A Scalable Microservices-Based Architecture for E-learning Platforms

[1]Pedro Guedes Monteiro and [2]Paulo Tomé

[1] Polytechnic of Viseu, Portugal
[2] CISeD - Research Centre in Digital Services, Polytechnic of Viseu, Portugal

[1]pv20272@alunos.estgv.ipv.pt, [2]ptome@estgv.ipv.pt

Abstract. E-learning platforms have become critical infrastructure for knowledge dissemination, flexible scheduling, and lifelong upskilling. Their relevance was underscored during the COVID-19 pandemic, which exposed recurring gaps in widely used systems: brittle scalability under surges, uneven customization across institutional contexts, fragmented support workflows, and limited, siloed analytics. These constraints hindered timely interventions, raised operational costs, and complicated governance and maintenance as deployments grew and heterogeneity.
This paper proposes a scalable architecture for e-learning that combines a microservices-based LMS with an integrated knowledge hub and gamification. The design supports institutional customization, observability, and elastic scaling. A comparative analysis with six leading platforms highlights the architectural advantages and addresses scalability and support gaps exposed during the pandemic.
To evaluate the relevance and advantages of this architecture, an integrated Artificial Intelligence layer provides conversational support and data-driven recommendations grounded in usage and performance signals. We complement the architectural description with a comparative analysis against six established platforms, using a set of evaluation variables defined in the methodology (architecture, licensing, implementation type, scalability, customization, ecosystem, integration/extensibility, and gamification), showing how the proposed approach addresses pandemic-revealed gaps while delivering SaaS-like elasticity without forfeiting institutional control.

Keywords: E-learning; Knowledge Hub; Gamification; Artificial Intelligence; Microservices; Digital Platform.

1 Introduction

The COVID-19 pandemic exposed critical limitations in existing e-learning platforms. Critical issues, as pointed in [1], include technological limitations, lack of adequate technical support, ineffective change management and cultural barriers. These gaps highlighted an urgent need for more effective and adaptable solutions.

Research Paper
DOI: https://doi.org/10.46793/eLearning2025.105GM
Part of ISBN: 978-86-89755-37-4

In response to these challenges, this led to the conception of a platform designed to combine e-learning features with an emerging knowledge center concept. This knowledge center refers to a collaborative digital space that integrates functionalities for creating, sharing and managing educational content. In contrast to traditional platforms, this implementation promotes an interactive and dynamic environment where users can consume information but also actively contribute to its enrichment.

To overcome challenges such as lack of interaction and inadequate support, the knowledge center includes features such as opening technical and pedagogical support tickets for quick resolution of problems, continuous interaction between users through comments and collaborative forums and direct publication of content by users, an approach aligned with evidence showing that user-generated content enhances collaborative knowledge building in online education [24].

The adoption of modern architectures, such as microservices supported by Docker and Fastify, has proved crucial to ensuring scalability and efficiency on large-scale platforms, as advocated by the literature [28]. According to these authors, microservice-based architectures, when combined with lightweight, high-performance tools such as Fastify, promote greater modularity and autonomy between services, allowing different modules to operate independently, which facilitates maintenance, incremental updates and the expansion of the platform as new needs arise.

This article is structured into sections that follow a logical sequence to ensure a clear and structured understanding of the platform being proposed. In the **Introduction section**, the context of the project is outlined, along with the motivation for its development.This section also defines the problem to be addressed and presents the main objectives established for the work. The **Methodology section** details the methods and procedures adopted for the platform's development. It covers the technical approaches, tools to be used and methodological strategies that support the implementation process. In the **Comparative Analysis section**, a range of current e-learning solutions available on the market is explored and analyzed. The goal is to understand their architectures, functionalities and and underlying technological choices. The **Proposed Architecture section** explains how the selected methodologies are to be implemented in an integrated and efficient manner. It describes the roles of each component within the platform and how they interact with one another. Within the **Comparison with Existing Solutions section**, the proposed platform is contrasted with the previously analyzed systems. This comparison highlights both the strengths and the potential limitations of the proposed solution. Finally, the **Conclusions and Future Work section** summarizes the key contributions and findings of the project. It highlights the strengths of the proposed architecture and suggests future directions that could enhance the platform's impact and capabilities.

2 Problem Statement

Despite the progress that has been made in the area of digital education, [1] highlight that there are still persistent shortcomings, such as the lack of interaction between users, the absence of adequate technical support and the lack of adapted pedagogical approaches, problems which, when analyzed as a whole, represent a major challenge that was highlighted in several of the platforms analyzed, which is the lack of integration of robust technologies and collaborative approaches. This obstacle directly compromises the effectiveness of e-learning platforms, since there is a restriction on offering personalized educational experiences aligned with users' needs.

Given the scenario described in the previous paragraph, the following central question arises: How can we develop an e-learning platform that is scalable, comprehensive and integrates a robust knowledge center in order to promote personalized and collaborative learning?

To meet the challenges of the issue raised, the proposed platform will rely on technologies capable of enabling user-generated content, but will also provide reactive technical support systems to help students face the challenges presented to them. This need has become especially evident during the COVID-19 pandemic, when many institutions found it difficult to effectively support students due to the lack of integrated technical assistance, as noted by the authors [15]. In addition, [26], in their article, demonstrate that the use of gamified elements has proven to be effective in improving student engagement and motivation in digital environments. Finally, to provide personalized learning experiences, the integration of Artificial Intelligence is increasingly seen as essential, allowing platforms to adapt content and resources to the individual needs of each student.

3 Methodology

In order to understand the current panorama of e-learning platforms, we carried out a comparative analysis of six solutions on the market. The aim of this analysis is to identify the architectural approaches adopted, the licensing models used and the functionalities that each platform offers. The following platforms were chosen for analysis: Moodle, Canvas, Open edX , Coursera, Udemy and Sakai.

As part of the comparative analysis of the Moodle, Canvas, Open edX, Coursera, Udemy and Sakai e-learning platforms, a set of variables were defined that were considered fundamental for a comprehensive and rigorous assessment of the solutions on the market. The purpose of selecting these variables was to ensure a balanced analysis of the technical, functional and strategic aspects of each platform. The variables analyzed can be seen in the table 1.

The selection of these variables was based on the need for a critical and complete analysis, which is not limited to the functionalities offered, but also considers technical sustainability, operational flexibility and user experience.

The comparison between the platforms was based on various criteria considered fundamental for evaluating e-learning solutions. In terms of *architecture*, the

Table 1: Description of Comparative Evaluation Variables

Variable	**Description**
Architecture	Refers to the technical foundation of the platform, including client-server models, layer separation, and development technologies. This variable assesses the platform's internal organization, robustness, and long-term sustainability.
Licensing Model	Distinguishes between open-source and proprietary solutions, evaluating implications for customization, cost, and community support, which directly affect adoption feasibility.
Deployment Type	Analyzes whether the platform uses a monolithic or microservices-based architecture, influencing scalability, maintainability, and modular updates.
Scalability	Evaluates the platform's capacity to support increasing user volumes and activities without performance degradation—crucial for institutional growth.
Customization	Assesses flexibility in adapting the interface, workflows, and features to institutional or pedagogical needs, ensuring contextual alignment.
Community and Ecosystem	Considers project vitality, including forums, documentation, third-party extensions, and development activity—indicators of maturity and sustainability.
Integration and Extensibility	Measures the ability to connect with external tools and systems via Application Programming Interfaces (APIs), LTI support, and interoperability standards.
Gamification	Analyzes the presence of motivational elements such as badges, levels, and challenges, reflecting the platform's potential for user engagement.

technical foundations of each platform were analyzed, such as the client-server model, the separation of layers and the technologies used in development.

With regard to *licensing*, an attempt was made to distinguish between open source solutions and proprietary platforms, reflecting on the implications of this distinction in terms of flexibility, costs and community. The *deployment dimension* was addressed by identifying monolithic or microservice-based architectures, considering their impact on scalability, maintenance and ease of updating.

Scalability, in turn, was assessed based on the platforms' ability to support a growing number of users, courses and interactions without compromising performance. With regard to *customization*, the aim was to understand the extent to which each platform allows the interface, functionalities and flows to be adapted to the specific needs of each organization or educational context.

The *community* and *ecosystem* analysis focused on the vitality of the project around the platform, including the existence of support forums, regular contributions, comprehensive documentation and extensions developed by third parties. With regard to *integration* and *extensibility*, compatibility with other tools and services, support for standards such as LTI (Learning Tools Interoperability) and the existence of APIs to expand functionality were considered.

Finally, the degree of *gamification* offered by each platform was also assessed, i.e. the presence of elements that promote user involvement through mechanisms such as badges, progression levels, rankings and motivational challenges.

4 Comparative Analysis

To establish a comprehensive understanding of the current landscape, we analyzed six prominent e-learning platforms widely adopted in both academic and corporate contexts. Each of these platforms demonstrates distinct architectural patterns, integration capabilities, and educational approaches. The following sections detail the characteristics of each platform, highlighting their strengths, limitations, and relevance to the development of a modern, scalable learning environment.

4.1 Moodle

Moodle is a widely adopted Learning Management System (LMS) globally and, according to its official documentation, [17], is licensed under the GNU General Public License v3 (GPLv3). This license allows the use, modification and redistribution of the source code, promoting a strong international development community .

On a technological level, the platform is mostly developed in PHP, with additional support for web technologies such as JavaScript, HTML, CSS and SQL, as described in [17]. This base ensures compatibility with web services (Apache HTTP Server, Nginx, IIS) and with database management systems such as MySQL, PostgreSQL, MariaDB, Oracle Database and Microsoft SQL Server (MSSQL).

Its modular architecture, as detailed in the official repository [18], allows extensive customization through plugins and themes. In addition, offers RESTful APIs and web services that enable integrations with external systems, such as academic information systems, digital libraries and videoconferencing tools [17].

With regard to the interface, Moodle uses the Boost theme, based on Bootstrap, guaranteeing responsiveness and accessibility on various devices [17]. In terms of security and authentication, it supports protocols such as LDAP, OAuth2, SAML and Shibboleth, allowing integration with federated identity infrastructures [17].

4.2 Canvas LMS

Canvas is an LMS developed by the US company Instructure and, according to its official documentation, [12], offers an open-source version under the Affero GNU General Public License v3 (AGPLv3). This license allows the use, modification and redistribution of the source code, requiring the sharing of modifications even in Software as a Service (SaaS) environments, promoting collaboration and transparency.

On a technical level, the platform is mostly developed in Ruby on Rails, complemented with JavaScript, HTML, CSS and SQL, as described in [12]. It supports web servers such as Apache HTTP Server and Nginx, and uses PostgreSQL as its database management system [13].

Its modular architecture allows extensions and integrations via RESTful APIs and support for the LTI standard, making it easy to connect to various external tools, [12]. It also provides webhooks and endpoints for real-time data synchronization [13].

In terms of the interface, Canvas adopts a responsive and accessible design, aligned with the Web Content Accessibility Guidelines (WCAG) 2.1, ensuring compatibility with multiple devices and promoting inclusion [12]. In terms of security and authentication, it supports protocols such as SAML, LDAP and OAuth2, ensuring integration with complex identity infrastructures [12].

4.3 Open EdX

Open edX is an open source LMS initially developed by the Massachusetts Institute of Technology (MIT) and Harvard University, and currently maintained by the Open edX community and the company tCRIL [19]. It is licensed under the Apache License 2.0, allowing wide freedom of use, modification and redistribution of the source code.

From a technological point of view, it is developed mostly in Python, using the Django framework, as well as JavaScript, HTML, CSS and SQL [19]. It supports web servers such as Nginx and Apache HTTP Server and uses databases such as MySQL and MongoDB [20].

The architecture is based on microservices and independent modules, allowing for higher scalability and better customization. It offers RESTful APIs, LTI support and integration with external tools via Open edX plugins [19]. It also offers webhooks for synchronizing data with external systems [20].

In terms of interface, Open edX adopts a responsive design, in line with the WCAG 2.1 guidelines, ensuring accessibility and compatibility with various devices [19]. In terms of security, it supports protocols such as OAuth2 and SAML, guaranteeing integration with institutional systems [19].

4.4 Coursera

Coursera is a SaaS-based commercial LMS, founded in 2012 by professors from Stanford University with the aim of making Massive Open Online Courses (MOOCs) available globally [5]. As a proprietary platform, it does not make its source code available, but operates under institutional and corporate licensing.

In terms of technology, it is developed with a distributed architecture based on microservices [4]. It uses languages and technologies such as Java, Python, JavaScript, HTML, CSS and databases such as MySQL, MongoDB and Cassandra, operating essentially on public cloud computing infrastructures (Amazon Web Services (AWS)).

Its integration with external systems is ensured through RESTful APIs and support for LTI, in addition to providing Software Development Kit (SDK)s and webhooks for educational partners and institutional platforms [6].

In terms of interface, Coursera adopts a responsive and accessible design, in line with WCAG 2.1 guidelines [5]. Security and identity management are supported by protocols such as OAuth2 and SAML, ensuring integration with federated authentication infrastructures [5].

4.5 Udemy

Founded in 2010, Udemy, [34], is an online course platform that operates under a closed commercial model. Unlike open-source platforms, it does not make its source code available and access is regulated by individual subscription or via Udemy Business, [32], aimed at organizations.

In technical terms, its infrastructure is based on a distributed microservices architecture [33]. The platform uses Python, React, JavaScript, HTML, CSS and databases such as MySQL and Redis, operating in Cloud Computing environments supported by AWS.

Although it does not offer an extensive public API, enterprise integrations are possible through authentication solutions such as Single Sign-on (SSO) and partial support for the LTI standard [32]. The platform favours stability and scalability, but with less scope for institutional customization.

Udemy's interface is responsive and user experience-oriented, meeting the criteria defined by WCAG 2.1 and security is ensured by protocols such as SAML and OAuth2, facilitating integration with corporate directories and external LMS platforms [34].

4.6 Sakai

Sakai is an open-source LMS, primarily oriented towards higher education, and is developed and maintained by the Apereo Foundation [2]. The platform is made available under the Educational Community License 2.0 (ECL 2.0), a variant of the Apache License 2.0, allowing wide reuse and modification of the code, in line with free software principles.

From a technical point of view, it is built in Java and follows a modular architecture based on independent components [3]. It uses application servers such as Tomcat and database systems such as MySQL, Oracle Database and PostgreSQL. The user interface is developed using web technologies such as

JavaScript, HTML and CSS, and the platform supports integration with LDAP and external services via RESTful APIs.

Sakai's extensibility is one of its main features: it supports the LTI standard for integrating external tools and has a system of built-in tools that can be activated, customized or removed according to the institution's profile. Additional web services allow interoperability with digital libraries and academic management systems [2].

The interface design is responsive and in line with the good accessibility practices established by WCAG 2.1, allowing fluid use on different devices [2]. Federated authentication mechanisms such as SAML, LDAP and OAuth2 are also supported, ensuring security and compatibility with complex institutional infrastructures.

5 Proposed Architecture

The proposed architecture aims to provide a robust, scalable, and modular e-learning platform capable of adapting to growing user demands without compromising performance or maintainability.

The platform is optimized for responsiveness and stability, even under peak load. The technology stack was selected based on maturity, community support, and alignment with agile and secure development practices. In parallel, the architecture supports extensive customization—both visual and functional—to meet the diverse needs of target user groups and institutional contexts.

Designed to address gaps in existing e-learning platforms, the solution combines scalability, resilience, and personalization with a collaborative knowledge center. The platform enables individualized experiences, supports real-time pedagogical intervention, and sustains engagement through gamification, all without compromising technical robustness. A modular approach ensures long-term adaptability, allowing for the system's evolution without full rewrites [16].

At its core, the platform adopts a microservices architecture, isolating responsibilities into independent services packaged in Docker containers [7] and orchestrated by Kubernetes [31]. This enables elastic scaling, fault tolerance, and controlled deployment cycles. Horizontal Pod Autoscaler [30] and dynamic load balancing ensure resource alignment with demand, while container isolation improves environmental consistency. This setup allows intensive components like analytics or content delivery to scale independently.

For the user interface, Next.js [35] was chosen, combining Server-Side Rendering (SSR) and Static Site Generation (SSG) [36], enhancing response times, indexing, and usability on constrained devices. Its automatic routing and reusable components facilitate rapid development while preserving visual and semantic coherence. Documentation confirms that SSR and SSG contribute both to performance and SEO, reinforcing its relevance in educational platforms.

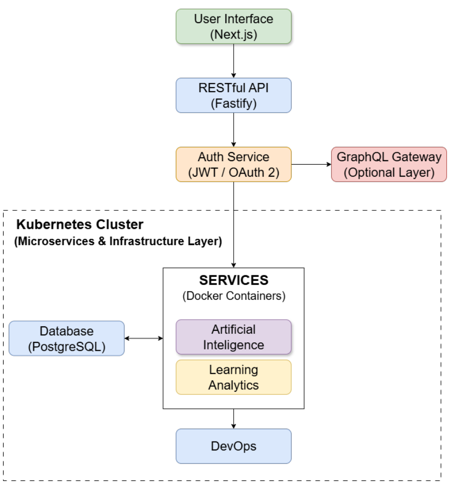
The services layer is implemented in Fastify [9], a lightweight Node.js framework with native JSON Schema validation. It enables the creation of secure, performant RESTful APIs in TypeScript. Benchmarks [29] highlight its performance advantage over other Node.js frameworks, making it ideal for large-scale platforms. Its plugin system also supports modular additions (e.g., authentication, rate limiting) without bloating the core.

PostgreSQL [21] was selected as the database for its ACID compliance, performance optimizations, and backup features, ensuring integrity and availability in high-throughput scenarios. For security, JWT [14] and OAuth2 [11] enable distributed authentication and authorization across services. GraphQL is optionally used for efficient data aggregation when REST endpoints are insufficient [10]. General Data Protection Regulation (GDPR) principles [8] guide personal data handling, emphasizing auditable logging and data minimization.

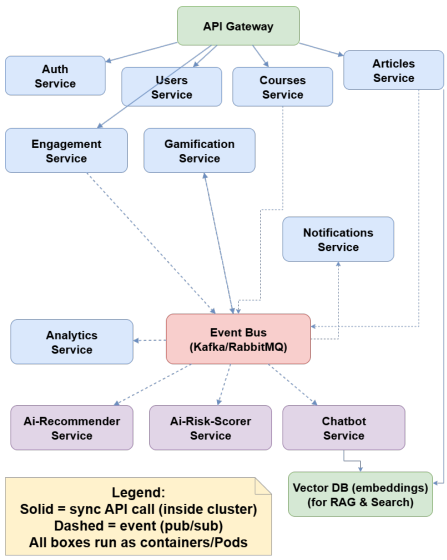
Artificial Intelligence and Learning Analytics modules analyze behavioral patterns to deliver personalized recommendations, identify at-risk learners, and support chatbot interactions [23]. Gamification reinforces engagement and motivation, with design guided by ongoing evaluation [25].

The platform's operation is sustained by DevOps practices, including CI/CD pipelines, automated testing, monitoring, and distributed tracing [27]. Observability is a key pillar in microservice ecosystems, supported through central metrics, logs, and correlation mechanisms [22]. Complexity is mitigated through automation and standardization, in line with microservices best practices [16].

Figure 1a presents an overview of the adopted architecture.



(a) Adopted Architecture



(b) Microservices Communication

Fig. 1: Overview of the proposed platform: (a) adopted architecture and (b) microservices communication.

5.1 Customization, Ecosystem, Integration/Extensibility, and Gamification

Ecosystem maturity varies markedly. Moodle and Canvas provide rich plug-in/LTI catalogs and public APIs; Sakai offers tool activation and LTI; Open-EdX supports modular extensions and webhooks; Coursera and Udemy expose narrower partner or enterprise interfaces. These differences shape how institutions tailor assessment, content, and analytics pipelines.

The proposed architecture presented in this paper, emphasizes contract-first composition (versioned REST, optional GraphQL) rather than in-process plugins, trading breadth-at-day-one for safer upgrades and clearer failure boundaries. Integration with SSO (OAuth2/SAML) and support for standards such as LTI sustain interoperability while keeping authorization and data minimization consistent across services.

Finally, the knowledge-center layer and an event-driven analytics/gamification stack speak to learner engagement and timely support: user-generated resources, ticketed assistance, and telemetry-based recommendations are integrated as first-class services; gamified elements motivate participation while remaining decoupled from core grading flows, in line with evidence that such mechanics can enhance engagement when carefully designed.

Taken together, the microservices approach delivers elasticity and operational visibility comparable to commercial SaaS, yet preserves institutional control over cadence, data, and customization. It complements—rather than merely replicates—the strengths of open-source incumbents by localizing change, aligning resource usage with real demand, and foregrounding support workflows, analytics, and motivation as integral, horizontally implemented capabilities. These outcomes map directly to the comparison variables defined in the methodology and to the shortcomings surfaced during the pandemic period.

The communication between the microservices that work together in Kubernetes can be seen in the figure 1b. There, synchronous interactions are terminated at the API Gateway, which applies JWT/RBAC and routes requests to core microservices (Auth, Users, Courses, Articles, Engagement, Gamification, Notifications, Analytics). Business events—such as "Enrollment Created", "Progress Updated", "Course Completed", and "Article Published" are published to the Event Bus (e.g., Kafka/RabbitMQ), where downstream subscribers consume them without tight coupling.

Gamification, Notifications, and Analytics process these events to award badges, trigger messages, and maintain metrics/warehouse views. AI services subscribe to the same streams: the Recommender updates personalized suggestions, the Risk Scorer flags at-risk learners, and the Chatbot performs RAG over a Vector DB (embeddings) for semantic retrieval. Solid arrows represent synchronous API calls; dashed arrows represent asynchronous pub/sub flows between containers/Pods.

Table 2: Comparative summary of the analyzed e-learning platforms

Criterion	**Moodle**	**Canvas**	**Open edX**	**Coursera**	**Udemy**	**Sakai**
Licensing	GPLv3	AGPLv3	Apache 2.0	Proprietary	Proprietary	ECL 2.0
Open source	✓	✓	✓	×	×	✓
Commercial model	Community (third-party SaaS available)	Hybrid (open-source & SaaS)	Community (with providers)	Closed SaaS	Marketplace / SaaS	Community
Base technology	PHP (+ JS)	Ruby on Rails (+ PostgreSQL)	Python/Django (+ MySQL/MongoDB)	Cloud microservices	Cloud microservices	Java (Tomcat)
Implementation type	Modular monolith	Cohesive web app with extensions	Service-oriented modules	SaaS (provider-managed)	SaaS (provider-managed)	Modular (cohesive deployment)
Scalability	Medium (tuning/cluster dependent)	Good (deployment-dependent)	Very good / Excellent	Excellent (SaaS)	Excellent (SaaS)	Good
Customization	High (plugins)	Good (LTI/APIs)	Good (XBlocks/APIs)	Limited (vendor-controlled)	Very limited (course/enterprise)	High (built-in tools/LTI)
LTI/API integration	✓	✓	✓	✓	Partial (enterprise)	✓
Primary focus	Formal education (multiple levels)	Higher education	University-scale MOOCs	Global MOOCs	Open marketplace / enterprise	Higher education
Installation complexity	Medium	Medium–High	High	SaaS (ready to use)	SaaS (ready to use)	High
Accessibility support	✓	✓	✓	✓	✓	✓
Active community	Very active	Active	Active	—	—	Smaller

6 Conclusions and future work

The proposed architecture offers a modular and scalable foundation for modern e-learning platforms. By adopting a microservices model with independent deployment, containerization, and orchestration, it ensures fault isolation, resource efficiency, and flexibility in evolving components. The use of a contract-first, CI/CD-enabled stack with observability and performance monitoring tools supports agile development and reliable operations.

Security, interoperability, and governance are treated as first-class concerns, with support for federated authentication, GDPR-aligned data practices, and integration via standards such as LTI, REST, and optionally GraphQL. Beyond infrastructure, pedagogical features such as a collaborative knowledge center, educational analytics, and gamification are embedded as core services to enhance learner engagement and enable timely interventions.

6.1 Future Work

Future development will focus on delivering a mobile client using cross-platform frameworks like React Native or Flutter. This tier will adopt an offline-first model and integrate with the existing identity and notification systems to ensure session continuity and user engagement under varying connectivity.

To maintain parity with the web experience, the mobile application will replicate accessibility guarantees, instrument telemetry, and leverage automated release strategies for progressive deployment.

Artificial Intelligence enhancements will span adaptive learning guidance, conversational assistance powered by retrieval-augmented generation from the knowledge base, and the refinement of gamification via A/B testing and behavioral analytics. All AI-driven features will adhere to privacy-preserving data practices, include human-in-the-loop mechanisms where appropriate, and undergo continuous efficacy evaluation to ensure responsible impact and trustworthiness.

Acknowledgements

"This work is funded by National Funds through the FCT - Foundation for Science and Technology, I.P., within the scope of the project Ref UIDB/05583/2020. Furthermore, we would like to thank the Research Centre in Digital Services (CISeD), the Polytechnic of Viseu, for their support."

Bibliography

[1] M. A. Almaiah, A. Al-Khasawneh, and A. Althunibat. Exploring the critical challenges and factors influencing the e-learning system usage during covid-19 pandemic. *EDUCATION AND INFORMATION TECHNOLOGIES*, 25(6):5261–5280, NOV 2020.

[2] Apereo Foundation. Sakai documentation. https://sakai.readthedocs.io/, 2025. Accessed on: Jun 11, 2025.

[3] Apereo Foundation. Sakai lms source code repository. https://github.com/sakaiproject/sakai, 2025. Accessed on: Jun 11, 2025.

[4] Coursera Inc. Coursera engineering. https://engineering.coursera.org/, 2025. Accessed on: Jun 11, 2025.

[5] Coursera Inc. Coursera for campus documentation. https://www.coursera.org/campus, 2025. Accessed on: Jun 11, 2025.

[6] Coursera Inc. Partners api documentation. https://partners.coursera.help/hc/en-us/categories/360002478312-Partners-API, 2025. Accessed on: Jun 11, 2025.

[7] Docker, Inc. Docker overview, 2025. Accessed on: Jul 23, 2025.

[8] European Parliament and Council. Regulation (eu) 2016/679 (general data protection regulation). Official Journal of the European Union, 2016. Accessed on: Jul 23, 2025.

[9] Fastify Core Team. Fastify — fast and low overhead web framework, 2025. Accessed on: Jul 23, 2025.

[10] GraphQL Foundation. Graphql specification, 2021. Accessed on: Jul 23, 2025.

[11] D. Hardt. The oauth 2.0 authorization framework. Technical Report RFC 6749, IETF, 2012.

[12] Instructure Inc. Canvas lms documentation. https://community.canvaslms.com/t5/Canvas-Basics-Guide/tkb-p/canvas-basics, 2025. Accessed on: Jun 11, 2025.

[13] Instructure Inc. Canvas lms source code repository. https://github.com/instructure/canvas-lms, 2025. Accessed on: Jun 11, 2025.

[14] M. B. Jones, J. Bradley, and N. Sakimura. Json web token (jwt). Technical Report RFC 7519, IETF, 2015.

[15] A. M. Maatuk, E. K. Elberkawi, S. Aljawarneh, H. Rashaideh, and H. Alharbi. The covid-19 pandemic and e-learning: challenges and opportunities from the perspective of students and instructors. *JOURNAL OF COMPUTING IN HIGHER EDUCATION*, 34(1):21–38, APR 2022.

[16] Martin Fowler. Microservices guide. https://martinfowler.com/microservices, 2019. Accessed on: Jul 23, 2025.

[17] Moodle HQ. Moodle developer documentation. https://moodledev.io/, 2025. Accessed on: Jun 11, 2025.

[18] Moodle HQ. Moodle open source lms source code. https://github.com/moodle/moodle, 2025. Accessed on: Jun 11, 2025.

[19] Open edX Consortium. Open edx documentation. https://docs.openedx.org/en/latest/, 2025. Accessed on: Jun 11, 2025.

[20] Open edX Consortium. Open edx source code repository. https://github.com/openedx/edx-platform, 2025. Accessed on: Jun 11, 2025.

[21] PostgreSQL Global Development Group. *PostgreSQL 16 Documentation*, 2024. Accessed on: Jul 23, 2025.

[22] Prometheus Authors. *Prometheus Documentation*, 2025. Accessed on: Jul 23, 2025.

[23] C. Romero and S. Ventura. Educational data mining and learning analytics: An updated survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(3):e1355, 2020.

[24] M. S. Roodposhti, E. Ezami, M. H. Hedayati, and A. Karimi. User-generated content effectiveness in co-creation of online higher educational services. *JOURNAL OF MARKETING FOR HIGHER EDUCATION*, 2024 MAY 8 2024.

[25] M. Sailer and L. Homner. The gamification of learning: A meta-analysis. *Educational Psychology Review*, 32:77–112, 2020.

[26] A. N. Saleem, N. M. Noori, and F. Ozdamli. Gamification applications in e-learning: A literature review. *TECHNOLOGY KNOWLEDGE AND LEARNING*, 27(1):139–159, MAR 2022.

[27] B. H. Sigelman, L. A. Barroso, M. Burrows, and P. Stephenson. Dapper, a large-scale distributed systems tracing infrastructure. Technical report, Google, 2010.

[28] F. Tapia, M. A. Mora, W. Fuertes, H. Aules, E. Flores, and T. Toulkeridis. From monolithic systems to microservices: A comparative study of performance. *APPLIED SCIENCES-BASEL*, 10(17), SEP 2020.

[29] TechEmpower. Web framework benchmarks (round 22), 2023. Accessed on: Jul 23, 2025.

[30] The Kubernetes Authors. Horizontal pod autoscaler, 2025. Accessed on: Jul 23, 2025.

[31] The Kubernetes Authors. Kubernetes documentation, 2025. Accessed on: Jul 23, 2025.

[32] Udemy Inc. Udemy business documentation. https://business.udemy.com/, 2025. Accessed on: Jun 11, 2025.

[33] Udemy Inc. Udemy engineering blog. https://tech.udemy.com/, 2025. Accessed on: Jun 11, 2025.

[34] Udemy Inc. Udemy terms of use. https://www.udemy.com/terms/, 2025. Accessed on: Jun 11, 2025.

[35] Vercel. Next.js documentation, 2025. Accessed on: Jul 23, 2025.

[36] Vercel. Rendering: Ssr vs ssg, 2025. Accessed on: Jul 23, 2025.