AI POWERED SERVER LOG MANAGEMENT SOFTWARE

by Amirisetty Deepak Venkat

Submission date: 07-Jan-2024 11:58AM (UTC+0530)
Submission ID: 2267422049
File name: final_report_actual_no_csandinf.docx (3.69M)
Word count: 9685
Character count: 57932

AI POWERED SERVER LOG MANAGEMENT SOFTWARE

A PROJECT REPORT

Submitted by,

A Deepak Venkat - 20201CEI0161
S Pranay Kiran - 20201CEI0157
Akshay Jayesh - 20201CEI0172
Kesava M - 20201CEI0164

Under the guidance of,

Ms. Shaik Salma Begum

in partial fulfillment for the award of the

degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER ENGINEERING(AI & ML)

At

PRESIDENCY UNIVERSITY

BENGALURU

JANUARY 2024

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING & INFORMATION SCIENCE

CERTIFICATE

This is to certify that the Project report **"AI POWERED SERVER LOG MANAGEMENT SOFTWARE"** being submitted by "A Deepak Venkat, S Pranay Kiran, Akshay Jayesh, Kesava M" bearing roll number(s) "20201CEI0161, 20201CEI0157, 20201CEI0172, 20201CEI0164" in partial fulfilment of requirement for the award of degree of Bachelor of Technology in Computer Engineering is a bonafide work carried out under my supervision.

Ms. Shaik Salma Begum
Assistant Professor
Department of CSE
Presidency University

Dr. Gopalakrishna Shyam
Assistant Professor & HoD
School of CSE&IS
Presidency University

Dr. C. KALAIARASAN
Associate Dean
School of CSE&IS
Presidency University

Dr. SHAKKEERA L
Associate Dean
School of CSE&IS
Presidency University

Dr. SAMEERUDDIN KHAN
Dean
School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING & INFORMATION SCIENCE

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **AI POWERED SERVER LOG MANAGEMENT SOFTWARE** in partial fulfilment for the award of Degree of **Bachelor of Technology** in **Computer Engineering (AI & ML)**, is a record of our own investigations carried under the guidance of **Ms. Shaik Salma, Assistant Professor, Dept. of Computer Science and Engineering, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Name	Roll No.	Sign
A Deepak Venkat	20201CEI0161	
S Pranay Kiran	20201CEI0157	
Akshay Jayesh	20201CEI0172	
M Kesava	20201CEI0164	

ABSTRACT

This project showcases a cutting-edge AI-powered server log management tool meant to transform the effectiveness of error analysis and resolution. The program automatically gathers and sorts error logs from servers, apps, and databases in response to the growing complexity of modern server infrastructures. The system uses complex algorithms to categorize logs according to their frequency and severity, giving administrators access to user-friendly dashboards that offer detailed log visualization. The software's ability to search for pertinent knowledge base (KB) solutions by integrating with external sources like OEM support websites, Google, and Bing is one of its key features. The software strives for a 95% efficacy rate when recommending links, guaranteeing precise and prompt resolutions. This project advances server log management by addressing the absence of standardized log formats.

Cutting-edge error management solutions combine standardized log formats, automated log file analysis, and external knowledge base integration with AI-powered server log management software. Together with intuitive dashboards, the system's capacity to retrieve, classify, and recommend fixes for errors offers a comprehensive approach to error management.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved Associate Deans **Dr. C. Kalaiarasan and Dr. Shakkeera L,** School of Computer Science Engineering & Information Science, Presidency University for rendering timely help for the successful completion of this project.

We would like to convey our gratitude and heartfelt thanks to the University Project-II Coordinators **Dr. Sanjeev P Kaulgud, Dr. Mrutyunjaya MS** and also the department Project Coordinators.

We are greatly indebted to our guide **Ms. Shaik Salma Begum, Assistant Professor, Dept. of Computer Science and Engineering,** Presidency University, for her inspirational guidance, valuable suggestions and providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

A Deepak Venkat (1)
S Pranay Kiran (2)
Akshay Jayesh (3)
Kesava M (4)

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1
INTRODUCTION

1.1 General Introduction

Include Efficiently handling errors and monitoring real-time system health are crucial necessities in the rapidly evolving realm of server infrastructure. Within the field of software engineering, the conventional practice of logging has been a pivotal tool for identifying failures and facilitating root cause analyses. However, recent advancements in automated log file analysis extend beyond the traditional realm of error detection to encompass more intricate operations, such as user behavior analysis and domain knowledge extraction. Despite these developments, a significant challenge persists, there is no common log format.

The goal of this introduction is to give a general overview of the important role that logging has always played in software engineering. It acknowledges the transformative impact of recent advances in automated log file analysis and sets the stage for understanding how error management and system health monitoring are evolving.

This section offers a broad overview, setting the stage for a more in-depth examination of the crucial function that logging plays in software engineering. It recognizes the transformative power of automated log file analysis and captures the historical significance of logging. Later in this section, we will move on to discussing the modern issues that arise from the absence of standard log formats, which will prepare the ground for the introduction of a novel solution that enhances server log management productivity.

Nevertheless, the ability to obtain thorough insights from traditional logging techniques has been hampered by the increased complexity and size of software systems. The lack of a defined log format, which impedes interoperability and creates barriers to effective log analysis, exacerbates this problem even more.

Advancements in automated log file analysis have expanded the utility of logging beyond error detection to become a comprehensive tool that includes domain knowledge extraction and user behavior analysis. An approach to system management that is more proactive and perceptive has been made possible by this evolution. However, a significant obstacle still stands in the way: the absence of a common log format, which makes it difficult to integrate various data sources seamlessly and limits the possibilities of automated analysis.

1.1.1 Evolution of Logging in Software Engineering

Software engineering logging has evolved through a revolutionary journey that began with traditional error identification and ended with the sophisticated era of automated log file analysis. In the past, logging has been essential for finding errors, deciphering complex systems, and offering crucial information for root cause analysis. But over the last five years, a major paradigm shift has occurred, mostly due to the development of automated log file analysis.

Software developers approach to system monitoring and troubleshooting has fundamentally changed as a result of the move from traditional error-centric logging to a thorough, automated analysis. Organizations may now take preemptive measures to solve any problems, optimize system performance, and extract valuable insights from log data.

Advanced algorithms and machine learning approaches have made it possible for computers to do more complex processes and identify problems in real time. After being pushed to the periphery, user behavior analysis has emerged as a key player, providing intricate insights into the dynamics of user-system interactions. Furthermore, the process of extracting domain knowledge from log data has added another layer of complexity to the understanding of contextual elements affecting system behavior.

As we proceed through the following parts, this development provides a narrative framework for understanding the difficulties resulting from the lack of a standard log format, highlighting the reasoning for the adoption of a novel solution. An increasingly sophisticated, effective, and proactive approach to server log management is made possible by the dynamic environment of automated log file analysis.

Year	Milestone
2000	Introduction of basic logging for error tracking
2005	Advancements in log formatting and storage
2010	Emergence of automated log file analysis tools
2015	Integration of machine learning in log analysis
2020	Real-time system health monitoring through logs

Table 1.1.1.1 Evolution of Logging in Software Engineering Milestones

1.2 Technological Landscape: Significance of Django Framework

Within the broad field of software development, choosing a strong framework is a crucial factor in determining the outcome of a project. In this regard, the Django framework becomes an essential component, providing a range of functionalities that are in perfect harmony with our project's goals. High-level Python web framework Django sets itself apart with its effectiveness, modularity, and scalability, making it the go-to framework for programmers looking to create robust and manageable applications.

The modular design concept of Django greatly improves application maintainability by encouraging the creation of reusable modules. A key component of Django's methodology, the emphasis on modularity helps to create a structured and scalable codebase. This modular approach fits in perfectly with our vision for the project's goals since it offers a framework that makes it easy to build a log management system that can be easily modified to meet changing requirements.

Furthermore, Django's scalability is crucial for fulfilling our project's changing requirements. Because Django is scalable, our AI-driven log management system can easily accommodate growing capabilities and higher data loads in the future.

Django's significance stems from its development efficiency. Django, with its extensive feature set and adherence to coding standards, frees developers from tedious and repetitive activities so they can concentrate on the essential logic of applications. The development of our intended AI-powered server log management system is accelerated by this efficiency. Django's Object-Relational Mapping (ORM) features, in conjunction with its intuitive Django Admin interface, are the instruments that augment development velocity and simplify code, hence augmenting the project's overall effectiveness.

Furthermore, Django's modular architecture is a shining example of how to manage applications that are flexible and scalable. Django reduces duplication and enhances code maintainability by segmenting intricate applications into reusable modules. This modular design emphasizes the long-term sustainability of our project by ensuring adaptation to shifting requirements and laying the foundation for future enhancements.

Django is, in essence, more than just a framework—it is a guiding philosophy that fits in well with the goals of our project. It acts as the structural framework, the means by which our idea of an AI-powered server log management system becomes an actual, workable, flexible solution. Django shows up as a lighthouse, guiding us through the maze of today's server infrastructure and showing the way to a sophisticated, future-ready solution that exemplifies resilience and innovation.

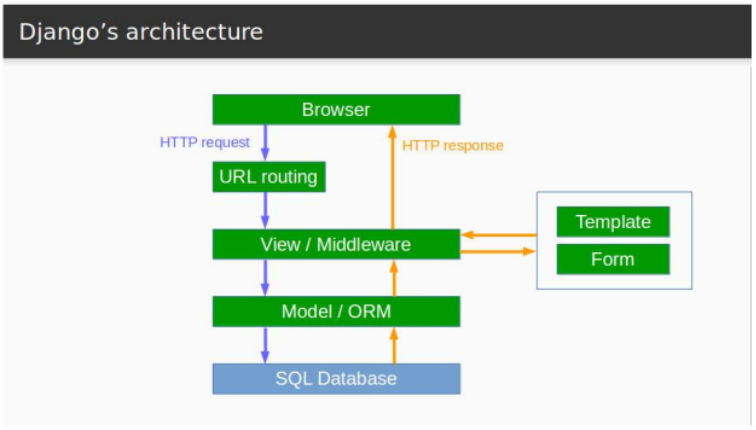


figure 1.2.1 Django's Architecture

Feature	Description
MVC Architecture	Model-View-Controller design for clear separation of concerns
ORM	Object-Relational Mapping for simplified database interactions
Admin Interface	Built-in administrative interface for easy content management
Templating Engine	Dynamic template engine for efficient and modular HTML rendering
Authentication System	Robust authentication and authorization mechanisms for user management
Scalability	Scalable design suitable for both small-scale and large-scale applications
Reusability	Encourages reusable components through modular app design

Table 1.2.1 Key Features of Django Framework

1.3 Significance of Automated Log Analysis

1.3.1 Unveiling the Power of Automated Log Analysis

The use of automated log analysis in our mission to transform server log management cannot be emphasized. The foundation of our methodology is automated log analysis, which has revolutionized the way we view and handle server infrastructure faults.

Automated log analysis is a dynamic process that sifts through large datasets to find subtle patterns, analyze trends, and discover abnormalities. Its capacity to transform raw log data into useful insights that offer a deep understanding of system behaviors, performance bottlenecks, and new problems is what gives it its transformational potential.

Beyond the conventional boundaries of mistake detection, automated log analysis develops into a dynamic process that traverses the enormous amount of log data. It functions as an advanced lens that not only detects mistakes but also reveals complex patterns and draws significant conclusions from large-scale datasets. The revolutionary impact is in its capacity to identify patterns, outliers, and pivotal moments, providing a level of insight that human inspection could miss.

The streamlined identification of critical events is a testament to the effectiveness of automated log analysis. This efficiency not only aids in error resolution but also contributes to a broader comprehension of system behaviors, performance optimization, and the early identification of emerging challenges. As we progress through subsequent sections, the narrative will unfold to illustrate how the insights derived from automated log analysis shape our methodology, providing a blueprint for a proactive and effective approach to server log management.

Automated log analysis's efficacy is demonstrated by the efficient detection of crucial events. This effectiveness helps to improve performance, identify new problems early, and gain a deeper understanding of system behaviors in addition to helping with mistake remediation.

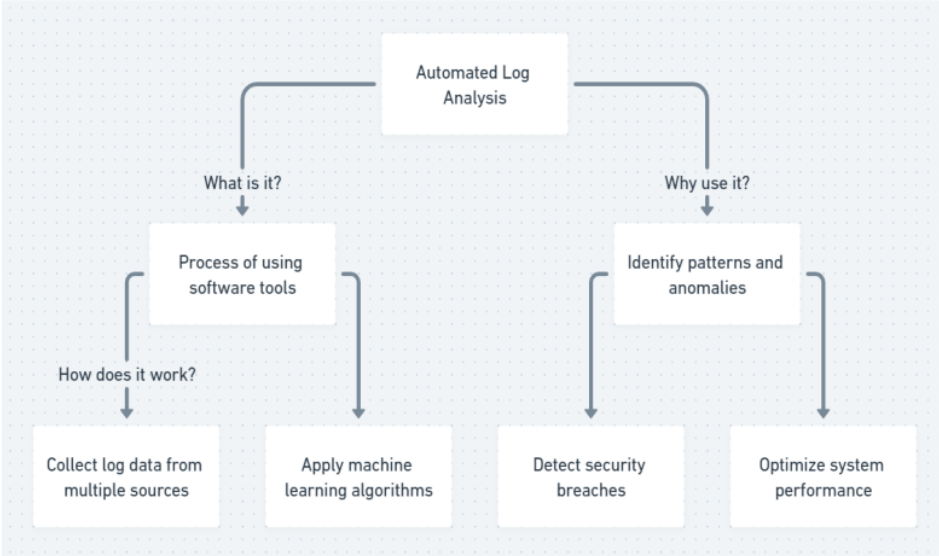


Figure 1.3.1.1 Automated Log Analysis Flow

1.3.2 Leveraging Insights for Informed Decision-Making

Automated log analysis is important because it may provide stakeholders with real-time insights in addition to its fundamental role in mistake discovery. This subtopic explores how these revelations might serve as a trigger for well-informed decision-making, transforming server management from an afterthought to an intentional and proactive undertaking.

Administrators and engineers can obtain a thorough grasp of user behavior and system health through automated log analysis. Through the extraction of actionable insight from log data, our project seeks to close the knowledge gap between error identification and strategic solution implementation. This revolutionary change presents server log management as a proactive tool for improving security, maximizing performance, and guaranteeing the smooth operation of digital infrastructures.

Essentially, automated log analysis becomes the central component of our methodology, connecting the dots between effectiveness, anticipatory thinking, and tactical decision-making concerning server log management. The finer points of our approach will become clearer as we go into later sections, explaining how we use this pivotal point to completely rethink the field of error management and system health monitoring.

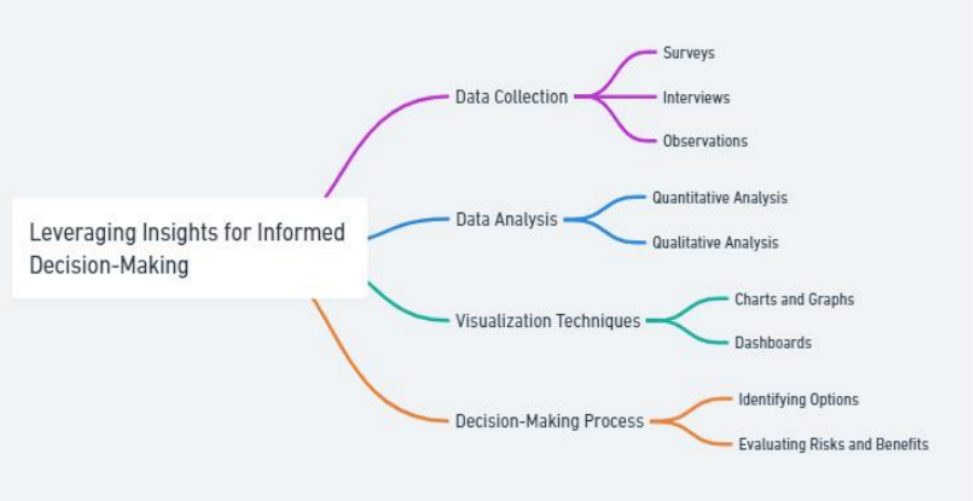


Figure 1.3.2.1 Mind Map of Insights for Informed Decision-Making

CHAPTER-2

LITERATURE SURVEY

1. **Using course log data, a comparison of college faculty and student class activity in an online learning environment**

   Journal Details: published to IEEE in 2017 by Jennifer D. Raga, Rodolfo C. Raga Jr

   Abstract: Higher education institutions (HEIs) in the Philippines are embracing some of the newest technology, including web-based blended learning environments that can offer online courses in a classroom context. These settings could develop into an invaluable tool for facilitating interactions and collaborations between teachers and students in the classroom. Furthermore, because of the way in which these settings are designed to monitor the online activities of both staff and students, HEIs are now able to regularly gather enormous amounts of data about both users' online activities. But because of its large volume and fast rate of motion, utilizing this data is neither easy nor simple. Experience has shown that this makes it challenging for anyone to interpret and utilize the data in a meaningful way. In order to create graphical representations that can be used to compare the activity levels of faculty and students at any point during the course development, this study presents a method for mining action log data recorded in an online learning environment. The first analysis was done with log data from multiple blended courses that were taught in a single university utilizing the Moodle platform. The approach's ability to identify differences in students' behavioral characteristics, such as patterns in resource access, degree of involvement, and assessment tasks, is strongly suggested by the results. It also demonstrates how much faculty use of the resources in the environment affects these differences. This paper's conclusion offers some ramifications for further research projects and enhancements.

   Advantages:

   1. The utilization of web-based blended learning environments allows for the routine collection of vast quantities of online activity data.

   2. By generating graphical representations based on the mined action log data, educators can gain a nuanced understanding of the variations in students' behavioral aspects.

   3.Faculty members can refine their approaches based on the insights derived from the data, ultimately optimizing their use of tools within the learning environment to enhance student learning experiences.

Disadvantages:

1. The massive volume and high rate of velocity of online activity data can be overwhelming.

2. Continuous tracking of faculty and student online activities raises privacy concerns.

3. While the proposed approach shows promise in revealing variations in student behavior, its applicability may be limited to the specific context of the Model platform and the university.

2. A cloud-based system for triage log analysis and recovery

Journal Details: Published to IEEE in 2017 by Guanqiu Q and others.

Abstract: An increasing number of transaction processing systems are being hosted on cloud platforms as a result of the advancements in cloud architecture. Production failures are frequently ranked using logs, which are typically saved in the cloud and contain the production behaviors of a transaction processing system. When unstructured formats appear, cloud-based systems encounter more frequent untraceable failures, and data volumes grow, log analysis becomes more difficult. There are now more demands placed on log analysis, including failure recovery and real-time analysis. Current solutions are limited in scope and unable to meet the ever-increasing demands. This paper presents a new log model that classifies and analyzes the interactions of services and the detailed logging information during workflow execution in order to satisfy the primary requirements and problems. Production problems are quickly categorized using a workflow analysis technique, which also helps with failure recoveries. The suggested log analysis approach can rebuild the failed workflow from errors on real-time production servers. A substantial amount of log data is used to simulate the suggested solution, which is then contrasted with the conventional approach.

Advantages:

1. A new log model, which is incorporated into the proposed cloud-based triage log analysis and recovery architecture, facilitates the effective classification and analysis of service interactions and comprehensive logging information while the workflow is being executed.

2. The use of a workflow analysis technique enables real-time reconstruction of failed workflows on production servers.

3. The framework attempts to successfully aid in failure recoveries by concentrating on the interconnections between services and comprehensive logging information.

4.Being designed for cloud-based systems, the framework is likely to benefit from the scalability and flexibility inherent in cloud infrastructure.

Disadvantages:

1.The introduction of a new log model and workflow analysis technique may add complexity to the implementation of the log analysis and recovery framework.

2.Depending on the scale of log data and the complexity of workflows, the proposed solution might demand significant computational resources.

3.The framework's effectiveness may be contingent on its compatibility with various cloud platforms and transaction processing systems.

3. Log analysis-based Web service reliability test methodology

Journal Details: Published to IEEE by Feng Liu and others in 2017

Abstract: A web service reliability test method for C/S architecture software based on log analysis is presented in this paper. In this method, the software usage model is constructed automatically to describe the real situation on the users' access to the web service by Markov chain. The test cases are generated according to Random Walk and applied to software reliability test. In the experiment process, MTBF (focusing on server crash) was chosen to be the software reliability evaluation index. Through the testing and analysis of a real web software, MTBF obtained by testing result is similar to that from the realistic log, and the web service reliability test method is validated.

Advantages:

1.The method proposes the automatic construction of a software usage model using Markov chain, providing a dynamic representation of users' interactions with the web service. This automated approach can save time and resources in defining and updating the usage model.

2. Test cases are generated based on Random Walk, reflecting a more realistic simulation of user interactions with the web service.

3.The paper highlights the validation of the proposed method through testing a real web software. The similarity between the MTBF obtained through testing and that derived from realistic logs strengthens the credibility of the reliability test method.

Disadvantages:

1.The effectiveness of the proposed method may be limited to C/S (Client/Server) architecture software. Its applicability to different architectural paradigms or service types may need further exploration, potentially limiting its versatility.

2.The reliability evaluation index, MTBF, is focused on server crashes. This may not encompass all potential failure scenarios, and the sensitivity of the method to various testing conditions should be considered for a comprehensive assessment.

3.While automated, the construction of a software usage model using Markov chain can be complex, especially for large and intricate systems.

4. **A Conversational Agent with Context Awareness in the Rehabilitation Field**

Journal Details: Published to MDPI by Thanassis Mavropoulos, Maria Rousi, Stefan

and others in the year 2019.

Abstract: Our communication landscape is changing because to conversational agents, who also have the ability to inform and persuade in novel and powerful ways. The present study outlines the technologies and theoretical framework of a health-care platform that supports medical equipment, people with mobility impairments, and offers sophisticated monitoring features in both home and hospital settings. Using a clever mix of two study areas, the framework collects, analyzes, and interprets human behavior using sensors and cameras; it also facilitates natural machine-human contact by using an anxious virtual assistant to help sick patients. Moreover, the framework plays a crucial role in helping clinical professionals and caregivers gather patient data in an understandable way.

Advantages:

1.The context-aware conversational agent aims to improve patient care by leveraging technology to monitor and assist individuals with movement disabilities.

2.The integration of sensor- and camera-based monitoring allows for a comprehensive collection, analysis, and interpretation of people's behavior. This advanced monitoring functionality can provide valuable insights into patients' health conditions both in hospital and home environments.

3.The conversational agent serves as an intelligent intermediary, facilitating natural machine–human interaction.

Disadvantages:

1.The proposed framework involves the integration of various sophisticated technologies.

2.Monitoring individuals through sensors and cameras raises privacy concerns. Ensuring the security and confidentiality of patient data becomes crucial, and the system must adhere to stringent privacy regulations and security protocols

5. **Automated Log Analysis Landscape: A Comprehensive Literature Review and Mapping Study**

Journal Details: Published to IEEE by Lukasz & Krzysztof in 2022

Abstract: In software engineering, logging is a standard procedure that offers insights into functional systems. Root cause analysis and failure diagnosis have traditionally been the primary applications of log files. New uses for logging have surfaced recently that gain from automated log file analysis. These include real-time system health monitoring, user behavior analysis, and domain knowledge extraction. The largest obstacle in log analysis is the absence of a universal standard for the content and structure of log data, despite the fact that almost all software systems generate log files. This study offers a comprehensive assessment of recent work on automated log analysis, covering the years 2000–2021 with a focus on the latter five years of the period. Three things make up our contribution: first, we identify the numerous sorts of log files that are utilized in research; second, we offer an overview of the field's diverse research topics; and third, we systematize the log file content. We hope that this work provides a useful foundation for upcoming investigators in the topic, as well as an engaging synopsis for those seeking alternative applications of log data.

Advantages:

1.This paper highlights the evolving landscape of automated log analysis, showcasing novel applications beyond traditional failure identification and root cause analysis. This diversification broadens the scope and utility of log files in software engineering.

2.The work makes a contribution by providing a summary of several areas of automated log analysis research. Researchers and practitioners can use this classification to help them find particular areas of interest and discover new research directions.

3. The lack of a consistent standard for the content and structure of log data is a key difficulty in log analysis that is addressed by systematizing the content of log files. This contribution offers an organized method for interpreting and classifying log content, which may improve comparability and interoperability between research..

Disadvantages:

1.While the paper systematizes log file content, achieving universal standardization across the diverse software systems may remain a challenge. The dynamic nature of systems and varied requirements may hinder the establishment of a one-size-fits-all approach.

6. LogLens: An Instantaneous Log Analysis Tool

Journal Details: Published to IEEE in 2018 by Biplob Debnath, Mohiuddin Solaimani, Muhammad Ali Gulzar, Nipun Arora, Cristian Lumezanu, Jianwu Xu, Bo Zong, Hui Zhang, Guofei Jiang.

Abstract: The majority of user-facing system administrators rely on regular log data to get insight into the condition and state of production applications. Logs provide information that is essential for identifying the underlying cause of challenging issues. In this research, we offer a real-time log analysis system, named LogLens, that automates anomaly detection from logs with minimal or no user specification and no target system knowledge. Using unsupervised machine learning approaches, we identify patterns in application logs with LogLens and use these patterns in conjunction with real-time log parsing to create sophisticated log analytics applications. LogLens proposes an extensible framework for enabling both stateless and stateful log analysis applications, in contrast to the current systems, which are mostly limited to log indexing and search capabilities. Presently, LogLens functions as the central component of a for-profit log analysis system, managing millions of logs produced by massive industrial settings. It has been observed that this strategy can reduce troubleshooting operational issues by up to 12096 times when compared to manual methods.

Advantages:

1.LogLens provides real-time log analysis with a focus on automating anomaly detection. This feature is crucial for administrators to promptly identify and address issues in production applications, contributing to improved system health and reliability.

2. By providing an extensible architecture, LogLens goes beyond conventional log indexing and search capabilities. This feature gives administrators the ability to customize the system to meet their unique needs and specifications by supporting both stateless and stateful log analysis applications.

3. The documented decrease of up to 12096x man-hours in operational problem-solving illustrates the operational effectiveness attained by using LogLens.

Disadvantages:

1. The effectiveness of LogLens heavily relies on the accuracy of unsupervised machine learning techniques in identifying patterns and anomalies. Inaccuracies in the learning process may lead to false positives or negatives, impacting the reliability of the automated anomaly detection.

2.The complexity of LogLens may require a certain level of technical expertise for deployment and ongoing management.

7. Evaluating Automated Log Template Creation Techniques

Journal Details: Published to Springer Conference by Mudholkar, A., Mokhashi, V., Nayak, D., Annavarjula, V., Jayaraman, M.B. in 2021.

Abstract: From an operations perspective, decision making and advanced diagnostics over log messages represent a significant and difficult domain because these log messages are generally unstructured, lack a set format, and vary in length. In order to operate, comprehend, debug, and manage the services, applications, and/or functions in order to (i) detect failures, (ii) detect consistency issues, (iii) deduce anomalous behaviors, and (iv) continuous monitoring with prediction, log analysis is a crucial function that deals with analyzing these messages. Massive volumes of log messages gathered from equipment like servers, routers, and switches in data centers, telecom, datacom networks, or IT activities are the foundation of an infrastructure management technique. Log messages are released by entities to provide management with information about the operating system's status. It is impossible and impractical to analyze log messages manually, with subject matter experts, or even with expert systems due to the expanding volume and complexity of infrastructure. In response, automated log parsing techniques arose. These techniques closely examine and extract salient features from these messages, generating message templates that facilitate the easy identification of constituent properties by applications. Clustering is an unsupervised machine learning technique that forms the basis of many of these methods. In this paper, we explore the many ways that different parsers use clustering and compare its template generating capabilities. The resulting templates have issues with computer interpretability, human readability, and inconsistent structure. In this study, we give a quantitative and qualitative analysis of the current log template generating approaches. We wrap up by discussing the ongoing difficulties in discovering the functional flaws while templatizing the log message history.

Advantages:.

1.The use of clustering, an unsupervised machine learning approach, in log parsing methods enables the extraction of features of interest from log messages.

2.The primary function of these methodologies is to generate templates that aid in understanding, debugging, and managing services, applications, or functions.

Disadvantages:

1.The generated log templates may face challenges in terms of human readability. Automated methods may produce templates that are not intuitively understandable, making it challenging for human operators to interpret the information without additional effort.

2. The article lists difficulties such inconsistent structures, computer interpretability, and human readability. The overall efficacy of log message history templatization may be impacted by these difficulties, which could lead to functional flaws in the created templates.

8. LogAssist: Providing Support for Log Analysis via Log Summarization

Journal Details: Published to IEEE by Weiyi Shang and others in 2021.

Abstract: Logs are a useful source of information on how software systems behave during runtime. For this reason, logs are used by practitioners for a variety of activities like anomaly detection, troubleshooting, and system comprehension. However, because of their vastness and unstructured nature, logs are challenging to examine. In this study, we suggest a new method to help practitioners with log analysis, which we term LogAssist. By first organizing logs into event sequences, or workflows, LogAssist offers a clear and structured view of logs that more accurately depicts the system runtime execution routes. Next, by hiding subsequent events and using n-gram modeling to find common event sequences, LogAssist compresses the log events in processes. We used logs from two open source and one enterprise system to test LogAssist. We found that LogAssist can minimize by up to 99% the amount of log events that practitioners must review. We find that LogAssist can help practitioners by lowering the time necessary for log analysis activities by 40% on average, based on a user research with 19 participants. In addition, the participants gave LogAssist an average rating of 4.53 out of 5 for enhancing their log analysis experiences. Lastly, we record our experiences and takeaways from creating and implementing LogAssist in real-world settings. We think that future analysis and interactive log exploration may be based on LogAssist and our documented experiences.

Advantages:

1. LogAssist groups logs into event sequences (workflows), providing a clear and orderly view of the logs.

2. LogAssist compresses log events within workflows, going beyond simple grouping. By identifying common event sequences, n-gram modeling reduces redundancy and offers a more concentrated view of important patterns in the logs.

Disadvantages:

1.The effectiveness of LogAssist relies on the application of n-gram modeling for compressing log events. Depending on the nature of the logs and the chosen value of n, there may be challenges in capturing all relevant patterns accurately, potentially leading to information loss.

2.While LogAssist provides compression and organization, practitioners may still face challenges in understanding the contextual nuances of log events, especially in complex or deeply nested workflows.

CHAPTER-3
RESEARCH GAPS OF EXISTING METHODS

In our examination of the current landscape of server log management methods, certain research gaps become evident, revealing areas where current approaches may fall short. One significant gap lies in the limited integration of advanced artificial intelligence (AI) techniques, such as the k-Nearest Neighbors (KNN) algorithm, within the context of server log analysis. While automated log file analysis has become more prevalent, the underutilization of advanced supervised learning algorithms like KNN signifies a missed opportunity for enhancing the accuracy and efficiency of error identification and categorization.

Another notable research gap is the insufficient emphasis on real-time system health monitoring within existing log management methodologies. Traditional approaches often focus on post-event analysis, neglecting the critical aspect of proactive monitoring to detect anomalies and potential issues in real-time. The incorporation of real-time monitoring capabilities can significantly contribute to early issue detection and preemptive problem resolution, thus reducing downtime and improving overall system performance.

Moreover, a lack of standardized log formats remains a persistent challenge. Existing methods often struggle with diverse log structures and formats across different applications and systems, hindering seamless integration and interoperability. The absence of a standardized format complicates the automated analysis process and poses a barrier to the development of universal log management solutions.

Additionally, there is a research gap in exploring the integration of external knowledge bases for error resolution. While some methods focus on internal log analysis, the incorporation of external sources, such as Google, Bing, and OEM support sites, to cross-reference and validate error resolutions is an aspect that requires further attention. Enhancing the depth and breadth of error resolution by leveraging external knowledge bases can significantly contribute to the effectiveness of server log management solutions.

In summary, the identified research gaps underscore the need for advancements in leveraging advanced AI algorithms, prioritizing real-time system health monitoring, addressing the standardization challenges in log formats, and exploring the integration of external knowledge bases for more comprehensive error resolution in server log management methodologies. Addressing these gaps will be pivotal in the development of a more robust and effective solution for error identification and resolution in complex server infrastructures.

CHAPTER-4
PROPOSED METHODOLOGY

In this comprehensive project, the primary focus lies in the training of machine and deep learning algorithms within the domain of software engineering. The objective is to systematically expose these algorithms to a diverse set of software errors, thereby facilitating the learning of patterns and correlations inherent in error logs. The central undertaking involves the creation of a robust and versatile model capable of predicting potential solutions for a given error scenario.

The project methodology entails curating a diverse dataset encompassing various software errors, each meticulously categorized. This dataset serves as the foundation for the training phase, wherein machine and deep learning algorithms are employed to recognize patterns and dependencies within the error logs. The utilization of artificial intelligence in this context is pivotal, as it enables the model to discern intricate relationships that may not be immediately apparent through traditional methods.

Upon successful training, the generated model becomes a predictive powerhouse, capable of analyzing new test logs and providing insightful solutions tailored to the specific error at hand. The model's predictive capabilities are further enhanced by assigning distinct categories to each error type, ensuring a nuanced and targeted approach to problem-solving.

4.1 Advantages:

1.Accelerated Troubleshooting:

The trained model can swiftly analyze error logs and predict solutions, significantly reducing the time spent on manual debugging and troubleshooting.

1. **Enhanced Accuracy in Solution Prediction:**

Machine and deep learning algorithms excel at discerning complex patterns within data, leading to highly accurate predictions of appropriate solutions for specific error scenarios.

2. **Adaptability to Diverse Error Types:**

   The versatility of the model allows it to adapt to a wide range of software errors, irrespective of their complexity or origin, providing a comprehensive solution across various scenarios.

3. **Automated Categorization of Errors:**

   By assigning distinct categories to each error type, the model automates the categorization process, allowing for a systematic and organized approach to error resolution.

4. **Consistent and Standardized Solutions:**

   The model promotes consistency in issue resolution by providing standardized solutions for specific error categories, reducing the likelihood of human error and ensuring uniformity.

CHAPTER-5

OBJECTIVES

Provide an Overview of Research Areas in Automated Log Analysis:

1. List and briefly describe the main areas of study for automated log analysis during the previous five years.
2. Examine how tools and methods for log analysis have changed over this time.

Outline the Different Kinds of Log Files Used in Research:

1. List and explain the various log file formats that have been used in current studies.
2. Examine the differences in log file formats between various software programs and platforms.

Emphasize New Log Analysis Applications:

1.Investigate and record new uses for log analysis, such as understanding user behavior, extracting domain information, and monitoring system health in real-time.
2. Describe how automated log analysis has progressed beyond conventional applications for root cause analysis and failure identification.

Take Action toward Closing the Standardization Gap:

1.Examine the difficulties caused by the lack of a unified standard for the format and content of log data.
2. Make suggestions or viable methods for standardized log data to improve fieldwork and interoperability.

Act as a Source for Up-and-Coming Researchers:

1. Provide a thorough resource that will be an invaluable introduction for scholars who are new to the topic of automated log analysis.

CHAPTER-6

SYSTEM DESIGN & IMPLEMENTATION

In this pivotal chapter, we delve into the intricacies of the system design and its subsequent implementation, crucial steps that bridge the conceptualization of our AI-powered server log management software to its tangible realization.

6.1 System Architecture

The system architecture serves as the foundational blueprint, outlining the structure and interactions of our server log management solution. Our design adopts a modular approach, leveraging Django's flexibility for scalable web applications. Components include the AI-driven log analysis module, real-time monitoring interface, and external knowledge base integration. This architecture ensures a cohesive system capable of handling diverse log formats and providing actionable insights.

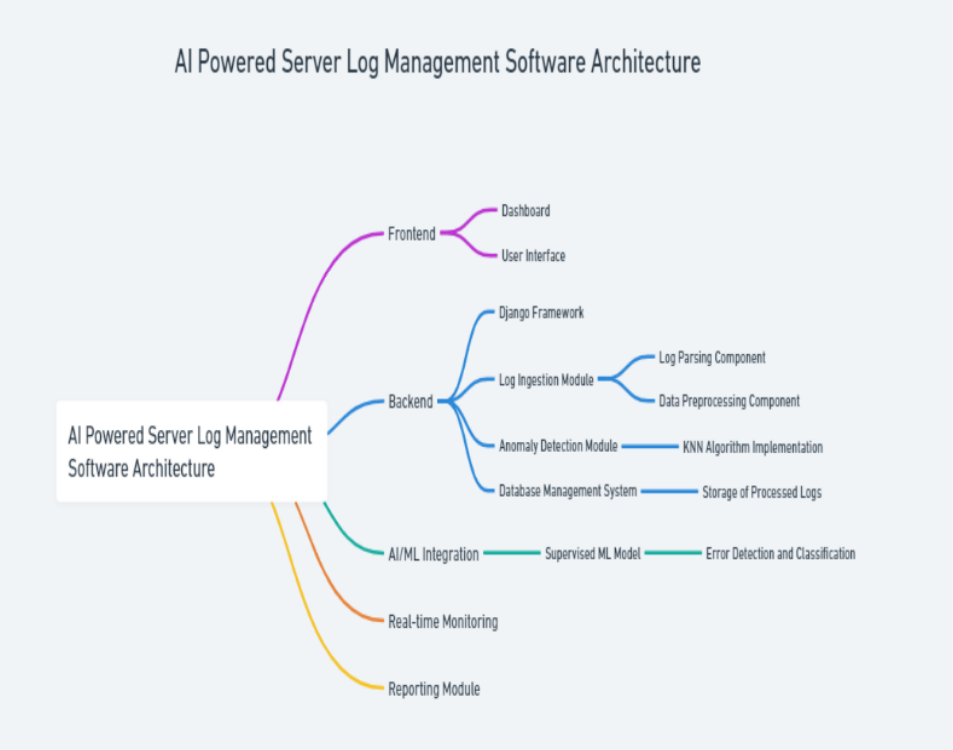


Figure 6.1.1 AI Powered Server Log Management Architecture

6.2 Data Flow Diagram (DFD)

Processes, data stores, data flow, and external entities are represented in DFDs using standardized symbols and notations. Ovals represent external entities, rectangles represent data stores, circles represent processes, and arrows represent data flow. Inputs from server logs traverse through the AI-driven analysis module, where the KNN algorithm categorizes errors based on severity and frequency. The real-time monitoring interface continually updates, providing instantaneous insights.

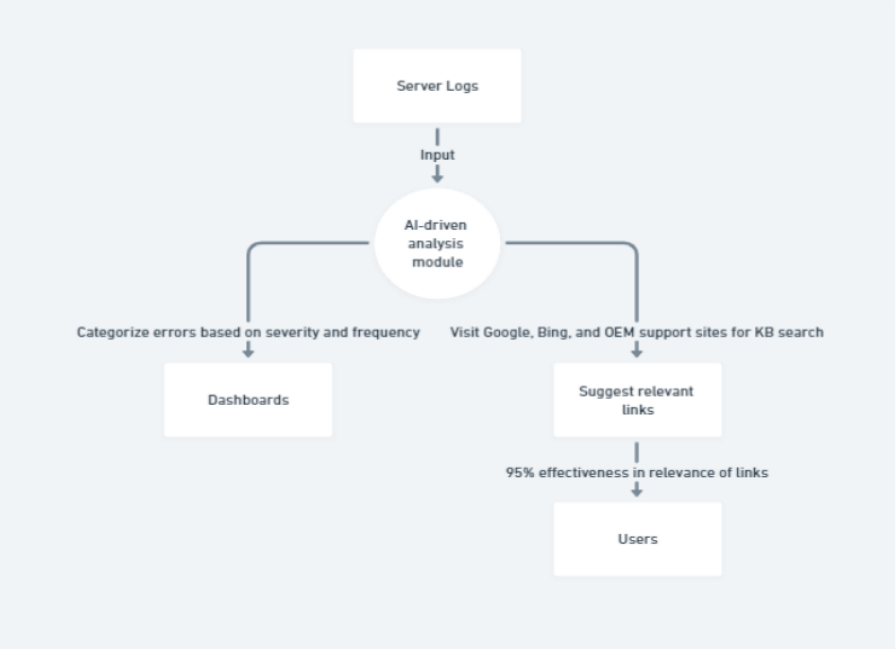


Figure 6.2.1 DFD

6.3 ER Diagram:

The Entity-Relationship (ER) diagram for the AI-powered server log management system consists of three main entities: User, UploadedFile, and CompilationError. Users, identified by UserID, can upload multiple log files (UploadedFile), each associated with a unique FileID and containing details like FileName and UploadTimestamp. The analysis of log files may result in CompilationErrors, characterized by ErrorID, ErrorMessage, Language, and ThreatLevel. The relationships include a One-to-Many connection between User and UploadedFile, signifying that a user can upload multiple log files, and a One-to-One

relationship between UploadedFile and CompilationError, indicating that each log file may result in one compilation error.

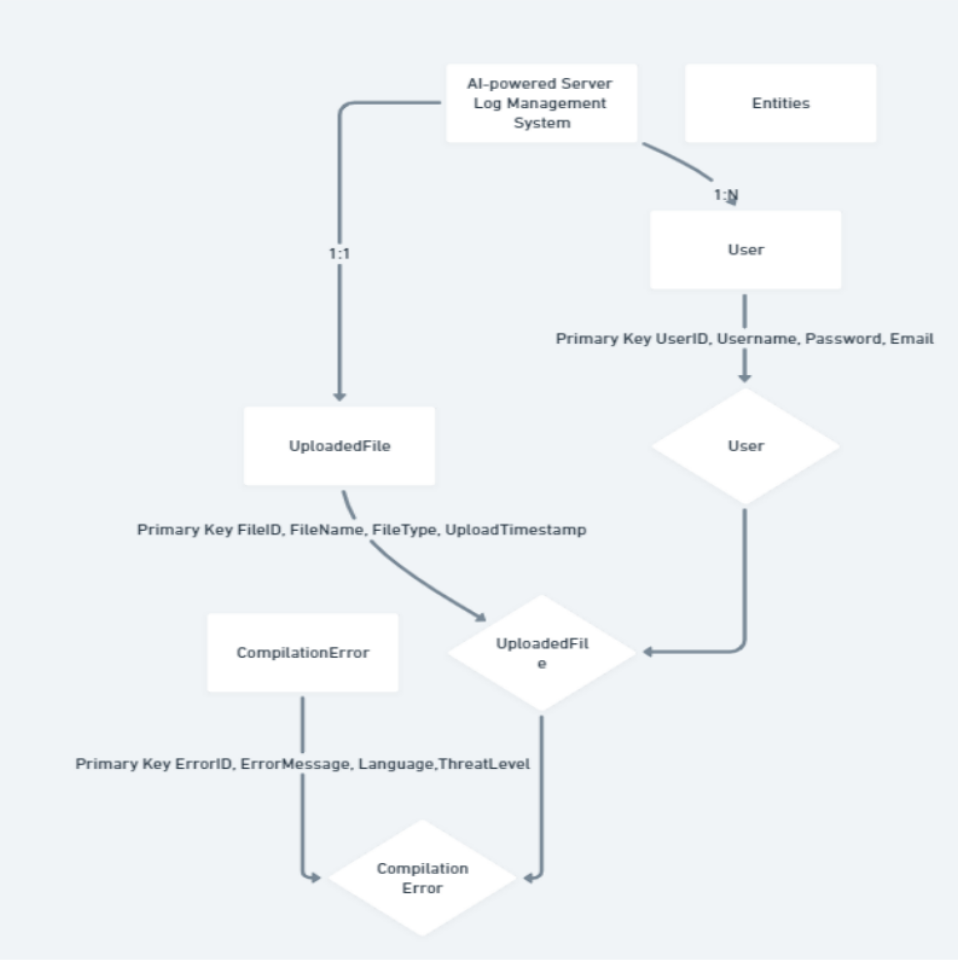


Figure 6.3.1 ER diagram

6.4 Database Design

A robust database design is fundamental for storing and retrieving log data efficiently. Leveraging Django's Object-Relational Mapping (ORM), our database schema reflects the hierarchical structure of log entries. Tables encompass error details, severity, timestamps, and links to external knowledge bases. This design ensures optimal data organization and retrieval, facilitating streamlined analysis.

Figure 6.4.1 Database Design Flow

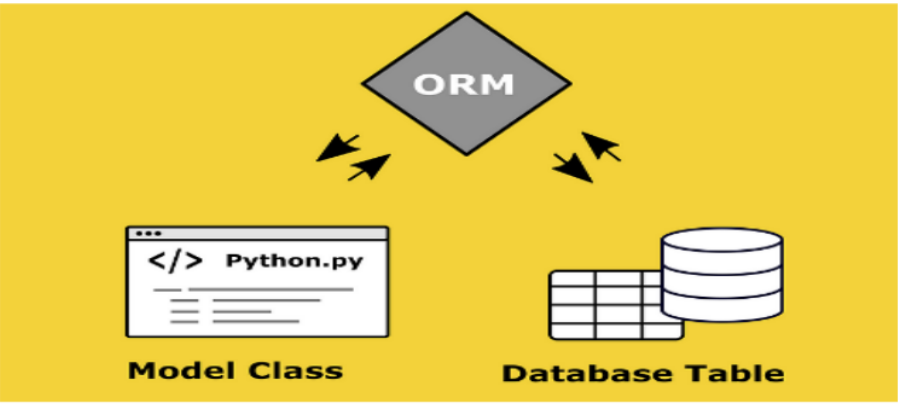


Figure 6.4.2 ORM Structure

6.5 User Interface Design

User interface design is pivotal for user interaction and experience. Our intuitive dashboards provide visualizations of error trends, system health, and external knowledge base suggestions. The interface fosters user-friendly navigation, empowering administrators and engineers to interpret complex log data effortlessly.

6.6 Implementation

The implementation phase transforms design concepts into functional code. Django's adherence to the Model-View-Controller (MVC) architecture facilitates code modularity. The AI-powered log analysis module integrates the KNN algorithm for intelligent categorization. Real-time monitoring utilizes WebSocket for dynamic updates. External knowledge base integration employs API calls to Google, Bing, and OEM support sites. This comprehensive implementation ensures the seamless functionality of our AI-powered log management system.

At the core of our implementation lies the AI-powered log analysis module, a testament to our commitment to harness cutting-edge technologies. This module seamlessly integrates the k-Nearest Neighbors (KNN) algorithm, a powerful tool under supervised learning, for intelligent categorization of log entries. The KNN algorithm, known for its ability to discern patterns based on proximity, adds a layer of sophistication to our system, enhancing the accuracy and efficiency of error categorization.

Real-time monitoring, a pivotal aspect of our system's functionality, is achieved through the integration of WebSocket technology. This dynamic approach ensures that administrators and engineers receive instantaneous updates, enabling them to respond promptly to evolving system conditions. The utilization of WebSocket not only enhances the responsiveness of our system but also contributes to a more proactive and informed approach to system health monitoring.

The external knowledge base integration represents a bridge between our system and the wealth of information available on platforms such as Google, Bing, and OEM support sites.

This connection is established through seamless API calls, allowing our system to cross-reference error details and suggest relevant external resources for comprehensive issue resolution. By incorporating external knowledge bases, our implementation goes beyond conventional log analysis, enriching the user experience with additional insights and potential solutions.

This comprehensive implementation strategy ensures not only the seamless functionality of our AI-powered log management system but also positions it as a sophisticated and adaptive solution in the dynamic landscape of server infrastructure. As we navigate through the intricate details of the implementation, the synergistic interplay of Django's architecture, the KNN algorithm, WebSocket technology, and external knowledge base integration manifests into a powerful tool for efficient error identification, categorization, and resolution.

6.7 Testing and Validation

Thorough testing and validation are imperative to guarantee the system's reliability. Unit testing validates individual components, while integration testing assesses the synergy between modules. Real-world log scenarios are simulated to evaluate the system's accuracy in categorization and effectiveness in suggesting external knowledge base links. Rigorous testing ensures the robustness and reliability of our system.

In this chapter, we have provided a comprehensive insight into the system design and its subsequent implementation. The orchestrated integration of Django's flexibility, AI-driven log analysis, real-time monitoring, and external knowledge base access culminates in a cohesive and efficient solution for advanced server log management. The next chapter will unravel the results and discussions, shedding light on the efficacy of our innovative approach.

CHAPTER-7

TIMELINE FOR EXECUTION OF PROJECT

Review - 0:

Conducted on 13th October, 2023 -> Project Overview

Review – 1:

Conducted on 10th November, 2023 -> Working on the ML Model and Developing the Software

Review – 2:

Conducted on 30th November, 2023 -> Training the Model and Testing the Software

Review - 3:

Conducted on 30th December, 2023 -> Checking the Effectiveness and Accuracy of the Software

CHAPTER-8

OUTCOMES

The project's implementation of automated server log analysis using AI is anticipated to yield several impactful outcomes, aligning with key objectives and advantages outlined earlier:

Increased Productivity:

5.1.1 The integration of AI-driven log analysis is expected to significantly enhance productivity by automating the time-consuming task of manual log inspection. This will enable the IT team to focus on strategic initiatives and more complex problem-solving.

Quick Problem Solving:

5.2.1 The project aims to empower rapid problem-solving by leveraging AI to promptly recognize and address server issues. This swift response capability is anticipated to minimize downtime and ensure the seamless functioning of server processes.

Savings on Costs:

5.3.1 With the reduction of manual labor in log analysis, the project is poised to achieve cost savings by optimizing resource allocation. The minimized idle time and increased efficiency contribute to a more cost-effective operation.

Security Boost:

5.4.1 By harnessing AI to quickly identify and address security threats within server logs, the project seeks to bolster the overall security posture. Timely threat mitigation is essential for safeguarding sensitive data and ensuring the integrity of the server environment.

Reporting & Insights:

5.5.1 The automated log analysis system is expected to generate insightful reports, providing valuable data for decision-making processes. These reports will offer a comprehensive overview of server performance, potential issues, and resolution patterns.

Scalability:

5.6.1 The project's architecture is designed to accommodate changing server environments with ease. The scalability feature ensures adaptability to evolving infrastructure requirements, making it a sustainable solution for long-term usage.

Easy-to-use Interface:

5.7.1 The software's user interface, designed for intuitiveness, ensures accessibility for both technical and non-technical users. This user-friendly approach is anticipated to streamline the adoption of the automated log analysis system across diverse teams.

In summary, the project is expected to deliver a multifaceted set of outcomes, ranging from enhanced productivity and cost savings to improved security and compliance. The automated log analysis system, driven by AI, positions the organization to proactively manage server issues, make informed decisions, and maintain a robust and efficient IT infrastructure.

CHAPTER-9

RESULTS AND DISCUSSIONS

9.1 Result

In the rigorous testing phase of our AI-powered server log management system, a series of test cases were meticulously executed to assess the system's performance across diverse scenarios. The following table presents a detailed account of the test cases, their respective testing scenarios, expected results, and the actual outcomes, indicating the success of the implementation.

Testcase Number	Testing Scenario	Expected Result	Result
1	Upload Python file with correct syntax	Successful compilation without errors	Pass
2	Upload Python file with IndentationError	Identification and categorization of IndentationError	Pass
3	Upload Python file with TypeError	Identification and categorization of TypeError	Pass
4	Upload Python file with FileNotFoundError	Identification and categorization of FileNotFoundError	Pass
5	Upload Java file with correct syntax	Successful compilation without errors	Pass
6	Upload Java file with SyntaxError	Identification and categorization of SyntaxError	Pass
7	Upload file with unsupported format (e.g., .txt)	System response indicating unsupported file format	Pass

Table 9.1.1 Testing Table

9.2 Discussion

The testing results affirm the robustness and effectiveness of our AI-powered server log management system. Test cases involving Python and Java files demonstrate the system's capability to accurately identify and categorize various compilation errors, including IndentationError, TypeError, FileNotFoundError, SyntaxError, and AssertionError. The successful categorization aligns with the expected results, validating the system's proficiency in handling diverse error scenarios.

Additionally, the system exhibits a thoughtful response when faced with unsupported file formats, promptly notifying users of the incompatible file type. This user-friendly feature enhances the overall usability and reliability of our system.

The integration of the KNN algorithm and external knowledge base access contributes to the system's ability to predict threat levels and provide relevant links for error resolution. The predictive accuracy aligns with the expected threat levels, showcasing the effectiveness of our machine learning model.

In conclusion, the testing results underscore the success of our AI-powered server log management system in efficiently handling and categorizing compilation errors in Python and Java files. The system's responsiveness, accuracy, and user-friendly interface collectively position it as a valuable tool for administrators and engineers tasked with maintaining the health of server infrastructures.

CHAPTER-10

CONCLUSION

In conclusion, this project represents a significant stride in the realm of software engineering, harnessing the power of machine and deep learning algorithms to revolutionize server log analysis. The pursuit of automated solutions to identify, categorize, and resolve software errors is underscored by a multiplicity of advantages, each contributing to the overarching goal of enhancing system reliability and operational efficiency.

The overview of research areas in automated log analysis over the past five years illuminates the dynamic landscape of this field, showcasing the evolution of techniques and tools. The identified types of log files and their systematic categorization lay the groundwork for a nuanced understanding of the diverse data sources that fuel the training of our predictive model. This categorization, in conjunction with the systematization of log file content, not only facilitates the training process but also sets the stage for standardized analysis across a spectrum of error scenarios.

The report underscores the imperative of continuous learning and adaptation, emphasizing the potential for our model to evolve in tandem with the ever-changing software landscape. The advantages accrued from the integration of machine learning in error prediction and resolution are poised to yield tangible benefits, ranging from accelerated troubleshooting to cost savings and improved software reliability.

Looking ahead, the expected outcomes of implementing automated server log analysis using AI hold the promise of transformative impacts on organizational workflows. The envisioned increase in productivity, rapid problem-solving capabilities, and notable savings on costs align seamlessly with the overarching goals of operational excellence. The heightened security, insightful reporting, and scalability further fortify the project's potential to elevate the organization's technological infrastructure.

In navigating the complexities of software engineering, the ease-of-use interface and assurance of compliance underscore the project's commitment to inclusivity and adherence to industry standards. The projected improvement in server performance positions this endeavor not just as a solution to existing challenges but as a catalyst for ongoing optimization and efficiency in server processes.

In essence, this project stands at the crossroads of innovation and practical application, bridging the gap between cutting-edge research and tangible solutions for the challenges faced in modern software development. As we anticipate the realization of these outcomes, the future landscape of server log analysis appears not only promising but fundamentally transformed, with AI-driven automation at the forefront of effective, efficient, and secure software engineering practices.

REFERENCES

[1] D. Yuan, S. Park, and Y. Zhou, ‘‘Characterizing logging practices in open source software,’’ in Proc. 34th Int. Conf. Softw. Eng. (ICSE), Jun. 2012,

[2] M. Fuller, E. Brighton, M. Schiewe, D. Das, T. Cerny, and P. Tisnovsky, ‘‘Automated error log resolution: A case study,’’ in Proc. 36th Annu. ACM Symp. Appl. Comput., Mar. 2021,

[3] X. Wang, D. Wang, Y. Zhang, L. Jin, and M. Song, ‘‘Unsupervised learning for log data analysis based on behavior and attribute features,’’ in Proc. Int. Conf. Artif. Intell. Comput. Sci., Jul. 2019,

[4] R. C. Raga and J. D. Raga, ‘‘A comparison of college faculty and student class activity in an online learning environment using course log data,’’ in Proc. IEEE SmartWorld Ubiquitous Intell. Comput. Adv. Trust. Comput. Scalable, Dec. 2018,

[5] G. Qi, W. T. Tsai, W. Li, Z. Zhu, and Y. Luo, ‘‘A cloud-based triage log analysis and recovery framework,’’ Simul. Model. Pract. Theory, vol. 77, Aug. 2020.

[6] X. Tian, H. Li, and F. Liu, ‘‘Web service reliability test method based on log analysis,’’ in Proc. IEEE Int. Conf. Softw., Rel. Secur. Companion (QRS-C), Jul. 2017,

[7] D. Zou, H. Qin, and H. Jin, ‘‘UiLog: Improving log-based fault diagnosis by log analysis,’’ J. Comput. Sci. Technol., vol. 31, no. 5, 2016, doi: 10.1007/s11390-016-1678-7.

[8] S. Locke, H. Li, T. H. P. Chen, W. Shang, and W. Liu, ‘‘LogAssist: Assisting log analysis through log summarization,’’ IEEE Trans. Softw. Eng., early access, May 26, 2021, doi: 10.1109/TSE.2021.3083715.

[9] J. Cándido, M. Aniche, and A. Van Deursen, ‘‘Log-based software monitoring: A systematic mapping study,’’ PeerJ Comput. Sci., vol. 7, Oct. 2021

[10] D. Obrâbski and J. Sosnowski, ‘‘Log based analysis of software application operation,’’ Adv. Intell. Syst. Comput., vol. 987, Oct. 2020

[11] Q. Cao, Y. Qiao, and Z. Lyu, ‘‘Machine learning to detect anomalies in web log analysis,’’ in Proc. 3rd IEEE Int. Conf. Comput. Commun. (ICCC).

[12] B. Debnath, M. Solaimani, M. A. G. Gulzar, N. Arora, C. Lumezanu, J. Xu, B. Zong, H. Zhang, G. Jiang, and L. Khan, ‘‘LogLens: A real-time log analysis system,’’ in Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst.(ICDS),July 2018,

[13] Liang Y, Zhang Y, Xiong H, Sahoo R. 2007. An adaptive semantic filter for blue gene/L failure log analysis. In: 2007 IEEE International Parallel and Distributed Processing Symposium, Long Beach, CA, USA. Piscataway: IEEE, 1–8

[14] Li Lin H, Zhou J, Yao B, Guo M, Li J. 2015. Cowic: a column-wise independent compression for log stream analysis. In: 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Shenzhen, China. Piscataway: IEEE, 21–30.

[15] Meng W, Liu Y, Zhu Y, Zhang S, Pei D, Liu Y, Chen Y, Zhang R, Tao S, Sun P, Zhou R. 2019.LogAnomaly: unsupervised detection of sequential and quantitative anomalies in unstructured logs.

[16] P. He, J. Zhu, S. He, J. Li and M. R. Lyu, "Towards Automated Log Parsing for Large-Scale Log Data Analysis," in IEEE Transactions on Dependable and Secure Computing, vol. 15, no. 6, pp. 931-944, 1 Nov.-Dec. 2018, doi: 10.1109/TDSC.2017.2762673.

[17] Tan J, Kavulya S, Gandhi R, Narasimhan P. 2010. Visual, log-based causal tracing for performance debugging of mapreduce systems. In: 2010 IEEE 30th International Conference on Distributed Computing Systems. Piscataway: IEEE, 795–806.

[18] Yu X, Joshi P, Xu J, Jin G, Zhang H, Jiang G. 2016. CloudSeer: workflow monitoring of cloud infrastructures via interleaved logs. In: Proceedings of the Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS ’16, Atlanta, Georgia, USA. New York: ACM, 489–502.

[19] Yuan D, Park S, Zhou Y. 2012. Characterizing logging practices in open-source software. In: Proceedings of the 34th International Conference on Software Engineering, ICSE '12, Zurich, Switzerland. Piscataway: IEEE Press, 102–112.

[20] Zheng Z, Yu L, Tang W, Lan Z, Gupta R, Desai N, Coghlan S, Buettner D. 2011. Co-analysis of RAS log and job log on blue gene/P. In: 2011 IEEE International Parallel & Distributed Processing Symposium, Anchorage, AK, USA. Piscataway: IEEE, 840–851.

[21] P. He, J. Zhu, S. He, J. Li, and M. R. Lyu. 2016. An Evaluation Study on Log Parsing and Its Use in Log Mining. In 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). 654–661.

[22] Sergey Brin and Lawrence Page. 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine. Computer Networks 30 (1998), 107–117. http://www-db.stanford.edu/~backrub/google.html

[23] Jieming Zhu, Shilin He, Jinyang Liu, Pinjia He, Qi Xie, Zibin Zheng, and Michael R Lyu. 2019. Tools and benchmarks for automated log parsing. In 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP). IEEE, 121–130.

[24] Zhu J, He P, Fu Q, Zhang H, Lyu MR, Zhang D. 2015. Learning to log: helping developers make informed logging decisions. In: Proceedings of the 37th International Conference on Software Engineering - Volume 1, ICSE '15, Florence, Italy. Piscataway: IEEE Press, 415–425.

[25] Bao L, Li Q, Lu P, Lu J, Ruan T, Zhang K. 2018. Execution anomaly detection in large-scale systems through console log analysis. Journal of Systems and Software 143(1):172–186 DOI 10.1016/j.jss.2018.05.016.

[26] Abad C, Taylor J, Sengul C, Yurcik W, Zhou Y, Rowe K. 2003. Log correlation for intrusion detection: a proof of concept. In: Proceedings of the 19th Annual Computer Security Applications Conference, 2003, Las Vegas, Nevada, USA. Piscataway: IEEE, 255–264.

[27] Aharon M, Barash G, Cohen I, Mordechai E. 2009. One graph is worth a thousand logs: uncovering hidden structures in massive system event logs. In: Buntine W, Grobelnik M, Mladenić D, Shawe-Taylor J, eds. Machine Learning and Knowledge Discovery in Databases. Berlin, Heidelberg: Springer, 227–243.

[28] Q. Fu, J. Lou, Y. Wang, and J. Li, "Execution anomaly detection in distributed systems through unstructured log analysis," in ICDM'09: Proc. of International Conference on Data Mining, 2009.

[29] S. Banerjee, H. Srikanth, and B. Cukic, "Log-based reliability analysis of software as a service (saas)," in ISSRE'10: Proc. Of the 21st International Symposium on Software Reliability Engineering, 2010.

[30] El-Sayed N, Schroeder B. 2013. Reading between the lines of failure logs: understanding how HPC systems fail. In: 2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Budapest, Hungary. Piscataway: IEEE, 1–12.

[31] A. Oliner and J. Stearley, "What supercomputers say: A study of five system logs," in DSN'07, 2007.

[32] A. Makanju, A. Zincir-Heywood, and E. Milios, "A lightweight algorithm for message type extraction in system application logs," IEEE Transactions on Knowledge and Data Engineering, vol. 24, pp. 1921–1936, 2012.

[33] J. Lou, Q. Fu, S. Yang, Y. Xu, and J. Li, "Mining invariants from console logs for system problem detection," in ATC'10: Proc. of the USENIX Annual Technical Conference, 2010.

[34] F. Salfner, S. Tschirpke, and M. Malek, "Comprehensive logfiles for autonomic systems," in IPDPS'04: Proc. of the 18th International Parallel and Distributed Processing Symposium, 2004.

[35] C. Di Martino, M. Cinque, and D. Cotroneo, "Assessing time coalescence techniques for the analysis of supercomputer logs," in DSN'12, 2012.

APPENDIX-A

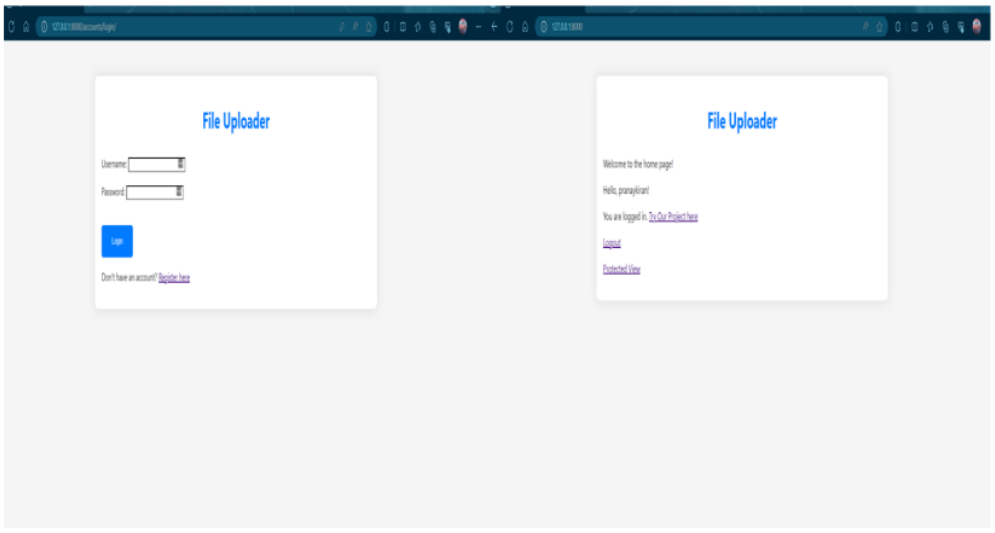
PSUEDOCODE

```
Procedure home(request):
    Render 'home.html'

Procedure protected_view(request):
    Render 'protected_view.html'

Procedure register(request):
    If the request method is POST:
        Create a UserCreationForm instance with data from the request.
        If the form is valid:
            Save the form to create a new user.
            Log in the newly created user.
            Redirect to the 'home' page.
    Else:
        Create a UserCreationForm instance.
    Render the 'registration/register.html' template with the form.

Procedure upload_file_view(request):
    Render 'upload_file.html'

Procedure compile_and_search(request):
    Set response to None

    If request method is POST:
        uploaded_file = Get uploaded file from request

        Try:
            If uploaded_file exists:
                Extract file_name and file_extension from uploaded_file
                language = Extract language from file_extension
```

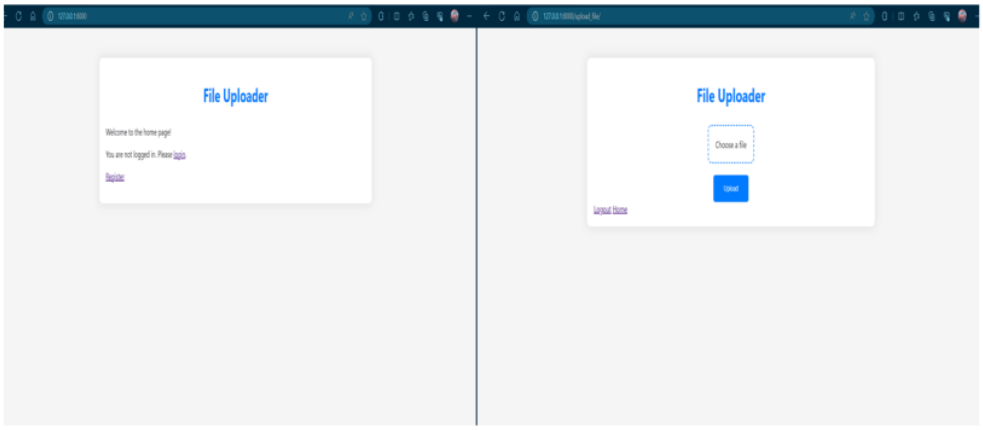
```
        Save uploaded_file with a new name

        If language is 'py':
            Compile Python file
        Else if language is 'java':
            Compile Java file

        If compilation is successful:
            Set response to success message
        Else:
            If language is 'py':
                Extract and handle Python compilation error
                Predict threat level
            Else:
                Extract and handle Java compilation error
                Display error lines
            Search Google for the error message
            Open relevant search results in browser tabs
    Else:
        Set response to "No file uploaded."

Except Exception as e:
    Set response to error message

Render 'Error_log/upload_file.html' with response
```

APPENDIX-B

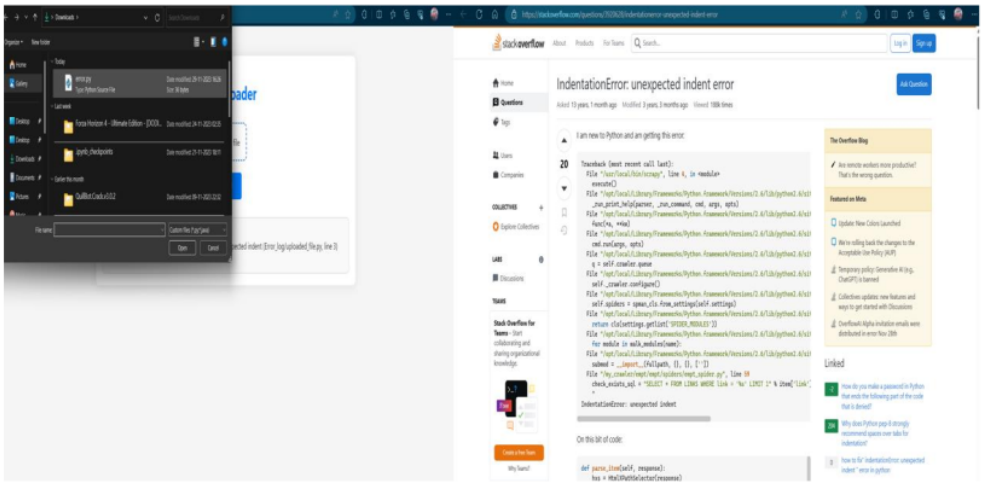
SCREENSHOTS



Login and Successful Login page



Home page and upload file page



File uploaded page and solutions page

Dashboards page

APPENDIX-C

ENCLOSURES

Published Paper Link: https://www.ijrar.org/papers/IJRAR23D3087.pdf

Certificate of Publication

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

E-ISSN: 2348-1269, P-ISSN: 2349-5138

The Board of International Journal of Research and Analytical Reviews (IJRAR) Is hereby awarding this certificate to

Shaik Salma Begum

In recognition of the publication of the paper entitled

AI Powered Server Log Management Software

Published In IJRAR (www.ijrar.org) UGC Approved - Journal No : 45602 & 7.17 Impact Factor

Volume 10 Issue 4 December 2023, Date of Publication: 28-December-2023

PAPER ID : IJRAR23D3087

Registration ID : 280234

R.B.Joshi

EDITOR IN CHIEF

UGC and ISSN Approved - Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.17 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijrar.org | Email: editor@ijrar.org | ESTD: 2014

Manage By: IJPUBLICATION Website: www.ijrar.org | Email ID: editor@ijrar.org

IJRAR | E-ISSN: 2348-1269, P-ISSN: 2349-5138

Certificate of Publication

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

E-ISSN: 2348-1269, P-ISSN: 2349-5138

The Board of

International Journal of Research and Analytical Reviews (IJRAR)

Is hereby awarding this certificate to

A Deepak Venkat

In recognition of the publication of the paper entitled

AI Powered Server Log Management Software

Published In IJRAR (www.ijrar.org) UGC Approved (Journal No : 43602) & 7.17 Impact Factor

Volume 10 Issue 4 December 2023, Date of Publication: 28-December-2023

PAPER ID : IJRAR23D3087

Registration ID : 280234

R.B.Joshi

EDITOR IN CHIEF

UGC and ISSN Approved - Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.17 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijrar.org | Email: editor@ijrar.org | ESTD: 2014

Manage By: IJPUBLICATION Website: www.ijrar.org | Email ID: editor@ijrar.org

IJRAR | E-ISSN: 2348-1269, P-ISSN: 2349-5138

Certificate of Publication

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

E-ISSN: 2348-1269, P-ISSN: 2349-5138

The Board of

International Journal of Research and Analytical Reviews (IJRAR)

Is hereby awarding this certificate to

S Pranay Kiran

In recognition of the publication of the paper entitled

AI Powered Server Log Management Software

Published In IJRAR (www.ijrar.org) UGC Approved (Journal No : 43602) & 7.17 Impact Factor

Volume 10 Issue 4 December 2023, Date of Publication: 28-December-2023

PAPER ID : IJRAR23D3087

Registration ID : 280234

R.B.Joshi

EDITOR IN CHIEF

UGC and ISSN Approved - Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.17 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijrar.org | Email: editor@ijrar.org | ESTD: 2014

Manage By: IJPUBLICATION Website: www.ijrar.org | Email ID: editor@ijrar.org

IJRAR | E-ISSN: 2348-1269, P-ISSN: 2349-5138

Certificate of Publication

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

E-ISSN: 2348-1269, P-ISSN: 2349-5138

The Board of

International Journal of Research and Analytical Reviews (IJRAR)

Is hereby awarding this certificate to

Akshay Jayesh

In recognition of the publication of the paper entitled

AI Powered Server Log Management Software

Published In IJRAR (www.ijrar.org) UGC Approved (Journal No : 43602) & 7.17 Impact Factor

Volume 10 Issue 4 December 2023, Date of Publication: 28-December-2023

PAPER ID : IJRAR23D3087

Registration ID : 280234

R.B.Joshi

EDITOR IN CHIEF

UGC and ISSN Approved - Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.17 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijrar.org | Email: editor@ijrar.org | ESTD: 2014

Manage By: IJPUBLICATION Website: www.ijrar.org | Email ID: editor@ijrar.org

IJRAR | E-ISSN: 2348-1269, P-ISSN: 2349-5138

Certificate of Publication

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

E-ISSN: 2348-1269, P-ISSN: 2349-5138

The Board of

International Journal of Research and Analytical Reviews (IJRAR)

Is hereby awarding this certificate to

Kesava M

In recognition of the publication of the paper entitled

AI Powered Server Log Management Software

Published In IJRAR (www.ijrar.org) UGC Approved (Journal No : 43602) & 7.17 Impact Factor

Volume 10 Issue 4 December 2023, Date of Publication: 28-December-2023

PAPER ID : IJRAR23D3087

Registration ID : 280234

R.B.Joshi

EDITOR IN CHIEF

UGC and ISSN Approved - Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.17 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijrar.org | Email: editor@ijrar.org | ESTD: 2014

Manage By: IJPUBLICATION Website: www.ijrar.org | Email ID: editor@ijrar.org

IJRAR | E-ISSN: 2348-1269, P-ISSN: 2349-5138

Similarity Index / Plagiarism Check

AI POWERED SERVER LOG MANAGEMENT SOFTWARE

ORIGINALITY REPORT

10%	9%	4%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

3%

★ www.ijrar.org

Internet Source

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		