# A Web-Based Bug Tracking and Issue Management System for Small and Medium Software Development Teams

**M. R. Chennoji[1], Yogeshwari Lomte[2], Aarti Lakade[3], Saurabh Meshram[4], Shashank Gadhe[5]**

[1]Assistant Professor, Department of Computer Science and Engineering, MGM's College of Engineering Nanded, India
[2,3,4,5]B.TECH, Dept. of Computer Science and Engineering, MGM's College of Engineering, Nanded , India

------------------------------------------------------------*****************-----------------------------------------------------------

## ABSTRACT

**Software development projects inevitably encounter defects and issues throughout the software development life cycle. Effective bug reporting and tracking play a crucial role in ensuring software quality, reducing development time, and improving team collaboration. However, many academic project teams, startups, and small software organizations still rely on informal bug reporting methods such as emails, messaging applications, spreadsheets, or verbal communication. These approaches often result in miscommunication, loss of bug information, lack of accountability, and delayed resolution.**

**This paper presents a Web-Based Bug Tracking and Issue Management System designed to provide a centralized, structured, and user-friendly platform for reporting and managing software bugs. The proposed system allows users or testers to submit detailed bug reports with severity, priority, reproduction steps, environment details, and attachments. Administrators or developers can review reported issues, update their status, and monitor progress through a dedicated dashboard. The system is implemented using the Flask web framework for backend processing and MongoDB for data storage.**

**By focusing on simplicity, minimal setup requirements, and essential functionalities, the proposed system serves as a lightweight alternative to complex enterprise tools such as Jira and Bugzilla. Experimental usage shows improved communication, better bug traceability, and increased efficiency in small development teams. The system is well suited for academic environments and small-scale software development projects.**

**Keywords: Bug Tracking System, Issue Management, Software Engineering, Flask, MongoDB, Web Application.**

## INTRODUCTION

Software systems have become increasingly complex due to evolving user requirements, rapid technological advancements, and frequent updates. As a result, software defects or bugs are unavoidable during development and maintenance phases. If not managed effectively, these bugs can lead to system failures, poor user experience, increased development cost, and delayed project delivery. Therefore, systematic bug tracking and issue management are essential components of modern software engineering practices. In many small organizations and academic environments, bug reporting is still performed using unstructured methods such as emails, WhatsApp messages, spreadsheets, or verbal communication.

These methods lack proper documentation, tracking mechanisms, and transparency, leading to lost issues and unclear responsibilities. As the number of reported bugs increases, managing them without a centralized system becomes increasingly difficult.[1] Although enterprise-level bug tracking tools such as Jira, Bugzilla, and Redmine provide comprehensive solutions, they are often complex, expensive, and require significant setup and training.

These factors make them unsuitable for students, beginners, and small development teams. Hence, there is a need for a simple, lightweight, and user- friendly bug tracking system that focuses on essential functionality without unnecessary complexity.[2]

## LITERATURE REVIEW

Various bug tracking and issue management systems have been developed and adopted in the software industry. Jira is a widely used commercial tool that provides advanced features such as sprint planning, automation, and analytics. However, its subscription-based pricing and complexity make it unsuitable for small teams. Bugzilla is an open-source alternative but requires server configuration and technical expertise. Redmine provides project management features along with issue tracking but offers a less intuitive user interface for non-technical users.

Recent studies emphasize the importance of usability and simplicity in software tools. Research indicates that overly complex systems reduce adoption rates, particularly among students and small organizations. Lightweight web-based systems developed using modern frameworks such as Flask offer improved accessibility, faster deployment, and ease of customization. The literature highlights a research gap in providing a bug tracking solution that balances simplicity, usability, and essential functionality. The proposed system aims to address this gap by offering a structured yet easy-to-use platform tailored for small-scale software development environments.[3],[4]

## 1. PROBLEM STATEMENT

Despite the availability of numerous bug tracking tools, small development teams and academic projects face challenges due to the complexity and cost of existing solutions. Informal bug reporting methods lead to miscommunication, loss of bug data, and inefficient resolution processes. Therefore, there is a need for a simple, free, and efficient bug tracking system that enables structured reporting, transparent tracking, and effective communication among team members.

## 2. PROPOSED SYSTEM

The proposed **Web-Based Bug Tracking and Issue Management System** provides a centralized platform for reporting, managing, and tracking software bugs. The system is designed with a focus on usability, minimal configuration, and essential features required for effective bug management.

### 2.1 Objectives

- To provide a structured platform for bug reporting
- To reduce communication gaps between users and developers
- To ensure transparency in bug status tracking
- To improve productivity in small development teams

### 2.2 System Features

- User authentication and login
- Detailed bug reporting with severity and priority
- Step-by-step reproduction details
- Attachment of screenshots or log files
- Administrative dashboard for bug management
- Real-time status tracking

## 3. SYSTEM ARCHITECTURE

The system follows a **client–server architecture** consisting of three layers:

1. **Presentation Layer:** User interface developed using HTML, CSS, and JavaScript
2. **Application Layer:** Flask backend handling business logic and routing
3. **Data Layer:** MongoDB database storing user data and bug records

Users interact with the frontend to submit bug reports, which are processed by the backend and stored in the database. Bugs are displayed in descending order based on creation time to ensure that newly reported issues appear at the top.[2]

## METHODOLOGY

The workflow of the system is as follows:

1. User logs into the system

2. User submits a detailed bug report
3. Backend validates and stores the data
4. Admin reviews the reported bug
5. Admin updates bug status
6. User tracks the status through the dashboard

This workflow ensures accountability and transparency throughout the bug lifecycle.

## 7. IMPLEMENTATION

The system is implemented using Flask due to its lightweight nature and ease of integration. MongoDB is used as the database for its flexibility and scalability. Each bug report is stored as a document containing attributes such as title, description, severity, priority, environment details, and status.

To improve usability, bugs are sorted using MongoDB's unique identifier in descending order, ensuring that the latest bugs are displayed first.[3]

## RESULTS AND DISCUSSION

The system was tested with multiple users reporting bugs simultaneously. The results indicate improved bug traceability, reduced communication overhead, and faster issue resolution compared to informal reporting methods. Users found the system easy to use, and administrators were able to manage bugs efficiently through the dashboard.

## 8. COMPARISON WITH EXISTING SYSTEMS

| Feature | Jira | Bugzilla | Proposed System |
|---|---|---|---|
| Cost | Paid | Free | Free |
| Complexity | High | Medium | Low |
| Setup | Required | Required | Not Required |
| User Friendly | Medium | Low | High |
| Best Suited For | Enterprises | Developers | Students & Small Teams |

## CONCLUSION

This paper presented a lightweight web-based Bug Tracking and Issue Management System tailored for small development teams and academic projects. Unlike enterprise tools, the proposed system emphasizes simplicity, usability, and cost-effectiveness. The system successfully improves communication, prevents loss of bug data, and enhances overall software development efficiency.

## FUTURE SCOPE

Future enhancements include email notifications, role-based access control, bug prioritization analytics, and mobile application support to further improve usability and scalability.

## REFERENCES

[1]. Pressman, R. S., Software Engineering: A Practitioner's Approach, McGraw-Hill.
[2]. Sommerville, I., Software Engineering, Pearson Education.
[3]. Jira Software Documentation, Atlassian.
[4]. Bugzilla: An Open Source Bug Tracking System.