WEB-Based Platform for Activity Scheduling Management

Cristian-Gabriel COLȚAN and Ștefania-Loredana NIȚĂ

***Abstract*—Web-based activity schedule management platforms are applications designed to organize, manage and optimize tasks. The resulting software product is a web application that was designed and developed according to the best practices of software engineering, leveraging the most modern and advanced technologies, such as React, Node.js, MongoDB, which make it possible to build scalable, fast and accessible applications from any device connected to the Internet. The implementation of the informatic system resulted in an efficient and adaptable digital platform that ensures high accessibility, scalability and simplified maintenance, making the platform suitable for real-world corporate environments.**

***Index Terms*—digital collaboration tools, full-stack development, scalable web applications, task scheduling platform, web-based activity management.**

I. Introduction

The 21st century is often referred to as the "century of speed." Analyzing the technological advances that have grown exponentially in recent years, we see that web applications have become essential tools, representing a solution for streamlining the management of activity schedules. They contribute decisively to the digitization of work tasks, facilitating progress monitoring and communication in collaborative environments. Managing collaborative activities across dynamic teams has been extensively analyzed in literature, revealing the need for flexible and traceable task coordination mechanisms [1].

Given the current technological context, the topic of this paper assumes particular importance, as it offers a solution to the real needs of the organizational environment: efficient management of work tasks and facilitation of real-time collaboration through a modern, scalable, and user-friendly IT system. The proposed digital solution also deals with the organization of the hierarchy of development teams, offering the possibility of defining the roles and responsibilities of each member within the assigned activities.

Thus, by clearly establishing tasks, increased productivity, reduced confusion, and effective progress monitoring are ensured. The application also significantly reduces the volume of repetitive and time-consuming activities for team leaders, facilitating their focus on strategic decisions and coordination, resulting in faster achievement of set goals.

The first forms of task automation appeared with the development of computers in the second half of the 20th century. In the 1970s, computer systems became more accessible, and organizations began looking for ways to automate repetitive activities in order to increase operational efficiency. Initially, schedules were managed manually, using physical diaries, printed calendars, approaches that were prone to errors, time-consuming [2], and increasingly difficult to manage as the volume of data grew.

The first software solutions for activity management appeared with the popularization of personal computers, allowing the digital storage of scheduling information in databases. Users employed simple "to-do list" applications that offered a limited number of functions. As computing technology advanced, software products dedicated to complex tasks began to be developed, gradually replacing the rudimentary physical methods that had become inefficient in the context of organizational requirements. An important moment in the history of the organizational field was the emergence of the first dedicated software, Artemis, developed in 1977 by Metier Management Systems [3]. The widespread adoption of these digital solutions began in the 1980s with the advent of the IBM Personal Computer, which contributed to the popularization of computing devices in the organizational environment. The introduction of Microsoft Project to the market was one of the most important milestones in the evolution of project management information systems. It allowed managers to plan projects and tasks graphically, schedule activities, and monitor progress [4].

Since the 2000s, there has been a massive shift towards cloud-based systems in the project management software industry. These online solutions have brought numerous benefits, such as real-time collaboration between team members, instant updates visible to all users, and access to the web platform from any location or device with Internet access. This has eliminated synchronization issues and significantly improved the efficiency and coordination of project teams. During the same period, there was also a paradigm shift in project management methodologies, which influenced the evolution of software platforms. Initially, these were managed sequentially, according to the Waterfall approach. As requirements became more complex and the need for rapid adaptation became more pronounced, Agile methodologies such as Scrum or Kanban gained momentum. Web applications such as Jira or Backlog were designed to support the implementation of Agile methods [5]. They emphasize iterative development, close collaboration, and continuous feedback, allowing teams to frequently deliver concrete results to customers.

A major development is due to the emergence of a large number of mobile devices, which has led to the adaptation of platforms to accessible working environments optimized

C. G. COLȚAN is with the Military Technical Academy "Ferdinand I", Bucharest, Romania (e-mail: gabriel.coltan@mta.ro).

Ș. L. NIȚĂ is with the Military Technical Academy "Ferdinand I", Bucharest, Romania (e-mail: stefania.nita@mta.ro).

for use on smartphones and tablets. The early 2010s brought the first scheduling management applications, which allow end users to access and manage their activities via their phones.

Currently, customer expectations have increased in terms of the functionalities offered by digital platforms. Several cloud-based task management platforms have been proposed to enhance distributed software development environments [6]. Automation and artificial intelligence have become essential components in the field of task management, being used to anticipate project delays, reduce manual effort, and optimize planning processes [7].

The contributions of the paper are as follows:

- The design and implementation of a modular web-based platform tailored for activity scheduling and team coordination in modern organizations.
- The integration of a scalable technology stack (React, Node.js, MongoDB) with cloud deployment (Railway) to ensure accessibility and maintainability.
- The development of real-time monitoring, notifications, and a detailed audit system using the ELK stack, enhancing transparency and accountability.
- The implementation of advanced features such as role-based access control (RBAC), two-factor authentication (2FA), and task recovery mechanisms.
- A performance validation process including simulated user testing and usage scenarios, confirming the platform's stability under concurrent load.

II. Proposed Solution

The proposed solution is a full-stack web platform designed to support activity scheduling, monitoring, and team coordination in a digital environment. It offers a role-based access model, allowing users to log in securely, view or assign tasks, and track progress across multiple teams.

Key features include:

- User and team management, including role assignment and member promotion;
- Task lifecycle management, from creation and editing to deletion and restoration;
- Interactive dashboards that visualize performance metrics, work hours, and team efficiency;
- Real-time notifications and an activity timeline to enhance communication;
- Audit and monitoring tools using ELK for full traceability of user actions and system status.

Built with React (frontend), Node.js/Express (backend), and MongoDB (database), the platform ensures fast load times, high availability, and intuitive use. It is deployed via Railway for easy continuous delivery and integrates modern security practices such as two-factor authentication (2FA) and role-based access control (RBAC).

This solution is designed similarly to other web-based project task management platforms [8], but introduces several innovations such as integrated monitoring and AI assistance.

The CCTask platform is designed for efficient task management in collaborative environments, offering a modern interface that ensures a smooth and optimized browsing experience. The system architecture follows the classic client-server model, in which users interact with the system through the web interface (client component), sending their requests and transmitting them to the server. The server receives the requests, processes them, accesses the information in the database, and returns the necessary data in a structured and efficient manner. To ensure traceability and effective monitoring, CCTask contains a logging mechanism that automatically transmits and centralizes user interactions with the application. As a priority, security is not neglected, with the platform implementing a role-based authentication mechanism, with the possibility of activating an additional step in the process in the form of two-factor authentication.

This approach helps separate responsibilities, contributes to system scalability, facilitates efficient task management, while the integration of the monitoring component significantly improves the process of detecting errors or non-conformities, making the platform a reliable tool.

Fig. 1 shows the general architecture of the CCTask platform, structured around two major components: Client Side and Server Side. The user interacts with the application through a web browser, sending HTTPS (HyperText Transfer Protocol Secure) requests and receiving corresponding responses.

To facilitate scalability and flexible development, the CCTask platform adopts a modular and extensible architecture, in which each module is clearly separated by responsibility and supports independent development, testing, and maintenance. This design choice enables future scalability and rapid integration of new features while maintaining system stability.

The innovative character of the platform lies in the way traditional components—authentication, task management, and user interaction—are enhanced by modern technologies such as AI-based assistance, real-time logging via ELK stack, and advanced administrative tooling, all tightly integrated into a coherent system. Below we detail the modules, highlighting their key functionalities and the novel aspects that contribute to the paper's value.

A. Visitor Module

The visitor module (Fig. 2) is intended for unauthenticated users, providing access to the platform's presentation page, which provides the reader with general information, briefly describes the main features available, the customer reviews, and the contact section. This isolated component allows unauthenticated users to access the presentation area of the platform. While standard in many applications, its clean separation from core infrastructure improves security and aligns with the principle of least privilege. It also serves as an onboarding path toward full user integration.

B. Authentication and Authorization Module

This module controls access to the platform. It is common to all user categories, as the authentication process does not differ depending on the user's role. Visitors can create a new account either by filling in the required fields (first name,

 Digital Object Identifier 10.32754/JMT.2026.1.13

last name, username, email, password) or by using external authentication via their Google account, integrated through Firebase. This approach facilitates the accessibility and security of the registration process.

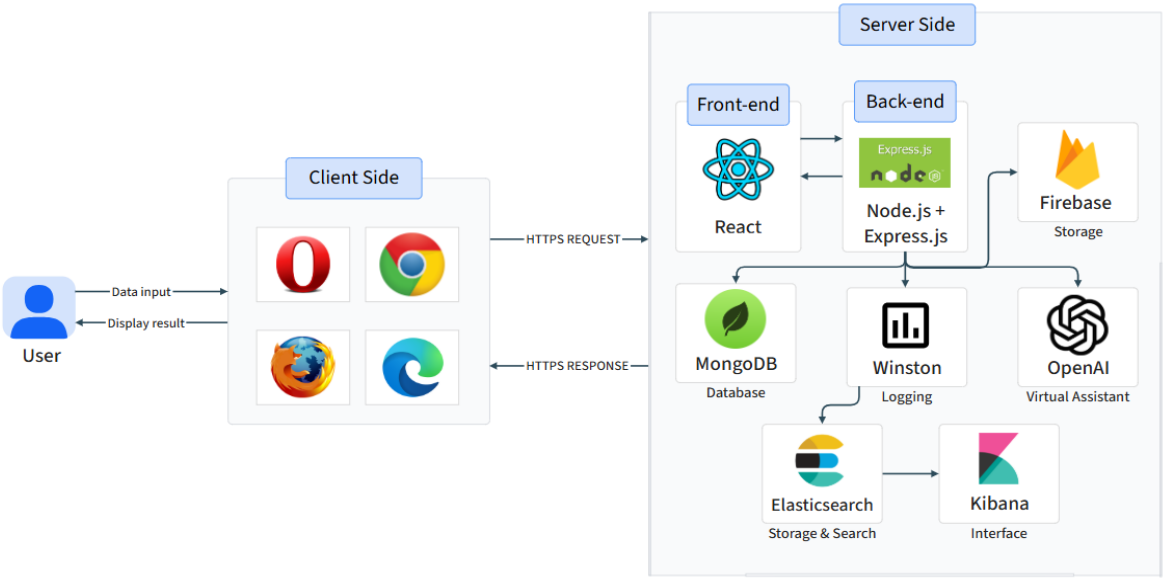

Figure 1. System architecture

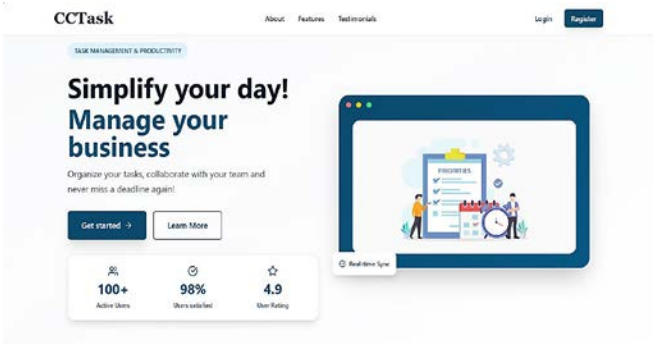

Figure 2. Visitor module

C. *Profile Management Module*

It can be accessed via the menu on the right side of the application. By accessing the Personal Details section (Fig. 3), the user can update certain personal data, such as name, surname, gender, and address.

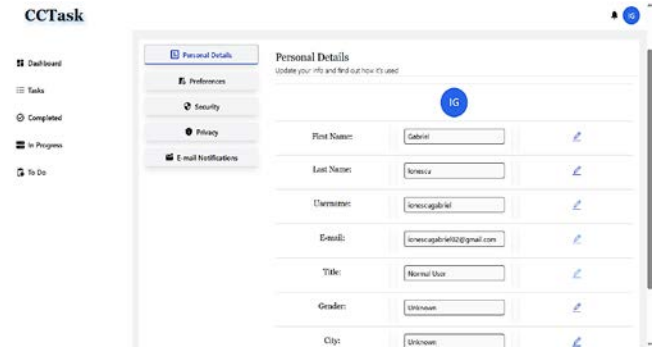

Figure 3. Personal Details section

In addition to standard personal data management, this module includes a two-factor authentication (2FA) mechanism, allowing users to increase account protection – an advanced feature not commonly included in lightweight task management tools. Furthermore, the system is designed to support customizable privacy and preference settings in future versions.

D. *Team Leader Management Module*

This module is accessible exclusively to personnel with Administrator roles, allowing for complete management of users designated as Team Managers, and includes numerous features that contribute to maintaining a coherent hierarchical structure. Both managers and the tasks they create are presented in tabular form, containing the most relevant information.

E. *Notifications Module*

The notifications module is responsible for automatically sending important information to users, ensuring real-time awareness and collaboration, reducing communication friction and preventing delays.

Notifications are generated in the following situations:

- When a task or subtask is created, the assigned team members are notified;
- When the Team Manager adds a new member to their team, they are notified;
- Any comment posted in a task's activity log generates a notification sent to the team manager;
- Team members are also notified when important details of a task are changed;
- Changing or assigning a role within the activity generates a notification to the user concerned.

Notifications are classified into two categories: Alert for events that require immediate attention, such as creating or modifying tasks and Message for informative, routine events (assigning a role, adding a user to the team).

F. *Task Management Module*

The task management module is the central component of the application and also the most complex. It is dedicated to managing all activities in the system, offering specific functionalities tailored to the user's role. The Tasks page (Fig. 4) displays, in the form of cards, both the tasks created by the user and those in which they are part of the team.

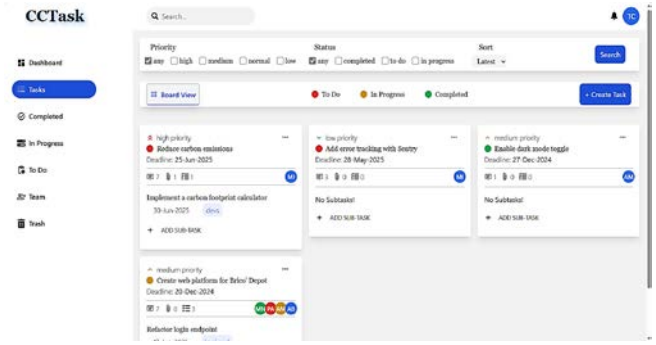

Figure 4. Tasks page

A search and filter function is also available to the user, allowing tasks to be filtered according to certain criteria (priority, stage) and sorted according to deadline. Each card briefly presents the details of the activity and includes an extensive menu of actions, which allows you to view the task in detail, edit the associated information, duplicate it, and delete it. Future versions may also integrate intelligent task scheduling algorithms based on machine learning, as explored in [9].

Once accessed, a task is structured into four main sections, each containing essential aspects. The Task Details section provides a complete overview of the title, deadline, team members involved, assigned roles, as well as subtasks and their current status.

The Timeline section shows a chronological log of all actions performed within the task (comments, time allocation, work on subtasks, error marking, etc.), facilitating traceability and real-time collaboration between team members.

The Statistics section contains summarized data that helps to objectively evaluate the activity carried out (Fig. 5).

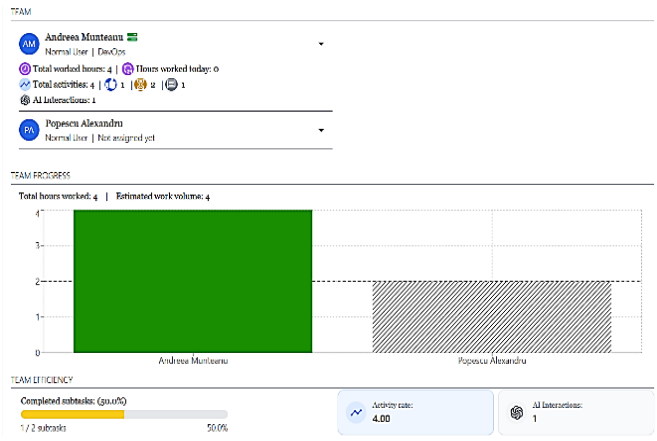

Figure 5. Statistics section

The Assistant section integrates an AI-based virtual assistant that helps users with personalized recommendations, answers to questions, or guidance in the task development process.

G. *Administration Module*

The module is intended exclusively for users with the Administrator role and represents the central control and configuration component of the application. It ensures the management of the entire ecosystem and allows direct intervention on users, tasks, and resources associated with the application. It is divided into 7 sections, as follows:

Users: manages the accounts of users registered in the application and provides features such as editing personal data, activating/deactivating accounts, viewing roles within tasks, and generating activity reports.

Tasks: provides an extensive interface for managing activities on the platform, with the ability to view, edit, or delete them. It can also manage secondary tasks, activity logs, and additional resources for each one.

Notifications: facilitates efficient monitoring of notifications issued on the platform, allowing sorting by notification type and creation date.

Team Managers: similar to the user management section, it allows editing personal data, viewing team composition, and generating detailed reports on their activity.

User Roles: allows you to customize the roles to be assigned to users within tasks. Task Coordinator cannot be deleted, as it is a default role in the platform.

Database backups: the administrator has access to a section dedicated to restoring the database from backup files, with the option to delete versions that are no longer relevant.

Kibana: the administrator is redirected to a specialized dashboard used for monitoring logs, tracking user actions, and analyzing application performance.

H. *Performance Logging and Monitoring Module*

An essential component for the continuous evaluation of the efficiency, security, and scalability of the IT system is the integrated logging and monitoring module. This module is responsible for collecting, indexing, analyzing, and visualizing performance and usage data related to the application. Its main objective is to provide full observability, reduce downtime, and enable real-time diagnostics of potential system failures or inefficiencies.

The platform integrates a professional-grade monitoring stack based on the ELK architecture (Elasticsearch – Logstash – Kibana), using a complete integration of the following technologies:

- Winston: serves as the main logging utility on the server side, capturing structured log events such as user interactions, API calls, errors, and warnings. Logs are automatically forwarded to Elasticsearch in real-time, ensuring durability and traceability [10].
- Elasticsearch: is the search and indexing engine for data collected by Winston. Its ability to process and search large volumes of semi-structured data is key in identifying performance trends and anomalies [10]. In addition, Elastic APM (Application Performance Monitoring) (Fig. 6) has been integrated to complement the logging mechanism, contributing to the rapid detection of any malfunctions [11].
- Kibana: acts as the visualization interface for the above data (Fig. 7). Through customizable dashboards, administrators can explore user sessions, identify frequent errors, track service health, and observe the efficiency of critical operations (e.g., task creation, authentication) [12].

 Digital Object Identifier 10.32754/JMT.2026.1.13

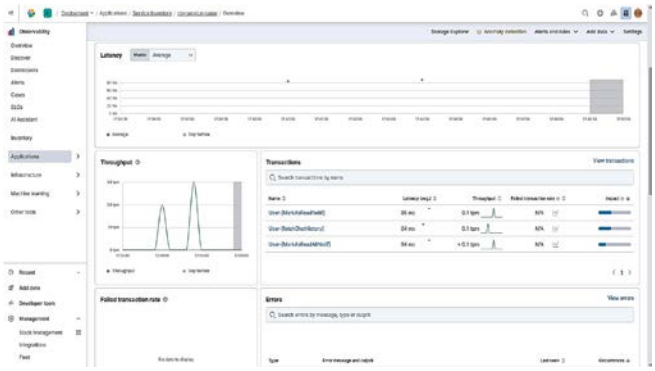

Figure 6. Application Performance Monitoring

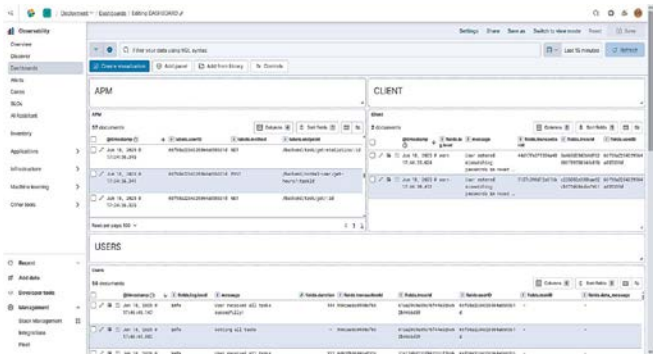

Figure 7. Kibana interface

In recent literature, full-stack monitoring using ELK and APM is acknowledged as a best practice in cloud-native and microservices-based systems [12-13]. Its integration in this platform aligns with state-of-the-art architectures adopted by enterprise-grade applications.

This architectural approach demonstrates how modern web development principles, such as modularity, scalability, observability, and security, can be effectively combined to deliver a task management platform tailored for real-world organizational needs.

III. Discussion

The results obtained following the implementation of the task management application highlight the efficiency of the implemented architecture and the system's ability to respond to user needs. This section details the results obtained, as well as how organizational processes are improved.

Following the analysis of data collected through Application Performance Monitoring (APM), it was observed that the application has an average response time of 117 ms for frequent operations (displaying pages and task information, authentication, etc.). This reduced response time is due to the use of the JSON format for data transfer and the optimization of back-end component routes through Express.js.

The modular architecture, the use of the MongoDB non-relational database management system, and Firebase for storing external resources enable stable operation over an extended period of time and facilitate the expansion of the platform without degrading its performance.

By integrating the Winston – Elasticsearch – Kibana technology stack, the IT system provides complete traceability of user interactions with the platform, enabling real-time analysis and anomaly detection. These tools facilitate effective performance monitoring and help reduce the time needed to identify potential vulnerabilities or operational issues.

The application is built with React and provides an intuitive, unified user experience, accessible from a wide range of devices, thanks to its adaptive design. In terms of security, the platform allows both classic authentication and Google account authentication, with the option of enabling two-step authentication. Access to features is restricted using a role-based access control (RBAC) model, with each user having different permissions depending on their assigned responsibilities (Guest, User, Team Manager, Administrator), helping to protect sensitive data and limit unauthorized access to critical areas of the app. In addition, SonarQube Cloud, a static code analysis platform, was used to ensure the quality of the source code and identify potential security vulnerabilities. By running the analysis periodically, issues such as code smells, injection vulnerabilities, or outdated dependencies that could have affected the IT system were identified and remedied promptly. These results confirm the viability of the IT system and demonstrate its applicability in industry, especially in organizational environments where it is necessary to coordinate activities, monitor the progress of development teams, and distribute the workload evenly.

IV. Conclusion

By integrating support for the Kanban methodology with external authentication services, a granular access-control model, and a modern logging and analysis stack based on Winston, Elasticsearch, and Kibana, the proposed system demonstrates the feasibility of a secure and scalable project-management platform. The adoption of a modular architecture and technologies such as React, Node.js, MongoDB, and Firebase underscores the contribution of this work to the design of contemporary web applications that balance performance, flexibility, and robust security guarantees.

Future work may explore advanced analytics, adaptive access-control policies, and enhanced monitoring capabilities to further improve the system's security posture, scalability, and usability.

References

[1] C. Nundlall and S. D. Nagowah, "Task allocation and coordination in distributed agile software development: A systematic review," *Int. J. Inf. Technol.*, vol. 13, no. 1, pp. 321–330, Feb. 2021. doi:10.1007/s41870-020-00543-4

[2] "Appointment scheduling software," *Wikipedia*, Feb. 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Appointment_scheduling_software&oldid=1275851288.

[3] M. N. Uddin, "What is task management software: Boost productivity effortlessly," *Macroter*, Nov. 2024. [Online]. Available: https://macroter.com/what-is-task-management-software/.

[4] E. Froud, "The rise of project management software: A history," *The Digital Project Manager*, Aug. 2025. [Online]. Available: https://thedigitalprojectmanager.com/project-management/history-of-project-management-software/.

[5] P. Karhapää *et al.*, "Evidence-based quality-aware agile software development process: Design and evaluation," *IEEE Access*, vol. 12, pp. 86487–86512, Jun. 2024. doi:10.1109/ACCESS.2024.3414614

[6] S. Ewenike, E. Benkhelifa, and C. Chibelushi, "Cloud based collaborative software development: A review, gap analysis and future directions," in *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, 2017, pp. 901–909. doi:10.1109/AICCSA.2017.220

[7] "The evolution of project management software: From traditional to Agile methodologies," *Nulab*, Oct. 2023. [Online]. Available: https://nulab.com/learn/project-management/evolution-project-management-software/.

[8] Y. A. Hambali, A. Anisyah, and A. S. Arianti, "Analysis and design of web-based project management system," *Journal of Engineering Science and Technology*, vol. 19, no. 3, pp. 996–1009, Jun. 2024.

[9] M. E. Khatib, M. AlQurashi, S. AlHashemi, M. AlKetbi, and S. AlHarmoodi, "Digital platforms' influence on project management," in *2023 International Conference on Business Analytics for Technology and Security (ICBATS)*, 2023, pp. 1–7. doi:10.1109/ICBATS57792.2023.10111200

[10] B. Lyu, "Design and implementation of real-time log analysis based on ELK," *Applied and Computational Engineering*, vol. 77, no. 1, pp. 124–131, Jul. 2024. doi:10.54254/2755-2721/77/20240666

[11] T. M. Ahmed, C.-P. Bezemer, T.-H. Chen, A. E. Hassan, and W. Shang, "Studying the effectiveness of application performance management (APM) tools for detecting performance regressions for web applications: An experience report," in *Proceedings of the 13th International Conference on Mining Software Repositories*, 2016, pp. 1–12. doi:10.1145/2901739.2901774

[12] J. Kosińska, B. Baliś, M. Konieczny, M. Malawski, and S. Zieliński, "Toward the observability of cloud-native applications: The overview of the state-of-the-art," *IEEE Access*, vol. 11, pp. 73036–73052, Jun. 2023. doi:10.1109/ACCESS.2023.3281860

[13] F. Basciftci and F. Aydemir, "Strategies for request-response logging in microservices architecture," in *2022 IEEE 20th Jubilee International Symposium on Intelligent Systems and Informatics (SISY)*, 2022, pp. 000121–000126. doi:10.1109/SISY56759.2022.10036308