10.14445/22312803/IJCTT-V70I10P105.

[2] A. Mohammed, Z. Solomon, and D. M. Nurideen, “A Web-Based Student’s Complaints Management System,” International Journal of Technology, vol. 11, no. 2, pp. 35–43, 2021, doi: 10.52711/2231-3915.2021.00005.

[3] E. Z. Muzones and J. R. Pomperada, “A Complaint Management System for Enhanced Efficiency at West Visayas State University Himamaylan City Campus,” Philippine Social Science Journal, vol. 8, no. 2, 2025, doi: 10.52006/main.v8i2.1343.

[4] O. M. T.-Manuel and J. N. Amadi, “Design and Implementation of a Web-Based Student Complaint and Response Management System for Enhancing Administrative Responsiveness in Tertiary Institutions,” American Journal of Science and Learning for Development, vol. 3, no. 8, pp. 100–111, 2024, doi: 10.51699/ajsld.v3i8.1116.

[5] I. N. Kabelele, K. K. Hhalla, Y. Mwamasangula, J. R. Mkakilwa, I. R. Mlangali, I. F. Mwandemani, and H. S. Athumani, “Design and Implementation of a Web-Based Student Complaint Management System for Universities,” International Journal of Engineering and Information Systems, vol. 10, no. 4, pp. 106–111, 2026.

[6] M. A. Insan, F. Octanouva, N. Vitriati, S. W. Pratama, and A. S. Ardiansyah, “Development of a Web-Based Campus Facility Complaint Information System Prototype Using Rapid Application Development,” Technema: Journal of Intelligent Engineering and Computing, 2026.

[7] W. D. Pratama and S. Widiono, “Development of a Web-Based Student Complaint System with Activity Dashboard,” Journal of Scientific Research, Education, and Technology, vol. 4, no. 4, 2025, doi: 10.58526/jsret.v4i4.967.

[8] "Automation Tool for Grievances Using Artificial Intelligence," Procedia Computer Science, vol. 283, pp. 2770–2779, 2026, doi: 10.1016/j.procs.2026.06.346.

[9] Z. Ismail, "Analysing the Gaps in the Conventional E-Complaint Method for Maintenance Management at Malaysian Polytechnics," Social Responsibility Journal, 2019, doi: 10.1108/SRJ-12-2019-0404.

[10] "Maintenance Management System (MMS) to Support Facilities Management at Malaysian Polytechnic," Smart and Sustainable Built Environment, 2017, doi: 10.1108/SASBE-08-2016-0022.

[11] W. H. DeLone and E. R. McLean, "The DeLone and McLean Model of Information Systems Success: A Ten-Year Update," Journal of Management Information Systems, vol. 19, no. 4, pp. 9–30, 2003, doi: 10.1080/07421222.2003.11045748.

[12] International Organization for Standardization, ISO 41001:2018—Facility Management: Management Systems—Requirements with Guidance for Use. Geneva, Switzerland: ISO, 2018.

[13] International Organization for Standardization, ISO 9241-210:2019—Ergonomics of Human-System Interaction: Human-Centred Design for Interactive Systems. Geneva, Switzerland: ISO, 2019.

[14] International Organization for Standardization and International Electrotechnical Commission, ISO/IEC 25010:2023—Systems and Software Engineering: Systems and Software Quality Requirements and Evaluation (SQuaRE)—Product Quality Model. Geneva, Switzerland: ISO, 2023.

[15] World Wide Web Consortium (W3C), Web Content Accessibility Guidelines (WCAG) 2.2. W3C Recommendation, 2023.

[16] Open Worldwide Application Security Project (OWASP), Application Security Verification Standard (ASVS), Version 5.0.0. 2025.

[17] J. Nielsen, Usability Engineering. San Francisco, CA, USA: Morgan Kaufmann, 1993.

[18] J. Nielsen, Designing Web Usability: The Practice of Simplicity. Indianapolis, IN, USA: New Riders Publishing, 1999.

[19] R. S. Pressman and B. R. Maxim, Software Engineering: A Practitioner's Approach, 9th ed. New York, NY, USA: McGraw-Hill, 2019.

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