

Bug Tracker System – A Web-Based Application for Bug Reporting and Real-Time Issue Tracking

Shinde Rohit Ram¹ Purevesh Wagh²

¹Student ²Assistant Professor

^{1,2}Master of Computer Applications

^{1,2}Anantrao Pawar College of Engineering & Research, Pune Affiliated to Savitribai Phule Pune University, India

Abstract — Software bugs and technical issues are common challenges during software development and maintenance work. In many cases, traditional bug reporting methods create problems such as slow issue resolution, repeated bug entries, poor communication, and less transparency between users and developers. To reduce these difficulties, this paper introduces a Bug Tracker System, which is a web-based application developed for software bug reporting, issue tracking, and better coordination between users and developers through a centralized platform. The system follows a structured process where users can submit software issues, administrators can review reported bugs, and developers can track as well as resolve assigned problems effectively. The application includes several useful features such as secure user login, role-based access, bug status tracking, priority-wise issue handling, screenshot upload support, and live issue status updates. The Bug Tracker System is implemented using web technologies such as HTML, CSS, Bootstrap, JavaScript, Python, Flask framework, and SQLite database. The application provides a simple and responsive interface that helps users report and monitor software issues easily. Security-related features including protected authentication and authorized user access are implemented to maintain safe and reliable handling of system data. The application helps improve debugging activities by lowering manual work, improving interaction between users and developers, and maintaining proper transparency during the issue tracking process. The obtained results show improved bug observation, quicker issue resolution, better workflow handling, and efficient processing of software-related problems. The proposed system shows how centralized web technologies can improve software quality, issue tracking activities, and overall software development performance.

Keywords: Bug Tracking System; Software Issue Management; Web-Based Issue Tracking; Role-Based Access Control; Software Quality Assurance; Flask-Based Web Application;

I. INTRODUCTION

Software bugs and system-related errors are common problems during software development and maintenance processes. Improper handling of software issues can decrease system performance, delay project work, affect user satisfaction, and increase maintenance efforts. In many organizations, bug reporting and issue tracking are still managed manually through emails, spreadsheets, or informal communication methods. These methods often result in repeated bug entries, delayed responses, reduced transparency, and weak interaction between users, testers, administrators, and developers.

To address these difficulties, the Bug Tracker System is developed as a web-based application that provides a centralized environment for reporting, tracking, and monitoring software bugs efficiently. The application allows users to submit bug-related information including bug title, issue description, priority level, and screenshots. Administrators and developers can review reported bugs, update issue status, and maintain proper tracking during the debugging process.

The application includes important functionalities such as secure user authentication, role-based access control, issue status monitoring, and priority-wise bug handling. Different issue conditions such as Open, In Progress, and Closed help users and developers understand the current stage of reported software bugs. The centralized workflow improves communication and reduces confusion during software maintenance activities.

From a technical perspective, the Bug Tracker System is implemented using HTML, CSS, Bootstrap, JavaScript, Python Flask framework, and SQLite database. The application provides a responsive and simple interface for smooth bug handling and issue monitoring activities. Secure login functionality is implemented to protect user information and maintain authorized system accessibility.

By integrating centralized bug reporting and live issue tracking, the Bug Tracker System provides an effective and reliable solution for handling software-related problems. The application improves transparency, reduces manual effort, strengthens coordination between developers and users, and supports improved software quality and maintenance processes.

II. PROBLEM STATEMENT:

Software bugs and application-related errors create major challenges during software development and maintenance activities. In many organizations and software development environments, bug reporting and issue tracking are still handled manually using emails, spreadsheets, or informal communication methods. These traditional approaches often create communication problems, repeated issue reports, delayed bug resolution, reduced transparency, and weak coordination among users, testers, administrators, and developers.

One of the main problems in existing systems is the absence of a centralized platform for handling software bugs properly. Users often face difficulties while reporting issues accurately, while developers experience challenges in monitoring bug status and identifying high-priority issues. Due to weak monitoring methods, many software bugs remain unresolved for long periods, which negatively affects software quality and user experience.

Another important limitation is the lack of real-time issue tracking and status visibility. In several situations, users are unable to understand whether their reported issue is under processing, resolved, or still pending. This creates confusion and reduces system reliability. In addition, manual handling of software-related problems increases workload and slows down debugging activities.

Security and access control are also major concerns in traditional bug tracking methods. Many existing applications do not provide secure authentication and proper role-based access functionality, which may result in unauthorized access or incorrect modification of issue-related information.

To solve these limitations, there is a need for a centralized and efficient web-based application that simplifies software bug reporting, issue tracking, and status monitoring. The proposed Bug Tracker System is developed to provide secure user authentication, role-based access control, priority-based issue handling, and real-time issue status updates within a simple and user-friendly environment. The application improves transparency, reduces manual work, improves coordination between users and developers, and supports effective software maintenance and debugging activities.

III. OBJECTIVES:

The main objective of the Bug Tracker System is to develop an efficient, reliable, and simple web-based application for reporting, tracking, and monitoring software bugs. The application is designed to improve communication between users and developers, reduce manual effort during bug handling, and support software maintenance activities through a centralized issue tracking process.

The specific objectives of the system are as follows:

- To develop a centralized application for reporting and tracking software-related issues effectively.
- To provide secure user registration and login functionality for authorized access to the application.
- To implement role-based access functionality for users, administrators, and developers to maintain proper security and smooth system operation.
- To allow users to report software bugs by providing details such as bug title, issue description, priority level, and screenshots for better understanding of software problems.
- To enable administrators and developers to monitor, update, and maintain bug conditions such as Open, In Progress, and Closed.
- To support priority-based bug handling for identifying and resolving important software issues quickly and effectively.
- To improve transparency by allowing users to check the current status of their reported issues in real time.
- To reduce duplicate issue reports and improve coordination between users and developers through a structured workflow process.
- To maintain bug-related information securely using SQLite database and protected access functionality.

- To develop a responsive and easy-to-use interface using modern web technologies for smooth bug reporting and issue monitoring activities.

IV. LITERATURE SURVEY:

Bug tracking and issue reporting applications play an important role in modern software development and maintenance processes. As software applications continue to become more complex, proper handling of software bugs has become necessary for maintaining software reliability, performance, and user satisfaction. Different researchers and developers have introduced multiple bug tracking applications to simplify issue reporting, bug monitoring, and debugging activities. However, many existing applications still face problems related to usability, transparency, security, and coordination between users and developers.

In earlier stages, bug tracking activities mainly depended on manual methods such as spreadsheets, emails, and written records for maintaining software issue details. These traditional approaches were difficult to manage and often resulted in repeated bug reports, delayed communication, and inefficient issue tracking processes. Users also had limited information regarding the latest condition of reported software bugs.

Later, several web-based bug tracking applications were introduced to improve software issue handling and maintenance activities. These applications provided functionalities such as bug submission, issue categorization, and status monitoring. Although these applications improved software debugging activities, many of them lacked secure user authentication, proper role-based access functionality, and real-time issue tracking features.

Research related to “Web-Based Bug Tracking Applications” explained the importance of centralized issue reporting for improving communication between users and developers. These applications simplified bug monitoring activities but still depended heavily on manual status updates and lacked effective workflow coordination functionality.

Similarly, studies associated with “Issue Tracking and Defect Observation Systems” mainly focused on improving software maintenance activities by categorizing issues according to severity and priority. Even though these systems improved bug organization, many failed to provide a simple and user-friendly interface suitable for academic projects and small-scale software environments.

Another research area concentrated on “Online Defect Tracking Applications” developed using technologies such as PHP, Python, and SQL-based databases. These applications provided basic functionalities including login authentication, issue reporting, and status updates. However, several limitations were observed, including:

- Lack of proper centralized monitoring
- Reduced transparency during issue tracking
- Delayed communication between users and developers
- Improper handling of duplicate issue reports
- Limited security and access control functionality
- Complicated user interfaces in some existing applications

Recent studies highlight the importance of lightweight, secure, and user-friendly web-based bug

tracking applications that can improve issue reporting efficiency and software maintenance processes. Modern applications mainly focus on centralized workflows, secure authentication, and structured issue handling methods to improve developer productivity and software quality.

To overcome the limitations of previous applications, the proposed Bug Tracker System provides a centralized and efficient solution for software bug reporting and issue tracking. The application allows users to submit bug reports containing details such as bug title, description, priority level, and screenshots. Administrators and developers can monitor bug conditions through stages such as Open, In Progress, and Closed, which improves transparency and coordination during debugging activities.

The proposed application is developed using HTML, CSS, Bootstrap, JavaScript, Python Flask framework, and SQLite database. Secure login authentication and role-based access functionality are implemented to provide reliable and protected system usage. By integrating centralized bug reporting, structured issue tracking, and easy-to-use interfaces, the application improves software maintenance efficiency and supports better software quality activities.

V. SYSTEM DESIGN AND ARCHITECTURE:

The Bug Tracker System is developed as a centralized web-based application for reporting, tracking, and monitoring software bugs effectively. The application follows a simple three-tier architecture that includes frontend, backend, and database layers. The frontend is designed using HTML, CSS, Bootstrap, and JavaScript to provide a responsive and easy-to-understand interface for users, administrators, and developers. The backend is implemented using the Flask framework in Python, which manages system logic, user authentication, request handling, and communication between different modules. SQLite database is used for storing bug records, user details, and issue status information in a secure and organized manner.

The application consists of different modules such as user authentication, bug reporting, issue tracking, status updating, and user-related operations. The authentication module allows only authorized users to access the system through secure login functionality. The bug reporting module enables users to submit software issue details including bug title, description, priority level, and screenshots. The issue tracking module helps administrators and developers monitor reported bugs and update their status efficiently.

The architecture of the application supports a structured workflow and smooth interaction between users and developers. Different issue conditions such as Open, In Progress, and Closed are maintained to represent the current state of software bugs. Priority-based issue handling helps identify important software problems and improves debugging efficiency. The centralized workflow reduces confusion and supports better transparency during issue resolution activities.

The application also includes secure access functionality through login authentication and role-based access control to prevent unauthorized usage and protect sensitive system information. SQLite database integration

provides lightweight, reliable, and efficient data storage suitable for academic projects and small-scale software environments.

The working process of the application begins when a user submits a software issue through the web interface. The reported issue is stored in the database and becomes visible to administrators and developers. Developers can analyze the issue, modify its status, and mark it as resolved after successful debugging. Users can also check the latest condition of their reported bugs, which improves communication and transparency between users and developers.

Overall, the Bug Tracker System architecture combines centralized bug reporting, secure authentication, structured issue tracking, and responsive web technologies to provide an efficient, reliable, and user-friendly solution for software debugging and maintenance activities.

VI. IMPLEMENTATION:

The Bug Tracker System is implemented using a web-based technology stack that provides simplicity, reliability, and efficient issue tracking functionality. The frontend of the application is developed using HTML, CSS, Bootstrap, and JavaScript, which provides a responsive and user-friendly interface for users, administrators, and developers. Through the web interface, users can create accounts, log in, submit software bugs, check reported issues, and monitor bug status updates. The backend is developed using the Flask framework in Python, which manages application processing, request handling, authentication, and communication between different system modules. SQLite database is used to store and maintain user details, bug records, issue status information, and related system data in an organized format.

The application follows a modular implementation structure where separate modules are developed for authentication, bug reporting, issue monitoring, status updating, and user-related operations. The authentication module provides secure login access for authorized users. Role-based access functionality is implemented to ensure that users, administrators, and developers can access only the permitted features available within the application.

The bug reporting process starts when a user submits issue details through the web interface. Users can provide information such as bug title, issue description, priority level, and screenshots related to the software problem. After submission, the issue information is stored securely in the SQLite database and becomes available for tracking and debugging activities.

Administrators and developers can access all reported software issues and update their current status according to the debugging process. The application supports different issue conditions such as Open, In Progress, and Closed, which help in monitoring issue progress effectively. Priority-based issue handling is also implemented to identify important software problems and support faster resolution of critical bugs.

The centralized workflow improves communication between users and developers by maintaining transparency throughout debugging activities. Users can monitor the latest condition of their reported issues without depending on

traditional communication methods such as emails or spreadsheets. The application also helps reduce duplicate bug reporting and improves software maintenance efficiency.

The Bug Tracker System is designed to provide lightweight and efficient performance suitable for academic projects and small-scale software environments. SQLite database integration supports fast data access and simplified deployment without requiring complex database server configuration. The responsive web interface allows users to access the application easily through standard web browsers.

Overall, the implementation of the Bug Tracker System combines secure authentication, centralized bug reporting, structured issue tracking, and responsive web technologies to provide an efficient and reliable solution for software debugging and maintenance activities.

VII. RESULTS AND DISCUSSION:

The implementation of the Bug Tracker System shows noticeable improvement in software bug reporting, issue tracking, and debugging activities. The application was tested using different user operations such as bug submission, login authentication, status modification, and issue monitoring. The obtained results indicate that the application provides better coordination between users and developers when compared with traditional manual bug handling methods.

One of the important outcomes of the application is the simplification of the bug reporting process. Users can easily submit software-related issues by providing details such as bug title, description, priority level, and screenshots. This helps developers understand software problems more clearly and reduces confusion during debugging activities. The centralized storage of issue reports also minimizes the possibility of losing important bug-related information.

The issue tracking functionality improves transparency within the application. Different issue conditions such as Open, In Progress, and Closed help users and developers monitor the present progress of reported software bugs. This structured workflow reduces communication gaps and supports better handling of software maintenance activities.

The application also improves issue handling efficiency through priority-based bug processing. Critical software issues can be identified quickly and resolved earlier compared to low-priority bugs. This helps improve software reliability and increases user satisfaction.

Secure login authentication and role-based access control provide protected system accessibility and prevent unauthorized usage. Only authorized users are allowed to report software issues or update bug status, which improves data security and system reliability. SQLite database integration supports smooth data storage and retrieval while maintaining lightweight system performance suitable for academic projects and small-scale software environments.

The responsive web interface developed using HTML, CSS, Bootstrap, and JavaScript provides smooth navigation and better usability for users. The application performs efficiently on standard web browsers and supports proper interaction between frontend and backend modules implemented using the Flask framework.

Compared with traditional manual methods such as spreadsheets or email-based reporting approaches, the proposed Bug Tracker System provides a more organized, transparent, and efficient solution for handling software-related issues. The centralized workflow reduces manual effort, improves communication between users and developers, and supports faster debugging activities.

Overall, the obtained results demonstrate that the Bug Tracker System successfully improves software issue handling by providing secure authentication, structured bug tracking, centralized monitoring, and user-friendly interaction. The application offers an effective and reliable solution for software maintenance and debugging activities in academic and small-scale development environments.

VIII. FUTURE SCOPE:

The Bug Tracker System can be improved further by adding advanced technologies and additional functionalities to enhance system performance, usability, and software maintenance activities. Some possible future improvements are described below:

A. Advanced Bug Analysis and Prediction:

Future versions of the application can include AI-based bug analysis and prediction features to identify commonly occurring software issues and help developers reduce software failures before deployment.

B. Real-Time Notification Functionality:

The application can be improved by adding email or SMS notification features for bug assignments, issue status updates, and resolution alerts, which can improve communication between users and developers.

C. Mobile Application Support:

A dedicated Android or iOS mobile application can be developed to provide easier access for users and developers to report and monitor software bugs directly from mobile devices.

D. Improved Dashboard and Analytical Reports:

Advanced dashboards and graphical reports can be implemented to analyze bug patterns, developer activities, issue resolution duration, and software quality improvement statistics.

E. Cloud Database Integration:

The current SQLite database can be upgraded to cloud-based or enterprise-level databases such as MySQL or PostgreSQL to provide better scalability and support for large-scale software applications.

F. Enhanced Security Functionality:

Additional security features such as two-factor authentication and encrypted data storage can be implemented to improve application protection and reduce unauthorized system access.

G. Improved File and Screenshot Handling:

Future versions of the application can support advanced file handling functionalities for uploading multiple screenshots, debugging reports, and log files for improved issue analysis.

H. Integration with Software Development Tools:

The application can be integrated with version control and project coordination platforms such as GitHub or Jira to improve software development workflow and team collaboration.

I. Multi-Language Support:

Support for multiple languages can be implemented to improve usability for users from different regions and technical backgrounds.

Overall, these future improvements can make the Bug Tracker System more secure, scalable, intelligent, and suitable for real-world software development and maintenance environments.

IX. CONCLUSION:

The Bug Tracker System is developed to provide an efficient, reliable, and easy-to-use solution for reporting, tracking, and handling software bugs. The application successfully simplifies software issue handling by providing a centralized platform where users, administrators, and developers can interact and monitor bug-related activities in an organized manner.

The implementation of the application improves communication between users and developers by reducing dependency on manual methods such as emails, spreadsheets, and unstructured reporting processes. Users can submit software bug reports with proper issue details, while developers can monitor, analyze, and update issue status effectively. The structured workflow using issue conditions such as Open, In Progress, and Closed improves transparency and supports better software maintenance activities.

The application also provides secure login authentication and role-based access control, ensuring that only authorized users can access specific functionalities within the system. The integration of Python Flask framework and SQLite database supports smooth backend processing and reliable data handling while maintaining lightweight and efficient system performance.

The responsive user interface developed using HTML, CSS, Bootstrap, and JavaScript improves usability and provides better interaction for users across different web browsers and devices. Priority-based issue handling also helps developers identify important software bugs and resolve them efficiently.

Overall, the Bug Tracker System demonstrates how web technologies and structured issue tracking methods can improve software debugging and maintenance processes. The application reduces manual effort, improves coordination between users and developers, and provides better visibility of software issues throughout their lifecycle.

With future improvements such as real-time notifications, advanced analytical reports, cloud database integration, and mobile application support, the application can become more scalable and suitable for real-world software development environments. The project provides a strong foundation for effective software issue handling and contributes toward improving software quality and maintenance efficiency.

ACKNOWLEDGMENT:

I would like to express my sincere thanks to the Department of Master of Computer Applications at Anantrao Pawar College of Engineering & Research (APCOER), Pune, for providing the required facilities, technical resources, and academic support needed for the successful completion of this research work.

I am deeply thankful to my project guide, Purevash Wagh, for his valuable guidance, continuous encouragement, technical support, and useful suggestions throughout the development of the Bug Tracker System. His support and motivation played an important role in completing this project successfully.

I would also like to thank Savitribai Phule Pune University for encouraging practical learning, innovation, and research activities during the MCA program.

I sincerely thank all the teaching and non-teaching staff members of the MCA department for their cooperation and support whenever required during the project work.

Finally, I would like to express my heartfelt gratitude to my family and friends for their continuous motivation, encouragement, and moral support throughout the academic year 2025–2026. Their support helped me complete this research work successfully.

REFERENCES:

- [1] M. Grinberg, *Flask Web Development: Developing Web Applications with Python*, O'Reilly Media, 2018.
- [2] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 8th ed., McGraw-Hill Education, 2015.
- [3] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 7th ed., Pearson Education, 2016.
- [4] I. Sommerville, *Software Engineering*, 10th ed., Pearson Education, 2015.
- [5] Ramez Elmasri and Shamkant B. Navathe, *Fundamentals of Database Systems*, 7th ed., Pearson Education, 2016.
- [6] R. Steven F. Lott, *Modern Python Cookbook*, O'Reilly Media, 2020.
- [7] Jon Duckett, *HTML and CSS: Design and Build Websites*, Wiley Publications, 2014.
- [8] S. Ethan Brown, *Web Development with Bootstrap*, O'Reilly Media, 2017.