TrackNFix : Institution – Centric Issue Management Platform

Aishwarya Laxmi Y[1], Nithyasri K[2], Richytha H S[3]

Under the guidance of **Ms.M.Buvana** (Asst. Professor)

Department of Information Technology,Sri Shakthi Institute of Engineering and Technology,Coimbatore

Abstract: The Institution – Centric Issue Management Platform is a web-based application designed to manage and streamline technical issue tracking within institutions such as schools, labs, or offices. The system enables different user roles—Students, Approvers, Technicians, and Admins—to interact through a centralized platform to report, review, and resolve issues efficiently. Students can raise new issues, view their status, and verify resolved ones. Approvers review and approve submitted issues and monitor technician performance. Technicians receive assigned issues, update progress, and mark them as completed. Admins manage users, labs, equipment, inventory, and issue types through a dedicated backend interface. Developed using Python with an SQLite database, the application runs over a local Wi-Fi network and is accessible via any modern web browser, eliminating the need for internet connectivity. It offers a lightweight, cost-effective solution suitable for small to medium-sized environments requiring efficient support operations.
This system improves communication, reduces response time, and provides a clear workflow from issue reporting to resolution. Its modular design also supports future enhancements like mobile access, cloud deployment, and advanced reporting features.

I.Introduction

In today's institutional and organizational environments, the smooth functioning of operations depends heavily on timely support for technical and infrastructure-related issues. Whether it's faulty lab equipment, IT support, or maintenance tasks, the absence of an efficient tracking and resolution system can lead to delayed workflows, reduced productivity, and user dissatisfaction. Many institutions still rely on verbal communication or paper-based reporting methods, which are not only inefficient but also lack transparency and accountability.

The Institution – Centric Issue Management Platform is designed to address these gaps by offering a centralized, role-based digital platform that allows different stakeholders to manage support-related tasks in an organized manner. By providing specific functionalities for Students, Approvers, Technicians, and Admins, the system ensures a structured flow of information and responsibilities. Each user interacts with the system based on their role, reducing confusion and improving the turnaround time for issue resolution.

This project is developed using Python for backend development and SQLite as the database, offering a lightweight and easy-to-install solution. The system operates within a local area network, making it highly suitable for schools, colleges, labs, or organizations that prefer internal deployment without relying on the internet. Users simply need to connect to the same Wi-Fi and access the system through any web browser, ensuring accessibility with minimal technical setup.

What makes this system especially practical is its focus on core usability and modular design. Students can raise and verify issues, Approvers can validate and assign tasks, Technicians can update issue progress, and Admins can manage all backend data such as users, labs, inventory, and equipment. This clear division of roles not only streamlines the workflow but also maintains a record of each step in the resolution process for future reference.

Furthermore, the system lays a strong foundation for future development. With its scalable architecture, features like cloud deployment, mobile application integration, real-time notifications, and analytics dashboards can be added to extend functionality. In essence, this project provides a robust and flexible solution to modernize support systems within controlled environments, improving efficiency, transparency, and user satisfaction.

II.Primary Objective

The primary objective of the Institution – Centric Issue Management Platform is to develop a centralized, web-based platform that streamlines the process of reporting, tracking, and resolving technical and operational issues within an institution or organization. The system aims to:

1.Facilitate smooth communication between Students, Approvers, Technicians, and Admins through a clearly defined role-based access system.

2.Automate the issue resolution workflow, from issue creation to final verification, ensuring transparency and accountability at each stage.

3.Reduce manual dependency and errors by replacing traditional support systems such as paper logs or informal reporting with a structured digital solution.

4.Enable easy access to the system via modern web browsers over a local Wi-Fi network, eliminating the need for internet connectivity or complex infrastructure.

5.Provide a scalable and lightweight platform that can be easily deployed, maintained, and enhanced with future features like mobile access, cloud deployment.

III.LITERATURE SURVEY

A literature survey is essential to understand existing solutions, their limitations, and how the proposed Field Support Management System contributes to the improvement of issue tracking and technical support processes within organizations and institutions.

1. Existing Manual Support Systems

Many institutions still rely on manual methods for handling technical support—such as phone calls, emails, or logbooks. While these methods are simple, they are prone to miscommunication, lack of traceability, and delays in resolving issues. Manual systems also fail to maintain historical records, making it difficult to evaluate technician performance or analyze issue trends.

2. Enterprise-Level Helpdesk Solutions

Commercial helpdesk software like Jira Service Desk, Zendesk, and Freshdesk offer robust features such as ticketing systems, SLA management, and automation. However, these platforms are often cloud-based, require ongoing subscription costs, and include complex features unnecessary for small institutions. Their enterprise-level architecture may be excessive for educational or single-location setups, making them less suitable for internal, local-use-only environments.

3. Role-Based Access Control (RBAC)

The concept of RBAC has been extensively studied and implemented in many systems to manage permissions based on user roles. Research shows that RBAC improves security and accountability by ensuring users only access information and features relevant to their responsibilities. Implementing RBAC in a local issue-tracking system provides a secure and structured workflow, as seen in academic works focusing on campus management and IT support systems.

4. Local Network Applications

Several studies have emphasized the advantages of offline-first applications in environments with limited or no internet access. SQLite, a serverless and lightweight database, has been proven suitable for such applications. Web-based platforms running over LAN (Local Area Network) provide accessibility without the need for cloud infrastructure, making them ideal for schools, labs, and small offices.

5. Need for Institutional Issue Tracking Systems

Research conducted in academic institutions has highlighted the need for digitized issue tracking systems tailored for internal use. Systems developed in universities often focus on lab equipment tracking, IT maintenance, or general service requests. These studies underline the importance of integrating user roles, tracking history, and generating reports to maintain operational efficiency.

IV. PROPOSED SYSTEM

The **Institution – Centric Issue Management Platform** is proposed as a centralized, web-based application designed to overcome the limitations of existing manual or semi-digital support systems used in institutions. The system introduces a structured, role-based workflow that allows Students, Approvers, Technicians, and Admins to perform their tasks efficiently and transparently within a single platform.

The proposed system allows **Students** to raise new issues, view their status, and verify whether the problem has been resolved. Once an issue is raised, it is reviewed by an **Approver**, who either approves or rejects the request and assigns it to a **Technician**. The Technician then updates the progress, marks the issue as completed, and logs important events throughout the resolution process. An **Admin** user manages system-wide data such as users, labs, equipment, inventory, and issue categories.

This system is built using **Python** as the backend language, **SQLite** as the database, and runs on a **local Wi-Fi network**, allowing users to access it via modern web browsers without requiring internet connectivity. The simple deployment model makes it ideal for schools, labs, and small organizations.

The proposed system provides several benefits:

- A **centralized platform** for issue tracking and resolution
- **Clear role-based access control** to avoid confusion and overlap
- **Real-time tracking** of issue status and technician performance
- **Lightweight and easy deployment** within local environments
- A **scalable design** that allows for future enhancements like mobile access, cloud deployment, or advanced analytics

Overall, the proposed system modernizes the support process, reduces delays, improves accountability, and ensures a more efficient operational workflow across the institution.

V. SYSTEM ARCHITECTURE

The Institution – Centric Issue Management Platform is designed with a simple, modular three-tier architecture that ensures maintainability, scalability, and efficient role-based communication. The system operates on a local area network (LAN), making it ideal for institutions with limited infrastructure. Below is a breakdown of the architecture:

1. Presentation Layer (Frontend/User Interface)

This is the interface through which users interact with the system. It is accessed via a web browser and built using standard web technologies like HTML, CSS, and optionally JavaScript for better interactivity.

•Accessible by all roles: Student, Approver, Technician, and Admin.

•Role-based dashboards and navigation.

•Forms for submitting issues, viewing status, and performing respective tasks.

•Responsive design for use across desktops and laptops connected to the same network.

2. Application Layer (Backend/Business Logic)

This layer contains the core logic of the system and is developed using Python (likely with Flask or Django framework). It handles:

•User authentication and session management.

•Role-based access control and redirection.

•Issue handling logic (submission, approval, assignment, status updates).

•Validation of inputs and error handling.

•Communication between the frontend and database.

3. Data Layer (Database)

The data layer uses SQLite, a lightweight, file-based relational database system suitable for single-machine deployments. It stores all essential data, including:

•User details and roles

•Issue records and logs

•Inventory, lab, and equipment details

•Event histories related to each issue

•System configuration data

The database is accessed securely through the backend using standard SQL queries or ORM (Object-Relational Mapping) methods.

4.Network Setup

•The entire system is hosted on a single computer (server) connected to a local Wi-Fi network.

•All client machines (users) access the system using a web browser by entering the IP address of the host server.

•No external internet or cloud services are required, keeping the system secure and isolated.

5.Security and Access Control

•Role-based login restricts access to functionalities depending on user type.

•Password change features and session controls are implemented.

•Only authorized users can perform sensitive operations like editing users or modifying equipment data.

VI.RESULT OBTAINED

The development and deployment of the **Institution – Centric Issue Management Platform** successfully met the core objectives defined at the start of the project. The system was tested within a local Wi-Fi network environment and performed efficiently across all intended user roles—Student, Approver, Technician, and Admin. Below are the key results obtained:

1.Successful Role-Based Access Implementation
Each user was able to log in and access their designated dashboard. Role-specific functionalities—like issue submission for students, approval and performance monitoring for approvers, and progress tracking for technicians—worked seamlessly without cross-role interference.

2.Smooth Issue Workflow Execution
The system accurately handled the complete issue lifecycle: from submission and approval to technician assignment, resolution, and final verification by the student. Status updates and event logs were displayed correctly at each stage.

3.Efficient Local Network Access
Users connected to the same local network could access the system in real time using a web browser. This validated the system's suitability for offline or local institutional environments without relying on internet access.

4.Data Management and Admin Control
Admin users were able to manage system data effectively, including adding/editing users, labs, inventory, and equipment. The ability to update issue types and reset passwords ensured the system could be maintained easily.

5.User-Friendly Interface and Performance
The frontend interface was found to be responsive. Test users

reported navigating between sections and completing tasks was intuitive and did not require any prior technical training.

VII.Conclusion

The **Institution – Centric Issue Management Platform** successfully addresses the need for a structured, role-based platform to manage technical and operational issues within an institution. Through the development and implementation of this system, the manual and unorganized methods of handling support requests have been replaced with a streamlined digital workflow that ensures better communication, accountability, and efficiency.

By integrating key functionalities such as issue submission, approval, technician updates, and admin-level data management, the system allows all stakeholders—Students, Approvers, Technicians, and Admins—to perform their roles effectively within a centralized platform. Its deployment over a local Wi-Fi network without the need for internet connectivity makes it especially suitable for small to medium-sized institutions and organizations.

The project demonstrated successful execution in terms of functionality, performance, and usability. It has laid a strong foundation for future enhancements, such as cloud integration, mobile app development, real-time notifications, and advanced reporting features. Overall, the Field Support Management System is a reliable and scalable solution that significantly improves internal support operations and contributes to smoother institutional workflows.

VIII.References

[1] Sandhu, R.S., Coyne, E.J., Feinstein, H.L., & Youman, C.E. (1996). *Role-Based Access Control Models*. IEEE Computer, 29(2), 38–47.
— This paper provides the foundational concept of role-based access control (RBAC), which is a core principle in the system's design.

[2] Silberschatz, A., Korth, H.F., & Sudarshan, S. (2020). *Database System Concepts* (7th ed.). McGraw-Hill Education.
— This book was referred to for designing and understanding the relational database schema used in the system (SQLite).

[3] Flanagan, D. (2006). *JavaScript: The Definitive Guide* (5th ed.). O'Reilly Media.
— Basic JavaScript logic was referenced for enhancing user interactions in the web interface.

[4] Python Software Foundation. (2023). *Python 3 Documentation*. Available at: https://docs.python.org
— Official reference for Python syntax, libraries, and backend integration used in the project.

[5] Palanisamy, R., & ThamaraiSelvi, M. (2019). *A Framework for an Academic Help Desk System in Higher Education Institutions*. International Journal of Computer Science and Mobile Computing (IJCSMC), 8(6)