

## Complaint Escalation Tracking Portal

<sup>1</sup> Bhargavi Mohana Maddu, <sup>2</sup> Evuri Hari Krishna, <sup>3</sup> Chadalavada Rajkumar, <sup>4</sup> Mrs.M. Anusha

<sup>1,2,3</sup>U. G Student, Dept COMPUTER SCIENCE AND ENGINEERING, St. Ann's College Of Engineering and Technology, Nayunipalli (V), Vetapalem (M), Chirala, Bapatla Dist, Andhra Pradesh – 523187, India

<sup>4</sup>Assistant professor, COMPUTER SCIENCE AND ENGINEERING, St. Ann's College Of Engineering and Technology, Nayunipalli (V), Vetapalem (M), Chirala, Bapatla Dist, Andhra Pradesh – 523187, India

### ABSTRACT

A web-based program called the Complaint Management System makes managing customer concerns easier. It provides a safe login and registration process that enables users to file complaints via an easy-to-use interface. MongoDB Atlas, a scalable cloud database, is where complaints are kept, guaranteeing dependable data access. The solution has an admin dashboard for effective case management and a user dashboard for real-time tracking. Unresolved complaints that exceed time limitations are flagged by an automatic escalation process. By adjusting to different organizational circumstances and setting the stage for future improvements like mobile application support and AI-based complaint classification, the system improves openness, accountability, and efficiency in complaint management.

### KEY WORDS:

Complaint Management, Escalation Management, Python Flask, MongoDB, Automation, Accountability, Service Quality.

### INTRODUCTION

A new platform called the Complaint Escalation System was created to help modern corporations handle complaints better. It is a transparent, centralized, and automated platform that enables users to file complaints, handle them instantly, and implement preset escalation guidelines. A complaint is automatically sent to the next higher authority if it is not handled within a predetermined time frame, guaranteeing that no problem goes unresolved. By attaching pertinent information to each complaint, including the complainant's details, issue description, designated handler, status, and timestamps, the system preserves an unambiguous chain of accountability. This methodical approach not only expedites the resolution process but also offers beneficial insights into persistent issues, allowing management to pinpoint the underlying causes and put preventative measures in place.

The system, which was created with Python web development frameworks, provides scalability, flexibility, and the capacity to integrate with

other organizational tools. All complaint records

are safely stored in its database, which is run by a reliable backend like MongoDB and facilitates effective search and retrieval functions. The Complaint Escalation System acts as a link between consumers and service providers by expediting complaint processing and implementing automated escalation. This guarantees improved communication, quicker problem-solving, and a notable increase in service quality. Over time, this strategy may increase organizational performance, build a more responsive service culture, and cultivate trust.

### LITERATURE REVIEW

Recent years I explored some related papers to Complaint Escalation System. In this I got some limitations. To check papers, I studied recent papers. In that first paper is A Framework of Customer Complaint Handling System will be proposed by Amy

J. C. Trappey in this paper they do not explore the escalation system where the customer didn't satisfy also taken the second paper is Complaint Management- Review And Additional Insights will be proposed by Amit Kumar in this paper they do not used the security based verification and storage of the complaints and I taken Smart Complaint Management System will be proposed by Devika Radhakrishnan in this paper they don't provide the user interface to interact the effectively and

.more difficult to handle and I taken Public Online Complaint Registration and Management System will be proposed by Divya Hiremath but in this the escalation system will be perform more effectively

.This limitations will be solved by proposed method.

### RELATED WORK

For both service quality and corporate communication, complaint handling is essential. Conventional systems frequently rely on email correspondence or manual processing, which results in low customer satisfaction, delayed replies, and a lack of transparency. Web-based solutions that automate complaint management

through defined processes and real-time monitoring have been proposed by researchers and developers. These solutions, however, frequently have limited scalability for larger user groups or rely on predefined escalation procedures. Cloud computing and web technology advancements have made it feasible to create complaint management systems that are adaptable, scalable, and lightweight. Frameworks such as Python Flask may simplify systems, provide robust backend support for RESTful APIs, and provide extensions for automatic escalation and role-based access. The NoSQL document-oriented design, high availability, and scalability of MongoDB and its cloud variant, MongoDB Atlas, make them well-liked data management systems. Real-time updates, fault tolerance, and data retrieval performance are all enhanced when Flask and MongoDB are combined.

#### EXISTING METHOD

The majority of complaints are handled and resolved manually or with some automation in the complaint management systems in place today. Usually, users use email, paper forms, or simple online platforms to register their complaints. These complaints are reviewed by administrative staff, who then manually forward them to the

appropriate people or departments. Manual entry is required to update the status of complaints on a regular basis, which causes delays, reduces transparency, and makes it more difficult for users and authorities to communicate effectively. Furthermore, a lot of the systems in use today lack an automated escalation feature; unresolved problems are left dormant until they are manually evaluated, which irritates users and provides subpart service.

#### PROPOSED METHOD

The proposed Complaint Escalation System uses MongoDB Atlas for cloud storage and Python Flask for backend development to address the shortcomings of existing complaint management systems. The proposed method ensures faster resolution and higher customer satisfaction by automating the complaint processing and escalation process, which sets it apart from traditional solutions. Through a web interface, users may submit complaints, and the system will automatically allocate them to the appropriate staff members based on user roles and pre-established regulations. A lightweight, flexible backend that supports role-based access control, secure authentication, and RESTful APIs may be constructed with Flask.

#### ARCHITECTURE

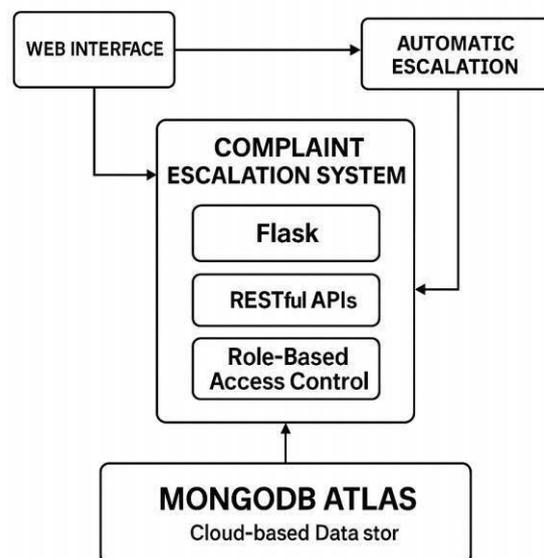


Fig1: Block Diagram

**METHODOLOGYDESCRIPTION**

**Complaint Escalation System:** The Complaint Escalation System is the central logic hub built on the Flask framework for managing integrity and flow of all complaint data. It exposes RESTful APIs which serve as the formal communication contracts for all other components.

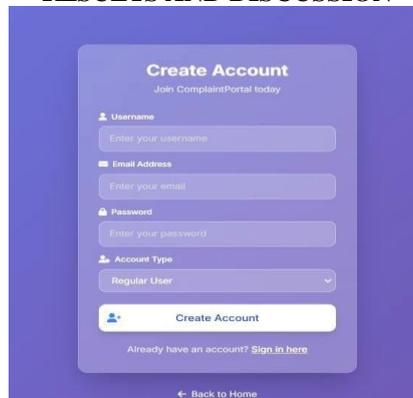
**MongoDB Atlas:** MongoDB Atlas serves as the scalable, cloud-based data store for the entire application. Everything related to a complaint the original, the agent notes, status changes, and history - is stored here permanently as flexible NoSQL documents. It is the Core System that manages all CRUD (Create, Read, Update, Delete) operations.

**Web Interface :**The Web Interface is the main front end component and serves as the portal for

customer and supporters alike. The Web Interface's primary function is to present a means of submitting new complaints and presenting their status in the user interface. Through the Web Interface user input is then packaged into standardized payloads to be submitted to the Core System over their RESTful APIs.

**Automatic Escalation:** The Automatic Escalation module functions as the dynamic means by which the outlined system function (automatic escalation) is executed. This service operates autonomously, performing continuous checks or being triggered in response to events, before applying business rules (e.g., checking whether a complaint has exceeded the time- to-resolve limit

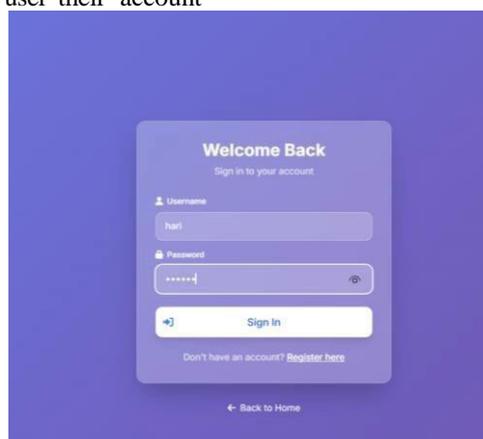
**RESULTS AND DISCUSSION**



**Fig2: Register page**

Where the user can fill the details with their uesrname, mail, password and user their account

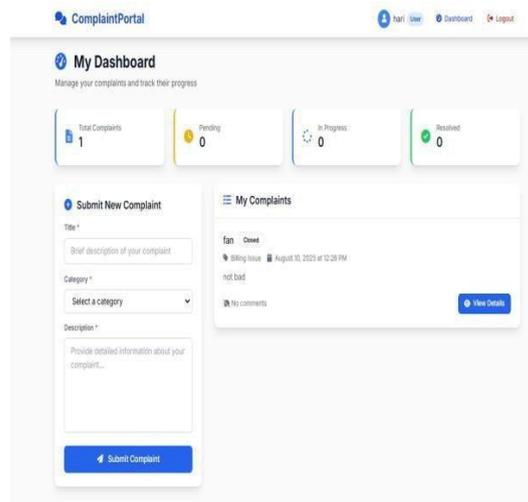
and then can login with there details



**Fig3: Login Page**

In this page where the user can be login there details with username and password and they can

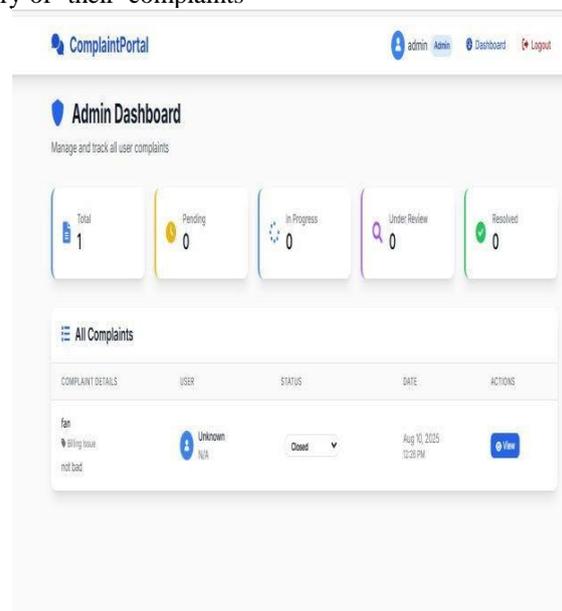
perform the operations



**Fig4: User Dashboard**

Where the user can raise the complaints by using the title and select the category of their complaints

and they can raise their complaints



**Fig5: Admin Dashboard**

In this dashboard where the admin can be done when the user raise the complaints by selecting closed, pending, completed and completed the complaints.

### CONCLUSION

The successful design and implementation of the Complaint Escalation System established a centralized and efficient platform for managing user complaints. With secure user authentication, only registered users can access the system, ensuring data integrity and privacy. The inclusion of a real-time dashboard allows users and administrators to monitor complaint progress and resolution status effectively. Furthermore, the automated escalation mechanism ensures timely action by alerting higher-level authorities when issues exceed the defined Service Level Agreement (SLA) period.

The use of MongoDB Atlas as the database enhances scalability, reliability, and security, making the system robust and adaptable for organizational needs. Overall, the system fulfills its objective of streamlining complaint management, improving accountability, and ensuring prompt issue resolution.

### FUTURE ENHANCEMENT

In the future, the system can be further enhanced to improve accessibility, intelligence, and user engagement. The development of cross-platform mobile applications for Android and iOS can extend accessibility, enabling users to manage and track complaints on the go. The integration of AI-powered complaint classification using machine learning and Natural Language Processing can

automate the categorization of complaints, predict resolution timelines, and prioritize urgent cases. Additionally, incorporating a multi-language interface will increase inclusivity, allowing users from diverse linguistic backgrounds to interact with the system seamlessly. These enhancements will transform the Complaint Escalation System into a more intelligent, user-friendly, and globally accessible platform for efficient grievance management.

#### REFERENCES:

1. MongoDB, *MongoDB Atlas Documentation*. [Online]. Available: <https://www.mongodb.com/docs/atlas>
2. Flask, *Flask Web Development Framework*. [Online]. Available: <https://flask.palletsprojects.com>
3. Bootstrap, *Bootstrap Official Documentation*. [Online]. Available: <https://getbootstrap.com/docs>
4. Python Software Foundation, *Python Documentation*. [Online]. Available: <https://docs.python.org>
5. W3Schools, *HTML, CSS, JavaScript, and Flask Tutorials*. [Online]. Available: <https://www.w3schools.com>
6. GeeksforGeeks, *Flask Tutorials and Python Web Development*. [Online]. Available: <https://www.geeksforgeeks.org>
7. DigitalOcean, *Understanding Web Application Security*. [Online]. Available: <https://www.digitalocean.com/community/tutorials>
8. Kumar, R., et al., "Automated Escalation Mechanisms in Complaint Management Systems," *International Journal of Computer Applications*, vol. 975, no. 8887, 2020.
9. Smith, J., & Clark, M., "Improving Service Quality through Complaint Escalation."