TICKET RAISING SYSTEM

E.Aarthi,

M.Sc.(Computer Science),

Department of Computer Science and Applications,

Auxilium College (Autonomous), Gandhi Nagar, 632006, Vellore District, TamilNadu, India.

E.Mail:aarthielangovan16@gmail.com

Dr.S.Tharani,

Assistant Professor,

Department of Computer Science and Applications,

Auxilium College (Autonomous), Gandhi Nagar, 632006, Vellore District, TamilNadu, India.

E.Mail:tharanica@auxiliumcollege.edu.in

ABSTRACT:

In today’s software development, it’s very important for different people involved to communicate clearly so that projects get completed on time. A ticket raising system serves as a main place where problems can be reported, progress can be tracked, and everyone’s responsibility can be clear throughout the software development process. This paper talks about designing and building a ticket raising system using Laravel, which includes a role-based access system with roles like Admin, Developer and Tester. The system uses Laravel’s MVC setup, Eloquent ORM and built-in tools for user authentication to create an application that is easy to scale, organized and safe. Using a clear process, users can create tickets, assign them, watch their progress and fix problems, making sure that bugs, tasks, and feature ideas are dealt with properly. The system is meant to help teams work together better, avoid delays and keep a clear record of all software issues from the beginning until the end.

I. INTRODUCTION

A ticket raising system is a web application that allows users to report problems, ask for help or make requests in a structure and easy-to track way. These systems are commonly used in customer support, IT service management, and internal help desks to help users and support staff communication better. When someone has a problem or needs help, they can create a "ticket" that includes details like what the issue is, how urgent it is, and any relevant information. The support team can then check the status of each ticket, such as "Open", "In Progress", or "Closed" assign it to the right person, reply to the user, and fix the problem. This helps make sure that sure that no requests are missed or forgotten, and both users and support agents can follow the progress and keep communication clear.

Laravel is a strong PHP web framework that works well for building a ticket raising system. It has a modern design, friendly tools, and many built-in features that make development easier. Laravel uses the Model-View-Controller (MVC) design pattern, which helps organize code and separate tasks clearly. It includes tools for user sign-in, managing routes, checking form details, handling databases, and sending notifications, all of which are important for a support system to work well. Using Laravel's Eloquent ORM, developers can easily create models for users, tickets, replies, and agents, and set up how they connect (for example, one user can create many tickets, and one ticket can have multiple replies). Laravel's Blade templating engine or Livewire components help make dynamic user interfaces that let users create, see and manage their tickets smoothly.

Laravel also supports extra packages and tools that can enhance the ticketing system. For example, the Spatie Laravel Permission package helps manage user roles like "customer", "agent" and "admin" making sure users only have access to the parts of the system that are relevant to their role. Laravel Notifications can send real-time updates to users when their tickets are changed or resolved, via email or in-app alerts. Background jobs and queues can handle sending emails or processing ticket assignments without slowing down the user interface. Admin dashboards can be made using packages like Laravel Nova or Filament to help support teams filter, sort, and manage incoming tickets efficiently. Overall, Laravel gives all the tools needed to build a secure, scalable and easy-to-use ticket raising system that can be customized to fit various business needs.

II. LITERATURE REVIEW

Title: Reducing user input requests to streamline the IT support ticket resolution process.

Authors: Gupta, M., Asadullah, S., Padmanabhuni, A., Serebrenik, A.

Year: 2018

Abstract: Managing and maintaining IT infrastructure such as hardware, software, and networks is a critical aspect of software development and maintenance initiatives. Service delivery ensures that user-submitted tickets, such as those from software developers, are addressed within agreed-upon resolution timelines. If these deadlines are not met, service providers may incur penalties. To avoid penalties for delays caused by non-working hours, such as waiting for user input, these periods are excluded from the service-level agreement (SLA) and the system pauses during these periods, though they still impact resolution timelines and user experience. This study aims to analyze the impact of user input during the ticket lifecycle. To address this, the research categorizes user request types, such as real-time and tactical user engagement, where tactical requests are aimed at pausing the SLA without requiring user input. Interventions include proactive request management and user engagement to reduce the number of user input requests and improve resolution time for ticket processing. A case study of a large multinational IT company revealed that around 57% of ticket resolution time is affected by user input. Our findings show that user input delays impact ticket resolution time, leading to higher SLA violations. Our solution significantly improves ticket resolution across proposed system features, including tiered resource usage and user communication improvements. User input resolution is crucial for efficient and SLA mandated service improvements. The improvement is revealed through proactive user engagement and tactical methods to reduce service level resolution time. Both traditional and proposed approaches focus on reducing SLA violations and improving resolution time. The SLA management includes SLA management includes SLA cycle management and better ticket resolution. The impact of user input on ticket resolution is significant, with the proposed solution out performing traditional SLA methodologies.

Title: ESSMArT way to manage customer requests.

Authors: Nayebi. M., L. Dicke, R. Ittyipe, C. Carlson and G. Ruhe.

Year: 2019

Abstract: The quality and acceptance of software products in the market are significantly influenced by our responsiveness to customer inquires. Upon receiving a customer request, it is essential to determine whether it should be escalated to the development team. Following the escalation, we are required to document the request as a development task and allocate it to a developer. To enhance this process and minimize the duration from receipt to escalation of a customer request, we aim to automate the entire customer request management system. It introduces a method known as ESSMArT. This method encapsulates text, forecasts the likelihood of a ticket being escalated, formulates the title and content of the ticket, and ultimately assigns it to an available developer. It commenced the evaluation of this method by performing an internal review of 4,114 customer tickets from Brightsquid's secure healthcare communication platform, Secure-Mail. Subsequently, it conducted an external evaluation to gauge the effectiveness of the approach. Our findings indicated that i) Supervised learning utilizing context-specific data is the most efficient for extractive summarization; ii) Random Forest, trained on a combination of conversation data and extractive summarization, is optimal for predicting ticket escalation, achieving the highest precision (0.9) and recall (0.55). In the external evaluation, it was also revealed that ESSMArT produces suggestions that are 71% in agreement with those proposed by humans. When the prototype was applied to 315 customer requests, it observed an average time savings of 9.2 minutes per request. ESSMArT accelerates ticket management ana alleviates the workload for human experts. It was concluded that ESSMArT not only expedites ticket management but also reduces the effort required from individuals. It can assist Brightsquid in (i) mitigating the impacts of staff turnover and (ii) decreasing the time from when an issue is reported to when a developer is designated to address it.

Title: Adaptive MoD Chatbot: Aiming to Provide Contextual Corporate Summarized Documents as Suggestions and Route Reported Issue Tickets.

Authors: Nayak, S.P., A. Rai, K.

Year: 2021

Abstract: With the emergence of the Internet and cloud computing technologies, numerous software solutions are now deployed on cloud infrastructure (SaaS). This model enables customers to pay for services based on their usage. Consequently, non-functional requirements of software, such as availability and reliability, have gained significant importance, as customers anticipate continuous support for their hosted applications. In this scenario, chatbots or conversational AI serve a crucial function as the initial point of contact for customers inquires. This approach minimizes the necessity for multiple interactions between the DevOps team and customers. At present, the Manager on Duty (MOD) acts as the primary contact for any incoming customer inquiries. If required, these inquiries are forwarded to the Developer on Duty (DoD) for prompt resolution. This paper intents to develop an innovative solution featuring an adaptive chatbot that develop an innovative solution featuring an adaptive chatbot that delivers contextual Q&A responses. Theis will encompass built-in natural language template generation, a technical document corpus with a designated knowledge base, and document text summarization tailored to the chatbot's interface. Furthermore, it will establish an indexing system for corporate technical documents to provide essential links and summarized text as responses. The system will also generate support tickets for reported issues, incorporating automated classification to facilitate quicker resolution of customer inquiries by the DoD. The methodologies employed will derive from information retrieval, information extraction, natural language processing, natural language generation, text summarization, text classification and ontology. This paper introduces a customized text ranking algorithm that prioritizes document sentences containing n-gram linked entities from the ontology knowledge base.

Title: Web-Based IT Helpdesk Ticketing System

Author: PT. Dayacipta Kemasindo

Year: 2022

Abstract: An IT helpdesk refers to a software or system designed to assist users in resolving their IT-related issues through one or multiple contact points. It enables users to report problems, monitor their status, and obtain support for products or services. Pt Dayacipta Kemasindo depends on IT for administrative functions and office operations. When IT issues arise, assistance from the IT department becomes essential. Nevertheless, several challenges exist. One significant concern is that the department reporting disturbances and malfunctions often lacks knowledge regarding the duration required to resolve their issues. The queuing process for tasks is not transparent, which can create an impression of sluggishness. The result of this research will be a system that the IT department can utilize to manage incoming assignments and tasks.

Title: Creating a Helpdesk Ticketing System to Enhance the Efficiency of IT Support Services on an XYZ Company Website Utilizing the Laravel Framework.

Author: Mohamad Nur Wahyuddin.

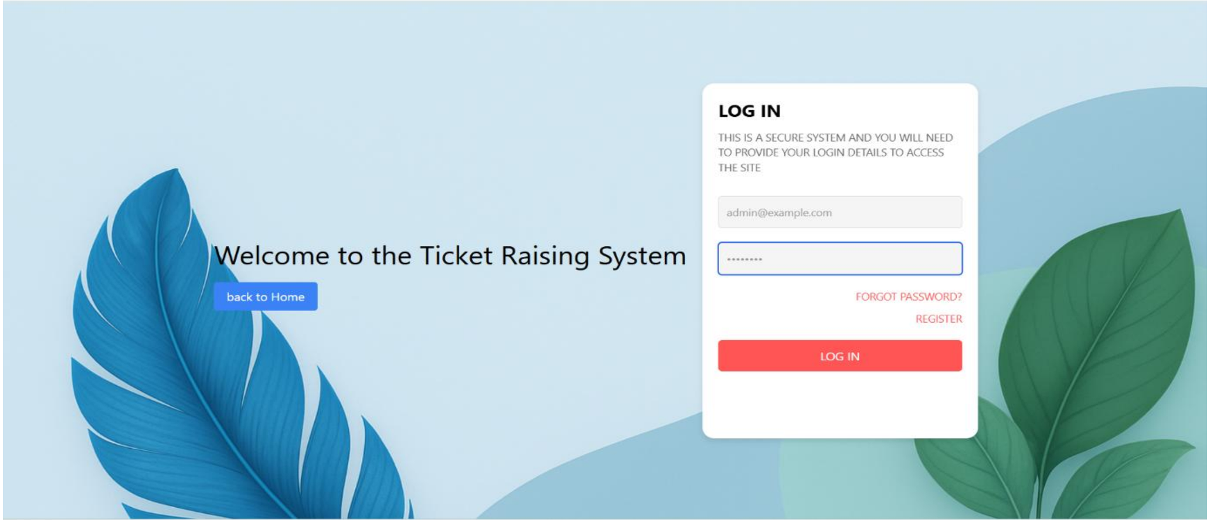
Year: 2024

Abstract: The XYZ company requires a Helpdesk Ticketing System that can enhance the efficiency of its IT support services. This research aims to design and implement a web-based Helpdesk Ticketing System using the Laravel Framework. The system assists the IT department in effectively responding to and resolving employee issues. The research methodology encompasses user requirement analysis, system design, and application development utilizing the Laravel framework. During the requirement analysis phase, the author identifies user needs and determines the essential features required in the system. Subsequently, the system is designed with an emphasis on effective organization and design principles. In the development phase, the author employs the Laravel framework to construct the Helpdesk Ticketing System. This framework was selected for its capability to expedite the development process and provide the necessary functionalities for the system. The system enables employees to report issues via the website, allowing the IT team to efficiently respond to and manage incoming tickets.

III. TICKET RAISING SYSTEM

A Ticket raising system is a crucial software platform extensively utilized in IT companies to enhance project management, issue tracking and team collaboration by facilitating effective communication among various roles involved in the software development lifecycle. The system generally comprises three primary dashboards: admin, developer and tester each with distinctly defined responsibilities and features to uphold transparency and efficiency. The admin dashboard functions as the central control panel where the administrator oversees users, projects and the overall communication flow between developers and testers. Administrators have the capability to create new projects, add or remove users, assign roles, and display the number of projects and users currently active in the system, providing a comprehensive view of ongoing operations. The developer dashboard enables developers to access the issues reported by testers, and take necessary actions to resolve them by writing code fixes or implementing solutions. After a ticket has been addressed, the developer can update its status and add pertinent comments or notes to communicate back with the tester, ensuring a transparent resolution cycle. This collaborative workflow guarantees that testers and developers remain synchronized throughout the bug-fixing process under the oversight of the admin. The tester dashboard is structured to allow testers to create tickets whenever issues, bugs or error are detected during software testing. These tickets encompass details such as project name, issue description, priority level and supporting documentation, offering developers clear insights into the nature of the problem. Testers can also monitor the progress of the tickets they have raised, track their current status, and review any comments or updates made by developers. By consolidating the processes of ticket creation, assignment and resolution into these three dashboards, the ticket raising system enhances accountability, streamlines communication between testers and developers, increases visibility for administrators and allows IT companies to provide more dependable software solutions in a structured and efficient way.

IV. APPENDIX



Welcome to the Ticket Raising System

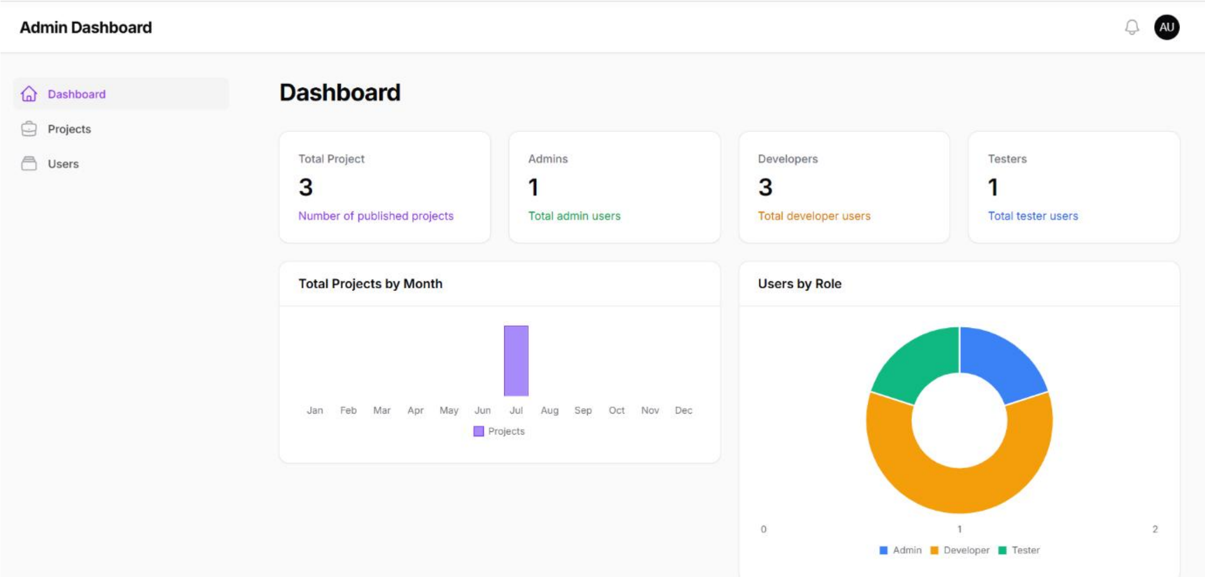
LOG IN

THIS IS A SECURE SYSTEM AND YOU WILL NEED TO PROVIDE YOUR LOGIN DETAILS TO ACCESS THE SITE

admin@example.com

••••••••

Figure 1. Login Page

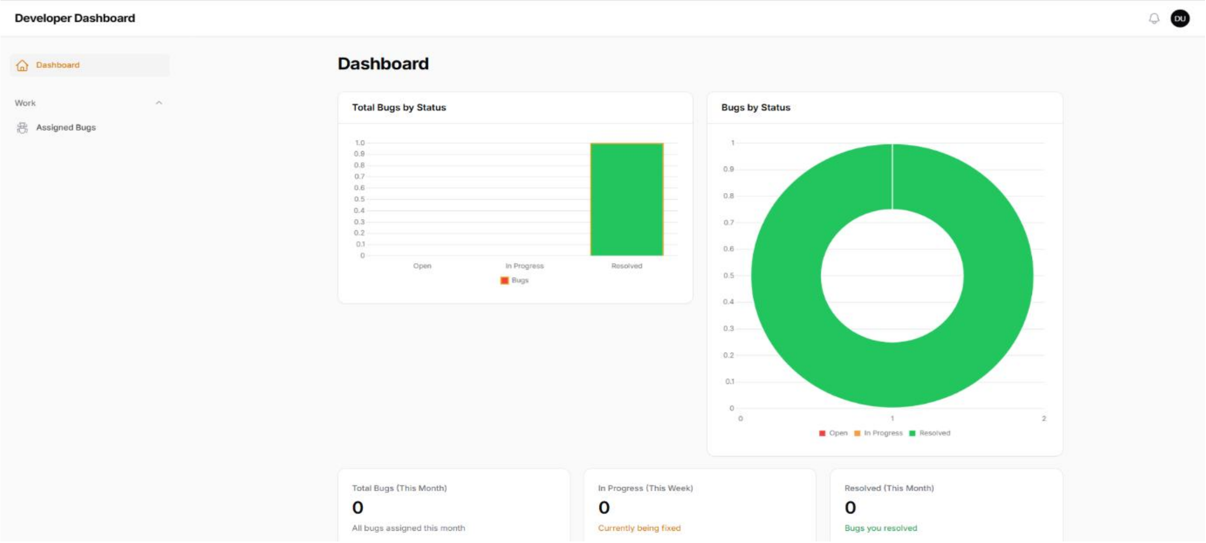


Admin Dashboard

Dashboard

Total Project	Admins	Developers	Testers
3	1	3	1
Number of published projects	Total admin users	Total developer users	Total tester users

Figure 2. Admin Dashboard



Developer Dashboard

DU

Dashboard

Work

Assigned Bugs

Dashboard

Total Bugs by Status

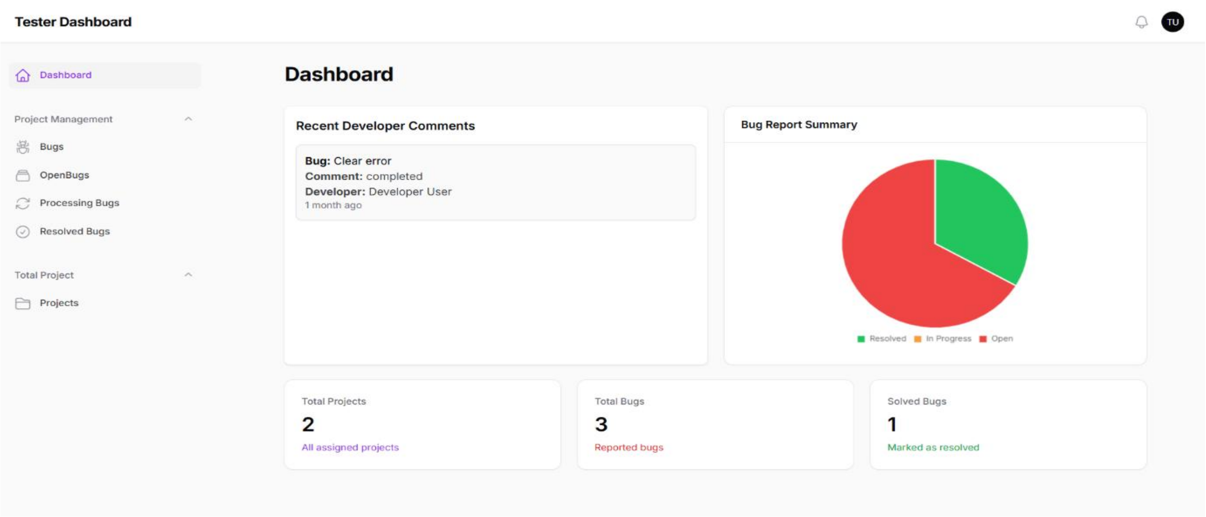
Bugs by Status

Total Bugs (This Month)
0
All bugs assigned this month

In Progress (This Week)
0
Currently being fixed

Resolved (This Month)
0
Bugs you resolved

Figure 3. Developer Dashboard



Tester Dashboard

TU

Dashboard

Project Management

Bugs

OpenBugs

Processing Bugs

Resolved Bugs

Total Project

Projects

Dashboard

Recent Developer Comments

Bug: Clear error
Comment: completed
Developer: Developer User
1 month ago

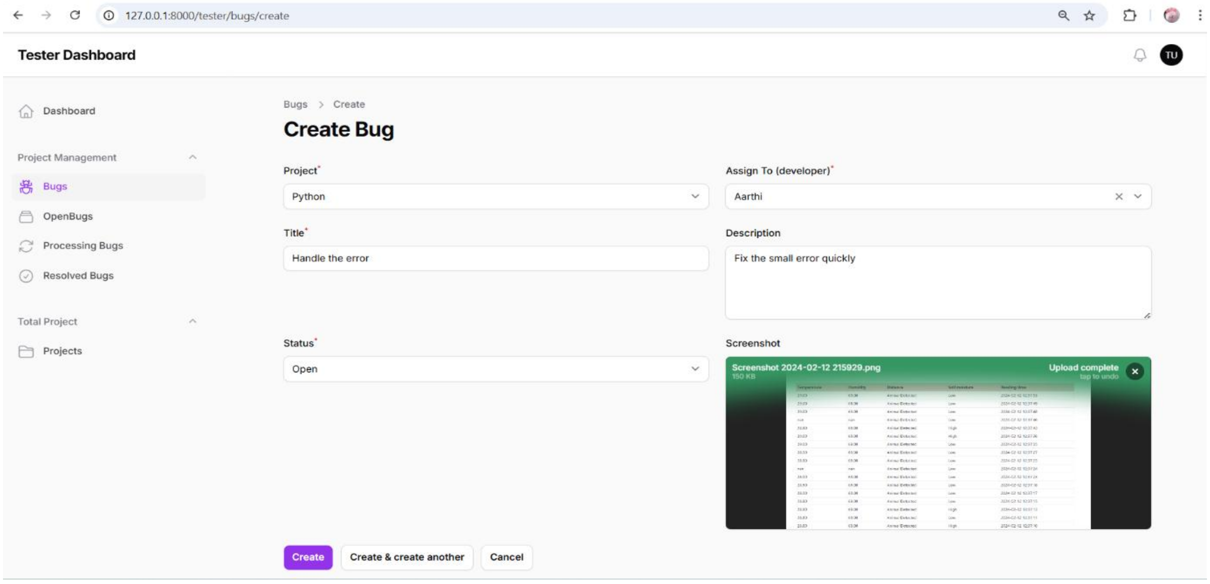
Bug Report Summary

Total Projects
2
All assigned projects

Total Bugs
3
Reported bugs

Solved Bugs
1
Marked as resolved

Figure 4. Tester Dashboard



Create Bug

Project* Python

Assign To (developer)* Aarthi

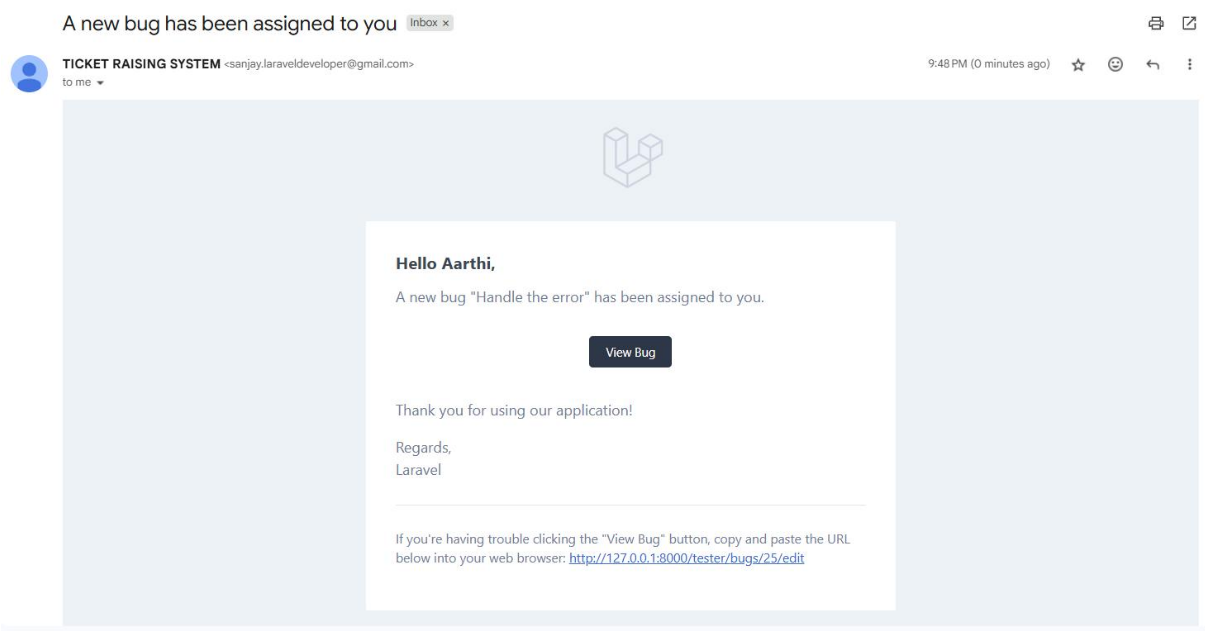
Title* Handle the error

Description Fix the small error quickly

Status* Open

Screenshot Screenshot 2024-02-12 215929.png 150 KB Upload complete tap to undo

Figure 5. Ticket Form



A new bug has been assigned to you

TICKET RAISING SYSTEM <sanjay.laraveldeveloper@gmail.com>
to me

9:48 PM (0 minutes ago)

Hello Aarthi,

A new bug "Handle the error" has been assigned to you.

Thank you for using our application!

Regards,
Laravel

If you're having trouble clicking the "View Bug" button, copy and paste the URL below into your web browser: http://127.0.0.1:8000/tester/bugs/25/edit

Figure 6. E-Mail Notification

V. CONCLUSION

The Ticket Raising System, developed using the Laravel framework, offers a roust and effective platform for managing software issues and facilitating communication among project stakeholders. Utilizing Laravel's MVC architecture, routing capabilities and integrated authentication features, the application guarantees secure and systematic task management. The system allows users to create, assign, monitor and resolve tickets within a well-defined workflow. Administrators are pivotal in supervising the entire process, managing user permissions, assigning roles and tracking ticket statuses. Developers are tasked with addressing technical issues by working on tickets allocated to them, while Testers confirm fixes and ensure that issues are adequate resolved prior to closure. This distinct separation of roles not only fosters accountability but also enhances collaboration among team members. By incorporating role-based access control and real-time ticket tracking, the system minimizes communication delays and boosts overall team productivity.

Laravel's elegant syntax and powerful backend capabilities facilitate the scaling and maintenance of the system as project requirements change. Moreover, the user-friendly interface guarantees seamless navigation and usability for all users, irrespective of their technical expertise. The Ticket Raising System is an invaluable asset in the software development lifecycle, encouraging transparency, improving issue resolution times and supporting ongoing quality assurance within development teams.

VI. REFERENCES

1. Mostafa, M. (2021), *Mastering Laravel 8*. Packet Publishing.

2. Aglibar, K.D., Alegre, G.C., Cel Mundo, G., Duro, K.F. and Rodelas, N., 2022. *Ticketing System: A descriptive research on the use of ticketing system for project management and issue tracking in IT companies*.

3. Otwell, T. (2023). *Laravel: Up & Running (3rd ed.)*. O'Reily Media.