PM

Development of an E-learning Platform
Application to Sleep Medicine

MASTER'S DEGREE PROJECT

Bruno Rodrigo Faria Gonçalves Rocha
MASTER IN INFORMATICS ENGINEERING

UNIVERSIDADE da MADEIRA
A Nossa Universidade
www.uma.pt

August | 2025

(This page intentionally left blank.)

Development of an E-learning Platform
Application to Sleep Medicine

MASTER'S DEGREE PROJECT

Bruno Rodrigo Faria Gonçalves Rocha
MASTER IN INFORMATICS ENGINEERING

SUPERVISOR
Fernando Manuel Rosmaninho Morgado Ferrão Dias

CO-SUPERVISOR
Fábio Ruben Silva Mendonça

(This page intentionally left blank.)

FCEE

Master Thesis in Informatics Engineering

Development of an E-learning Platform: Application to Sleep Medicine

Bruno Rocha

supervised by

Morgado Dias and Fábio Mendonça

November 8, 2025

(This page intentionally left blank.)

Abstract

This thesis addresses the need for specialized educational tools in sleep medicine by designing an e-learning platform for clinicians, students, and the public interested in sleep quality analysis (as the considered use case). Despite increasing research and public awareness about sleep disorders, a persistent gap in accessible, interactive, and effective educational resources. The platform was developed using Docker, ReactJS for its intuitive front-end, Laravel for robust back-end functionality, PostgreSQL for reliable database management, and Nginx for efficient server performance. Adopting a single-page application architecture also ensures a seamless, responsive user experience across devices. Extensive quantitative evaluations assessed performance metrics such as application programming interface response time, memory usage, central processing unit utilization, network latency, database efficiency, and web performance. Qualitative assessments validated the platform's effectiveness through user feedback, usability testing, and surveys. Users reported increased engagement, improved comprehension of sleep concepts, and enhanced interaction through dynamically generated quizzes and modules. The integration of large language models further elevated educational interactions by generating context-sensitive quiz questions and real-time evaluations. The work highlights the platform's role in providing new ways to improve educational experiences in sleep medicine. Furthermore, the platform's flexibility and scalability also suggest adaptability across diverse medical education settings, supporting constant learning and better clinical results. Additionally, the platform can be adapted to other e-learning scenarios and scientific fields, with minimal adjustments. Regarding future recommendations, robust security, optimized scalability, and user-centric design are pointed out as important points to guarantee long-term sustainability and broader adoption in medical education.

Keywords: Sleep Medicine · E-learning Platform · Sleep Analysis · Docker Containerization · Single Page Application · Large Language Models

(This page intentionally left blank.)

Resumo

Esta tese aborda a necessidade de ferramentas educacionais especializadas em medicina do sono, através da conceção de uma plataforma de e-learning destinada a clínicos, estudantes e ao público em geral interessado na análise da qualidade de sono (caso de estudo considerado). Apesar do aumento da investigação e da consciencialização pública sobre os distúrbios do sono, permanece uma lacuna significativa em recursos educativos acessíveis, interativos e eficazes. A plataforma foi desenvolvida utilizando Docker, ReactJS para um front-end intuitivo, Laravel para uma robusta funcionalidade back-end, PostgreSQL para uma gestão segura da base de dados, e Nginx para desempenho eficiente do servidor. A adoção de uma arquitetura de aplicação de página única assegura também uma experiência de utilizador fluida e responsiva em diversos dispositivos. Avaliações quantitativas extensivas analisaram métricas de desempenho como tempo de resposta da interface de programação de aplicações, uso de memória, utilização da unidade central de processamento, latência da rede, eficiência da base de dados e desempenho web. Avaliações qualitativas validaram a eficácia da plataforma através do feedback dos utilizadores, testes de usabilidade e inquéritos. Os utilizadores relataram maior envolvimento, melhor compreensão dos conceitos relacionados com o sono e interações melhoradas por meio de módulos e questionários gerados dinamicamente. A integração de grandes modelos de linguagem fortaleceu ainda mais as interações educacionais ao gerar questões contextuais e avaliações em tempo real. Este trabalho sublinha o papel da plataforma na disponibilização de novos métodos para melhorar as experiências educativas na área da medicina do sono. Além disso, a flexibilidade e escalabilidade da plataforma sugerem a sua adaptabilidade a diversos contextos de educação médica, promovendo a aprendizagem contínua e melhores resultados clínicos. Adicionalmente, a plataforma pode ser adaptada a outros cenários de e-learning e áreas científicas com ajustes mínimos. Quanto às recomendações futuras, salienta-se a importância da segurança robusta, escalabilidade otimizada e design centrado no utilizador, como aspetos fundamentais para garantir a sustentabilidade a longo prazo e uma adoção mais ampla na educação médica.

Palavras-chave: Medicina do Sono · Plataforma de E-learning · Análise do Sono · Docker · Aplicação de Página Única · Grandes Modelos de Linguagem

(This page intentionally left blank.)

Acknowledgments

I would like to convey my profound appreciation to my thesis supervisors, Prof. Morgado Dias and Prof. Fábio Mendonça, for their constant support, guidance, and insightful advice throughout the entire development of this work. I am especially thankful for their tolerance and comprehension during occasional delays, and for continuously encouraging me throughout this journey.

I would also like to thank the Foundation for Science and Technology (FCT) for their support at the early stages of this project through the summer scholarship, which provided the foundation to initiate this research.

My thanks extend to the academic community, the University of Madeira, and in particular to Engineer Diogo Freitas, for his technical assistance and for providing the infrastructure that supported the implementation of the platform.

I am grateful to all the participants who contributed to the usability testing of the platform and shared valuable feedback to improve its quality.

To my mother Fátima and father António (in memoriam), thank you for your unconditional support, encouragement to overcome myself, and for being lifelong role models. To my sister Nádia, thank you for your patience and resilience. I am also deeply thankful to my extended family and close friends for standing by me during difficult times and celebrating the joyful moments along the way.

Finally, I would like to express my deepest appreciation to my girlfriend, Laura Silva, for her constant support, patience, and understanding throughout this entire process. Your encouragement was essential in helping me complete this work.

A heartfelt thanks to all those - named and unnamed - who, in any way, contributed to the realization of this thesis.

(This page intentionally left blank.)

List of Figures

List of Tables

Table of Contents

(This page intentionally left blank.)

1 Introduction

Sleep can be defined as a natural repetitive state with reduced response to external stimulation [1], and can be influenced by many factors such as age, sex, diet, physical and psychological health. It can be split into 4 stages: N1, N2, N3, and Rapid Eye Movement (REM). N1 is considered the starting phase and lightest stage of sleep, and N2 represents a deeper stage as the heart rate and body temperature drop. Following, N3 is considered the deepest sleep stage, most commonly associated with body repair and strengthening of the immune system. Normally, the REM stage is associated with dreaming, with more erratic and irregular breathing rates. Sleep analysis is typically performed through a polysomnogram, basically measuring electroencephalography (EEG), eye movements, pulse oximetry, muscle movement, nasal airflow and pressure, examining and assessing each stage of sleep [2]. So, sleep analysis is essential to determine the quality of sleep, which highly affects the quality of life and health.

Interest in sleep quality is increasing in the scientific community and the general population, since it is today recognized as a predictor of physical and mental health, wellness, and overall vitality [3]. As its definition is sometimes subjective, and it is used widely by researchers, physicians and patients [4], and so, to determine the sleep quality of a subject it is necessary to implement several objective measures which can be obtained through methodologies that can be considered for sleep quality characterization [5], including the EEG spectral analysis of sleep, Cycling Alternating Pattern (CAP), among others. The better this analysis is carried out, the better the medical specialist will do an objective and precise diagnosis. Sometimes, a single methodology is not enough to reach the expected efficiency, so there are several approaches to that measurement, for example, combining different types of measures or subgrouping individuals based on certain characteristics [5], which means an increasing overload of complex measurements to a single medical specialist over time.

However, there is a lack of consensus in the scoring of numerous sleep metrics [6]. This issue is particularly pronounced with CAP, where inter-expert agreement frequently falls below 80%. These discrepancies are primarily attributable to inherent difficulties associated with the accurate interpretation and scoring of complex physiological signals obtained during sleep, specifically EEG. The precise identification and classification of patterns within EEG data usually necessitate extensive training and practical experience, establishing a considerable learning curve for both new and established professionals.

Consequently, there is a need for an accessible mechanism capable of facilitating knowledge transfer, standardizing training protocols, and advancing the acquisition and dissemination of best practices in sleep signal scoring. An accessible e-learning platform would not only facilitate the pedagogical process for new practitioners but also constitute a valuable resource for experienced experts seeking to refine methodologies and thus improve inter-scorer reliability. Therefore, e-learning emerges as a suitable approach, providing a platform for disseminating complex information, facilitating interactive learning experiences, and providing standardized training modules. Its inherent flexibility allows for asynchronous access, accommodating diverse learning paces and schedules, which is relevant for professionals with demanding clinical or research responsibilities. Moreover, e-learning platforms can integrate multimedia content, such as annotated examples of EEG signals

and guided scoring exercises, to improve comprehension and practical skill development. The scalability of e-learning also means that comprehensive training can be delivered to a wide audience, effectively addressing the current deficit in consistent educational resources for sleep metric scoring.

The e-learning process and usage have also been growing over the years in multiple fields as it proves to be flexible, engaging, and learner-centered [7]. With such capabilities, it can be used in several areas of study, not as a replacement, but as a supplement to an existing educational process. Unfortunately, some issues need attention while using this supplement, including isolation, home distractions, lack of shared knowledge and practice, but once the nature of online distance learning and the need for e-learning are understood, it allows for a learning experience as rich as any campus experience [7].

1.1 Problem Statement

With the rise of efforts on finding a reliable automatic sleep classification for patients, there are not enough data to prove the efficiency of those efforts, mostly due to the lack of their clinical application or for not achieving a reasonable accuracy in the classification to support the diagnosis. Adding to this, there is also a lack of specialized physicians to perform an adequate analysis of sleep quality, which limits the application of sleep quality in clinical diagnosis [8].

The shortage of specialized physicians and limited data indicate that there is a gap in the process of education and access to reliable information among physicians, which can be complemented with an e-learning platform that fulfills those needs. This is especially relevant for scoring challenging procedures related to sleep quality, especially CAP scoring.

1.2 Study's Objective

This thesis introduces a web-based educational platform to assist specialized physicians, students, and the public to learn and easily accessing and exploring information in the analysis and assessment of sleep. This platform has focused core features based on the need for medical education:

1. Accessibility: A browser-based interface that allows easy access across various devices, ensuring availability for users regardless of location or operating system.
2. User-Centered Interface: An intuitive and user-friendly graphical interface that simplifies the navigation and operational learning curve, accommodating users of varying technical abilities.
3. Optimized Learning Workflow: A simplified process that facilitates efficient learning and practical application of the principles of sleep quality assessment, promoting a focused and effective educational experience.

The platform is designed with a modular and scalable structure and is devised as an infrastructure for future educational tools in different clinical/medical educational areas (in this work, however, it will be sleep, one of the most difficult disciplines). To ensure its readiness for widespread use, the platform will be systematically evaluated across multiple areas:

1. Security: Ensuring data protection and user confidentiality.

2. Performance: Analyze how quickly the content is loaded, server downtime, service availability, and overall responsiveness to ensure a better user experience.
3. Stability: The capacity to reduce errors and ensure consistency in platform functions over the long term.
4. Flexibility: Allowing new tools, features, and other adjustments with minimal development changes.
5. Responsiveness: Have compatibility with the majority of devices and screen sizes.

This modular and scalable approach guarantees a flexible and robust solution that can satisfy the different demands for medical education in multiple areas.

1.3 Research Questions

This study's goal is to answer the following research questions:

1. How can a web-based educational tool help improve the learning of students and/or physicians in sleep quality analysis?
2. What are the major features and design aspects that are essential to have in a user-friendly and effective e-learning platform for medical education in sleep analysis?

1.4 Significance of the Study

The importance of this thesis is that it has the potential to address relevant educational, clinical, and technological challenges in medical training. Specifically, this thesis seeks to contribute to:

1. Addressing the Education Gap: By offering e-learning tools for students, this study directly addresses the identified need in education for medical students and physicians specializing in sleep quality analysis.
2. Scalability in Medical Education: The platform's ability to handle dynamic content allows for potential expansion into other areas of medical education, addressing similar needs in different clinical scopes and specialties.
3. Technological Integration in Medical Education: This study shows the potential of web-based technologies for enhanced specialized medical education, offering a concept that can lead to further innovations in this area.

1.5 Proposed Solution

This subsection outlines the complete solution architecture, emphasizing the pedagogical, technological, and security considerations that guided the development. The platform merges a modular Single Page Application (SPA) with a Large Language Model (LLM) enhanced learning layer, establishing a scalable blueprint for AI-driven medical e-learning.

1.5.1 Conceptual Overview

The system is conceived as a browser-based platform that integrates three functional pillars, namely, content delivery, assessment, and community interaction, within a containerized microservice stack. A reverse-proxy gateway (Nginx) provides a single entry point, behind which isolated containers (React front-ends, Laravel, PostgreSQL database, and LLM service) communicate through REST endpoints.

1.5.2 Large Language Models as an Innovative Layer

LLMs remain underutilized in domain-specific medical e-learning applications. Deploying an LLM container inside a specific context introduces three distinctive capacities not yet documented for sleep-medicine education:

1. Dynamic question generation - use web resources to generate clinically relevant multiple-choice or short-answer questions.
2. Automatic grading and formative feedback, evaluating free-text answers against rubric prompts and returning constructive feedback instantly upon submission.
3. Context-aware chatbot, providing helpful assistance regarding the platform theme subject.

All model processing is conducted within a network-isolated environment, ensuring that only de-identified prompt data are handled.

1.5.3 Technical Explanation

- **React SPA**: eliminates full page reloads, enhancing responsiveness and mobile usability.
- **Tailwind CSS**: utility-first styling keeps the bundle size small while enabling rapid, consistent UI prototyping.
- **Laravel API**: offers eloquent object-relational mapper (ORM), robust token authentication (Sanctum, or others), and RESTful routing.
- **PostgreSQL Database**: ACID-compliant relational store with full-text search for fast retrieval of library resources and forum posts.
- **Nginx Reverse Proxy**: single ingress point that terminates transport layer security (TLS), serves static assets, and load-balances internal containers.
- **Docker orchestration**: ensures environment parity and permits horizontal scaling (e.g., spawning additional LLM workers).
- **LLM Service Container**: encapsulates all inference logic, allowing model upgrades or domain fine-tuning without touching the core stack.

1.5.4 Security and Privacy

Ensuring strong security and privacy of user data on the platform implies deploying multiple protective strategies. The system relies on short-lived tokens for authentication, reducing the windows of vulnerability and increasing user data protection. Additionally, sensitive information is

being securely stored with encryption standards, and the data integrity can be ensured via the use of immutable logs, with accountability and traceability being assured. All information exchanges and container interactions take place only inside the secured internal network, further reducing the risk associated with external data breaches or unauthorized access.

Furthermore, the platform is developed in compliance with the General Data Protection Regulation (GDPR) and relevant data protection laws. It applies concepts such as data minimization, data portability, the right not to be subject to automated decision-making processes, correction, and deletion requests. Privacy-by-design is core to the platform's architecture, which includes security and privacy considerations in early phases of development. Such preventive measures as periodic checks of the security system, vulnerability tests, and a monitoring program are scheduled to proactively discover, analyze, and reduce threats and vulnerabilities. When these security measures are combined, such a combination greatly reduces the risk of the loss of protected information, protecting user privacy and data veracity.

1.5.5 Pedagogical Modules, Alignment, and Expected Impact

The pedagogical modules developed within the platform combine different educational methods to facilitate user engagement and support effective learning. The **Interactive Academy** is composed of short, media-rich modules built with active recall and spaced repetition methods to efficiently memorize and retain critical information. Moderators generate different content blocks, such as videos, texts, explanations, and personalized questions. In addition to the academy, the **Knowledge Library** can be used as a central repository with interactive content that adds value to thematic themes. Moderators can manage these resources dynamically, in which case the moderator can add, update, or rearrange modules so that the content remains current and available. Additionally, threaded and tag-based interactions on the **Discussion Forum** support collaborative learning among peers and archive expert inputs, leading to a community of dynamic knowledge exchange. Finally, the **Personalized Dashboard** helps learners to improve based on self-assessment, offering insights into their quiz performances, course completion rates, and detailed learning history. This aspect provides learners with the ability to monitor their learning progress.

These modules are intentionally formatted based on established educational theories and strategies. In particular, the platform leverages micro-learning, delivering content in bite-sized, manageable units that minimize cognitive overload and support efficient retention through spaced repetition. It is presented in a case-based learning format with real-world examples that stimulate clinical reasoning and facilitate the practical application of theoretical knowledge. Additionally, the incorporation of social constructivism principles is evident through peer forums and chatbot-facilitated interactions to facilitate active collaboration and social learning dynamics. Finally, the adaptive mastery principle is realized through LLMs by creating content and dynamically adjusting difficulty levels, ensuring learners' promotion only until there is evidentiary mastery.

It is anticipated that implementing these pedagogical modules and methodologies has a high potential for impact. The LLM layer of the platform allows fast content creation that enables rapid production and update of educational materials. It also provides personalized feedback, going beyond traditional assessments to offer specific and encouraging feedback, thereby improving learner comprehension and engagement. The system design focus is to offer an engaging community

experience, which encourages meaningful interactions and the sharing of knowledge among users. Importantly, the architecture remains flexible, enabling easy adaptation of the solution to different clinical areas, with a modern AI-driven approach to medical education.

1.6 Scope and Limitations

The scope of this study displays the primary objectives and areas of focus for developing an educational tool focused on medical learning. Specifically, the scope elements are:

1. Development of a browser-based educational tool focused on sleep quality analysis, accessible throughout various devices.
2. Implement and design a user-friendly graphical interface to meet the needs of specialized physicians and students in medical areas.
3. Testing the platform for security, performance, stability, flexibility, and responsiveness in diverse usage scenarios.

However, the limitations of this work are scenarios where the platform may be constrained by factors that may influence its usability, performance, and impact. Specifically:

1. Focus on Sleep Analysis: While designed to be adaptable, the initial implementation is specifically developed for education in sleep quality analysis.
2. Target user group: The platform is primarily designed for students and specialized physicians, which may limit its immediate applicability to other healthcare professionals.
3. Technology Dependence: The effectiveness of the platform relies on users having access to and familiarity with web-based tools, potentially restricting usability for those with limitations to access or experience.
4. Indirect Clinical Impact: Although the platform aims to improve education, its direct clinical impacts on patient care, diagnosis, and outcomes are outside the scope of this study and would require further research.

1.7 Structure of the Work

The research is classified into 7 chapters. The first chapter introduces the subject of the research and synthesizes a review of the relevant literature in the area of sleep analysis, articulating the purpose and research questions that this thesis aims to answer and their significance. The second chapter examines the related work in sleep quality analysis and e-learning, including state-of-the-art existing methods, challenges, and current technological solutions. The methodology process of the platform development is explained in Chapter 3, as well as the evaluation approach. The fourth chapter presents the system architecture and design decisions, while the development and deployment are discussed as well. Chapter 5 presents the results of the usability testing and system measurements, followed by Chapter 6, which visually displays the final product. Finally, Chapter 7 concludes this part with the experimental results on the usability testing, performance evaluation, and flexibility assessment of the platform, and then the contribution and findings are summarized, and the limitations, which may lead to potential areas of future research, are discussed.

2 Related Work

The following chapter discusses the related work associated with Web-Based Learning (WBL) systems and the conjunction between WBL and medical education, either on a theoretical level or described in a certain tool.

Sleep medicine is a specialized field that involves the diagnosis and treatment of sleep disorders, which also have a profound influence on general health and quality of life. This discipline has undergone significant transformations in recent years, with new advancements in technology and research methodologies that are reshaping the understanding, evaluation, and treatment of sleep disorders.

Sleep medicine also involves educating healthcare providers about sleep disorders and their management. Studies have shown that a significant portion of medical residents feel inadequately trained in sleep medicine, implying the need to enhance the educational curricula that incorporate sleep medicine topics throughout the medical training period [9][10][11]. This educational gap is significant because the proper management of sleep disorders is best achieved by an integrative approach which draws from multiple medical disciplines, including psychiatry, neurology, and pulmonary medicine [12][13]. Accordingly, this creates the demand for appropriate web-based supporting platforms for knowledge sharing, and WBL seems to be an appropriate solution.

2.1 Web-based Learning

Cook and Dupras [14] described several preparatory steps necessary to develop a proper website design and highlighted several good practices throughout some of the development stages (development, implementation, maintenance). It supports the idea of combining the principles of active learning with some web features to implement them more effectively through those steps (including web design, resource management, learning self-assessment, etc).

The differences between the Traditional Classroom Learning (TCL) environment and the WBL environment in several dimensions of learning such as time, space, place, interaction, control, and technology, described by Picolli, Ahmad and Ives [15], are highlighted by Sharma and Garg [16], as they conclude that WBL offers significant results in terms of performance, flexibility, and interaction between the learner and the instructor. Nevertheless, broader reviews on online learning in medical education [17] emphasize that, despite these advantages, such approaches may still face challenges such as reduced social presence, learner isolation, and dependence on technological infrastructure if not properly supported.

In the article of Hamzah, Ariffin and Hamid [18] it is discussed how some existing characteristics in the WBL environment, as they encourage active discussion, cooperative learning, and flexible learning assessment, can increase motivation for learning a subject as learners can interact more with the teacher and their colleagues, emphasizing the student-centeredness in this environment (not depending only on the instructor). It summarizes that the educator must supervise the usage of the learning environment to guarantee its efficiency, leading to a positive impact on students.

An educational learning system developed by Anghel, A. Florea, Gellert, and D. Florea [19] shows how some features of WBL should be presented to increase system efficiency, as well as

some concerns regarding security features of using and storing data in the database. The authors underlined that collaboratively merging the TCL and WBL is mandatory and that visibility of main functions for e-learning objects is key throughout the development.

David Cook [17] discussed the main advantages and disadvantages of Internet-based or online learning approaches in medical education, which include many principles applicable to Web-Based Learning (WBL). Some advantages include:

- Flexible scheduling, allowing asynchronous participation in online groups and flexibility in a physical location.
- Individualized student learning, by allowing them to select multiple learning opportunities and evolve at their own pace.
- Share and save resources by eliminating redundancy in the development of course materials.
- Easily updated perpetual resource, as the learners can return to access the course materials when needed (for studying, seeing patients, etc.).
- Facilitated evaluation, providing the same flexibility and timing, as it adds customized feedback to the learner.

Cook also referred to some disadvantages, as a few are related to the advantages, which are listed as follows:

- With the increase of the flexibility, it means also that the learner studies alone, increasing the perception of social isolation.
- Although the WBL aims for individualized learning, it fails more often to correspond to the needs of the user, in which the instructions seem to be more predetermined than personalized.
- Technical problems are inevitable (bugs, server errors, misconfigurations, etc), resulting in user dissatisfaction and decreasing user participation in the course.
- The cost of developing each WBL experience can be very expensive.
- Poor instructional design may occur when trying to implement effective designs without an instructor for needed clarification.

Although some points of WBL can be very appealing compared to TCL, the role of WBL is still quite unclear in medical education, as it is necessary to develop the use of WBL, either alone or in integration with other instructional designs. It should be noted that WBL must only be used when its advantages outweigh its disadvantages.

Having regard to the preceding information, according to Chiemeke and Imafidor [20], with the sudden shift to WBL due to the current state of education, there is a knowledge gap among teachers and learners that increases the demand for systems that can reduce it, also highlighting key issues to improve the effectiveness of learning under such conditions. Although a general drop in the quality of education is expected in the near term, the quality is expected to increase in the long term as education progresses and adjusts itself to the shift.

2.2 E-learning in Medical Education

According to Ruiz, Mintzer, and Leipzig [21], medical educators can use e-learning to improve the efficiency of educational interventions, which, in graduate medical education, e-learning materials can be integrated into the education of residents and fellows, even replacing lectures and other synchronous methods of educational instruction. It is agreed that the integration of e-learning into undergraduate, graduate, and continuing medical education will promote a shift toward adult learning in medical education.

Huynh [22] stresses that even though e-learning has been demonstrated to be as effective as conventional didacticism, and encourages a medical student to exert greater control over their learning by employing flexibility over the content, the students do not see it replacing the traditional didactic methods.

In the article of O'Doherty et.al [23], some barriers and solutions are identified among the educators, including, for example, the following:

- Skill Deficit, which can be solved through engaging with e-learning, to gain skills for teaching practice
- Lack of time and pressure, as the educators need to accustom themselves to the e-learning tool.
- Lack of infrastructure in countries with low medical budgets.
- Miscommunication, as the asynchronous environment does not support active communication.
- Negative attitude of some educators, as they can feel inadequacy or/and limited knowledge. Promoting change in standards and behaviors is, therefore, an important solution.

The authors also underlined that online learning can provide easier and more effective access to a wider range of information, preparing students and educators for the new challenges facing the digital age in medical education.

According to Preim and Botha [24], the process of combining the two learning methodologies is described, defining what the e-learning system should provide and how to present it in natural language, to create a self-directed and directed method of learning, emphasizing the role of user motivation throughout usage of the developed system, allowing exploration of all the material available, interaction with it by solving tasks that involve manipulation of graphical objects or answering multiple choice questions.

Likewise, the authors Schneider and Binder [25] refer that online education plays an increasing role in all of the stages of medical education, being more effective, interactive, and adaptive to the learner's needs. It also states that this learning methodology will become a key element in post-graduate training, since doctors, after leaving their work environment, isolate themselves from structured institutional education, which e-learning education can fill this gap.

Currently there are multiple platforms already developed in multiple areas of medical education, but the neither one of them specifically address the domain of sleep medicine or integrate all the recommended functionalities for an effective online learning process.

- **MedSimAI** [26]: Focused on clinical simulation using LLM dialogue and real-time feedback, but lacks specialization for specific medical fields.
- **Virti** [27]: Offers virtual AI avatars and performance analytics for healthcare training, emphasizing interpersonal skills rather than specialized knowledge.
- **Amplifire** [28]: Employs adaptive algorithms for personalized learning but provides generic medical content.
- **MediTools** [29]: Incorporates automated question generation and AI tutoring, though still in prototype phase with limited specialization.

2.3 Security and Privacy in E-learning Platforms

In e-learning platforms, it is important to always consider the importance of the security of the information and the privacy of stored and accessed data inside the platform, particularly when handling sensitive information. Its advancement has significantly transformed educational practices, offering flexibility and accessibility to learners. However, security is one of the main concerns, especially regarding e-assessments. Security is mainly an organizational and management issue, and improving security is an ongoing process in e-learning. [30]. According to Miguel, Caballé, and Xhafa, to assess the overall security from learners' and evaluators' perspectives, multiple key factors need to be satisfied, which encompass availability, integrity, identification and authentication, confidentiality and access control, and non-repudiation.

The implementation of a fully secure system that has all these factors properly implemented is very difficult. A practical approach focuses on critical properties considered essential in the evaluation context: identification and integrity. Ensuring these properties, so that students are accurately identified and the submitted information is unmodified, forms the foundation of trust in the e-assessment process [30].

Platform or system providers must implement robust security measures such as granular access controls, strong authentication mechanisms, data encryption to reduce the risks associated with common attacks or vulnerabilities, such as Malware or phishing attacks, (Application Programming Interface) API, and integration vulnerabilities, Unauthorized access, among others [31]. Also, the end users should be reminded that they also play a crucial role in maintaining security by following the general best practices. The implementation of security measures must be aligned with international or regional data protection regulations [31].

2.4 Design, Scalability, Adaptability of Educational Platforms

The platforms must be designed to accommodate the increasing demand for digital learning environments and to be effective. Current e-learning systems are designed with a learner-centered approach, focusing on intuitive navigation, accessibility, and engagement features [32]. Modern educational platforms have prioritized user-centered design approaches as the use of desired user experience (UX) and user interface (UI) design serves to help students reach their desired learning outcomes [33]

To address scalability challenges resulting from the growing demand for e-learning solutions, modern architectures increasingly adopt cloud-based and modular infrastructures that enable dynamic allocation of computational resources according to usage patterns and institutional requirements. This approach allows educational institutions to handle both peak and regular operational loads efficiently, optimizing resource utilization while maintaining consistent performance and service availability. Scalability, cost-effectiveness, and reliability are interrelated factors that influence the overall quality of e-learning environments, as demonstrated by recent studies showing that cloud-based implementations can significantly increase simultaneous user capacity and operational efficiency without compromising service quality [34].

2.5 Evaluation Methods for E-learning Effectiveness

Assessing the effectiveness of e-learning platforms is of utmost relevance for ensuring quality standards in education and to achieve the desired learning outcomes for the users (learners).

According to Ertl et al. [35], the evaluation of e-learning includes formative and summative methods, each having different purposes, to improve and to adapt the learning environment to learners' needs and to prove the quality of the learning environment.

The study from Miguel, Caballé and Xhafa compared peer-to-peer (automatic) assessments with continuous (manual) assessments, which their findings suggest the hybrid assessment model shows to be promising, but needs to be adapted to improve reliability and effectiveness. The higher discrepancy in deviation of user behavior raised concerns over potential cheating and forced the need for e-learning platforms to implement identification and integrity measures to detect and address academic dishonesty.

There are several goals (or stages) in the development and assessment phases that need to be achieved according to Ertl et al. [35], which include quality of instruction, organizational benefits, and cost-effectiveness.

A comprehensive evaluation strategy typically incorporates both qualitative and quantitative methods to capture the general view of the e-learning experience. Qualitative methods, such as in-depth interviews, group discussions, and observational studies, can be useful to understand the motivations, challenges, and perceptions of the learners. These methods can be more useful in an early stage of development to gather feedback on content, clarity, and design. This is complemented by quantitative methods, which supply measurable data like user satisfaction surveys, learning patterns, performance metrics, and usage analytics.

The authors also suggest the usage of Kirkpatrick's Four-Level Model, which is a framework to evaluate training and e-learning initiatives. Examines four levels of evaluation: Reaction, Learning, Behavior, and Results. Each level of the model represents a more precise measure of the effectiveness of a training program. This could help the development team trace the influence of the system from the initial learner to professional practice or institutional performance.

2.6 Summary

This chapter has provided a detailed insight of the current state of the literature surrounding the topic of WBL environment within the context of medical education. The examined literature highlights the increasing role of digital/adaptive technology in enhancing accessibility, interactivity, and engagement, as well as pointing to the limitations in the areas of specialization, testing, and scalability. In this context, a distinct need has been detected concerning the lack of specialized online learning support in the subject of sleep medicine education. The identified findings point both to challenges and opportunities in the design of effective online learning systems, thereby underlining the need for a specialized learning environment, which establishes the rationale for the next chapter.

3 Methodology

This study employs a mixed-methods research design, combining quantitative and qualitative approaches to develop and evaluate an e-learning platform for the education of sleep analysis. The research design is developed to support the main goals involved in the development of the browser-based online educational tool with a simple graphical user interface to ease the learning process for specialized physicians in the analysis of sleep quality.

The research is based on a developmental research process that is suitable for creating and testing educational interventions in real-world settings. This approach allows for the iterative development of the e-learning platform while collecting data on efficacy and user experience. The study design consists of several phases:

1. Literature Review: A review of the existing literature on web-based learning environments, e-learning in medical education, and sleep analysis was conducted to inform the development of the platform.
2. System Development: The e-learning platform was developed using a Docker-based client-server architecture, incorporating ReactJS for the front-end, Laravel for the back-end, PostgreSQL for the database, and Nginx as the web server. This phase followed software engineering best practices and used a Single Page Application (SPA) approach.
3. Quantitative Evaluation: Various quantitative metrics were employed to assess the performance, stability, flexibility, and responsiveness of the developed system. These metrics provide objective data on the technical aspects of the platform.
4. Qualitative Evaluation: User feedback and experiences were gathered through qualitative mechanisms to identify potential issues within the system and gather ideas for improvement in both system operations and design.
5. User testing: The platform was tested to evaluate its effectiveness in facilitating the mediation learning process for sleep quality analysis.

3.1 Quantitative and Qualitative Metrics

To evaluate the effectiveness and user experience of the e-learning platform, the proposed solution is to use a mix-method approach, combining objective data (quantitative metrics) and subjective, experiential feedback (qualitative metrics). This allows to have a more comprehensive perspective on how the users interact with the platform, the faults and the benefits in a broader vision.

3.1.1 Quantitative Metrics

Quantitative evaluation was based primarily on system performance and user experience metrics. These were collected using tools such as Google Lighthouse and Thunder Client, under standardized network and hardware conditions to ensure reliable, reproducible results.

– First Contentful Paint (FCP): Time from page load to when the first visible content appears.

- Largest Contentful Paint (LCP): Time to render the largest visible content element, typically a text block or image.
- Total Blocking Time (TBT): Duration when the page is blocked and unable to respond to user input, due to long-running tasks.
- Speed Index: Measures how quickly visual content is displayed during page load, impacting perceived performance.
- Cumulative Layout Shift (CLS): Captures layout instability by quantifying unexpected shifts of page content during loading.
- API Response Time: Time between sending a request to the backend and receiving a complete response, measured using Thunder Client on typical endpoints.
- Database Query Performance: Measured using PostgreSQL's *'EXPLAIN ANALYZE'* to assess typical query times and efficiency.

These metrics were selected to provide an objective assessment of the platform's expected technical behavior during evaluation. Since the study focuses on the methodological validation of performance measurement rather than exhaustive benchmarking, the chosen metrics aim to demonstrate how the platform's responsiveness and efficiency can be effectively evaluated in upcoming production-level analyses.

3.1.2 Qualitative Metrics

Qualitative metrics offer critical insights into user experience through subjective feedback. These metrics provide essential information on user perceptions, satisfaction levels, usability, and overall interaction with the platform.

The qualitative data will be obtained through an online survey, which was structured to evaluate the user experience in the different roles of the platform (Admins, Moderators, and Users), using Likert-scale items and open-ended questions. From the users, the goal was to understand their challenges, experience satisfaction, usability, engagement with the platform, and social engagement. From the Admins and Moderators, the goal is to understand the alignment with the platform goals, the implementation of the interactive elements, modularity, and system administrative usage.

3.2 Development of the Application

The development process of the platform was initiated with two foundational objectives identified: firstly, to create an educational tool dedicated to sleep analysis with a user-friendly and intuitive graphical interface; secondly, to consolidate comprehensive training procedures tailored for physicians into a single, cohesive tool. These general objectives served as a high-level template for subsequent decomposition and detailed expression of specific needs.

Upon closer examination of the requirements specification, these original objectives were generalized into a detailed collection of functional, non-functional, and technical requirements. These specifications enabled coherent implementation, ensured alignment with user needs and expectations, and supported rigorous verification during the final evaluation phase.

The main **functional requirements** defined are as follows:

- User Authentication and Authorization: The platform shall provide secure authentication mechanisms and clearly defined user roles (e.g., administrator, moderator, and learner) and manage respective permissions.
- Interactive Learning Modules: The platform shall support the creation, organization, and delivery of interactive and engaging learning content, including multimedia elements such as videos, images, and simulations, to accommodate diverse learning styles.
- Progress Tracking: The platform shall record and display users' progress across courses and quizzes, including analytics such as progress charts and performance reports, to enhance learners' self-assessment and instructors' monitoring capabilities.
- Learning Evaluation: The system shall provide automated evaluation tools capable of assessing learners' knowledge and understanding consistently and accurately, reducing manual grading effort and minimizing assessment errors.
- Centralized Library: The platform shall maintain a centralized repository of educational and clinical resources related to sleep analysis, ensuring efficient organization, searchability, and retrieval of content.
- Knowledge Sharing: The system shall include a structured communication environment that enables peer-to-peer discussions, collaborative learning, and organized information exchange through tagged and categorized content.

The following **non-functional requirements**, which are needed to specify the behavior and performance of the system and to ensure a good user experience, are:

- Usability and Accessibility: The platform shall provide a logically organized, user-friendly interface that meets accessibility standards and complements the abilities of the widest range of users.
- Performance: The system shall maintain low latency and high availability under typical operational loads, delivering fast response times while supporting concurrent user sessions efficiently.
- Scalability: The system architecture shall support future expansion, allowing the integration of additional educational modules, increased user capacity, and extended functionalities without major redesign or code replacement.
- Security: The platform shall ensure protection of user data through secure authentication, high-level encryption, and safe data-handling practices to prevent unauthorized access and data breaches.
- Cross-platform Compatibility: The platform shall offer consistent functionality and user experience across major devices and operating systems, including desktop, tablet, and mobile environments.
- Maintainability: The system shall be developed using modular, well-documented, and clearly structured code to facilitate efficient updates, debugging, and future enhancements.

To guarantee technical coherence and development effectiveness, specific **technical requirements** were established:

- Technology Stack: The system shall employ a standardized and cohesive technology framework used across development, clearly specifying chosen technologies and ensuring compatibility throughout the development cycle.
- Containerization: The platform shall utilize Docker-based containerization to provide consistent and isolated environments for development, testing, and deployment, thereby minimizing compatibility issues and streamlining continuous integration and delivery processes.
- Database Management: The system shall implement optimized and well-structured database schemas that support efficient data storage, retrieval, and analysis, ensuring reliability and scalability in handling large datasets.

3.2.1 Implementation Planning

The implementation was performed following a plan with 5 elements and 4 phases, specifically:

1. Initial setup: Establishment of the development environment, GitHub repository, and versioning.
2. Core development: Administrative Front-end and Back-end simultaneous development, having consistency in the functionalities during the development.
3. Front-end development: Front-end development with a simple UI, only to get the necessary information on the screen.
4. System Integration: Development of integration procedures to integrate the multiple parts of the platform (back-end, front-end, administrative front-end, and database system).
5. Design Implementation: Adapt the front-end development to the proposed wireframe for the platform.
6. Alpha Testing Phase: End-to-end tests for every functionality implemented, for early defect detection and correction.
7. First Deployment Phase: Structure the domain, production environment, permissions, access, and deployment pipeline.
8. Beta Testing Phase: General tests with final users gathering valuable information to fix or adapt the functionality according to the feedback.
9. Final Deployment Phase: Deployment of the application and configuration of the system production environment variables.

3.2.2 Deployment and Maintenance Strategy

The development and deployment approach with Docker containers, maintaining consistency throughout the development and production environments, allows an easy deployment pipeline, either automatic or manual, depending on the server network limitations. If the server is in the cloud

(like AWS, DigitalOcean, etc) or in a local server, the process can be simplified by GitHub Actions. But if the server is only available via a Virtual Private Network (VPN), the initial deployment process will need to be manual.

After the final deployment, a structure maintenance is planned, with minimal development effort, which includes security audits, database optimization, and system updates when needed, ensuring consistency, performance, and reliability in the system provided.

3.3 System Architecture

The system architecture is defined by the set of necessary structures to reason about the system and how these structures are connected through a relationship. It must be taken into account that the architecture omits certain information about the elements that are not useful for the reasoning on the system.

3.3.1 Defining the Architecture

The system architecture of this platform is based on the current architecture of Docker, being a client-server architecture [36]. Docker architecture was chosen because it successfully meets the basic requirements for the project: enhanced data sharing capabilities, easier maintenance, improved security controls, and centralized management of data resources. The overall architecture can be briefly organized into separate components, each being responsible for specific functionalities within the system, including the Docker Client, Docker Host, Docker Objects, and Docker Registry. Such a modular and component-based architecture has the advantage that it makes the system easily scalable, provides easy flexibility in development, and is easy to maintain, thereby making the system robust from the point of sustainability in diverse educational environments. The overall architecture can be seen in Figure 1.

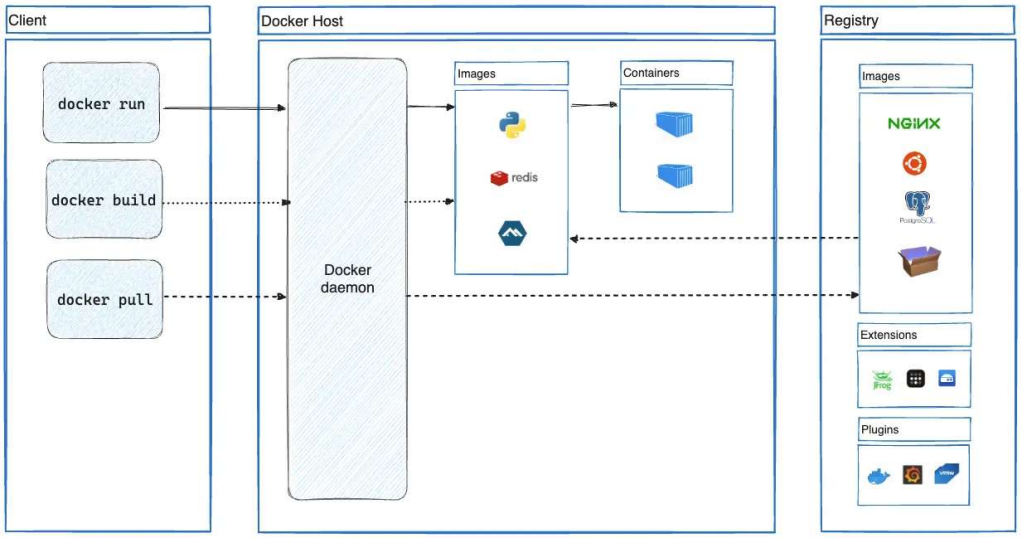

Fig. 1. Docker architecture diagram.

The Docker client is how its users interact with Docker, as it can reside on the same host as the Daemon or can connect to a remote host. Provides a Command Line Interface (CLI) that allows controlling the application through the daemon. The Docker host is made up of the Docker daemon, images, networks, containers, and volumes. The daemon is responsible for managing the requests between and for the existing Docker objects.

The Docker images consist of a simple template with instructions to create a Docker Container. Those images define the basis of the system as they can be created from scratch, registered, and used to create other containers with the same characteristics. The images are stored by the Docker Registry entity.

Containers can be defined as a relatively isolated environment in which an application or a part of it is executed. These objects only have access to resources that are defined in the image on which they are created. Sometimes these objects are misunderstood with Virtual Machines (VM), as they can run in the same infrastructure, with the hypervisor placed above the host Operational System, coordinating the guest systems [37] and abstracting the entire hardware server, while containers abstract the operating system kernel, resulting in a much faster and more lightweight instance. In Figure 2 [38], it is possible to observe the architectural difference between the two components.

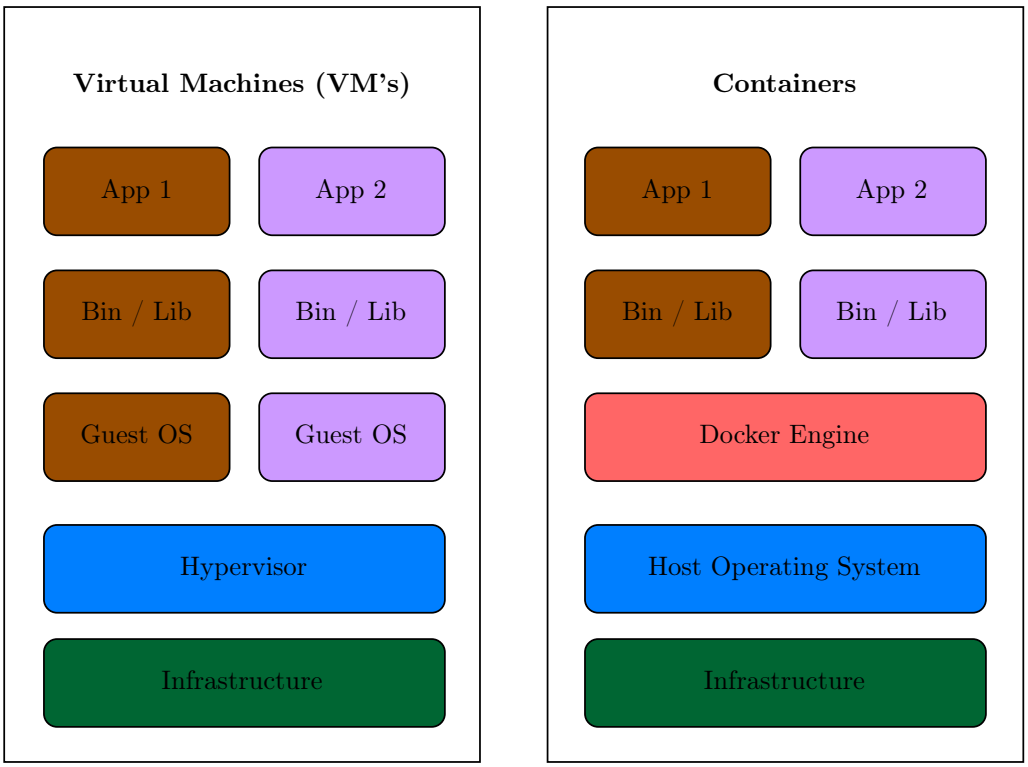

Fig. 2. Architectural comparison between Virtual Machines and Containers.

Even though the Docker image and the container are closely related, the difference between them should be underlined, as the image, as stated previously, is a set of metadata required to run an application, and the container is an image's instance in execution.

As the Docker CLI is used to manage individual containers on a Docker engine, in which the client uses the command line to access the Docker daemon, the Docker-compose CLI is used to manage a multicontainer application. For that purpose, a YAML file is used to configure the application services and work in all environments (production, staging, development, and testing) [39]. To use Compose properly, it is necessary to follow the following steps:

- Define the Dockerfile.
- Define which services will be used in the application in the YAML file, running them all together in an isolated environment.
- Run the Docker command to start and run the entire application.

Using this approach, it is possible to develop the system in an automated testing environment, easily creating and destroying the testing environment with a few commands.

The Dockerfile is a file that contains a list of instructions to build a container image, which then can be used to produce images to be deployed to production by one or several applications.

3.3.2 Frameworks

As the default architecture was settled, the frameworks that would make up the website needed to be chosen based on the main objectives of this thesis. Therefore, some frameworks were considered for the different functionalities of the system, which were front-end, back-end, database, and web server.

Front-end

For the front-end, the following frameworks were considered: React, Angular, and VueJS. These are widely regarded as some of the best tools for building modern web applications due to their performance, scalability, and active community support. To find the most appropriate framework for this project, the following selection criteria were used:

1. **Design flexibility:** ability to create dynamic, responsive, and visually attractive user interfaces with a high degree of customization.
2. **Complexity:** The inherent complexity of the framework in terms of setup, development process, and maintenance.
3. **Reusability:** The framework's support for creating reusable components and modules that can be leveraged across different parts of the application.
4. **Documentation:** The existence and quality of official documentation, as well as community-driven resources.
5. **Learning Curve:** The time and effort required for developers to become proficient with the framework, particularly for those with JavaScript experience.

As such, React was chosen as the most appropriate framework for the project. The architecture of React is component-based, which provides a high degree of reusability, enabling developers to build encapsulated components that manage their own state and can be easily reused across the application. This modularity eases development and helps maintain a consistent codebase.

In addition, React has a large and well-established community that has created many learning materials, libraries, and tools to assist developers in addressing common challenges. The official documentation is excellent and frequently updated, which contributes to a smoother learning curve for developers familiar with JavaScript.

Back-end

The selection of an appropriate back-end framework for this project involved a comprehensive evaluation of several established frameworks, including Laravel, Django, Ruby on Rails, Flask, and Express. The evaluation process was guided by the following key criteria:

1. **Documentation:** The quality, clarity, and comprehensiveness of available documentation were crucial factors. Robust documentation facilitates efficient problem-solving, feature understanding, and implementation.
2. **Features:** The range and depth of built-in functionalities were assessed, including database interaction, middleware support, templating engines, and routing capabilities. A feature-rich framework can significantly accelerate development by reducing the need for custom component creation.
3. **Performance and Scalability:** The framework's ability to maintain stability and performance under high load and during scaling was a critical consideration.
4. **Security:** Built-in security features and protections against common vulnerabilities (e.g., SQL injection, cross-site scripting, cross-site request forgery) were carefully evaluated to ensure adherence to best security practices.
5. **Community and Ecosystem:** The size and activity of the developer community, availability of third-party packages, and long-term support were considered to ensure ongoing development and problem-solving resources.
6. **Learning Curve and Development Speed:** The ease of adoption for the development team and the framework's ability to facilitate rapid development were important factors.

After a thorough evaluation, Laravel was selected as the back-end framework for this project. The decision was based on several key advantages:

- **Elegant ORM:** Laravel's Eloquent ORM provides an intuitive and powerful interface for database operations, simplifying data management and querying.
- **Artisan CLI:** The built-in command-line interface tool, Artisan, facilitates rapid development by automating many common tasks.
- **Robust Security Features:** Laravel includes strong security features out-of-the-box, including protection against various common web vulnerabilities.

- **Scalability:** Laravel's architecture supports scalability, making it suitable for both small projects and large-scale applications.
- **Database Migrations:** Laravel provides an intuitive interface for database schema management through migrations, enabling smooth schema evolution over time.
- **Extensive Third-Party Package Support:** Laravel has extensive support for integrating third-party plugins, allowing a flexible and modular development.
- **Comprehensive Documentation:** Laravel's documentation is extensive, well-maintained, and user-friendly, supporting efficient development and problem-solving.

These factors, combined with Laravel's emphasis on developer productivity and elegant syntax, made it an ideal choice for developing a robust, secure, and scalable educational platform for sleep quality analysis. Furthermore, Laravel's strong community presence, large-scale documentation, and active discussion groups provide ample resources for continuous learning and problem-solving [40].

Database

For the database, the choice resided in the selection of a Structured Query Language(SQL) database or a NoSQL database. As SQL databases such as MySQL, Oracle, PostgreSQL, and Microsoft SQL Server, among others, are widely recognized for their ability to define and manipulate structured data using SQL, a powerful and standardized query language [41].

On the other hand, No-SQL databases such as MongoDB, Cassandra, and Amazon DynamoDB offer a more flexible approach to data management. They do not rely on fixed schemas, allowing for dynamic data structures, which makes them a popular choice for handling large volumes of unstructured or semi-structured data. Furthermore, NoSQL databases are a great choice for horizontal scalability and large, ever-changing data sets, making them suitable for applications with rapidly changing data and requirements for scaling across distributed systems.

After evaluating the system's needs for structured data management, data integrity, scalability, and speed, PostgreSQL was selected as the most suitable database solution. Although both PostgreSQL and MySQL provide table-based relational structures and SQL support suitable for web applications, PostgreSQL was favored due to its stability, advanced data-integrity mechanisms, usage of a buffer cache, and reliable replication features, which better align with the platform's need for consistent and extensible data handling [42].

Web Server

A reliable and efficient web server is essential for managing client-server communications, processing requests, and delivering content in a web-based application. Even though there are some options such as Apache HTTP Server, Nginx, Lighttpd, or others, two of them stand out as they are the most used nowadays in web development, which are Apache and Nginx, due to their stability, performance, and features.

Nginx was ultimately chosen as the web server for this project. Its lightweight and efficient architecture allows it to handle multiple concurrent requests within a single thread, providing faster content delivery and improved performance under high traffic loads. Nginx is particularly well-suited for modern applications that require high concurrency and low memory usage.

Additionally, Nginx has a strong market presence due to its security features, offering built-in mechanisms to mitigate Distributed Denial of Service (DDOS) attacks, API authentication, and protection at the application layer.

3.3.3 The System

In short, the base components chosen were the following:

- Frontend: ReactJS (v18.3.1)
- Administration Frontend: ReactJS (v18.3.1)
- Backend: Laravel (v9.0) running on PHP (v8.1)
- Database: PostgreSQL (v13)
- Web Server: Nginx (v1.29.3)

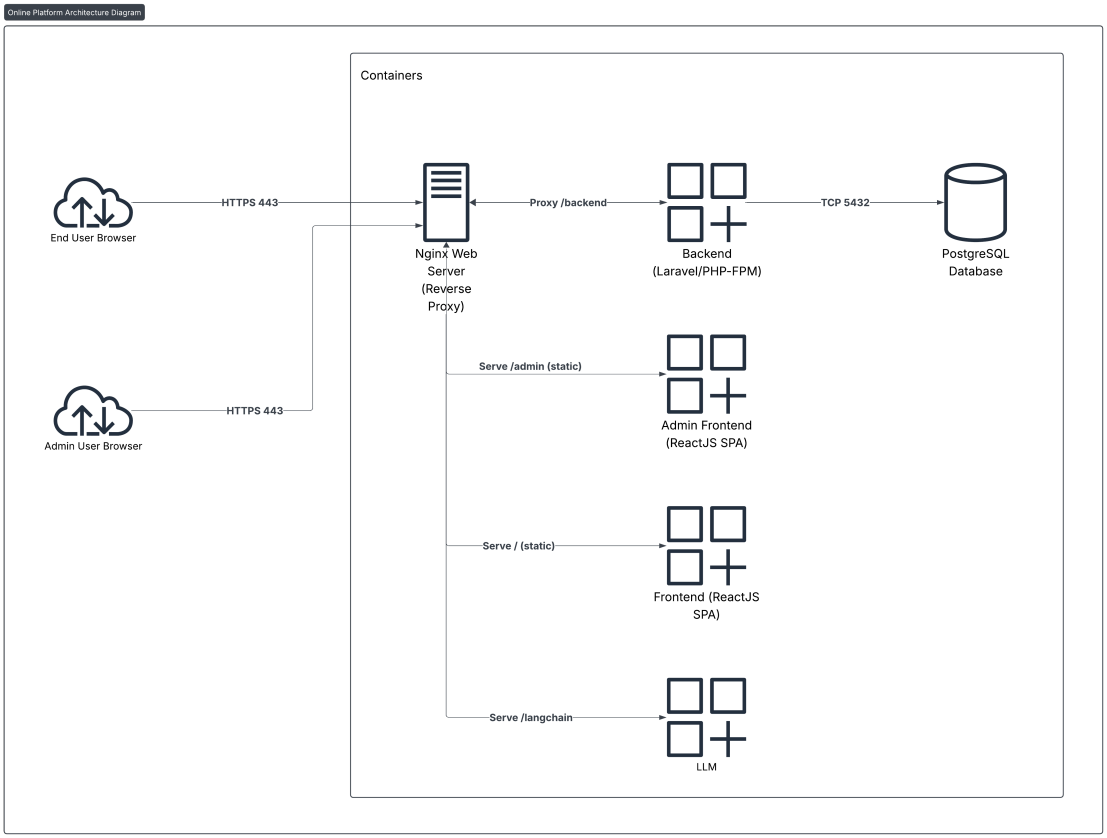

Fig. 3. Overall system architecture of the e-learning platform.

With the combination of Laravel and ReactJS, it is possible to develop complex and high-performance applications with a full-stack development strategy, as it is evolving with the web and has integrated several features like API authentication, real-time communication, and more, which can be useful in the long run.

The backend, using the Laravel framework, exposes a RESTful API [43] to communicate with the frontend. REST (Representational State Transfer) is an architectural style designed to create lightweight, scalable web services as an alternative to the Simple Object Access Protocol (SOAP) architecture. Some of the principles to be followed are as follows:

- The client-server mandate, where there is a clear separation between the client and the server.
- Statelessness, which defines the kind of commands to be used between the client and the server, should contain all the necessary information to obtain a successful response from the server.
- Layered system, in which multiple layers are used to expand the interface and the system.

Although it was possible to transition to CRUD (an acronym for Create, Read, Update, and Delete) from a RESTful API, it would increase the cost of development without substantially increasing the main characteristics already provided by the REST API.

On the client side, a Single Page Application (SPA) approach was chosen as an alternative to the Multipage Application (MPA) model. Unlike MPAs, which reload the full page with each navigation event, SPAs load resources once and dynamically update components as needed. This leads to faster user interactions, reduced server load, and lower maintenance overhead over time [44].

React, as the chosen front-end framework, has a good performance in creating SPAs. It's a virtual Document Object Model (DOM) that efficiently updates only the necessary parts of the page, resulting in faster rendering and a smoother user experience. React's component-based architecture aligns well with the SPA model, allowing for the creation of reusable UI elements that can be dynamically loaded and unloaded as needed.

The key benefits of using React for SPA development in this project include:

- Efficient DOM manipulation through the virtual DOM, leading to improved performance.
- Reusability of components, promotion of consistent design, and reduction of development time.
- Rich ecosystem of libraries and tools, facilitating the implementation of complex features.
- Robust state management options, such as React Context or Redux, for handling application-wide data.
- Server-side rendering capabilities, which can improve initial load times and SEO performance.

3.4 User Interface and User Experience Design

The design of the UI and UX significantly influences how effectively users interact with an e-learning platform. Given the complexity and importance of educational content in sleep analysis,

particular attention was paid to creating an intuitive, engaging, and accessible design tailored specifically to medical professionals and students.

3.4.1 Design Objectives

The primary design objectives to guide the UX/UI design were established by initial research and literature. The design should aim at providing ease of use, ensuring intuitive navigation through the platform, reducing the cognitive load, and minimizing the learning curve, allowing the user to focus on the provided content. Accessibility should be prioritized, compliant with standard accessibility guidelines, to ensure that the platform can adapt to the needs of users effectively.

The device's responsiveness is an important goal, as it should provide at all times the best user interaction regardless of the device used in the process, including tablets, smartphones, laptops, or others. Therefore, consistency across the platform creates a predictable and linear user experience, which also supports the user's proficiency. There should also be a focus on incorporating interactive elements to sustain user motivation and interest, motivating the continuous usage, and enhancing the overall learning effectiveness.

3.4.2 Design Methodology

The design process should have multiple stages, from the initial research through iterative refinement. Each stage is constructed to ensure continuous user interaction, with the integration of best practices in user-centered design. The process should have the following key stages:

- Research and analysis - Review academic literature and similar e-learning platforms to analyse standards and gaps in the current solutions.
- Information architecture - develop a user flow diagram to map the main user interaction scenarios and optimize their experience. In this process, the navigation and content layout should be structured for the end user.
- Wireframing and prototyping - creation of low-fidelity wireframes to explore interface layout and content hierarchy.
- Usability testing - conduct structured usability testing with target users, observe navigational behavior, and identify interface issues.
- Iterative refinement - Apply the feedback from the usability testing phase to improve the visual consistency in content and navigation structure.
- Design validation and handoff - Ensure all the feedback is considered and end the design process.

3.4.3 Key Components

There are a few key interface components needed in the most basic e-learning system to maximize user interaction and learning effectiveness. The user dashboard, presenting the information of user progress, recommended interactions, and saved information, is essential to work as a central

hub for the user data. The navigation structure needs to consider a well-structured menu system, easily responsive to other devices, with easy accessibility on all platform pages.

A data hub (or informational center), working as a library or a knowledge sharing hub, should be a consolidated interface where users could access learning materials on the theme, summaries, recent news, and articles. The platform's interactive module should be enriched with quizzes and courses (dynamically created by a platform moderator) supporting the active and passive learning styles, and encouraging self-guided exploration.

3.5 Data Collection Methods

To evaluate the effectiveness of the e-learning platform, a mixed-methods approach was used for data collection. Specifically:

1. System Logs: Automated collection of user interaction data, including time spent on modules, quiz scores, and feature usage patterns.
2. Surveys: Pre-test and post-test questionnaires to assess user knowledge, confidence, and satisfaction with the learning experience.
3. Performance metrics: Collection of technical performance data such as load times, response rates, and system stability indicators.
4. Learning Analytics: Analysis of user progress through the course, identifying areas of difficulty and success in the learning materials.

3.6 Usability Testing

The usability testing involved university students, as they represent one of the platform's primary targets. Medical staff or professionals, although being another primary user target, are not included.

The testing methodology consisted of two structured questionnaires: a pre-test and a post-test (see appendices B and C respectively). These were constructed using principles from usability frameworks and tailored to the context of digital learning in the medical domain. The choice of questions for each questionnaire was guided by the need to assess both general digital literacy and specific usability attributes, such as clarity, consistency, and perceived educational value. Also, the questions were chosen with caution to not gathering any personal information.

The pre-test questionnaire was developed to check participants' baseline familiarity with e-learning systems, confidence in digital tools, and interest in the area. The post-test questionnaire focused on usability criteria such as navigational ease, visual design coherence, content clarity, responsiveness, and educational effectiveness. The statements were formulated according to the best practices in UX research and rated using a five-point Likert scale to allow the interpretation of perceived usability in quantitative analysis.

Participants are invited to interact freely with the platform, without predefined tasks, to promote natural exploration and spontaneous feedback. They are encouraged to ask questions or

voice concerns, fostering exploratory dialogue while minimizing the influence of structured guidance. During the sessions, no personal data is collected, and the feedback collected through the post-test questionnaires is used to identify areas of improvement and inform iterative design refinements. This feedback loop ensures the platform's development remains aligned with the user's needs and expectations in a real-world scenario.

3.7 E-learning Methods

The e-learning platform incorporates various pedagogical strategies to enhance the learning experience. Through multimedia integration, the platform supplies a combination of text, images, videos, and interactive simulations to reach various learning preferences and ease the comprehension of complex subjects and concepts. In addition, an adaptive component was implemented through algorithms that modify the difficulty of learning materials according to each user's performance and progression. This mechanism was designed to dynamically tailor the learning path, allowing for individualized instruction that reflects the learner's current proficiency level. The principles of microlearning are constantly applied at the content structure. Although it depends on the content designer's approach, it can be broken down into more focused and manageable units, facilitating user retention and accommodating any possible time constraints of users (mainly professionals).

To stimulate collaborative learning through professional discourse and among peers, discussion forums and peer-review mechanisms are employed, which in turn enhance the learning atmosphere as well as the knowledge generated by the course. Using case-based learning also helps close the gap between theoretical knowledge and practical application, fostering expertise and developing critical thinking skills among learners.

Repetition, a fundamental component of effective learning, is systematically implemented through algorithms that periodically reintroduce previously covered material, reinforcing knowledge retention over the long term. Central to the platform's functionality is the outcome of the learning evaluation. Having a sensitive theme, like sleep (or sleep medicine), it is essential that the evaluation process is capable of assessing the user in their interactions, but also provides personalized feedback, to encourage continuous learning. The methodology proposed combined traditional e-assessment strategies with AI-driven techniques, more particularly LLM, improving the evaluation of user performance.

The evaluation process aims to provide real-time, constructive feedback that helps in information reinforcement and addressing knowledge gaps, to adapt the learner's progress through continuous assessment, making a progression path suitable for the user's capabilities. But to do that, one should also accurately measure the user's comprehension of key concepts in the subject provided.

The platform, at its start, should support several types of learning assessments:

- Binary Choice: Yes or No, True or False, or any other pair of responses.
- Multiple Choice: used for quick checks of factual knowledge.
- Free Text Responses: Evaluated using LLM, enabling a deeper insight into user understanding.

- Case-based scenarios: Real or simulated scenarios where learners should construct a complex response (diagnosis or procedural decisions).
- Progress Quizzes: Creation of quizzes to evaluate the long-term retention, detecting the learning curve (increasing or decreasing).

3.8 Learning Evaluation

The learning evaluation in the platform is made by using the LLM to apply and grade the quizzes individually, leveraging advanced natural language processing capabilities to improve assessment efficiency and provide real-time individualized feedback.

LLMs are sophisticated AI systems that understand, interpret, and generate text-based language that is human-like. They were selected to be included in this e-learning environment as they can automate complicated assessment activities, offer immediate individual feedback, and enrich the interactive educational process. LLMs support sophisticated text understanding, enabling deeper analyses of user-generated content, particularly in free-text responses, case-based scenarios, and complex reasoning problems. This feature greatly enhances assessment efficiency, objectivity, and accuracy. LLM must be implemented as an embedded container in the Docker architecture, where the communication between LLM and Backend containers is based on the REST API. The integration of an LLM can facilitate various educational activities such as:

- **Learning content generation**: Generate learning content (questions or general text) based on the needs of the user to ensure content relevance and personalized engagement.
- **Automated grading**: Automatically grade user responses to quizzes and exercises, reducing manual grading workload and ensuring consistent evaluations.
- **Real-time feedback**: Allow real-time personalized feedback concerning mistakes, misconceptions, and suggestions for improvement.
- **Adaptive learning paths**: LLM can dynamically adjust the difficulty and content of learning materials based on the user's performance metrics.
- **Interactive chatbot**: Use LLMs to establish an online chatbot to answer user questions and offer real-time help, effectively promoting user engagement.

With this strategic use of LLMs, the platform significantly enhances educational quality, engagement, and scalability, ensuring consistent service provision applicable to the dynamic and varied needs of its users.

3.9 Summary

Based on the related work presented in the previous chapter, this chapter describes the research design, development line, and evaluation methods used in developing the e-learning environment. A composite evaluation methodology, including quantitative measures on system performance and qualitative user experiences, was therefore defined to perform a thorough evaluation and gain different perspectives from users. The use of a modular, containerized design and incremental approach to development also allowed for scalability and flexibility.

With this methodological foundation in place, the following chapter endorses the actual adaptation and design process, translated into actual code in the form of the implementation section.

4 Development

The development process involved the use of web technologies to construct a modular, scalable, and accessible platform that provides dynamically generated content. The following development phases are not in order, since some of the processes were merged and executed simultaneously, as the Implementation and Design.

4.1 System Architecture

The platform uses a modular client-server model with Docker-based containerization. Each component operates independently, ensuring flexibility, maintainability, security, and deployment consistency. The overall system architecture is illustrated in Figure 3 (see Section 3.3.2).

The main reason for such implementation is due to modularity itself. The implementation suggests that the same solution can be modular and replicated without conflict of information, since there is isolation of services to each cluster. This enhances the platform security as there is only a single point of entry through Nginx, which minimizes the potential attack vectors.

To reinforce this protection, a rate-limiting middleware was configured at the API gateway layer to control the number of requests originating from a single Internet Protocol (IP) address within a defined time window. This mechanism mitigates denial-of-service (DoS) and brute-force attacks by preventing excessive or malicious traffic, helping maintain stable performance and reliable access for legitimate users.

As mentioned, the scalability with this architecture elevates the platform's potential to other subjects or application areas, as it can be scaled independently. This also means that the developer effort can be reduced due to the containerized setup, which is the same for every environment.

In this architecture, it's important to mention the database and backend security regarding access. Only the backend is authorized to access the database information, since the container is not publicly accessible, reducing also the risk of direct SQL access threats.

4.2 Implementation

Docker containerization was needed for this platform development and deployment process. Having each service encapsulated (backend, frontend, frontend-admin, database, langchain, and web server) within its own container, it was possible to easily develop, test, debug, and prepare for deployment, since the environment was consistent across the environment.

Each service was defined by a dedicated Dockerfile, which specifies the image, necessary packages, and environment configurations. Some services can have a dedicated Dockerfile for production or another development stage.

The YAML configuration file, which described how the containers should work with each other, was configured as:

```
services:
  app-backend:
```

```
    user: root
    build:
      context: ./app-backend
    container_name: app-backend
    restart: always
    ports:
      - "5090:9000"
    volumes:
      - ./app-backend:/var/www
      - /var/www/vendor
    environment:
      - DB_CONNECTION=pgsql
      - DB_HOST=postgres
      - DB_PORT=5432
      - DB_DATABASE=db_name
      - DB_USERNAME=db_user
      - DB_PASSWORD=db_secret
    networks:
      - app-network

  frontend:
    build:
      context: ./frontend
      dockerfile: ./Dockerfile.prod
    container_name: frontend
    restart: always
    ports:
      - "5300:3000"
    environment:
      - CHOKIDAR_USEPOLLING=true
      - FAST_REFRESH=false
      - WATCHPACK_POLLING=true
    networks:
      - app-network

  frontend-admin:
    build:
      context: ./frontend-admin
      dockerfile: ./Dockerfile.prod
    container_name: frontend-admin
    restart: always
    ports:
      - "5301:3001"
    volumes:
      - ./frontend-admin:/app
      - '/app/node_modules'
    environment:
```

```
    - CHOKIDAR_USEPOLLING=true
  stdin_open: true
  networks:
    - app-network
  command: npm run dev

nginx:
  build:
    context: ./nginx
  container_name: app-nginx
  ports:
    - "5004:80"
  volumes:
    - ./nginx/nginx.conf:/etc/nginx/nginx.conf
    - ./nginx/conf.d:/etc/nginx/conf.d
  depends_on:
    - app-backend
    - frontend
    - frontend-admin
    - postgres
    - langchain
  networks:
    - app-network

postgres:
  image: postgres:13
  container_name: postgres
  environment:
    POSTGRES_DB: laravel
    POSTGRES_USER: laravel
    POSTGRES_PASSWORD: secret
  ports:
    - "54320:5432"
  volumes:
    - pgdata:/var/lib/postgresql/data
  networks:
    - app-network

langchain:
  build:
    context: ./langchain
    dockerfile: ./Dockerfile.prod
  container_name: langchain
  restart: always
  environment:
    - PYTHONUNBUFFERED=1
    - LANGCHAIN_TRACING_V2=true
```

```
      - LANGCHAIN_ENDPOINT=langchain_endpoint
      - LANGCHAIN_API_KEY=langchain_api_key
      - LANGCHAIN_PROJECT=langchain_project
    ports:
      - "5400:4000"
    volumes:
      - ./langchain:/app
    depends_on:
      - app-backend
    networks:
      - app-network
networks:
  app-network:
    driver: bridge

volumes:
  pgdata:
```

This approach allowed to reduce the reduce the overhead during the development, orchestrating a full environment spin-up with a single command.

All traffic originating from outside was routed through a reverse proxy container (Nginx), forwarding requests to the respective service containers, based on the request route. This architecture ensured clear isolation, preventing unauthorized connections from bypassing security layers. Environment variables were defined in `.env` files (and passed at run time), providing essential content to be provided to a specific service.

4.2.1 Backend

The backend layer is responsible for handling all the data transactions, management, and integration by REST endpoint to all the necessary containers to integrate into the informational schema.

Laravel framework implements its MVC architecture to the platform, and along with the usage of the Eloquent Object–relational mapping (ORM), offers a simple way to handle data, implement rules (access or management), promoting scalability and maintainability.

The structure of the backend architecture was:

```
app-backend/
├── app/
│   ├── Helpers/
│   ├── Http/
│   │   ├── Controllers/
│   │   ├── Middleware/
│   │   └── Resources/
│   ├── Models/
│   └── Notifications/
├── config/
└── database/
```

```
    ├── migrations/
    └── seeders/
├── public/
├── routes/
├── composer.json
└── Dockerfile
```

All the routes used inside the backend container are grouped under the dedicated file web.php (which is exposed by the internal Laravel routing mechanism to `/`). This was made to provide better management by centralizing all the routes through the predefined route `/backend` by defining a route prefix in the group association.

All the routes necessary to get, update, save, and delete were simplified by using the resource(s) method from Laravel. This cleans the code, increases security, and lowers the possibility of human error. Only the dedicated functions were declared separately.

The controllers followed the naming nomenclature, based on the model entity, using a design aligned with the Laravel RESTful approach. When a controller is created with the `--api` flag in Laravel, five functions are always created: `index`, `show`, `store`, `update`, and `destroy`. This allows for a dynamic association of the route name to its function when using the `resources`. For example, in application to users:

```
Route::resource('users', 'App\\Http\\Controllers\\UserController');

// Internally maps to:
// GET    /users           -> index()
// GET    /users/{user}    -> show()
// POST   /users           -> store()
// PUT    /users/{user}    -> update()
// DELETE /users/{user}    -> destroy()
```

This automatic association of routes to controller methods enhances maintainability, since developers only need to extend or override the default logic within these methods whenever additional functionality or custom parameter validation is required.

To improve security in the platform, Laravel Sanctum was implemented for token-based authentication, ensuring that, when applying the rule, all the requests to the backend endpoint could be validated securely. Other security measures were implemented regarding the token expiration time. `CheckTokenExpiry` Middleware was created for that purpose, which consists basically of checking if the user is authenticated, is not an administrator, and if the token date is still valid.

To allow the users to access the system, a role-based authentication model was integrated, restricting access to certain endpoints to a certain role. One rule implemented regarding the authorization and access was the middleware: `CheckMaintenanceMode`. This middleware verifies if any of the administrators have enabled the maintenance mode. If so, it allows only the traffic for the administrator, sending a maintenance flag to the frontend.

4.2.2 Frontend

This layer has the objective of providing an intuitive, responsive, and accessible interface to all users engaging with the e-learning platform. This layer, as mentioned previously, was developed using ReactJS.

The structure of the frontend architecture was:

```
frontend/
├── public/
│   └── index.html
├── src/
│   ├── context/
│   ├── locales/
│   │   ├── en/
│   │   └── pt/
│   ├── providers/
│   │   ├── axiosConfig.js
│   │   ├── apiUrl.js
│   │   └── authProvider.js
│   ├── resources/
│   │   ├── components/
│   │   ├── media/
│   │   ├── routes/
│   │   └── utils/
│   ├── App.js
│   ├── routes.js
│   ├── i18n.js
│   └── index.js
├── package.json
├── tailwind.config.js
├── Dockerfile
└── Dockerfile.prod
```

This displays the general organization of the files and structure for several nuclear functionalities, like translations, routing management, and reusable code.

4.2.3 Database Design

PostgreSQL was chosen as the database of choice due to its proven performance, guaranteed data integrity, and support for complex SQL queries. It was well-suited for a scalable e-learning environment thanks to its strong support for advanced relational models, indexing strategies, and concurrent transactions.

The general schema was structured to handle the modularity necessary for this platform, allowing flexible association between users, courses, modules, quizzes, and other resources. Particular attention was paid to normalization and indexing to ensure both data consistency and performance under real-world usage.

The complete entity-relationship diagram (ERD) is presented in Appendix F, outlining the major entities, relationships, and constraints implemented. This diagram includes important junction tables and foreign key, preserving the relational data integrity for queries.

The user subsystem is a core piece of the schema, used to keep track of user profiles, preferences, favorites, and analytical data. This can range from personal data and role-based access to individual performance metrics for measuring the effectiveness of learning and engagement, and recording user-specific interactions such as bookmarked content, completed courses, and quiz history.

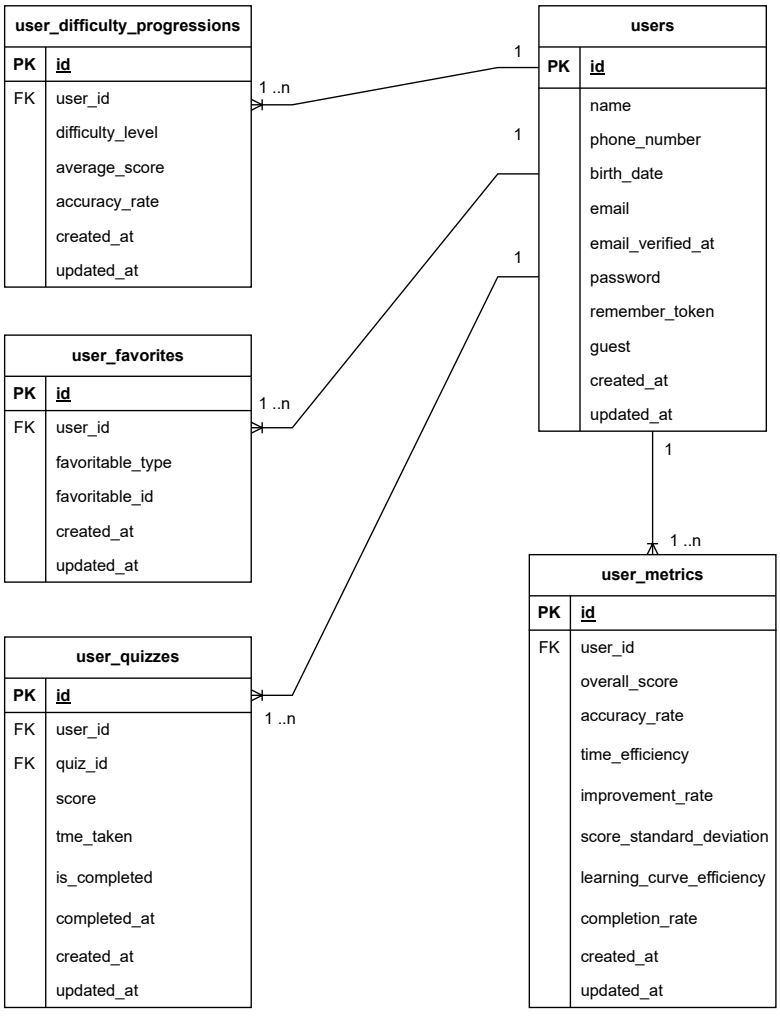

Fig. 4. Entity-Relationship Diagram for the User Subsystem.

This subsystem manages the dynamic quiz generation and quiz history track in the intermediate table `user_quizzes`. This subsystem includes the quiz's history, quiz metadata, questions, and topics. Relationships between users, quizzes, and their responses enable the platform to generate reports, track progress, and adapt future quiz difficulty to user performance.

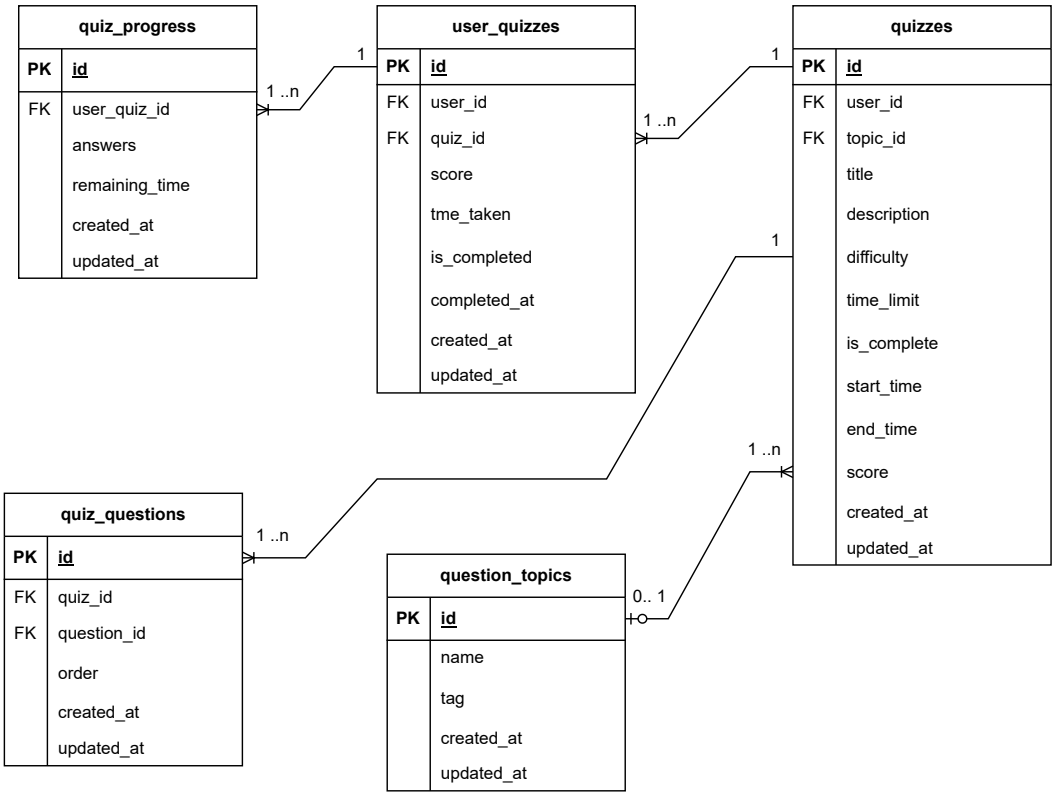

Fig. 5. Entity-Relationship Diagram for the Quizzes Subsystem.

The courses subsystem manages the hierarchical structure of the educational courses. This includes lessons, modular content, and interactive elements. It's characterized by progress tracking, subscriptions, ratings, and modular content. This structure ensures modularity and ease of content personalization.

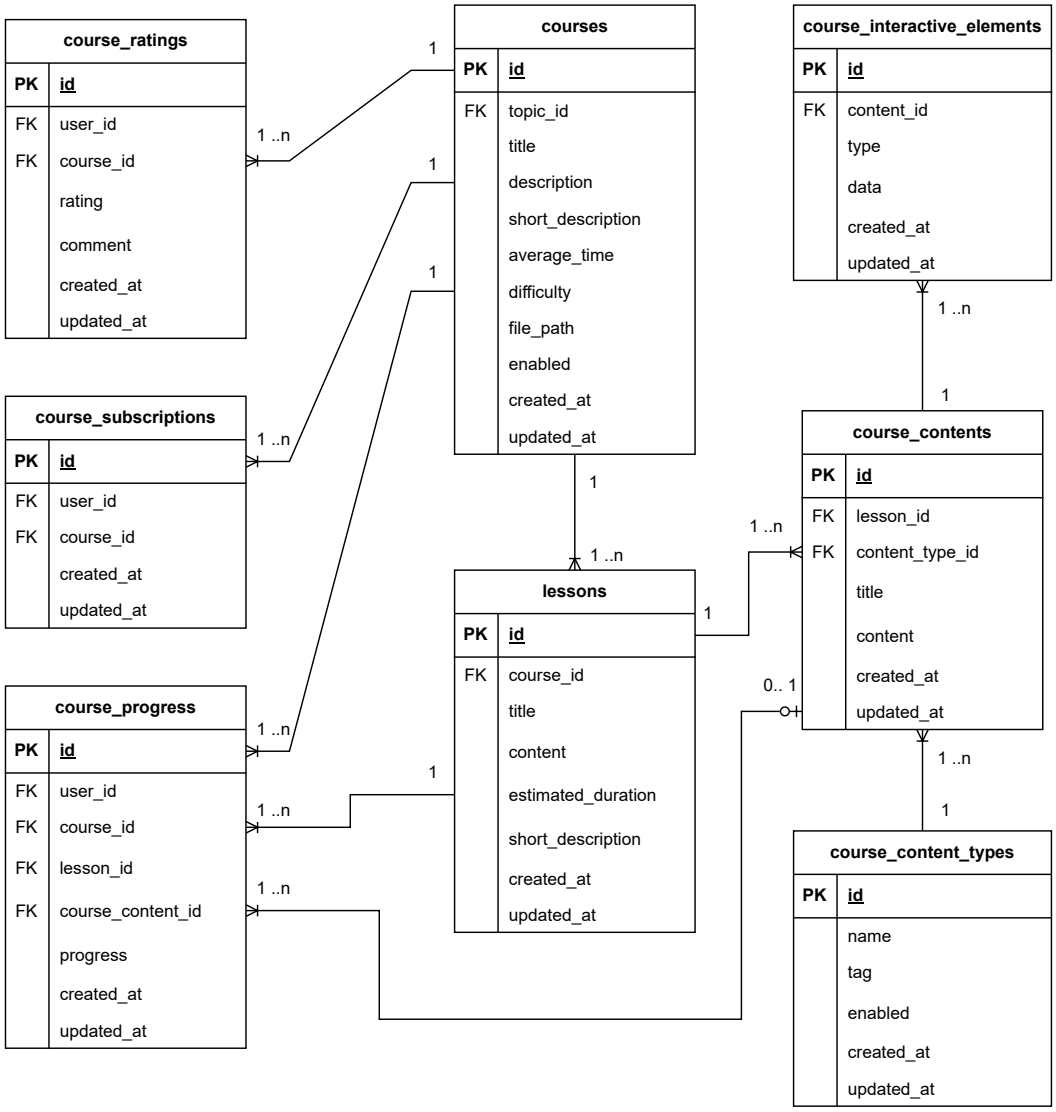

Fig. 6. Entity-Relationship Diagram for the Course Subsystem.

The library subsystem provides for the efficient handling, browsing, and presentation of static reference materials that are categorized and related to topical library pages. Every page consists of several content modules, promoting organized and modulated information delivery in the context of educational topics. The modular design allows for maximum flexibility, with content updated and maintained with little overhead, and users can easily navigate and find content. Although primarily designed as a non-interactive resource hub, the library subsystem provides comprehensive and curated educational e-learning content, significantly enriching learners' access to supplementary knowledge on the platform.

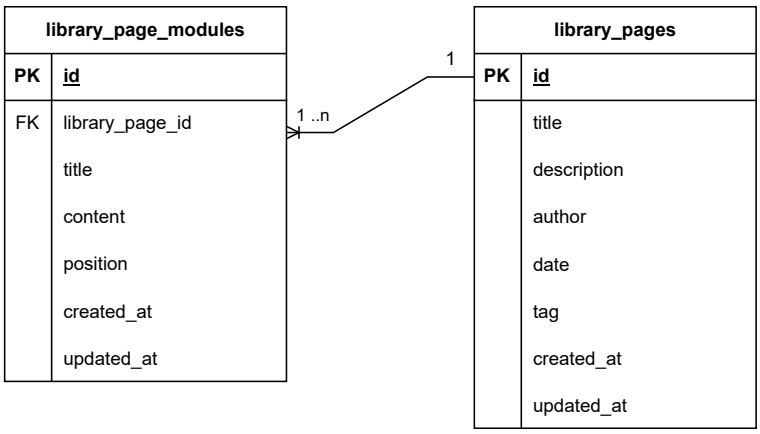

Fig. 7. Entity-Relationship Diagram for the Library Subsystem.

The forum subsystem is responsible for managing the social modules of the platform, which include discussion threads, user interactions, and content categorization within the platform. It is organized into five main entities: forum posts, forum threads, forum categories, and the corresponding like systems for posts and threads. This database design supports a flexible, interactive, and user-centered discussion environment promoting structured conversation.

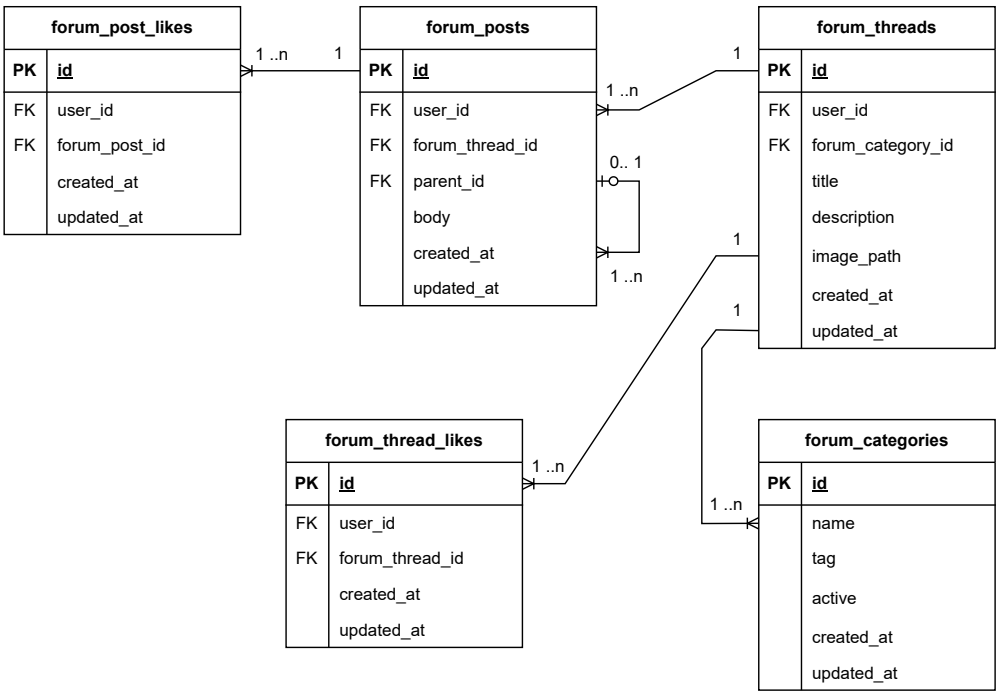

Fig. 8. Entity-Relationship Diagram for the Forum Subsystem.

The question subsystem is designed to support the dynamic management, categorization, and evaluation of assessment questions within the platform. The structure was built around the central question entity, storing detailed information about the individual questions, including the type, difficulty, textual context, and supplementary data to be used if needed, such as hints, explanations, and associated images.

The central entity is linked to relevant topics through a dedicated intermediate table `question_question_topic`, allowing flexible many-to-many associations with entries in the `question_topics` table. Each question can also be classified by its type, such as multiple-choice, free-text, or others, enabling the assessment logic.

Possible answers or options related to each question are managed in the `question_options` table, particularly essential for structured question formats like multiple-choice or yes/no questions.

User interactions and responses to these questions during quiz attempts are systematically recorded in the `question_responses` entity, capturing key assessment data such as correctness, response time, response quality scores, and providing suggested answers for feedback purposes. This subsystem is integrated with the quiz subsystem by the intermediate table user_quizzes, ensuring the learner tracking at the granular question level, supporting the adaptive learning techniques implemented.

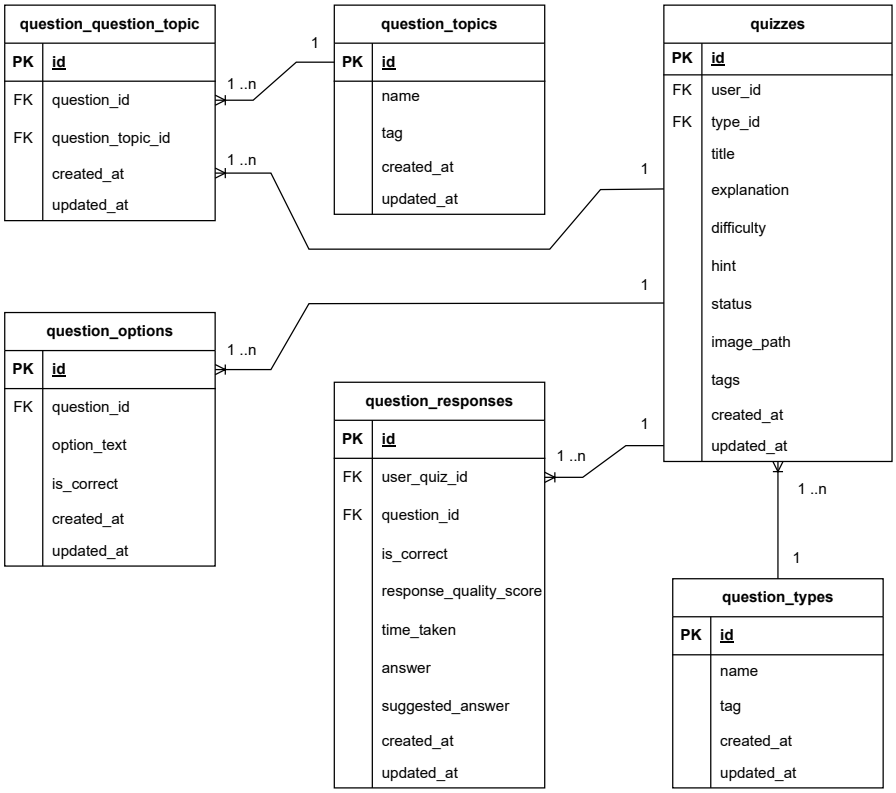

Fig. 9. Entity-Relationship Diagram for the Question Management Subsystem.

In such a complex structure, the performance was enhanced by indexing the most queried columns in tables, such as primary keys, foreign keys, or user IDs. This allows to have an optimized response time, having minimal effects on the platform processes.

4.2.4 Frontend Admin

This service layer was developed similarly to the frontend layer. Its structure is very similar, since the base is React, but adding the react-admin package, with a slight difference of using TypeScript. This is responsible for managing all the content provided within the platform and performing privileged tasks, such as managing users and content moderation. Here is also possible to manage all the contact requests and reports provided by the end users.

The react-admin package, being an open-source software, provides a predefined template for all the components and layouts, to easily manage and develop new components without much effort, with the ability to adapt the core functionalities to a specific task.

Because the Admin frontend is containerized separately, it runs independently from the main user interface, being only accessible through the prefix `/admin`. This approach enhances security by allowing network-level and role-based access controls specific to administrators. It also simplifies the development workflow, as updates to the admin interface can be deployed and tested independently, without any effect on the main service layer.

4.2.5 Web Server

The web server was developed using NGINX, chosen primarily for its lightweight architecture, high concurrency capabilities, and robust feature set to handle complex HTTP routing.

The routing and management behavior is primarily controlled by two files, which are necessary to override: `nginx.conf` and `default.conf`. They are responsible for setting general configurations and setting domain-specific directives, respectively.

The **nginx.conf** is:

```
user  nginx;
worker_processes auto;

error_log  /var/log/nginx/error.log warn;
pid        /var/run/nginx.pid;

events {
    worker_connections  1024;
}

http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    log_format  main  '$remote_addr - $remote_user [$time_local] '
                      '"$request" $status $body_bytes_sent '
```

```
                    '"$http_referer" "$http_user_agent" '
                    '"$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile        on;
    keepalive_timeout  65;

    gzip              on;
    gzip_comp_level   5;                        # Compression level (1  9 )
    gzip_proxied      any;                      # Compress responses for
        all proxied requests
    gzip_min_length   1024;                     # Only compress files   1
        KB
    gzip_buffers      16 8k;                    # Buffer configuration
    gzip_types        text/plain
                      text/css
                      application/json
                      application/javascript
                      text/xml
                      application/xml
                      application/xml+rss
                      image/svg+xml;            # MIME types to compress
    gzip_vary         on;

    # Include configuration blocks from conf.d
    include /etc/nginx/conf.d/*.conf;
}
```

This file configuration declares several global directives, including:

- Worker processes: Dynamically set to maximize the CPU utilization
- Logging: Defines custom log formats and where they are stored.
- File serving: Enables `sendfile` for efficient static file delivery.
- Keep-alive timeout: Balances the performance with resource usage by controlling the open time of an idle connection.
- Include files: Pulls additional configs from the specified folder.

The **default.conf** file is:

```
server {
    listen 80;
    listen [::]:80;

    server_name brocha.biesalab.org;
```

```
    location ~* \.(?:ico|css|gif|jpe?g|png|js|webp|woff2?|eot|ttf|svg|otf)$
        {
        proxy_pass        http://frontend:3000;
        proxy_http_version 1.1;
        proxy_set_header  Upgrade $http_upgrade;
        proxy_set_header  Connection "upgrade";
        proxy_set_header  Host $host;
        proxy_set_header  X-Real-IP $remote_addr;
        proxy_cache_bypass $http_upgrade;
        expires           1y;
        add_header        Cache-Control "public, max-age=31536000,
            immutable" always;
    }

    location / {
        proxy_pass http://frontend:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_cache_bypass $http_upgrade;
    }

    location /admin {
        proxy_pass http://frontend-admin:3001;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }

    location /backend {
        include fastcgi_params;
        fastcgi_pass app-backend:9000;
        fastcgi_param SCRIPT_FILENAME /var/www/public/index.php;
        fastcgi_param PATH_INFO $fastcgi_path_info;
        fastcgi_param PATH_TRANSLATED $document_root$fastcgi_path_info;
    }

    location ~ \.php$ {
        include fastcgi_params;
        fastcgi_pass app-backend:9000;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param PATH_INFO $fastcgi_path_info;
    }
```

```
    location /langchain/ {
        proxy_pass http://langchain:4000/;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

The presented configuration has the following routing directives:

- `location /`: The public traffic is directed to the frontend container, which listens to port 3000 in its container.
- `location /admin`: The traffic requests to the administrative frontend container (frontend-admin). This separation ensures that the container can be restricted with additional or separate authentication or network policies.
- `location /backend`: Uses FastCGI to pass requests into the Laravel application. This route is used for all RESTful API calls (prefixed with `/backend`.
- `location /langchain`: It proxies all the AI-generated content and requests to the langchain, ensuring the LLM features are modular and separate from the main application logic.
- `PHP handling`: Although it's redundant in this case scenario, it was added to handle legacy cases or edge cases using PHP scripts.

When a user visits the platform hosted in the domain `brocha.biesalab.org`, NGINX checks the `server_name` directive and listens on ports 80 (with or without SSL). Based on the `location` block, requests are then forwarded to the corresponding container, which makes it straightforward to scale or replace any service without altering external endpoints.

As the previous statement in the request flow, it can be unusual regarding the SSL requests, it's because the server's Apache configuration for the domain redirects to port 5004 (assigned to the NGINX in the docker-compose file), whether the request is by HTTP or HTTPS. Meaning that the SSL configuration is not set in the NGINX but in the Apache configuration, although it could be centralized in a single configuration file.

The advantages of the used solution are:

- **Single Entry Point**: All requests funnel through NGINX, simplifying logs, security policies, and potential load balancing in multi-container scenarios.
- **Isolation and Scalability**: The frontend, frontend-admin, backend, and LLM each run independently, enabling quick scaling of any service as demand grows.

– **Maintainability**: Container naming references (`frontend:3000`, `langchain:4000`, etc.), without specifying directly the hostname, can be updated via Docker Compose without rewriting domain-level server blocks.

4.2.6 Large Language Models

The LLM component, developed with Langchain, has the purpose of introducing modularity to the platform. This is the component responsible for all the dynamic generated content in the platform, including theme topics, questions (content and options, if they exist), and is also the component responsible for the assessment of the dynamic generated quizzes in the platform.

The integration is based on the **OpenAI GPT-3.5 Turbo** model, accessed remotely through the OpenAI API rather than a locally deployed instance. This setup allows the platform to leverage the model's natural-language understanding and generation capabilities without imposing computational demands on the local infrastructure. The backend communicates securely with the model through asynchronous REST API requests, ensuring that no personal or sensitive user data is transmitted during inference. This configuration also facilitates modular development using LangChain, enabling the creation of structured and reusable prompt chains that define specific tasks such as question generation, quiz evaluation, and interactive content responses.

Also incorporated in this component is the live chat, to generate information content based on the user's questions and answers. For these functionalities, prompt engineering techniques were used to maintain consistency, modularity, effectiveness of the responses, and contextual accuracy across the outputs. These prompts were developed to be easily integrated by the backend container over the internal Docker network. The LLM component has three main functions, namely the live chat, the question generator, and the quiz evaluation.

Live chat

This functionality enables users to ask questions about the platform theme. The chat remains context-aware by tracking conversation history and only allows theme-relevant dialogue. The prompt structure for this feature dynamically incorporates the ongoing conversation and ensures that any off-topic queries are gently redirected.

```
conversation_prompt = "\n".join([
    f"{msg.user}: {msg.text}" for msg in conversation
])
full_prompt = f"""You are an expert on {theme}. Only answer questions that
    are directly related to {theme}. If the user's message is irrelevant or
     attempts to divert from the theme, politely remind them to stay on
    topic in their language.

Ongoing conversation:
{conversation_prompt}

User: {message}
```

```
System:"""
```

Listing 1. Prompt for context-aware live chat

Question Generator

This component generates quiz questions dynamically for a specified topic and platform theme. It supports three question types: Yes/No, Multiple Choice, and Free Text. Each question is enriched with metadata, such as difficulty level (on a scale from 1 to 100), relevant tags, and optionally, topic tags. Multiple Choice questions include between 4 to 8 options, each labeled with correctness flags. The output is returned in a JSON format suitable for direct integration and handling by the backend.

```
topic_string = ""
if topic_tag:
    topic_string = f" Focus on the following topic: {topic_name}."

full_prompt = f"""You are an expert on {theme}.{topic_string}
You need to generate a variety of quiz questions.
There are three types of questions:
    1. **Yes/No**: A question where the answer can be either "Yes" or "No".
    2. **Multiple Choice**: A question with multiple predefined options,
        where one or more options can be correct, up to 8 options.
    3. **Free Text**: A question where the answer will be written by the
        user.

Generate 10 questions in the following JSON format, with a mix of question
    types. If the question is Multiple Choice or Yes/No, include the
    correct answer(s). For Free Text, leave the options blank.
For the multiple choice questions, you can provide up to 8 options, but no
    less than 4.
The difficulty of the questions can range between 1 and 100, to have a more
     specific assessment. The higher the number, the more difficult the
    question is.
Return only the JSON objects:

[
    {{
        "title": "The question text",
        "type": "multiple_choice", // or "yes_no" or "free_text"
        "difficulty": 1, // Difficulty rating from 1 to 100
        {f'"topic_tag": "{topic_tag}",' if topic_tag else ''}
        "tags": ["tag1", "tag2"], // Relevant tags
        "options": [ // If type is "multiple_choice" or "yes_no", provide
            options
            {{
                "option_text": "Option A",
```

```
                "is_correct": true // Mark true if this option is correct
            }},
            {{
                "option_text": "Option B",
                "is_correct": false
            }}
        ]
    }}
]
"""
```

Listing 2. Prompt for generating quiz questions

Quiz Evaluation

The quiz assessment is defined by the following code:

```
    full_prompt = f"""You are an expert teacher on {theme}.
    Evaluate the student's answer to the question below.
    Question: {question}
    Answer: {answer}
    Evaluate the student's answer and provide the following:
    1. Is the student's answer correct? Respond with 'Yes' or 'No'.
    2. Provide a response quality score between 0 and 1, where 1 means the
        answer is completely correct, and 0 means it is incorrect.
    3. Detailed feedback that outlines what the student did well, areas for
         improvement, and a complete suggested answer for the question.
    Your response should be in the following JSON format:
    [
        {{
            "is_correct": true or false,
            "response_quality_score": float between 0 and 1,
            "feedback": "Detailed feedback including a suggested answer to
                the question."
        }}
    ]
    """
```

Listing 3. Prompt for quiz answer evaluation

4.3 User Interface and User Experience

Following the guidelines and presented methodology defined previously, and building upon the previously defined design objectives, the development of the UX/UI required creating intuitive, coherent, and responsive user experiences. Attention was given to the adaptation of the interfaces for the various screen sizes and device contexts.

4.3.1 Information Architecture

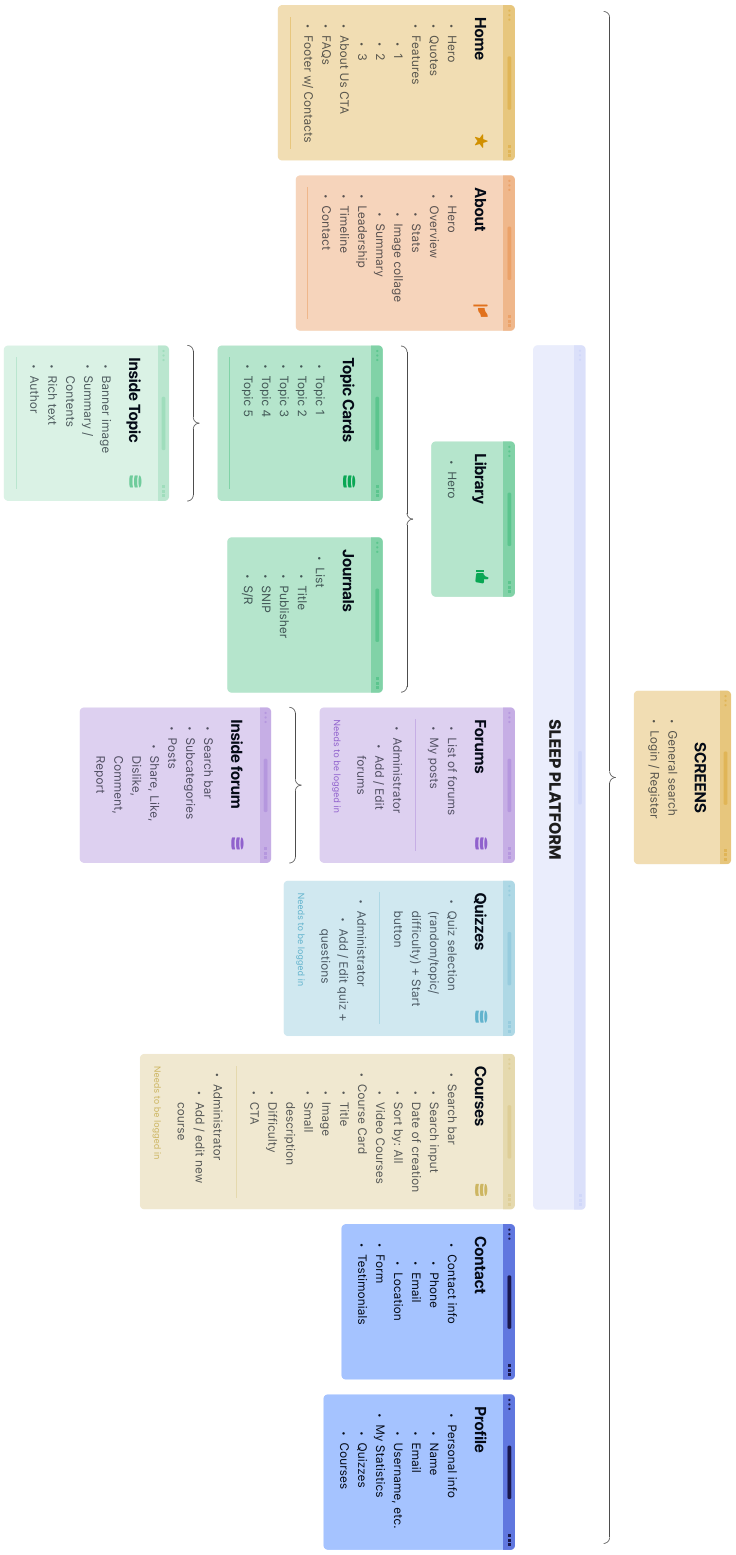

Fig. 10. Information Architecture Diagram.

The information architecture developed during the design phase was implemented through structured layouts and navigation components. It follows a hierarchical organization, starting with general unauthenticated pages (Home, About, and Contact), which provide introductory information and onboarding pathways for new users. Authenticated users access functional areas of the platform, including the Library, Forums, Quizzes, Courses, and Profile sections. This modular architecture ensures a consistent user experience and facilitates content exploration and learning engagement. Figure 10 illustrates the implemented model.

4.3.2 Low Fidelity Prototyping

The transition from design prototypes to functional interfaces was conducted by using Figma. These prototypes represented initial explorations of visual hierarchy, layout spacing, and content distribution, to be used as a reference for the frontend development.

Figure 11 illustrates the wireframes developed for the landing page. This design emphasizes clear information prioritization, readability, and intuitive navigation. In addition to the landing page, the low fidelity prototypes defined structural layouts for essential unauthenticated pages such as the about page, contact page, error page, and the login and registration pages.

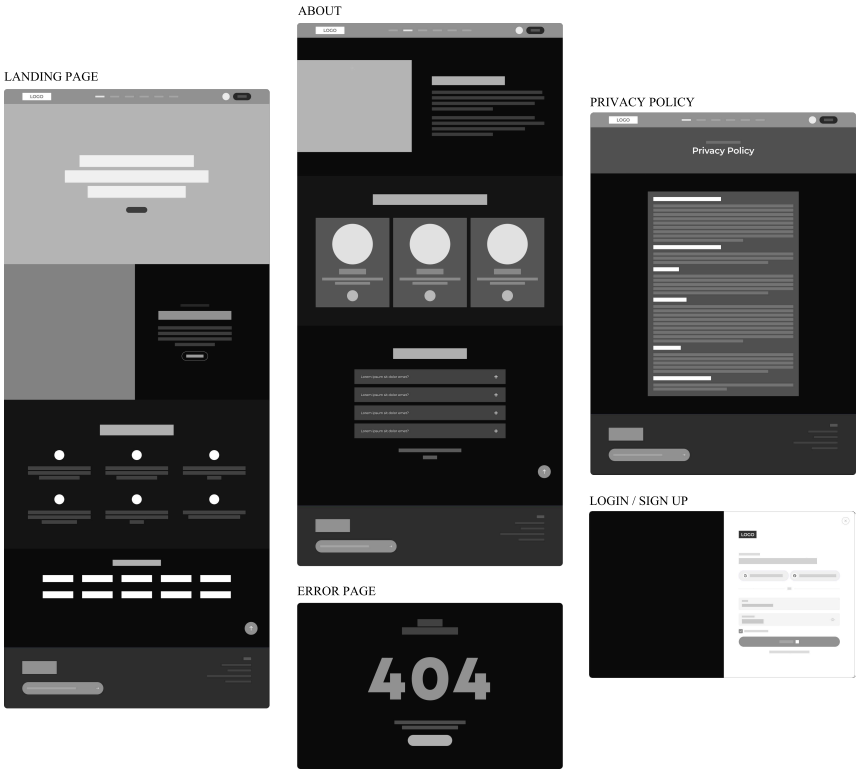

Fig. 11. Landing page view.

Wireframes representing authenticated user interactions were categorized into four main functional areas, each presented individually below.

Regarding the library section, the prototypes were focused on creating an easily navigable interface that supports efficient content filtering and organized resources. The emphasis was placed on facilitating straightforward access to knowledge materials (Figure 12).

Fig. 12. Library pages view.

For the academy, the wireframes prioritized structured learning experiences, incorporating intuitive course navigation, clear lesson layouts, and quiz functionalities. The designs emphasized user interaction, feedback mechanisms, and progress tracking throughout courses (Figure 13).

Fig. 13. Academy pages view.

Wireframes for the forum distinctly outlined user interactions, content creation, and thread management features. Particular attention was devoted to enhancing social interactions, including comment replies, rating systems, and content categorization (Figure 14).

Fig. 14. Forum pages view.

Lastly, the profile section prototypes included comprehensive layouts for the user's personal dashboard, configuration settings, and interfaces for managing quizzes, courses, and performance tracking. Design efforts focused on presenting detailed activity summaries to enhance personalized user experiences (Figure 15).

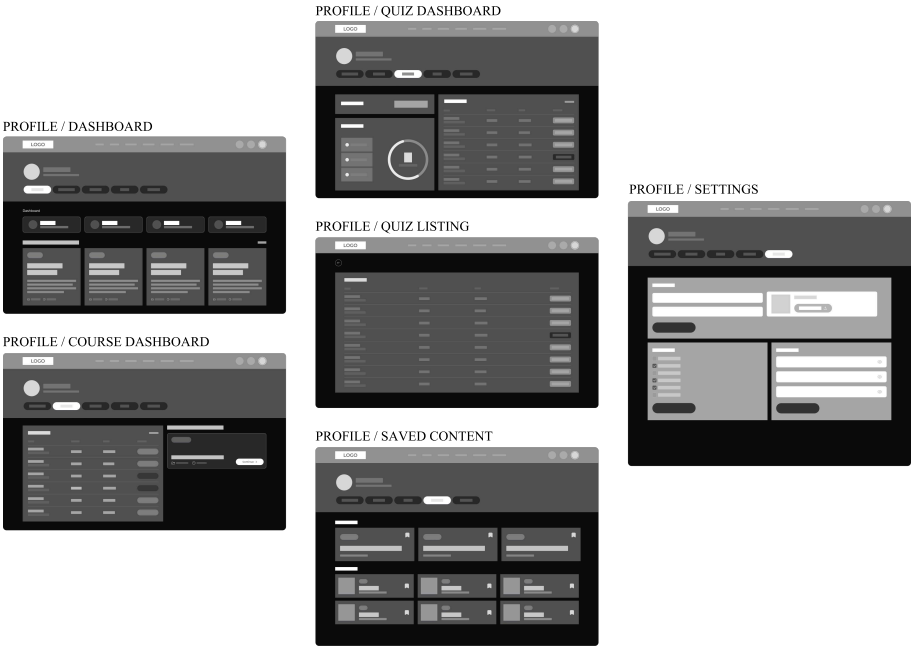

Fig. 15. Profile pages view.

These 'low-fidelity' prototypes proved to be indispensable for the subsequent rounds of high-fidelity prototyping to follow, as they helped maintain the design vision and overall user experience through the varying iterations of design.

4.3.3 High Fidelity Prototyping

Upon validation of the initial layout and user flow concepts from low-fidelity wireframes, the design flow moved onto high-fidelity prototyping. High-fidelity prototypes were also built in Figma, which included final design pieces: color schemes, typography, detailed component interactions, and responsiveness to different screen sizes. These proof-of-concept prototypes were very close to the final user experience, providing a clearer specification and visual concept to guide development.

As part of the high-fidelity prototyping phase, a uniform visual theme was established to improve user interaction and promote consistency across the platform. The visual identity was built around three key design components: color scheme, typography, and grid layouts.

A balanced and accessible color palette was selected, featuring a combination of primary, secondary, and neutral colors. Primary colors used include shades of blue, chosen for their calming and trustworthy qualities, great traits to be connected with a medical study environment. Accent colors, such as muted greens and teals, were used to emphasize interactive elements and guide user attention. Neutral shades (white, gray, and soft pastels) were used extensively for backgrounds, facilitating readability and visual order. The selected palette complies with the accessibility guidelines for contrast ratio for users with different visual abilities.

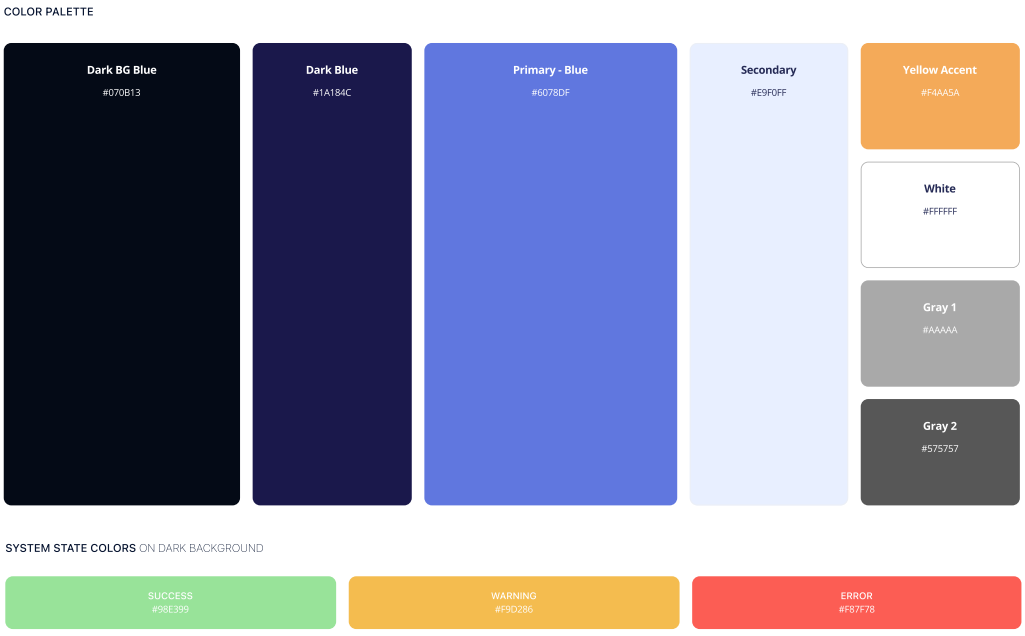

Fig. 16. color palette theme.

Typography decisions emphasized clarity, readability, and visual hierarchy. The primary font chosen for the platform was `Montserrat`, known for its clean, modern aesthetic and excellent readability across diverse screen resolutions. Specific font weights and sizes were defined to differentiate textual elements clearly:

- **Headings (H1, H2, H3, H4):** Set in `Montserrat Bold`, sizes ranged from 32pt to 80pt on desktop screens, and scaled responsively between 24pt and 72pt on mobile devices, providing prominent and clear visual hierarchy.
- **Subheadings, Captions, and Footnotes:** Rendered in `Montserrat Medium`, font sizes varied from 16pt to 22pt on desktop displays and were slightly reduced to 14pt to 20pt on mobile devices, ensuring consistent readability and clear structural differentiation.
- **Body Text:** Utilizing `Montserrat Regular`, the body text was typically set between 16pt and 20pt on desktop devices and adjusted to 14pt to 18pt for mobile readability. This provided optimal text legibility across varied screen dimensions and user interactions.
- **Interactive Elements (Buttons, Navigation):** Designed using `Montserrat Medium`, typically at 16pt for desktop interfaces and scaled to 14pt on mobile screens, clearly distinguishing interactive components from static textual elements to enhance usability and user interaction.

Furthermore, the serif font `Cormorant` was strategically incorporated into selected headings and prominent textual elements to visually highlight specific keywords and emphasize key concepts. Its elegant and refined appearance provided visual contrast to the sans-serif `Montserrat`, effectively drawing user attention to important information and enhancing overall readability and aesthetic appeal.

A consistent and responsive grid system was set up as well, concerning standard UX/UI practices. The layout structure used for the design is based on a 12-column grid system for desktops or laptops, to make the content alignment predictable and flexible. For the mobile phones, the grid system was based on a 4-column model. This led to a responsive design, where content is displayed altered depending on the user's device, whether it be a desktop screen or a phone screen. Consistent spacing, margins, and gutters (set by grid guidelines) ensured visual harmony between the pages and structural cohesion throughout the interfaces of the platform.

Table 1. Grid layout values definition by screens.

	Mobile (SM)	**Desktops (LG)**
Number of Columns	4	12
Margin	25px	160px
Padding	25px	20px
Gutter	15px	20px

The unauthenticated pages, including the landing page, about page, contact page, error screen, and login/registration forms, were designed to convey a strong visual identity and ensure a smooth experience. These screens incorporated finalized color schemes, typography, and interactive behaviors. Each page integrates clear, consistent typography and interactive elements such as expandable

FAQ sections, intuitive forms, structured content layouts, and engaging visual feedback to ensure seamless navigation and enhanced user interaction, effectively communicating core platform information and facilitating user onboarding. The final prototype outcome is represented in the high-fidelity prototype shown in Figure 17.

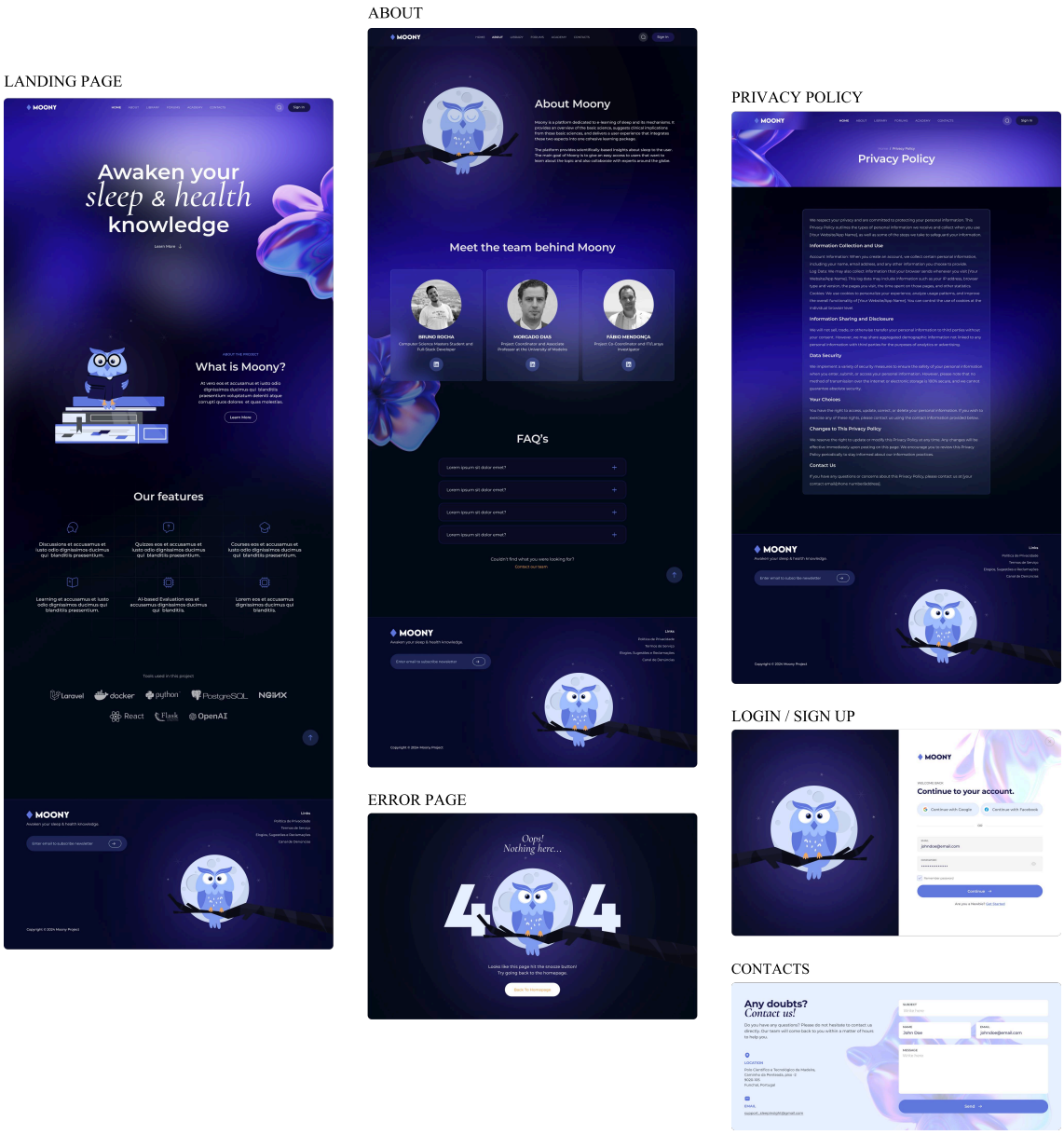

Fig. 17. High fidelity landing page prototype.

The following elements detail the high-fidelity screens developed for each main functional area. The Library high-fidelity prototypes refined the content presentation, visual hierarchy, and imple-

mented finalized visual styles including color schemes, font choices, and responsive layouts (Figure 18).

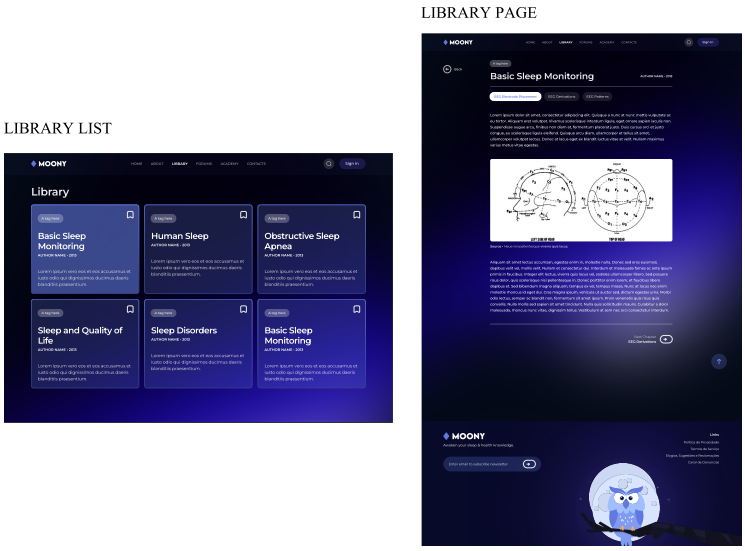

Fig. 18. High fidelity library pages prototype.

Then, the academy prototype focused on improving the course and quiz layouts, including the quiz interactions (back, forward, cancel, and finish) and user progress indicators, having interactive feedback throughout the process (Figure 19).

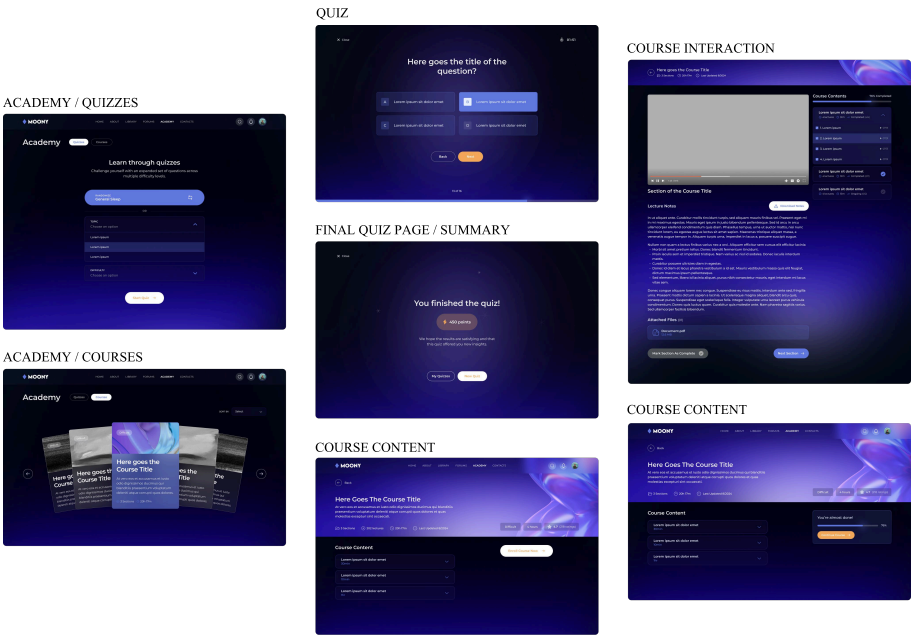

Fig. 19. High fidelity academy pages prototype.

The Forum prototype included interactive elements such as post creations, real-time feedback on posts and comments, and user interactions, including a thematic live chat (Figure 20).

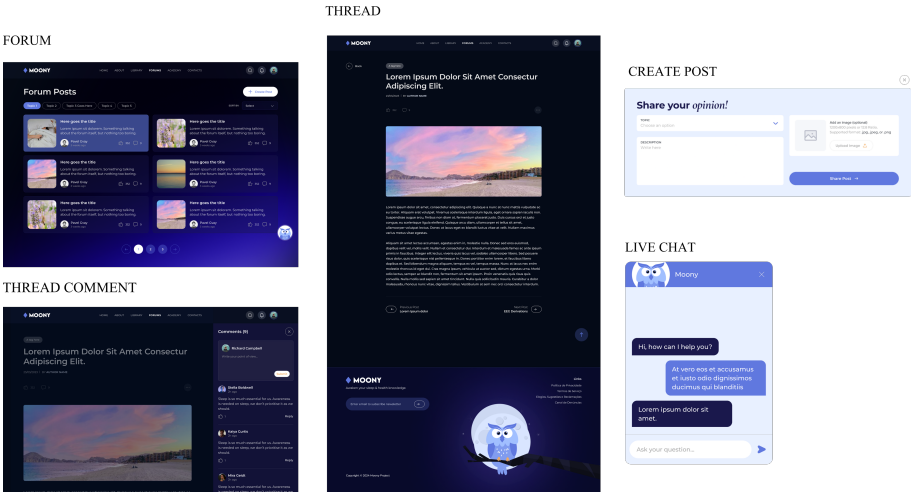

Fig. 20. High fidelity forum pages prototype.

The Profile prototype introduced an enhanced dashboard experience for a compact data visualization for quizzes, courses, user progress metrics, and settings configuration (Figure 21).

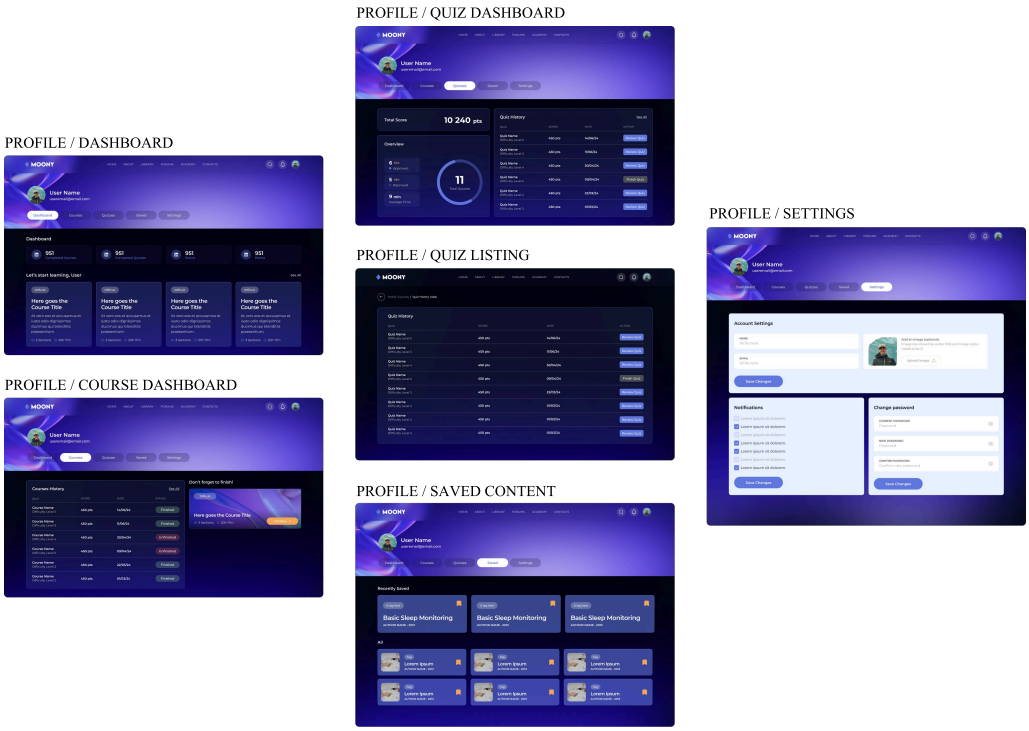

Fig. 21. High fidelity profile pages prototype.

Profile / Dashboard

This is the default landing page upon login. It shows a greeting, the user's avatar, and tabs to access different profile areas. Below the navigation bar is the main dashboard, which displays recommended courses as interactive cards. Each card is clickable and redirects the user to a course detail or content module. Hovering over a card reveals progress or resume actions.

Profile / Course dashboard

This page provides a summary of all courses the user has enrolled in. A table on the left lists course names, progress indicators, and last accessed timestamps. Each course row is clickable and takes the user to the last viewed module or quiz. On the right is a callout reminder to resume an in-progress course, encouraging continuity.

Profile / Quiz dashboard

Displays aggregate performance metrics including total score, quiz completion count, and a pie chart showing correct vs incorrect answer ratios. The user can navigate to individual quiz summaries via the right-hand table, which lists each quiz attempt with timestamps, scores, and a "Review" button to re-analyze performance.

Profile / Quiz listing

Provides a historical list of all completed quizzes. This is a more detailed and scrollable version than the dashboard. Each row includes a quiz name, score, date, and a direct link to a detailed review page. This allows users to track longitudinal performance and revisit incorrect answers.

Profile / Saved Content

A personal content archive where users can bookmark or save articles, modules, or quizzes. The items appear as tiles with preview titles and icons. Each tile links directly to the corresponding content. Users can manage saved items, delete entries, or filter by type using the toolbar above.

Profile / Settings

This configuration page allows users to update profile information (name, image), manage notification preferences, and change their password. All fields are editable in-place. The system gives instant feedback on saved changes, and validation messages appear when fields are incomplete or incorrect.

Interactive Navigation and Application Management

Across all pages, a consistent top navigation bar enables quick switching between sections: Dashboard, Courses, Quizzes, Saved, and Settings. Clicking any of these tabs reloads the corresponding page without requiring a full page refresh (SPA behavior). Each page component supports user interaction (click, hover, save, edit), and changes are persisted through backend API calls.

4.4 Learning Evaluation

4.4.1 Time Definition

To associate an estimated time for answering a question based on its difficulty, a static method was implemented to be publicly available for general use. This function maps the numeric difficulty value assigned to each question to a predefined difficulty level category, and accordingly to an estimated response time measured in seconds.

Table 2 displays the recommended time in seconds for the general questions based on the difficulty category.

Table 2. Estimated Time per Difficulty Level.

Difficulty Level	Description	Typical Expectation Time (seconds)
Very Easy	Very simple knowledge recall	15 - 30
Easy	Basic understanding	30 - 45
Normal	Moderate reasoning and thinking	45 - 60
Hard	Advanced problem solving	75 - 90
Very Hard	Complex reasoning and synthesis	90 - 120

The internal implementation of the function for time distribution is described in five levels, each fitted in the time expectation range in the Table 2:

- Very Easy: 20 seconds
- Easy: 40 seconds
- Normal: 60 seconds
- Hard: 90 seconds
- Very Hard: 120 seconds

If the difficulty value cannot be mapped directly, no value is assigned at the creation or removed (invalid or missing), the function defaults to the *Normal* level time (60 seconds).

4.4.2 Questions Assessment

The question assessment mechanism systematically evaluates user responses during quiz interactions, analyzing correctness, quality, and providing comprehensive feedback where applicable. To effectively accommodate different question formats, two primary evaluation methods were implemented. Firstly, **Predefined Answer Evaluation** is applied to structured questions such as multiple choice and yes/no types, wherein user responses are compared directly against a predefined set of correct answers. Secondly, **Dynamic Answer Evaluation**, which uses a Langchain-powered LLM as detailed in section 4.2.6, dynamically assesses free-text responses. This advanced evaluation provides nuanced evaluation and content-sensitive feedback, significantly enhancing the depth and personalization of user feedback.

The evaluation workflow follows these general steps:

1. User answers are received and matched to their respective questions.
2. The question type is identified to determine the appropriate evaluation method.
3. Answers are assessed either through predefined answer comparison or via dynamic LLM evaluation.
4. Results (correctness, response quality scores, feedback) are recorded and returned for further analysis.

Predefined answer evaluation is suitable for structured question types (e.g., multiple-choice, yes/no). The correctness determination involves comparing user answers directly against correct options stored in the system, allowing a faster response time instead of sending all the information to the LLM to be analyzed and corrected.

For multiple-choice questions, correctness validation involves verifying exact matches between user-selected options and correct options from the database. Yes/no questions involve straightforward comparison after normalizing text.

Example: Given a multiple-choice question with correct options: `{"Option A", "Option B"}`, a user's answer selecting "Option A" and "Option B" would be evaluated as correct, whereas selecting only "Option A" or adding "Option C" would be incorrect.

Code implementation:

```
private function evaluatePredefinedAnswer(Question $question, $answer):
    bool
{
    $correctOptions = QuestionOption::where('question_id', $question->id)
        ->where('is_correct', true)
        ->pluck('option_text')
        ->toArray();

    $allOptions = QuestionOption::where('question_id', $question->id)
        ->pluck('option_text')
        ->toArray();

    if ($question->type->tag === 'multiple_choice') {
        if (is_array($answer)) {
            foreach ($answer as $option => $optionValue) {
                if ($option == 'option_text' && !in_array($optionValue,
                    $allOptions)) {
                    return false;
                }
            }

            return empty(array_diff($correctOptions, [$answer['option_text'
                ]]))
                    && empty(array_diff([$answer['option_text']],
                        $correctOptions));
```

```
        } else {
            return in_array($answer, $correctOptions);
        }
    } elseif ($question->type->tag === 'yes_no') {
        return in_array(strtolower($answer), array_map('strtolower',
            $correctOptions));
    }

    return false;
}
```

Listing 4. Predefined answer evaluation

For free-text or open-ended questions, dynamic evaluation uses a LangChain-based LLM. This sophisticated method considers semantic correctness and answer quality, providing nuanced correctness and detailed feedback.

The detailed LLM implementation is described separately (see section 4.2.6). Here, the system captures correctness, response quality score, and feedback from the LLM API response.

Code implementation (Integration):

```
private function evaluateWithLangChain(Question $question, $answer)
{
    $data = [
        'theme' => SysConfig::tag('theme')->first()->value,
        'question' => $question->title,
        'answer' => is_string($answer) ? $answer : strval($answer)
    ];

    $responseLangchain = $this->callLangChainAPI($data);
    $response = json_decode($responseLangchain['response'], true);

    if ($response) {
        return [
            'is_correct' => $response[0]['is_correct'],
            'response_quality_score' => $response[0]['
                response_quality_score'],
            'suggested_answer' => $response[0]['feedback'],
            'time_taken' => null,
        ];
    } else {
        return [
            'is_correct' => false,
            'response_quality_score' => 0,
            'time_taken' => null,
        ];
    }
```

```
}
```

Listing 5. LangChain-based evaluation integration

The evaluation workflow integrates both methods seamlessly, selecting the appropriate strategy based on the question type. Structured responses are validated internally, while unstructured answers invoke the LLM. A combined evaluation handler manages this logic and formats results for further analytics or user display.

```
private function evaluateAnswers($questions, $answers)
{
    $correctAnswers = 0;
    $questionResponses = [];

    foreach ($answers as $questionId => $answer) {
        $question = $questions->firstWhere('id', $questionId);

        if (!$question) continue;

        if ($question->type->tag === 'free_text') {
            $evaluationResult = $this->evaluateWithLangChain($question,
                $answer);
        } else {
            $isCorrect = $this->evaluatePredefinedAnswer($question, $answer
                );
            $evaluationResult = [
                'is_correct' => $isCorrect,
                'response_quality_score' => null,
                'time_taken' => null,
            ];
        }

        if ($evaluationResult['is_correct']) { $correctAnswers++; }

        $questionResponses[] = [
            'question_id' => $questionId,
            'question_options' => $question->options,
            'answer' => $answer,
            'is_correct' => $evaluationResult['is_correct'],
            'response_quality_score' => $evaluationResult['
                response_quality_score'],
            'time_taken' => $evaluationResult['time_taken'] ?? null,
            'suggested_answer' => $evaluationResult['suggested_answer'] ??
                null,
        ];
    }
```

```
    return [
        'correctAnswers' => $correctAnswers,
        'questionResponses' => $questionResponses
    ];
}
```

Listing 6. Complete answer evaluation workflow

The hybrid evaluation approach ensures flexibility, accuracy, and detailed feedback tailored to each question type, improving the overall assessment quality and facilitating effective user learning through nuanced feedback. Structured evaluations maintain simplicity and efficiency, while dynamic evaluations leverage advanced AI-driven analysis to provide deeper insights into user understanding and performance. This balance allows to have a faster response time, and any other method or question type that needs to be integrated can easily be put in the existing structure without affecting the existing assessment subsystems.

4.4.3 User Learning Metrics

User learning metrics are needed for assessing the learner progression, performance, and overall engagement within the educational platform. These metrics, systematically tracked and recorded through data history, enable comprehensive analyses that inform adaptive learning strategies. By evaluating these metrics, the platform can tailor personalized learning pathways and provide actionable feedback, thereby enhancing user interactions and facilitating continuous improvement over time. Each metric is carefully defined to clearly illustrate its role in assessing user performance and engagement, accompanied by explicit descriptions of calculation methodologies and practical implementation strategies.

Accuracy Rate

Definition: The accuracy rate measures the percentage of quiz questions that a user answers correctly.

Rationale: This metric directly indicates the correctness of user responses, providing a clear reflection of a user's knowledge accuracy and understanding of the content. A consistently high accuracy rate suggests effective learning and retention.

Calculation approach: The accuracy rate is calculated by dividing the total number of correctly answered questions by the total number of questions answered by the user, multiplied by 100 to express the result as a percentage.

$$\text{Accuracy Rate (\%)} = \frac{\text{Number of Correct Answers}}{\text{Total Questions Answered}} \times 100$$

Example calculation: If a user answered a total of 150 questions, and out of those 120 were answered correctly, the accuracy rate is calculated as follows:

$$\text{Accuracy Rate} = \frac{120}{150} \times 100 = 80\%$$

Code implementation: The accuracy rate is computed programmatically as:

```
$correctAnswers = QuestionResponse::whereHas('userQuiz', function ($query)
    use ($userId) {
    $query->where('user_id', $userId);
})->where('is_correct', true)->count();

$totalQuestions = QuestionResponse::whereHas('userQuiz', function ($query)
    use ($userId) {
    $query->where('user_id', $userId);
})->count();

$accuracyRate = ($totalQuestions > 0) ?
    round(($correctAnswers / $totalQuestions) * 100, 2) : 0;
```

Listing 7. Accuracy rate implementation

Interpretation of Results: An accuracy rate above 85% indicates strong mastery, while rates below 60% might signal the need for intervention or additional learning resources.

Time Efficiency

Definition: Time efficiency is the measure of how effectively a user manages time during quiz completion, represented by the average duration a user spends per quiz.

Rationale: Evaluating time efficiency helps identify users who complete quizzes in an optimal time frame, potentially indicating strong content mastery or efficient decision-making skills. Conversely, prolonged durations might suggest areas needing reinforcement.

Calculation approach: Time efficiency is calculated by averaging the total completion times recorded across all quizzes a user has completed.

$$\text{Time Efficiency (seconds)} = \frac{\text{Total Time Taken for all quizzes}}{\text{Number of quizzes completed}}$$

Example calculation: If a user completed 3 quizzes, spending 120, 90, and 150 seconds respectively, the time efficiency calculation would be:

$$\text{Time Efficiency} = \frac{120 + 90 + 150}{3} = 120 \text{ seconds}$$

Code implementation: The backend calculation in PHP/Laravel:

```
$averageTimePerQuiz = UserQuiz::where('user_id', $userId)->avg('time_taken'
    );
$timeEfficiency = round($averageTimePerQuiz, 2);
```

Listing 8. Time efficiency implementation

Interpretation of Results: Shorter average times (without compromising accuracy) generally indicate higher efficiency. However, extremely short times may reflect rushed or guess-based responses, warranting further investigation.

Score Standard Deviation

Definition: Score standard deviation represents the variability or consistency of a user's quiz scores around the mean.

Rationale: Understanding variability in user performance can reveal inconsistencies in knowledge or skill areas. Lower standard deviation signifies consistent performance, while a higher deviation indicates fluctuating proficiency.

Calculation approach: The standard deviation (σ) is computed using the following statistical formula:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n}}$$

where x_i represents individual quiz scores, n the total number of quizzes, and μ the mean score.

Example calculation: For quiz scores: 80, 70, 90, and 60, the mean (μ) is 75.

$$\sigma = \sqrt{\frac{(80-75)^2 + (70-75)^2 + (90-75)^2 + (60-75)^2}{4}} = \sqrt{\frac{350}{4}} \approx 9.35$$

Code implementation:

```
protected function calculateStandardDeviation($scores)
{
    $n = count($scores);
    if ($n === 0) {
        return 0;
    }
    $mean = array_sum($scores) / $n;
    $sumSquaredDifferences = array_reduce($scores, function ($carry, $item)
        use ($mean) {
        return $carry + pow($item - $mean, 2);
    }, 0);
    return round(sqrt($sumSquaredDifferences / $n), 2);
}
```

Listing 9. Standard deviation calculation

Interpretation of Results: A smaller standard deviation (close to 0) indicates reliable and stable performance, whereas a larger value may suggest inconsistency or knowledge gaps.

Completion Rate

Definition: Completion rate is the percentage of quizzes a user fully completes compared to the total number of quizzes they have started.

Rationale: A high completion rate demonstrates strong engagement, perseverance, and motivation. Conversely, lower completion rates might highlight difficulties, frustration, or disengagement.

Calculation approach: The completion rate is calculated as the ratio of completed quizzes to quizzes started, expressed as a percentage:

$$\text{Completion Rate (\%)} = \frac{\text{Quizzes completed}}{\text{Quizzes started}} \times 100$$

Example calculation: If a user started 10 quizzes and completed 7:

$$\text{Completion Rate} = \frac{7}{10} \times 100 = 70\%$$

Code implementation:

```
$quizzesStarted = UserQuiz::where('user_id', $userId)->count();
$quizzesCompleted = UserQuiz::where('user_id', $userId)->where('
    is_completed', true)->count();
$completionRate = ($quizzesStarted > 0) ? round(($quizzesCompleted /
    $quizzesStarted) * 100, 2) : 0;
```

Listing 10. Completion rate implementation

Interpretation of Results: Completion rates above 85% typically signify strong user engagement. Rates below 60% may indicate motivational or usability challenges requiring intervention.

Improvement Rate

Definition: Improvement rate measures the percentage change between a user's initial quiz score and their most recent quiz score.

Rationale: Tracking improvement provides clear evidence of learning progression, effectiveness of the content provided, and the user's ability to retain or enhance their knowledge over time.

Calculation approach: The improvement rate is calculated using the following formula:

$$\text{Improvement Rate (\%)} = \frac{\text{Latest Score} - \text{Initial Score}}{\text{Initial Score}} \times 100$$

Example calculation: If the initial quiz score was 50 and the latest quiz score was 75:

$$\text{Improvement Rate} = \frac{75 - 50}{50} \times 100 = 50\%$$

Code implementation:

```
protected function calculateImprovementRate($scores)
```

```
{
    $numScores = count($scores);
    if ($numScores < 2) {
        return 0;
    }
    $initialScore = $scores[0];
    $latestScore = $scores[$numScores - 1];

    $improvementRate = ($initialScore != 0) ? round((($latestScore -
        $initialScore) / $initialScore) * 100, 2) : 0;

    return $improvementRate;
}
```

Listing 11. Improvement rate calculation

Interpretation of Results: Positive values indicate improvement, negative values suggest decline, and near-zero values imply stable performance without significant change.

4.4.4 User Progression

The user progression subsystem tracks all the performance metrics segmented by quiz difficulty levels. This enables detailed monitoring of a user's progress, facilitating adaptive difficulty adjustments and personalized learning paths.

The metrics tracked at each difficulty level include:

- **Average Score**: The mean quiz score achieved by the user for each specific difficulty level.
- **Accuracy Rate**: The percentage of correctly answered questions across quizzes within each difficulty level.

The implementation process involves aggregating quiz data based on the predefined difficulty categories described previously and calculating key performance indicators.

Code Implementation

The following Laravel backend function, written in PHP, illustrates the described logic:

```
protected function updateUserDifficultyProgression($userId)
{
    $userQuizzes = UserQuiz::where('user_id', $userId)
                           ->with('quiz')
                           ->get();

    $quizzesByDifficulty = $userQuizzes->groupBy('quiz.difficulty');

    foreach ($quizzesByDifficulty as $difficultyLevel => $userQuizzesGroup)
        {
        $averageScore = $userQuizzesGroup->avg('score');
```

```
        $quizIds = $userQuizzesGroup->pluck('id');

        $correctAnswers = QuestionResponse::whereIn('user_quiz_id',
            $quizIds)
            ->where('is_correct', true)
            ->count();

        $totalResponses = QuestionResponse::whereIn('user_quiz_id',
            $quizIds)
            ->count();

        $accuracyRate = ($totalResponses > 0)
            ? round(($correctAnswers / $totalResponses) * 100, 2)
            : 0;

        UserDifficultyProgression::updateOrCreate(
            ['user_id' => $userId, 'difficulty_level' => $difficultyLevel],
            [
                'average_score' => round($averageScore, 2),
                'accuracy_rate' => $accuracyRate,
            ]
        );
    }
}
```

Listing 12. User progression update implementation

Interpretation and Usage

These metrics, segmented by difficulty, provide a different insight into user performance trends, allowing the system to dynamically change the presented content difficulty, ensuring continuous learning effectiveness and motivation. The users which demonstrates consistently high performance can receive progressively harder questions, as the inverse can occur as well, having users at high levels have content recommendations at lower difficulty settings.

4.4.5 Security Measures

The platform has multiple layers of security controls implemented, both inherent to the Laravel framework and introduced through custom configurations and deployment restrictions. These measures are meant to protect the user's sensitive data, securely authenticate and authorize, and limit the external exposure.

Framework-level Protections

Laravel framework provides a robust security foundation through built-in features that were adopted throughout the backend development:

- **CSRF Protection:** Laravel's automatic Cross-Site Request Forgery (CSRF) protection was enabled for all forms and API routes using CSRF tokens.
- **SQL Injection Prevention:** All database queries were executed using Laravel's Eloquent ORM or Query Builder, which use parameter binding to prevent SQL injection attacks.
- **Password Hashing:** User passwords are securely hashed using the bcrypt algorithm via Laravel's 'Hash' facade.
- **Rate Limiting:** Laravel's built-in rate limiting middleware ('ThrottleRequests') was applied to sensitive endpoints to prevent brute-force attacks and abuse.
- **Route Middleware Enforcement:** Laravel's route middleware system ensures that API requests pass through layered checks (authentication, authorization, throttling, etc.) before reaching controllers.
- **Sanitized API Responses:** All outbound responses are manually constructed as JSON, ensuring full control over the output and eliminating accidental data leakage or model exposure.

Authentication and Authorization

The user's access is controlled through Laravel's authentication guards and policy-based authorization system:

- **Guarded Routes:** API routes are protected using middleware ('auth:sanctum') to enforce authenticated access.
- **Role-Based Access Control (RBAC):** User roles were implemented and tied to authorization policies to restrict actions based on assigned permissions.
- **Custom Policies:** Fine-grained control over resource access was enforced using Laravel policies for models such as quizzes, user metrics, and administrative content.

Input Validation and Sanitation

All user-submitted data is validated using Laravel's request validation mechanism ('FormRequest' classes), ensuring both structural integrity and type safety. For example:

```
$request->validate([
    'email' => 'required|email',
    'password' => 'required|min:8',
]);
```

This prevents malformed inputs from being processed or persisted to the database.

4.5 Deployment

This chapter describes the approach, implementation, and considerations involved when deploying the platform on the server to open access. The deployment was performed on a private server, secured through restricted VPN access from the University of Madeira, and orchestrated via Docker compose to ensure consistent, reproducible, and secure deployment.

4.5.1 Environment

The platform was deployed on a private server configured to meet security standards, performance, and scalability requirements. The server runs with the following configuration:

- **Operating System:** Ubuntu 20.04.2 LTS (GNU/Linux Kernel 5.4.0-208-generic x86_64)
- **CPU:** Intel(R) Core(TM) i7-9700F CPU @ 3.00 GHz, 8-core, single-thread.
- **RAM:** 32 GB
- **Storage:** Hybrid storage configuration with a total of 29 TB.

This configuration provided ample processing capacity and storage for containerized deployment and data management.

The deployment uses Docker and Docker Compose for container orchestration, which was implemented using Docker (v20.10.21) and Docker Compose (v2.32.4) to manage isolated application services efficiently. This approach enabled consistent deployments, simplified scaling, and enhanced portability across environments.

The internal network configuration was designed to ensure controlled service exposure. A dedicated subdomain was mapped to the deployment server, and DNS routing was configured to direct all incoming traffic through an internal NGINX reverse proxy container. This setup enhanced security by minimizing direct access to internal services and limiting the attack surface.

Access to the server is made by SSH, and external access to services is disabled by default to enforce an isolated and secure deployment environment. Regular monitoring of server performance metrics such as CPU, memory, disk space, temperature, and processes helps to ensure reliability and prompt response to potential issues.

Occasionally, the accumulation of unused containers and images during extended periods of inactivity led to memory bottlenecks. To resolve this, the following command was used periodically to clean up the system:

```
docker system prune -a
```

This command safely removes unused containers, images, and volumes, helping to reclaim disk space and improve system responsiveness.

4.5.2 Deployment Process

The initial deployment process was executed by the following steps:

1. **VPN Connection and SSH Access to Server:** Before any deployment operation, a secure VPN connection must be established. This ensures that all remote management activities, including SSH access to the server, are isolated from public exposure and comply with institutional security policies.
2. **Repository Cloning and Configuration:** Once connected to the server, the source code and Docker configuration files are pulled from the private Git repository:

```
git clone git@<repository-address>:project-name.git
cd project-name
```

The remaining configuration files (like environment files and server-specific settings) were configured in each container folder via a '.env' file, which includes credentials, API keys, and other container-specific settings.

3. **Docker Compose Setup:** Based on the file used in development, a new file for Docker compose configuration was created: `docker-compose.prod.yml`. This file defines all system services, including:
   - **frontend:** React-based user interface for learners.
   - **frontend-admin:** Administrative interface built on React Admin.
   - **backend:** Laravel-based API service.
   - **nginx:** Acts as a reverse proxy and routing gateway.
   - **postgres:** PostgreSQL database for persistent data storage.
   - **langchain:** Handles LLM-powered quiz generation and assessment.
4. **Building and Starting Containers:** Once the initial configuration was complete, the services were built and started using the command:

```
docker-compose -f docker-compose.yml -f docker-compose.prod.yml up -
    d
```

5. **Health Checks and Verification:** After deployment, each service is individually verified to ensure proper functionality. General system logs are inspected using:

```
docker compose logs -f
```

But, generally, when an error occurs inside a container, sometimes there's a need to inspect the logs directly inside the container:

```
docker compose logs <container-name>
```

Critical endpoints, such as API routes, frontend rendering, and LLM responses, are tested to confirm that services are running correctly and are reachable within the container network.

4.5.3 Maintenance and Updates

To apply updates or make configuration changes, containers can be stopped, rebuilt, and restarted with minimal downtime. Redeployment typically follows the pattern:

```
git pull
docker compose build
docker compose up -d
```

This approach ensures that only modified containers are recreated, allowing the rest of the services to remain continuously available

4.6 Challenges Encountered

The development of user interfaces and components within a design software has some practical challenges. The general software complexity, design concepts, and internal tools at the start were the main challenges in the development of the platform.

Furthermore, trying to keep the balance between the visual appeal and usability was essential because, as a developer tries to make a component usable and functional, at least to the primary testing stage, a designer tries to emphasize aesthetic considerations. Addressing this issue required interactive refinement between the functionality being developed and the aesthetic, focusing heavily on achieving clarity and simplicity through the user flow process.

In the initial approach to the prototype implementation, only the desktop resolution was taken into consideration when developing the main functionalities for the platform. Only when those were completed and working, was there an adaptation in the design to the mobile screen. The main challenge in this adaptation was the hierarchy of information, for example, if an image and text are presented side by side on the desktop screen, on the mobile screen should be presented the image at first and then the text. This adaptation was easily implemented using Tailwind CSS in the Frontend application.

Another challenge that occurred during the development phase involved engineering effective prompts for the LLM, which were responsible for the dynamic generation of content such as quiz questions, answers, assessments, and live chat responses.

The most vulnerable prompts to have end-user influence were the quiz assessments and live chat responses. These prompts needed to be re-engineered to enforce structure, define types, and set expectations. For example:

```
f"You are an expert on {theme}. Answer the following based on the ongoing
    conversation.\n\n{conversation_prompt}\n\nUser: {message}\n\nSystem:"
```

This prompt was vulnerable to user interactions trying to change the topic or perform any other action. Changing the prompt to:

```
full_prompt = f"""You are an expert on {theme}. Only answer questions that
    are directly related to {theme}.
    If the user's message is irrelevant or attempts to divert from the
        theme, politely remind them to stay on topic in their language.

    Ongoing conversation:
    {conversation_prompt}

    User: {message}

    System:"""
```

This re-engineered structure significantly improved output consistency and reduced topic deviation. By clearly stating role constraints and explicitly handling off-topic input, the LLM generated more accurate, focused, and context-aware responses. It also enhanced trustworthiness by reducing the system's susceptibility to manipulation, ultimately resulting in a more robust user experience. Despite these improvements, prompt engineering remains an iterative and nuanced task.

During the deployment phase, several challenges occurred regarding the routing inside the server. As the server is hosted through a VPN, its access can be very limited, due to safety reasons. However, the target was to set the main traffic port through port 5004. This part was easily solved by changing the configuration of the Docker-compose and the services Dockerfile, but the server itself had the routing made with Apache. This resulted in a conflict with SSL communication.

After deploying all the containers and having each running separately, there was an error that occurred when trying to access any route for the backend by the frontend containers. The error was the following: `No route found to host`. These issues normally occur when there are misconfigurations in the routing parameters in the web server. After trial and error, and checking that all the containers could have outbound communication, the error was an internal Docker configuration from the server, blocking the inbound communication, since the Docker user didn't have permission to access the containers through the new network. This problem had to be solved inside the server by executing the following set of commands:

```
sudo iptables -t nat -A POSTROUTING -s 172.20.0.0/16 ! -o br-<bridge_number
    > -j MASQUERADE
sudo iptables -I FORWARD 1 -s 172.20.0.0/16 -d 172.20.0.0/16 -j ACCEPT
sudo iptables -I DOCKER-USER 1 -s 172.20.0.0/16 -j ACCEPT
```

4.7 Summary

In this chapter, it was described how the educational platform was developed end-to-end, from system architecture, integration of backend and frontend, database, UI/UX prototyping, to functionality based on LMM. The implementation followed the previously described methodological framework and focused on performance, usability, and maintainability. Importantly, the implementation also featured elements of interactive learning and 'smart' automation via LLMs to provide a personalized and contextual experience. Having achieved a study design, the next chapter assesses the usability of the achieved platform by testing its effectiveness, system metrics, as well as user feedback, and the evaluation process.

5 Results

This section presents the results obtained from the usability surveys applied before and after using the platform in an alpha stage, following the initial deployment process. The primary objective of this evaluation was to assess how users perceived the platform's usability, navigational flow, visual design, responsiveness, and educational effectiveness.

The evaluation was guided by the same methodology described in the section 3.6, and was structured using two custom questionnaires: a pre-test to assess baseline digital familiarity and interest in the subject, and a post-test to gather user feedback on usability and learning experience. These documents are included in the appendix B (Pre-test) and C (Post-test). Additionally, this section includes system performance metrics collected from the production server hosting the platform.

5.1 General Characteristics of the Evaluation

In this study, 24 participants were invited to participate, of whom eight identified as female and sixteen as male. All participants were students from general academic fields.

Before participating, each was provided with information on the aims and methods of the study, as well as data protection guidelines. Each participant signed an informed consent document following the guidelines of the Ethics Committee. A copy of the form is included in Appendix A.

5.1.1 Testing Protocol

The evaluation followed this structured sequence:

1. The session began with a verbal explanation and signing of the informed consent form.
2. Participants completed a short pre-test questionnaire (Appendix B).
3. Participants were then asked to explore the platform freely for 10–15 minutes. No specific tasks were imposed to allow spontaneous interaction and natural usability discovery.
4. Observers monitored interactions, noting any usability difficulties or feedback.
5. Participants then completed the post-test questionnaire (Appendix C).

5.1.2 Questionnaire Design

The pre-test contained five statements assessing digital literacy, prior experience with online learning platforms, and interest in the topic. The post-test consisted of 11 statements focused on usability, structure, visual design, responsiveness, and perceived learning effectiveness. Both used a **5-point Likert scale**, where:

1 = Strongly Disagree 2 = Disagree 3 = Neutral 4 = Agree 5 = Strongly Agree

The full text of both questionnaires is presented in Appendices B and C. Detailed response distributions for each participant are presented in Appendices D and E.

5.2 Calculation of the Global Perception Index

For each participant, the average of the answers to the pre-test and post-test perception questionnaire responses was calculated to compare changes in perceived usability and satisfaction. Table 3 shows the overall means and standard deviations obtained from all participants (n = 24).

Table 3. Means and standard deviations of pre-test and post-test scores (n=24).

	Pre-test	**Post-test**	**Difference (Post - Pre)**
Mean	4.150	4.375	0.225
Standard Deviation	0.480	0.487	-

The slight increase in the mean score suggests a general improvement in the user perceptions following the platform interactions.

5.3 Statistical Analysis of the Difference

To assess whether user perceptions of the platform were improved after interaction, a paired samples t-test on the scores before and after the experiment was performed to test whether there is a difference between the pre- and post-test average scores of the participants and whether the mean of the differences is significantly different from zero. The goal of the analysis was to determine the change in perceived usability and educational value before and after hands-on experience.

The paired samples t-test is appropriate in this context as it assesses whether the mean difference between two related measurements (pre-test and post-test scores for the same users) is statistically significant. The test statistic is computed using the formula:

$$t = \frac{\bar{d}}{s_d/\sqrt{n}} \tag{1}$$

where:

- $\bar{d}$ is the mean of the differences between paired observations
- s_d is the standard deviation of those differences
- n is the number of paired observations (in this case, $n = 24$)

The test revealed a statistically significant improvement in user perception after using the platform:

$$t(23) = 2.18,\ p = 0.039$$

This means that collectively, participants thought of the platform as more usable and educationally impactful after exposure to it. The effect size is moderate, but the significance of the estimates indicates an average favorable impact on users.

5.4 Correlation Analysis of Pre-Test Factors

To better understand if any factors influenced the participant's final evaluation or learning gains, Pearson correlation coefficients were calculated between each pre-test question and the average score of post-test questions and the difference.

The Pearson correlation coefficient, denoted as r, quantifies the linear relationship between two variables and is calculated using the following formula:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \tag{2}$$

where:

- x_i and y_i are individual values of the two variables being compared
- $\bar{x}$ and $\bar{y}$ are the means of those variables

For this analysis:

- x_i corresponds to the score of a specific pre-test question for participant i
- y_i corresponds to either the participant's average post-test score or their difference score (post-test mean minus pre-test mean)

The strongest positive correlation with the post-test average score was found in:

- **P5 - Interest in learning about sleep**: $r = 0.681$
- **P2 - Comfort with digital devices**: $r = 0.637$

This correlation suggests that participants who initially indicated more interest in the topic and greater comfort with technology were more likely to rate the platform positively.

On the other hand, the strongest negative correlation with the score improvement (difference) was in:

- **P3 - Comfort with usability tests**: $r = -0.666$
- **P1 - Familiarity with e-learning platforms**: $r = -0.497$

This indicates that less experienced usability testers, as well as novice e-learning participants, presented a greater increase in their perception of the platform. This suggests the platform can be especially successful in transforming the experience of inexperienced or initially reluctant users.

5.5 Graphical Representation of Results

This bar chart displays the average score for each post-test question (PO1 to PO11), highlighting user perception across different usability and content-related dimensions.

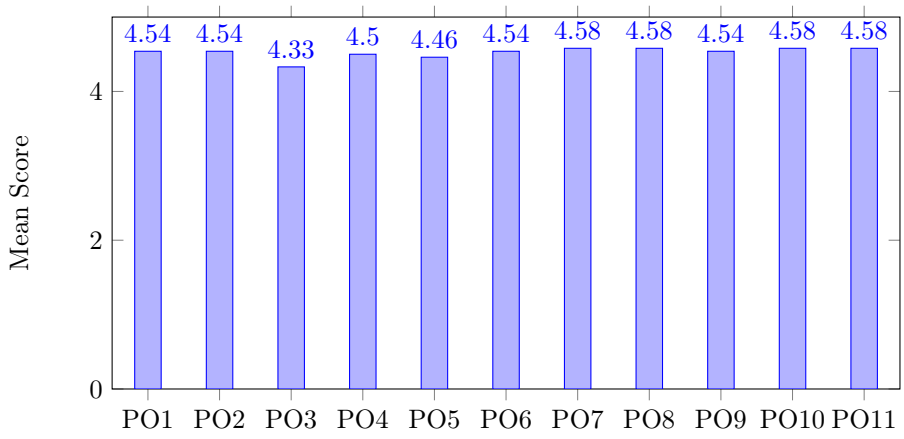

Fig. 22. Average post-test scores for each question (PO1–PO11), reflecting user perception across usability and content-related dimensions.

The average post-test scores for all perception items (PO1-PO11) are consistently high, ranging between 4.33 and 4.58 on a five-point scale. This narrow range of indicates minimal variability and strong agreement among participants regarding the usability and content of the platform.

This grouped bar chart compares the overall average scores before and after the platform usage, reflecting a general improvement in perception.

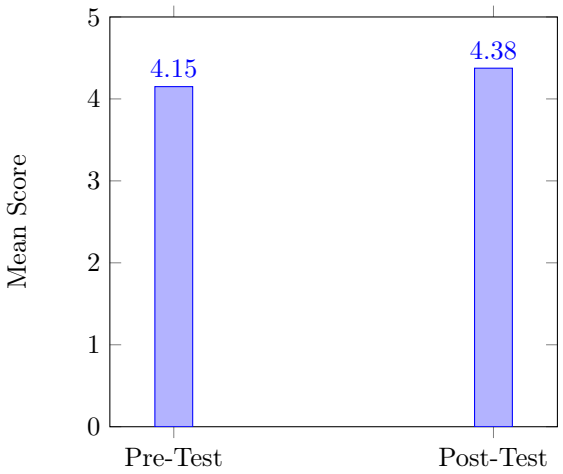

Fig. 23. Comparison of overall pre-test and post-test average scores, indicating improvement in user perception after using the platform.

5.6 System Metrics

Effective evaluation of system metrics is essential for ensuring optimal platform performance, stability, and scalability. To comprehensively assess the e-learning platform, this chapter will discuss essential metrics, their significance, methods for collection, and illustrative examples. The measurements were obtained directly from the deployed solution running in its operational environment described in Chapter 4.5, using the same configuration and infrastructure intended for real use. While the evaluation did not include large-scale or stress-testing scenarios, the results represent genuine operational performance under normal usage conditions and therefore provide a realistic view of the system's efficiency.

5.6.1 API Response Time

API response time measures the elapsed time between when a client issues a request to the back-end and when the full response is received. It is critical for maintaining a fluid user experience, especially when retrieving quiz data or submitting answers. This metric was measured using Thunder Client (similar to Postman) against an open route '/front/config/params' endpoint under representative load. An average response time below 200 ms is considered acceptable, and the platform achieved an average of **155 ms**.

5.6.2 Database Performance

Database performance is measured by query execution time, transaction throughput, and connection pool utilization. Slow queries can severely degrade the user experience and the platform's overall utility. Using PostgreSQL's 'EXPLAIN ANALYZE' and slow-query logs to profile typical read/write operations, the main common JOIN query ran in **0.219 ms** on average, showing the indexing capacity and database structure performance.

5.6.3 Web Performance

Web performance was assessed using **Google Lighthouse v10**, an open-source automated auditing tool developed by Google, which evaluates critical aspects of web user experience such as loading speed, interactivity, visual stability, and overall user experience quality. Lighthouse simulates realistic user scenarios, applying standardized network and hardware conditions to ensure consistent and objective measurements.

The performance metrics evaluated by Lighthouse include:

- First Contentful Paint (FCP): Measures the time elapsed from the initiation of page load until the first visual element appears on the screen, representing perceived loading speed.
- Largest Contentful Paint (LCP): Calculates the time required to load the largest visible content element (typically an image or text block), directly influencing user perception of page loading performance.
- Total Blocking Time (TBT): Captures the cumulative duration during which the page is unresponsive to user interactions (clicks, taps, keyboard input) due to task processing, reflecting interactivity responsiveness.

– Speed Index: Assesses the rate at which visual content is displayed during the loading process, indicating the perceived loading speed by users.

– Cumulative Layout Shift (CLS): Measures the visual stability by quantifying unexpected shifts of page elements during the loading phase, crucial for ensuring a consistent and user-friendly interaction.

Performance thresholds established by Lighthouse, and used to interpret metric scores, are detailed in Table 4.

Table 4. Lighthouse Performance Metrics - Recommended Thresholds (v10).

Metric	**Good Performance**	**Needs Improvement**
First Contentful Paint (FCP)	<1.8 s	1.8-3.0 s
Largest Contentful Paint (LCP)	<2.5 s	2.5-4.0 s
Total Blocking Time (TBT)	<200 ms	200-600 ms
Speed Index	<3.4 s	3.4-5.8 s
Cumulative Layout Shift (CLS)	<0.10	0.10-0.25

First Contentful Paint (FCP)

FCP measures the time between the moment the page begins loading to when the first content piece becomes visible to the user. A low FCP ensures that the users perceive the webpage as loading promptly, improving the initial engagement and user satisfaction. During controlled testing in a standardized development environment, the platform recorded an average FCP of approximately 0.5 seconds. However, actual performance may vary depending on network conditions, device capabilities, and content complexity.

Largest Contentful Paint (LCP)

LCP evaluates the render time of the largest block of visible content (normally an image or text block). This metric directly impacts the user perception of the loading performance. Ideal LCP values are below 2.5 seconds, ensuring users quickly perceive content as fully loaded. According to our test environment, the platform had an average LCP of 0.98s, jaw-breaking. Despite that, some differences in production conditions may vary, e.g., image dimensions, rendering time on low-end devices, etc.

Total Blocking Time (TBT)

TBT is the total duration of time that a page is unable to respond to user input, such as mouse clicks, screen taps, or keyboard presses. Lighthouse has a target for responsiveness of 200 ms or less. Under testing conditions, the platform gave an average TBT of 110 milliseconds, reflecting satisfactory interactivity. This is for optimized environments; real-world TBT in resource-lean devices or with heavy concurrent load may be higher.

Speed Index

Speed Index measures how quickly content is visually displayed during page load. Values under 3.4 seconds are generally considered acceptable, but top-tier websites often reach values under 1 second. In controlled testing, the platform recorded a Speed Index of 1.1 seconds, indicating

efficient visual rendering. While this result demonstrates solid performance relative to Lighthouse thresholds, direct comparison with high-performance benchmarks should be made cautiously, and ideally supported by empirical studies or large-scale audits, both of which are outside this thesis's scope.

Cumulative Layout Shift (CLS)

CLS measures the cumulative impact of unexpected layout shifts during the lifespan of a page, which affects visual stability and user experience. A CLS score less than 0.1 is good, 0.1 to 0.25 requires improvement, and greater than 0.25 is poor, according to Lighthouse guidelines. This system achieved a CLS of 0.032, which is an indication of strong layout consistency in controlled tasks. This is a marginally significant value, and one should view it with caution. Furthermore, as CLS depends very much on how and when content (e.g., assets) is loaded, and may be affected by asynchronous elements, it is recommended to continuously monitor during production usage, to get an impression of the layout shifts happening in practice in a range of real-world conditions.

5.7 Summary

This chapter reports on the evaluation results of the platform, including user-centered usability testing and system-level technical metrics. This platform showed positive results in terms of responsiveness, engagement, and learning efficacy, and participants raised their understanding as well as interaction level. Quantitative results indicated system technological deployment stability and responsiveness, and qualitative comments indicated high levels of user satisfaction and some potential for improvement. These findings confirm the contributions of the previous chapters concerning design and implementation. These results will be explained in the chapter 6, focusing on the possible implications of these results, as well as the limitations, and their relations with the research questions.

6 Final Product Overview

This chapter presents an illustrative description of the final e-learning platform developed in the context of sleep medicine education. Based on the methods and technical specifications presented in the previous chapters, this chapter presents the practical development and successful completion of the initial objectives, which include easy-to-use interfaces and advanced interactive features.

The main goal is to provide evidence of how abstract principles and ideas, with technical foundations, can be refactored to build a coherent, user-focused learning tool. The next sections will present the most common view of the different parts of the platform.

6.1 Unauthenticated screens

The unauthenticated screens include the pages that are accessible before the user goes through the authentication process. These primarily serve to introduce the platform, present important information, and facilitate the authentication process.

6.1.1 Landing page

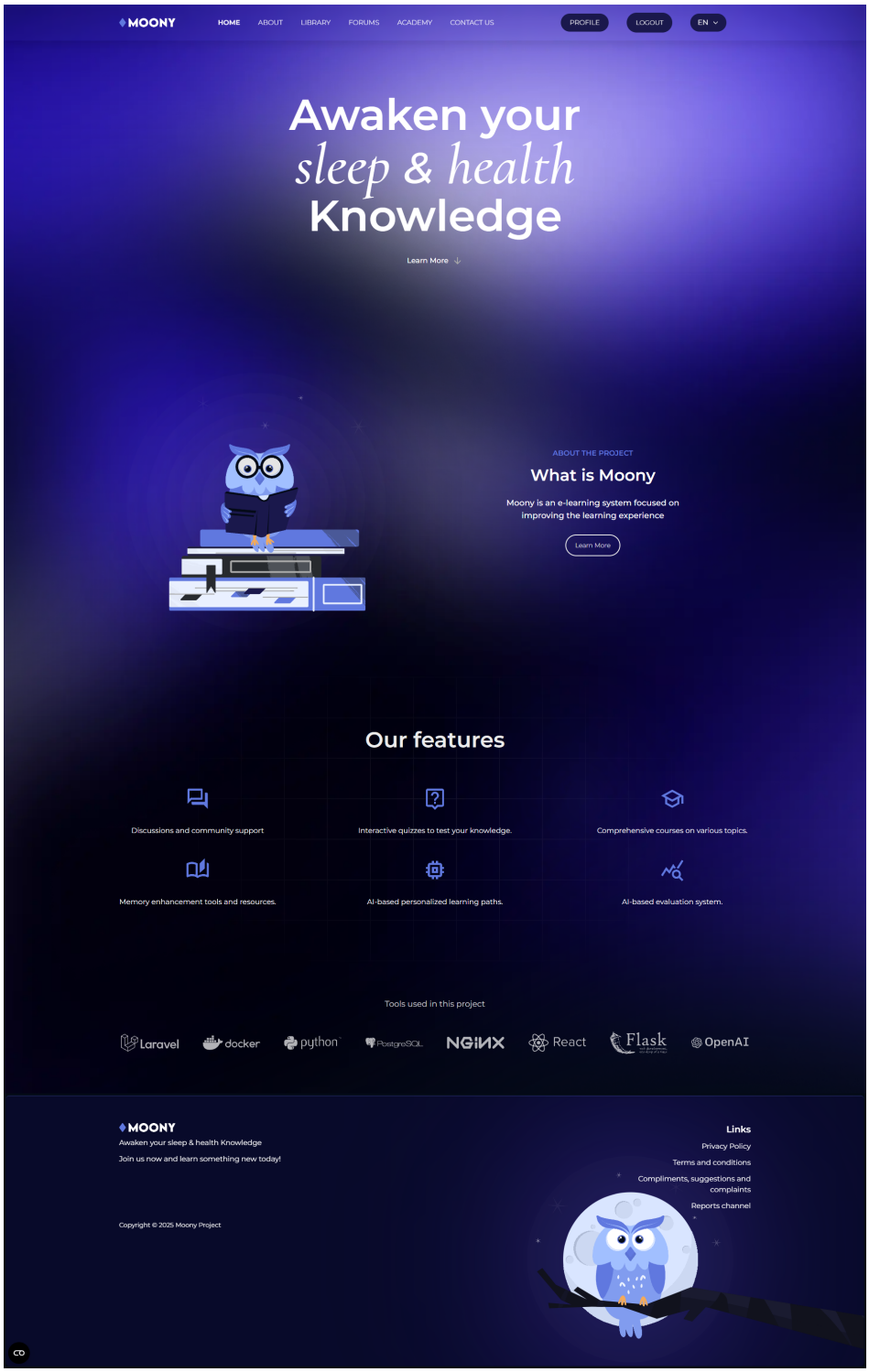

Fig. 24. Final preview of the Homepage.

6.1.2 About, Contact, Privacy Policy pages

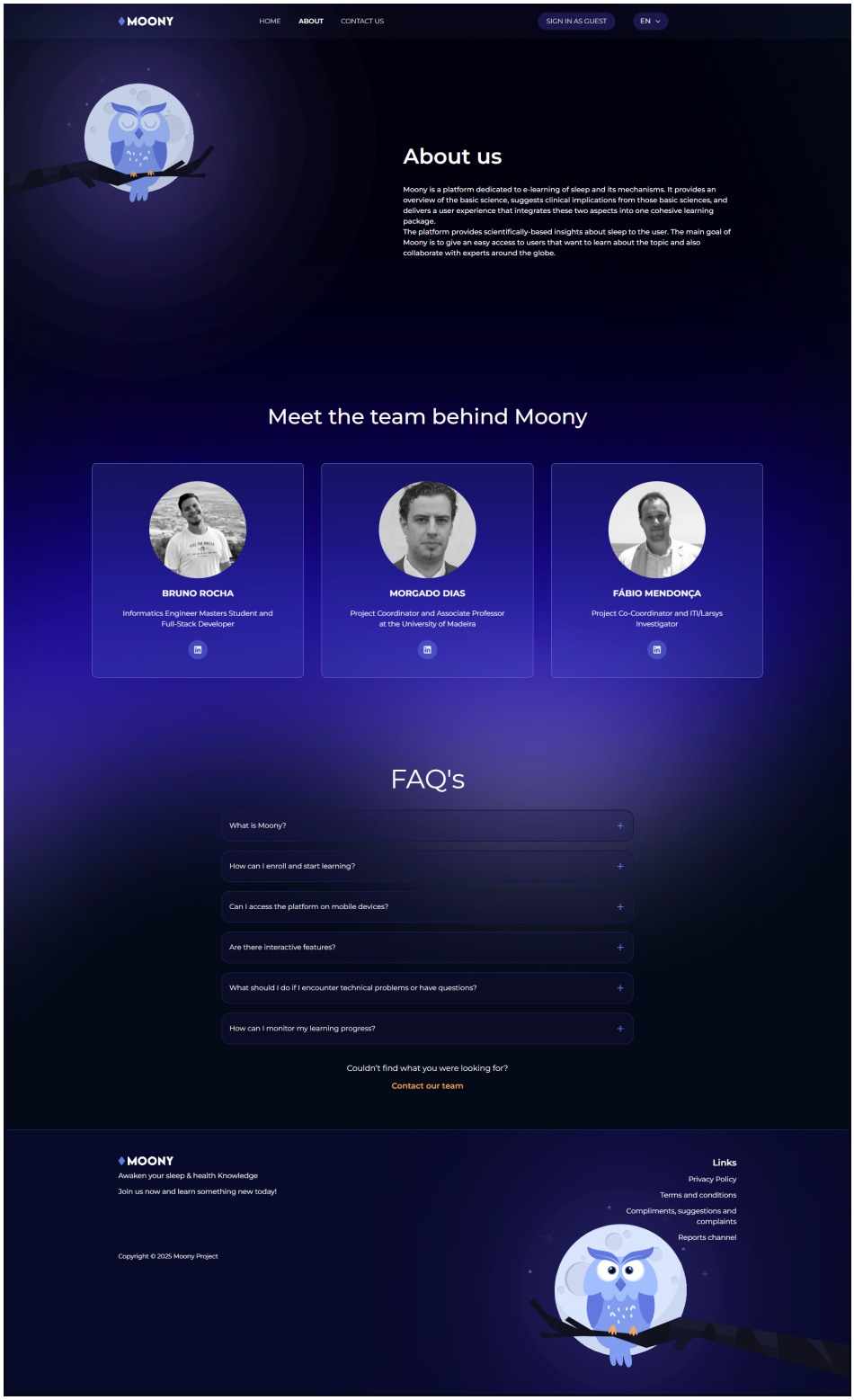

Fig. 25. Final preview of the About page.

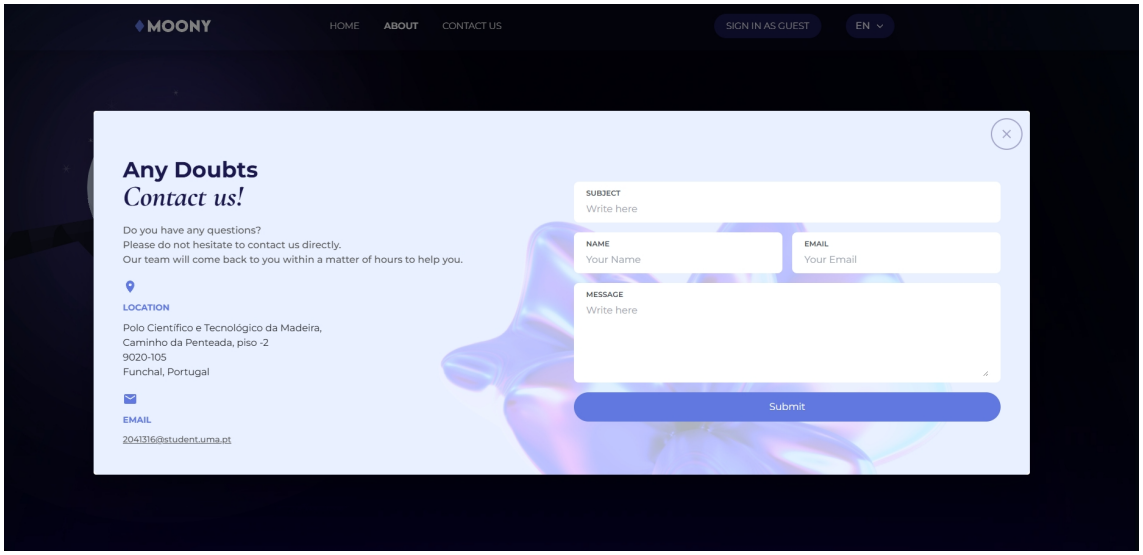

Fig. 26. Final preview of the Contact pop-up form.

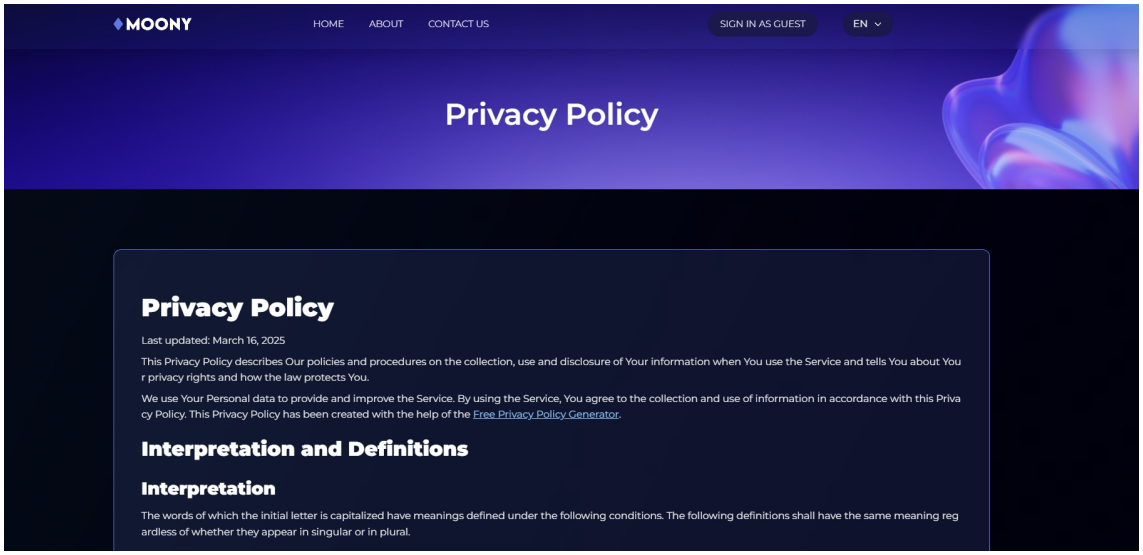

Fig. 27. Final preview of the layout for the privacy policy and terms and conditions page.

6.1.3 Login/Registration Pages

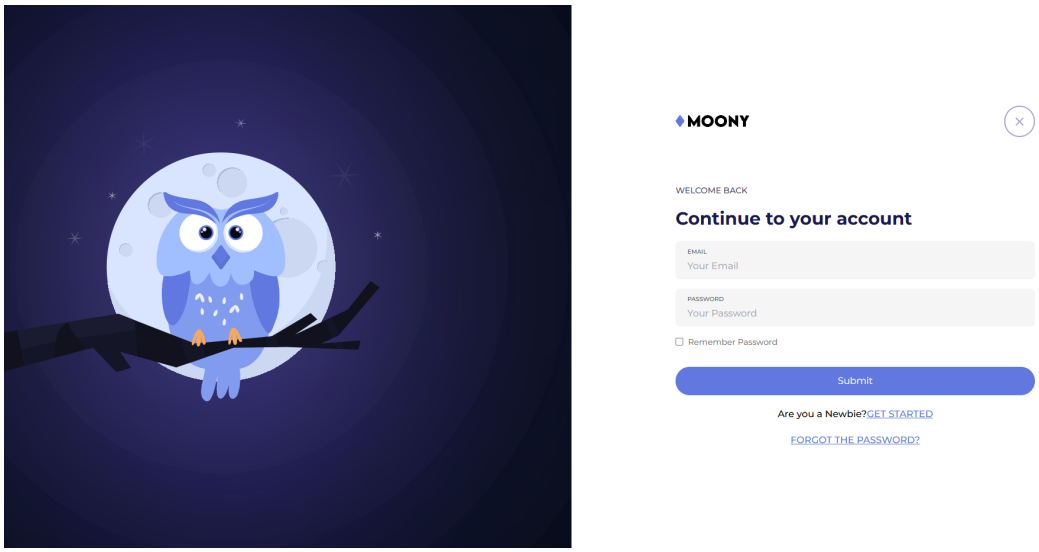

Fig. 28. Final preview of the login/registration page.

6.1.4 Maintenance page

Fig. 29. Final preview of the maintenance page.

6.2 Authenticated screens

The authenticated screens become accessible after users successfully log in, providing personalized and interactive educational content and features.

6.2.1 Library Module

The library module centralizes educational materials, including articles, journals, and news related to sleep medicine, enabling organized access and convenient browsing.

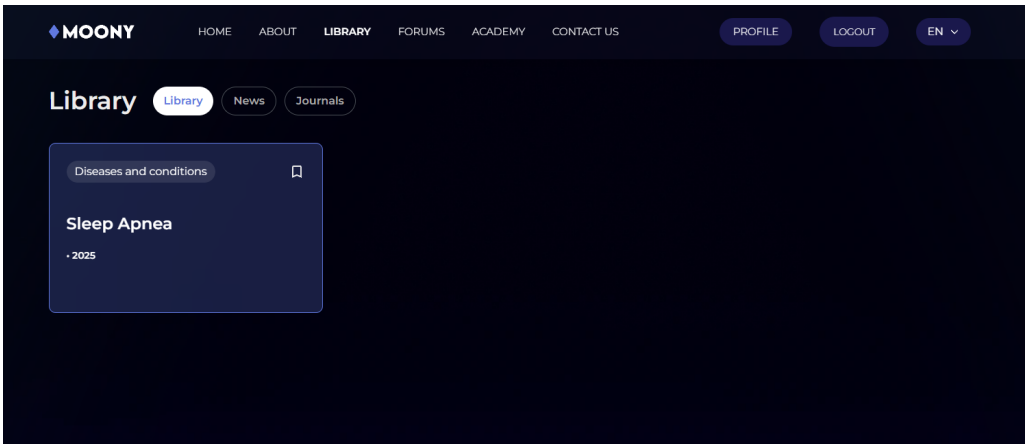

Fig. 30. Final preview of the library page.

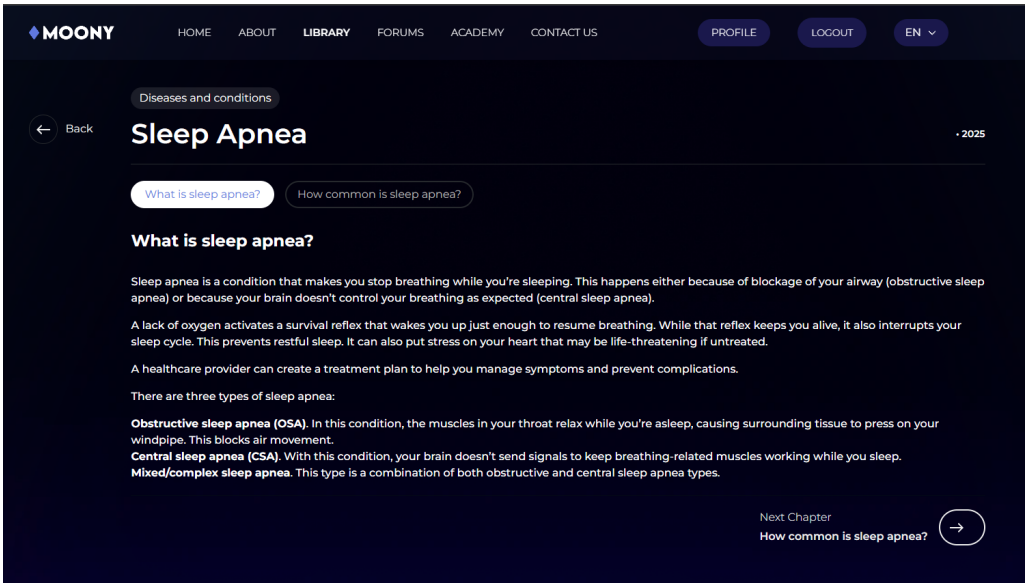

Fig. 31. Final preview of the library page content preview.

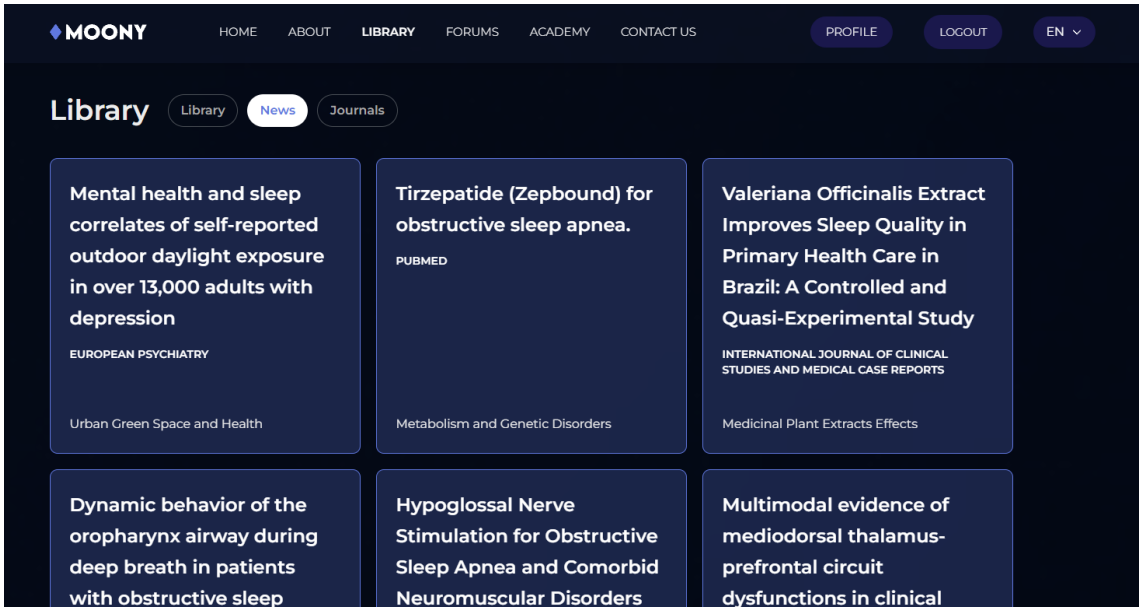

Fig. 32. Final preview of the news section inside the library module.

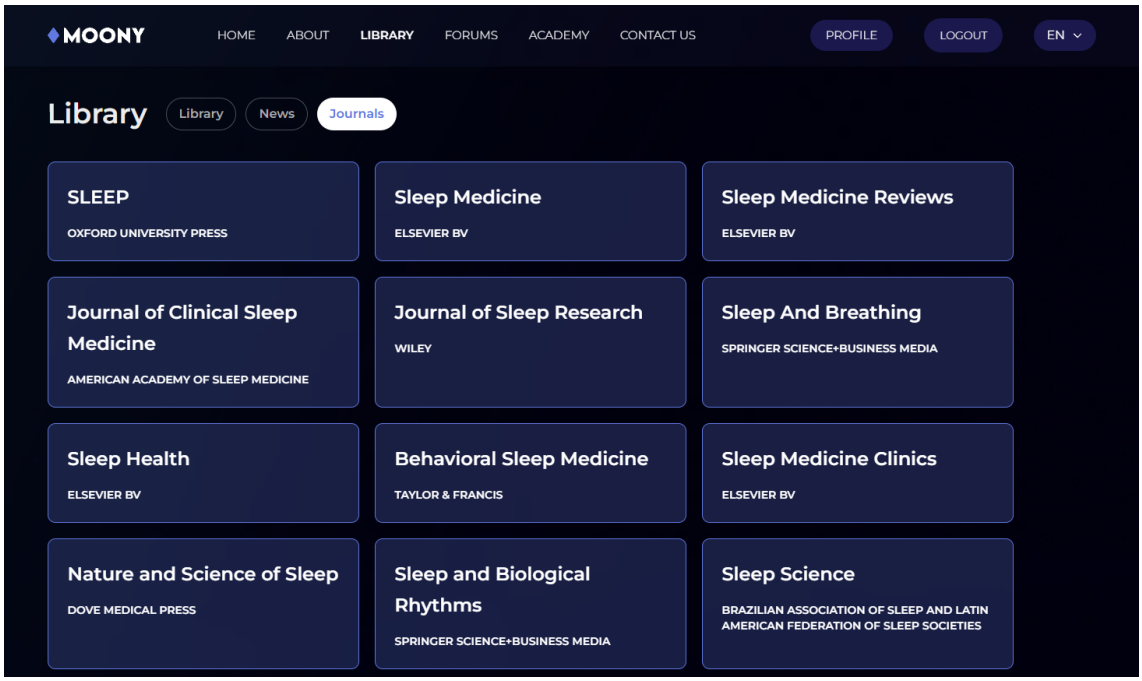

Fig. 33. Final preview of the journals section inside the library module.

6.2.2 Forum and Discussion Module

This interactive module facilitates communication and knowledge exchange among users through forums, discussions, and integration of chatbot support for enhanced interaction.

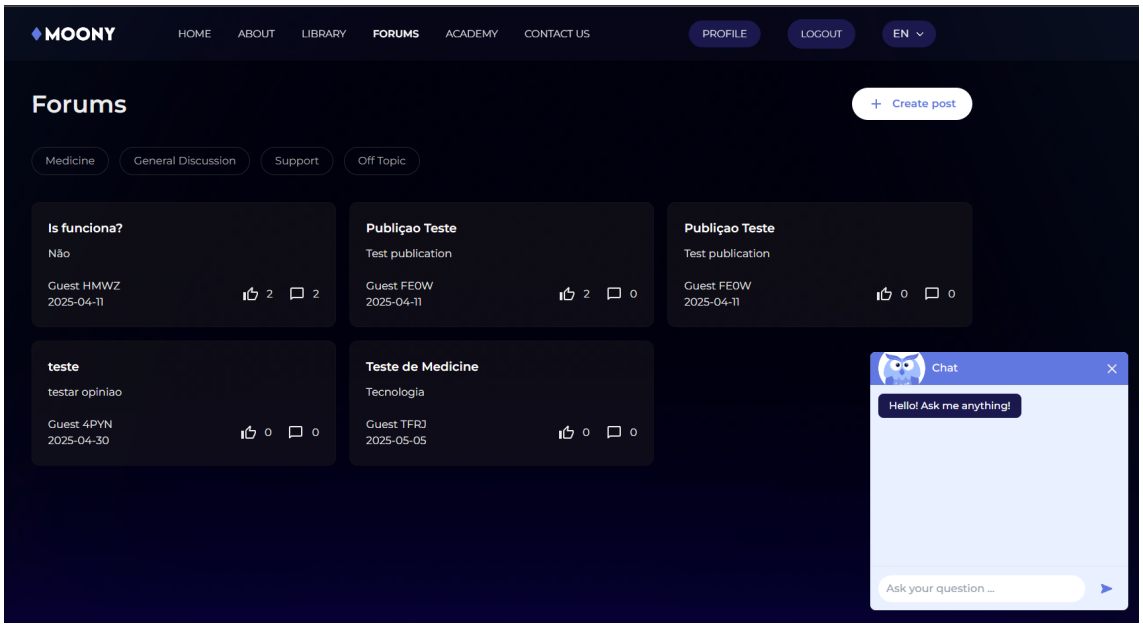

Fig. 34. Final preview of the forum dashboard with the preview of the chatbot in the right corner.

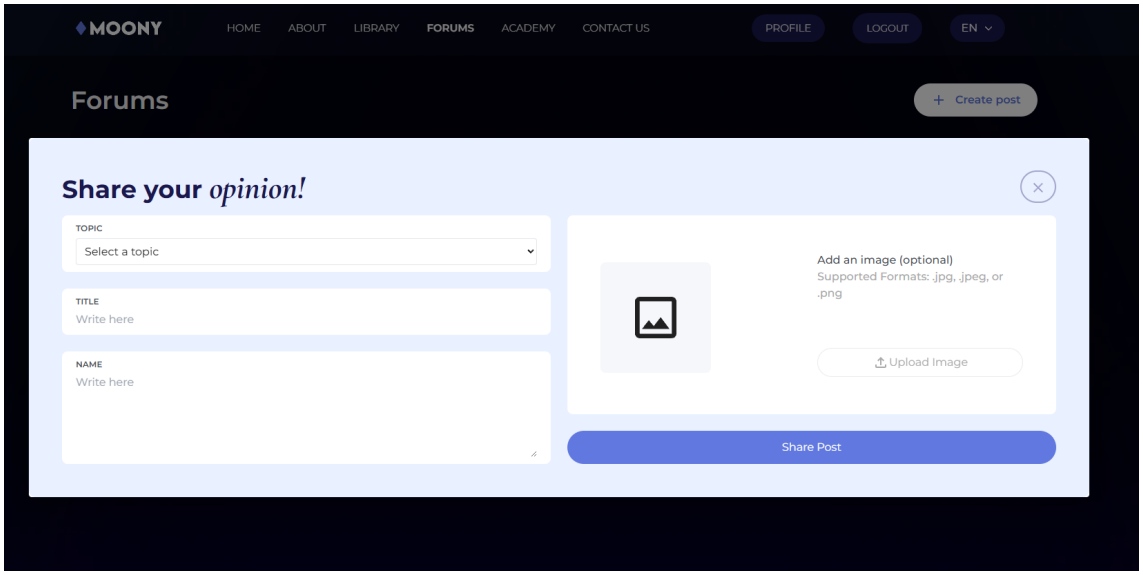

Fig. 35. Final preview of the thread registration form

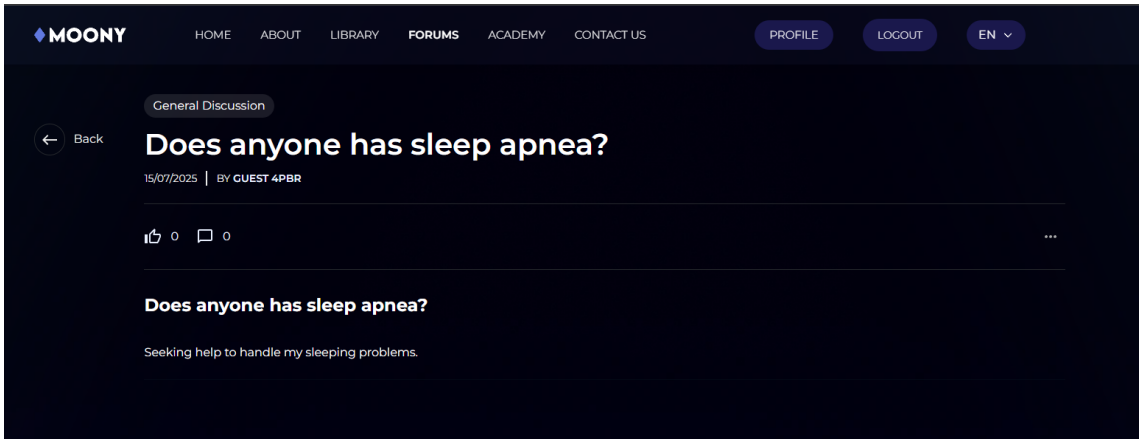

Fig. 36. Final preview of the forum thread content.

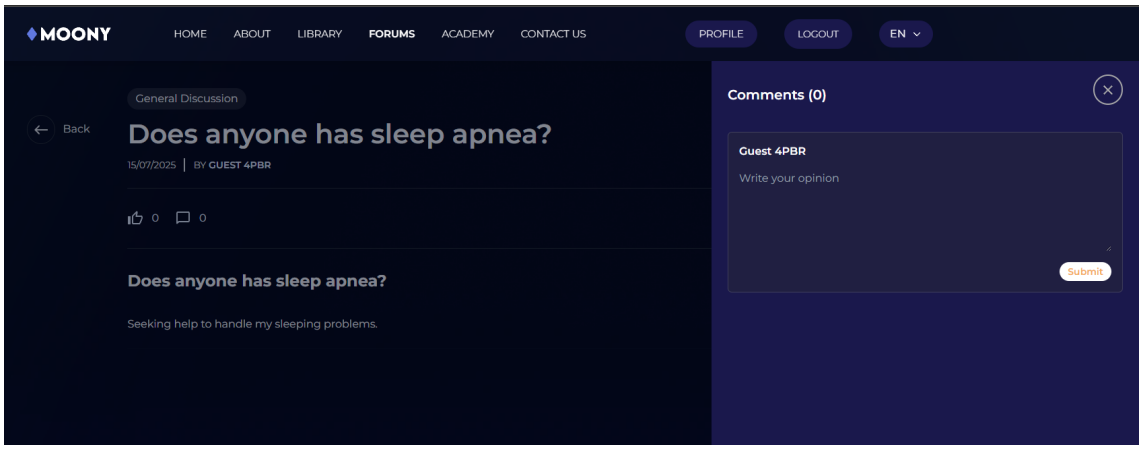

Fig. 37. Final preview of the forum thread content comment functionality.

6.2.3 Academy Module

The Academy module provides structured educational content, dynamic quizzes, automated assessments, and real-time feedback to optimize learning experiences.

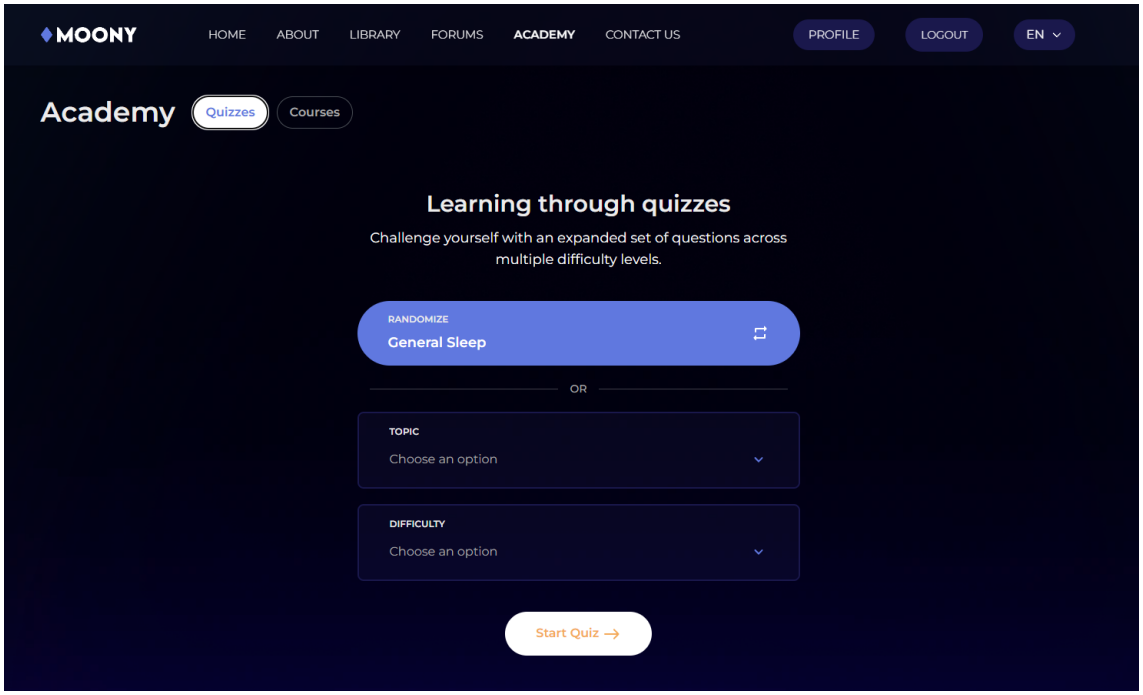

Fig. 38. Final preview of the quiz module in the academy page.

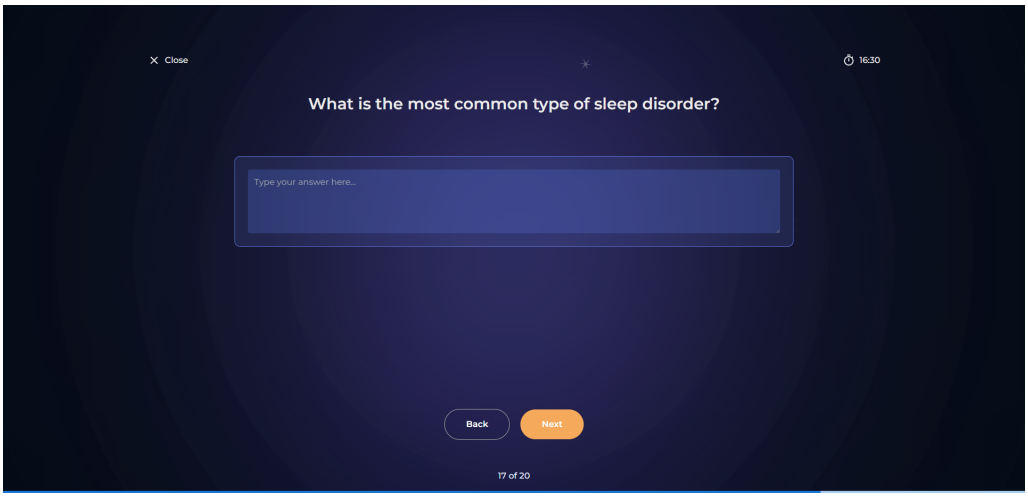

Fig. 39. Final preview of the quiz questions generation. It is possible to identify the timer, remaining questions, and iterative buttons.

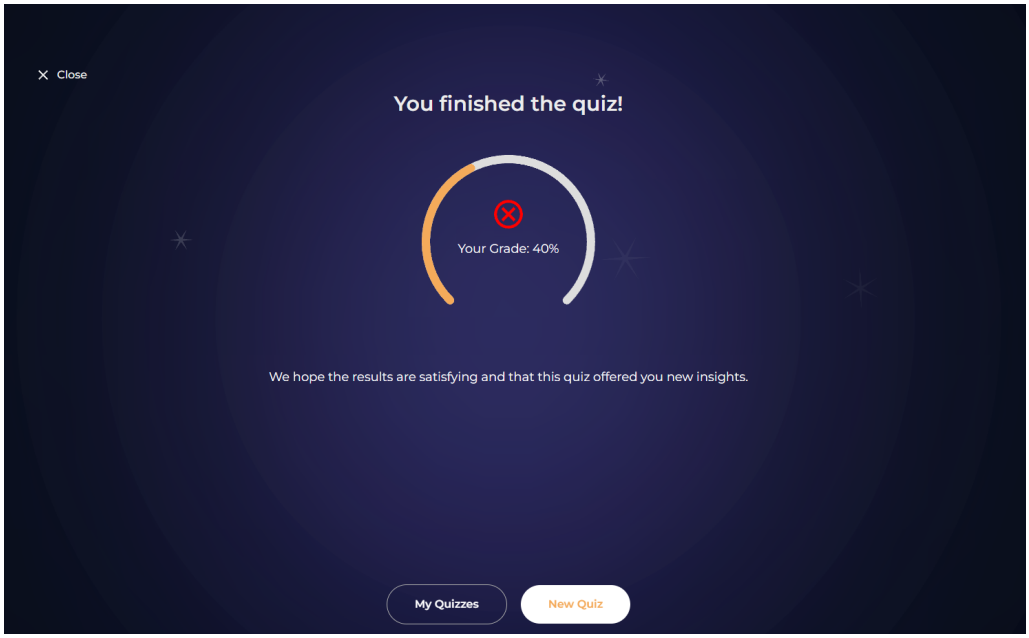

Fig. 40. Final preview of the quiz finished page. This page presents the grading after the automatic evaluation.

Fig. 41. Final page preview from the quiz review. This page presents all the responses and correct/suggested answers.

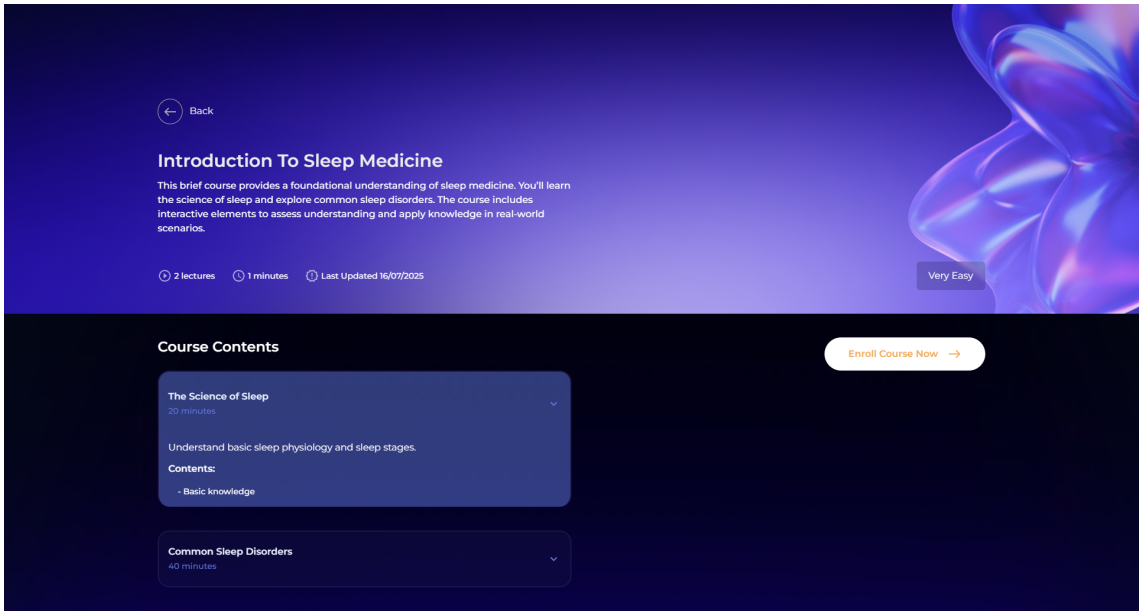

Fig. 42. Final preview of the course page.

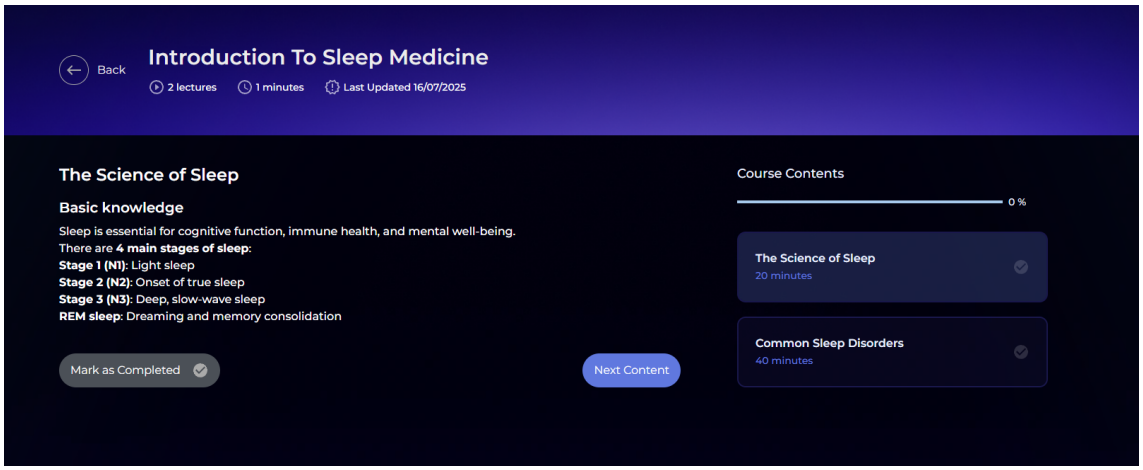

Fig. 43. Final preview of the course content page.

6.2.4 User Profile / Dashboard

Personalized dashboards offer a centralized view of user progress, courses, quizzes, and bookmarked resources, enhancing individual learning journeys.

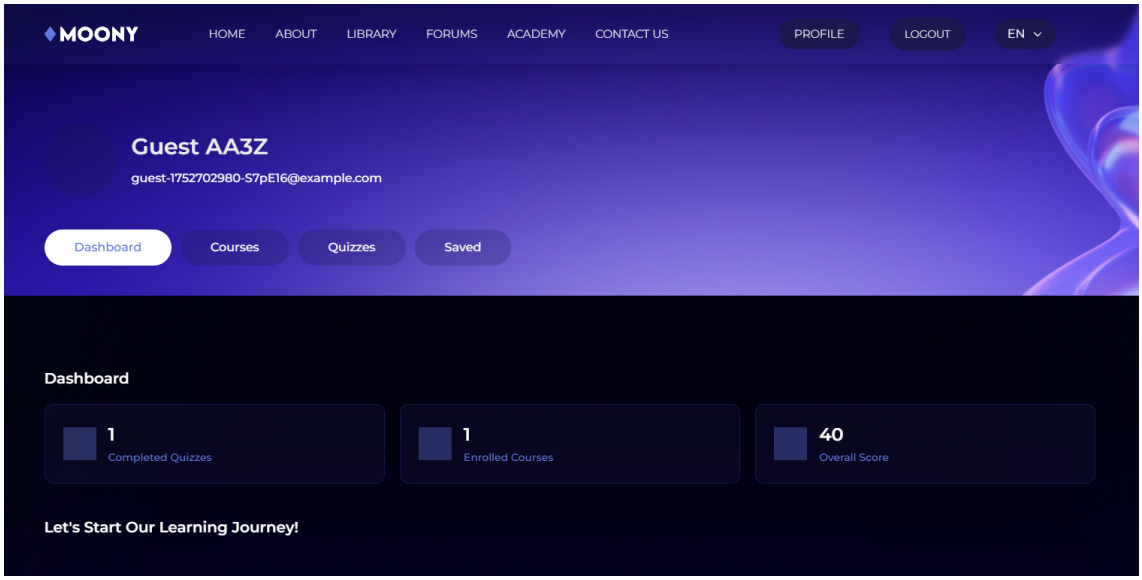

Fig. 44. Final preview of the main dashboard in the user profile.

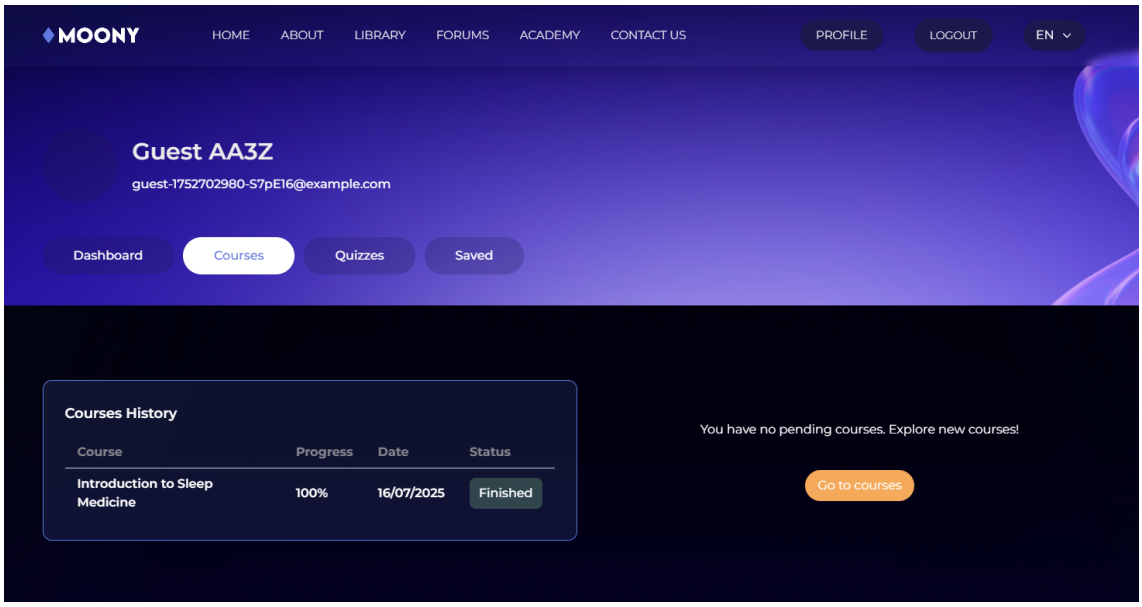

Fig. 45. Final preview of the courses dashboard. The user has an overview of the learning content currently enrolled, and recommended courses (when available).

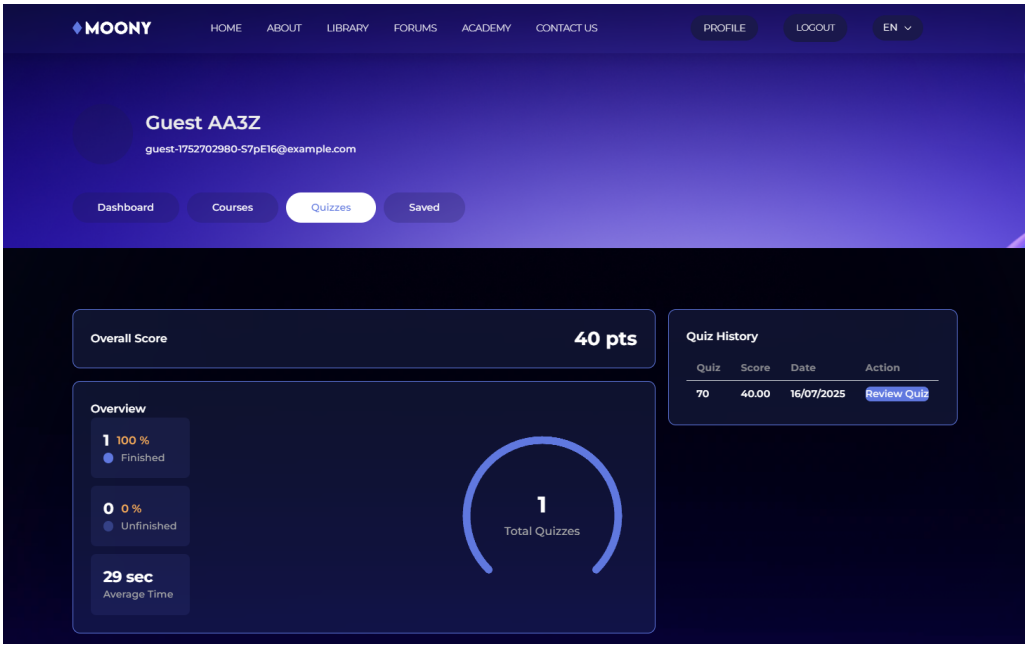

Fig. 46. Final preview of the quizzes dashboard. Here it lists the user's quiz history, making it possible to finish an uncompleted quiz or review the questions from previous quizzes.

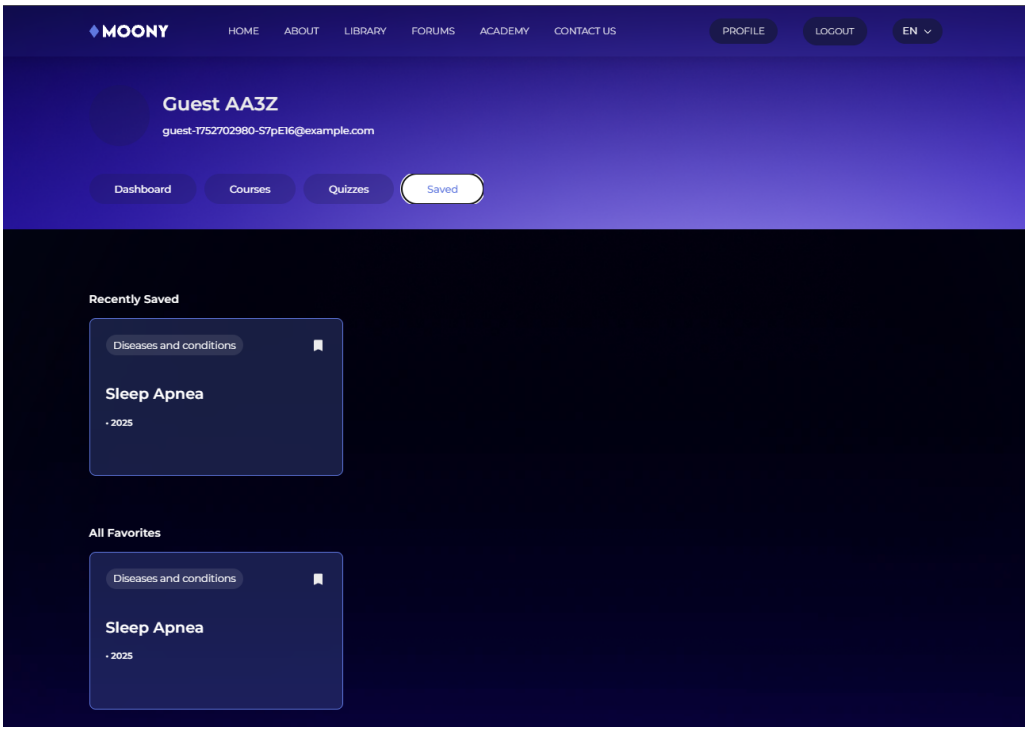

Fig. 47. Final preview of the saved contents dashboard. In this section, based on information from the library, the user can save by bookmarking relevant information.

6.3 Administrative screen

The administrative interface, implemented with the predefined CRM (see Section 4.2.3), serves as the primary control center for managing all platform content, user interactions and system configurations. Despite its simple and functional layout, this modules enable administrators perform a wide range of operations essential for maintaining the platform's adaptability and relevance.

Through this interface, administrators can perform all the CRUD (create, read, update and delete) operations to nearly all database entities, since the automatic CRUD functions imply the previous migrations and API controller. Therefore, its possible to change the content of FAQs, privacy policy, terms and conditions, library content, course material and quizzes items. User related functions as managing accounts, password, usernames, contact support and reports, are managed in this dashboard as well. Additionally, the module provides configuration options for system emails, platform themes, maintenance screens, and LLM-related parameters to ensure secure container operation. This level of control ensures that the platform remains flexible, modular and aligned with technical updates and pedagogical requirements.

The following images are small previews of the large background section developed to keep the platform content dynamic for the end user. In this section, although with a simple layout, it has the most important function, which is to manage all the content of the platform.

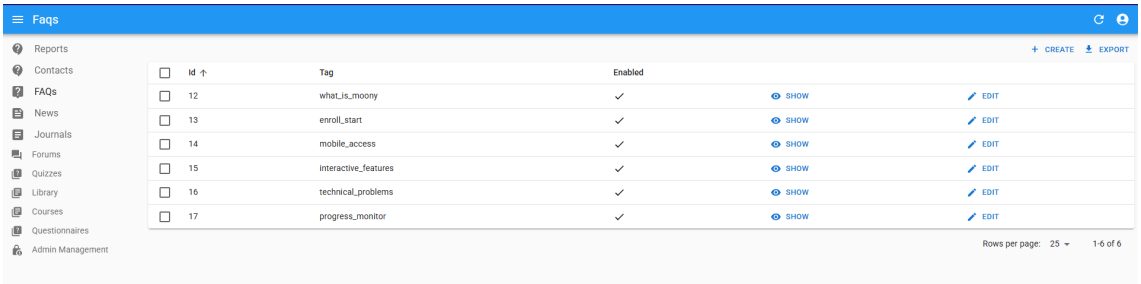

Fig. 48. Preview of the administrative FAQ, which is then presented on the About page.

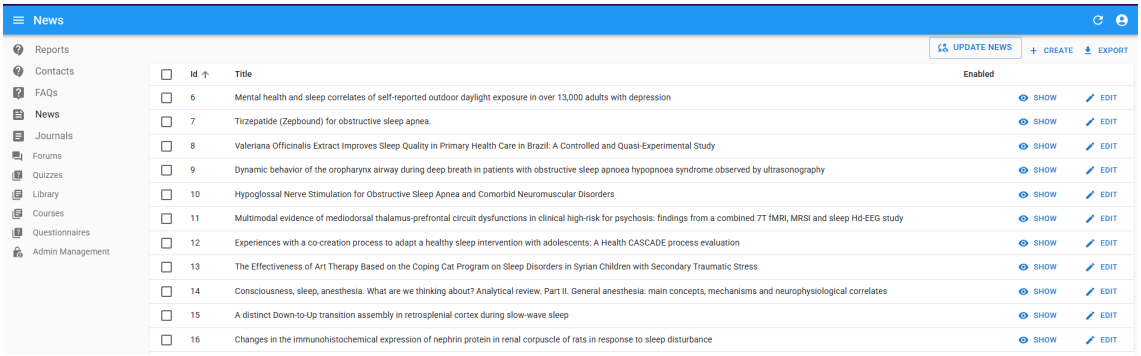

Fig. 49. Preview of the administrative page of news (in Library module).

7 Conclusion

This work discussed the design, development, and evaluation of an e-learning platform for sleep medicine education. The e-learning platform used Docker for service orchestration, the frontend interface uses ReactJS, the backend functionalities were implemented with Laravel, PostgreSQL is used for effective data storage, and Nginx as a reverse proxy, runs smoothly on all devices while maintaining simplicity concerning maintenance and scalability. The platform addressed a clear gap in medical education digital tools by incorporating interactive modules, an automated quiz generation system, and a user-centered design. Technical performance tests showed that API response times, database throughput, and overall web performance metrics consistently met established benchmarks, while user studies reported increased engagement, enhanced comprehension of complex sleep concepts, and positive interactions fostered through dynamically generated quizzes and modules. Notably, the LLM-powered quiz engine produced context-sensitive questions and real-time feedback with minimal human effort, illustrating a practical use of AI in medical learning.

Reviewing the functional and non-functional requirements identified in Section 3.2 confirmed compliance with the initial specifications. Functional features such as role-based authentication, progress tracking, and a central library of resources were all achieved and validated during the development and testing process. Also, all non-functional requirements were achieved, such as a level of usability and accessibility, a low response time, data protection complying with GDPR, and a reliable cross-platform experience. The modular and containerized architecture further supports maintainability and potential for scalability in the future.

The statistic analysis showed that the participant's perception of the stated improvement of the platform, was statistically significant for a sample of this size, and from pre-test to post-test the there was an average increase of 0.225 points on a Likert scale of five points, with a standard deviation of 0.505, suggesting that most participants had the perception of improvement. Post-test ratings were highest for clarity and relevance of content, ease of navigation, and cohesive visual organization, which indicated that usability features, centered in the design of the platform, led to a satisfactory experience for learners.

Correlation analysis further supported these findings, indicating that participants with greater initial interest in the subject and stronger digital skills provided higher evaluations. Interestingly, the greatest improvements were found among those with lower initial familiarity and confidence, suggesting the platform is especially effective for engaging and benefiting less experienced users, fulfilling its goal of widening access to sleep-medicine knowledge.

These findings align closely with current literature on best practices for medical e-learning, emphasizing interactive content delivery, concise micro-modules, and robust back-end frameworks to guarantee stability and security. Specifically, the automatic grading, as well as the content for quizzes traversing a pipeline of LLM prompts, extend prior work by demonstrating that AI-generated formative assessment can be both scalable and pedagogically effective.

Considering initial objectives from Chapter 1, the platform has effectively met the goal of creating an effective, accessible, and user-friendly online learning environment for sleep-medicine education. The study also met its research questions by identifying and validating important de-

sign components such as interactive modules, responsive layouts and interfaces, personalized dashboards, and AI-based assessments.

Despite the positive results, many limitations were encountered during both the evaluation and development that should be acknowledged:

- Focused User Group: The platform's primary targeting of clinicians or students specializing in sleep analysis may limit the generalization of findings to other medical fields without further adaptation and testing.
- Technology Dependence: Users unfamiliar with web-based technologies found it difficult to work with web technology, which may limit usability for digitally inexperienced segments of the medical community.
- Indirect Clinical Impact: While educational effectiveness was demonstrated, direct clinical impacts - such as improvements in patient diagnosis and treatment - were beyond this study's scope. Future longitudinal studies are needed to quantify clinical impacts.
- Platform Scalability: Although scalability is good in the experiments to date, extended real-world use over longer time scales may identify unanticipated performance bottlenecks with the current configuration of the platform. These could be mitigated through environment-specific configuration adjustments.

For future development, the order of priority will be integration of improved device support, offline access on any device, and multilingual content support to foster international accessibility. In addition, consistent reinforcement of data privacy protections, particularly about third-party LLM APIs, will be essential to maintain trust between users and ensure compliance with regulations.

In conclusion, this research showed how user-centered design, modern web tools, and AI-generated content are combined to enhance the quality of online learning in niche fields of medicine. Future research should directly assess the sustained impact on clinical outcomes. Additionally, to further increase global accessibility, ongoing development should focus on expanding the system's accessibility through broader device compatibility, offline functionality, and multilingual content. These improvements will enhance user inclusion across multiple healthcare and educational settings.

(This page intentionally left blank.)

References

[1] J. M. Ellenbogen, David Foulkes, Thien Thanh Dang-Vu, Melodee A. Mograss, and Rosalind D. Cartwright. Sleep, November 19 2020. Encyclopedia Britannica. Accessed October 25, 2020.

[2] AK Patel, V Reddy, and JF Araujo. *Physiology, Sleep Stages*. StatPearls Publishing, Treasure Island (FL), 2020. Updated 2020 Apr 29.

[3] Maurice Ohayon et al. National sleep foundation's sleep quality recommendations: first report. *Sleep Health*, 3(1):6–19, 2017.

[4] AG Harvey, K Stinson, KL Whitaker, D Moskovitz, and H Virk. The subjective meaning of sleep quality: a comparison of individuals with and without insomnia. *Sleep*, 31(3):383–393, March 2008.

[5] Andrew D. Krystal and Jack D. Edinger. Measuring sleep quality. *Sleep Medicine*, 9 Suppl 1:S10–7, 2008.

[6] Fábio Mendonça, Sheikh Shanawaz Mostafa, Fernando Morgado-Dias, Antonio G. Ravelo-García, and Thomas Penzel. A review of approaches for sleep quality analysis. *IEEE Access*, 7:24527–24546, 2019.

[7] Rachel Ellaway and Ken Masters. Amee guide 32: e-learning in medical education part 1: Learning, teaching and assessment. *Medical Teacher*, 30(5):455–473, 2008.

[8] J. E. Orr, P. N. DeYoung, and R. L. Owens. The future of the sleep lab: It's complicated. *Journal of Clinical Sleep Medicine*, 14(4):499–500, 2018.

[9] Demetra S. Stamm, Sandra Taylor, Uyen Thao Nguyen, and Kimberly A. Hardin. Survey evaluating sleep education catalyzed change in residency training. *International Journal of Clinical Medicine*, 2015.

[10] Thomas Penzel, Dirk Pevernagie, Zoran Đogaš, Ludger Grote, S de Lacy, Andrea Rodenbeck, Sören Berg, Fabio Cirignotta, Marie-Pia d'Ortho, Diego Garcıa-Borreguero, Patrick Lévy, Lino Nobili, Teresa Paiva, Philippe Peigneux, Thomas Pollmächer, Dieter Riemann, Debra J. Skene, Marco Zucconi, and C A Espie. Catalogue of knowledge and skills for sleep medicine. *Journal of Sleep Research*, 23(2):222–238, 2013.

[11] Simone B. Duss. The swiss sleep house bern - a new approach to sleep medicine. *Clinical and Translational Neuroscience*, 8(1), 2024.

[12] Thomas Penzel, Dirk Pevernagie, Philippe Peigneux, Tiina Paunio, Walter T. McNicholas, Zoran Đogaš, Ludger Grote, Andrea Rodenbeck, Fabio Cirignotta, Marie-Pia d'Ortho, Lino Nobili, Teresa Paiva, Thomas Pollmächer, Dieter Riemann, Marco Zucconi, Elizabeth A. Hill, Erna-Sif Arnardottir, and Liborio Parrino. Sleep medicine catalogue of knowledge and skills – revision. *Journal of Sleep Research*, 2021.

[13] Hartmut Schulz and Piero Salzarulo. The development of sleep medicine: A historical sketch.

Journal of Clinical Sleep Medicine, 12(7):1041–1052, 2016.

[14] David A. Cook and Denise M. Dupras. A practical guide to developing effective web-based learning. *Journal of General Internal Medicine*, 19(6):698–707, 2004.

[15] Gabriele Piccoli, Rami Ahmad, and Blake Ives. Web-based virtual learning environments: A research framework and a preliminary assessment of effectiveness in basic it skills training. *MIS Quarterly*, 25:401–426, 2001.

[16] S. Sharma and S. Garg. Web based learning: A research on assessing its effectiveness in contrast to traditional classroom learning environment. In *2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT)*, pages 32–34, New Delhi, 2016.

[17] David A. Cook. Web-based learning: pros, cons and controversies. *Clinical Medicine*, 7(1):37–42, 2007.

[18] Norhasyimah Hamzah, Arihasnida Ariffin, and Hashima Hamid. Web-based learning environment based on students' needs. *IOP Conference Series: Materials Science and Engineering*, 226:012196, 2017.

[19] T. Anghel, A. Florea, A. Gellert, and D. Florea. Web-based technologies for online e-learning environments. In *Seventh International Scientific Conference "eLearning and Software for Education"*, Bucharest, April 2011.

[20] S. Chiemeke and O. M. Imafidor. Web-based learning in periods of crisis: Reflections on the impact of covid-19. *International Journal of Computer Science & Information Technology (IJCSIT)*, 12, 2020.

[21] J. G. Ruiz, M. J. Mintzer, and R. M. Leipzig. The impact of e-learning in medical education. *Academic Medicine*, 81(3):207–212, 2006.

[22] Roy Huynh. The role of e-learning in medical education. *Academic Medicine*, 92(4):430, 2017.

[23] D. O'Doherty, M. Dromey, J. Lougheed, et al. Barriers and solutions to online learning in medical education – an integrative review. *BMC Medical Education*, 18:130, 2018.

[24] B. Preim and C. P. Botha. *Visual computing for medicine: theory, algorithms, and applications*, pages 805–823. Newnes, 2013.

[25] Thomas Binder. E-learning in medicine: Current status and future developments. *Hamdan Medical Journal*, 12:147, 2019.

[26] Yann Hicke, Jadon Geathers, Niroop Rajashekar, Colleen Chan, Anyanate Gwendolyne Jack, Justin Sewell, Mackenzi Preston, Susannah Cornes, Dennis Shung, and Rene Kizilcec. Medsimai: Simulation and formative feedback generation to enhance deliberate practice in medical education, 2025.

[27] Virti Ltd. Virti: Virtual reality and ai training platform for healthcare. `https://www.virti.com/industries/healthcare/`, 2024. Accessed: 2025-07-18.

[28] Amplifire Inc. Amplifire: Adaptive learning platform. `https://amplifire.com/`, 2024. Ac-

cessed: 2025-07-18.

[29] Amr Alshatnawi, Remi Sampaleanu, and David Liebovitz. Meditools – medical education powered by llms, 2025.

[30] Santi Miguel, Jorge nd Caballé, Fatos Xhafa, and Josep Prieto. Security in online web learning assessment. *World Wide Web*, 18(6):1655–1676, 2015.

[31] Essohanam Djeki, Jules Dégila, and Muhtar Hanif Alhassan. Best practices for ensuring security and privacy in e-learning environments, 2024.

[32] Florence Martin, Ting Sun, and Carl D. Westine. A systematic review of research on online teaching and learning from 2009 to 2018. *Computers & Education*, 159:104009, 2020.

[33] Thamsanqa Keith Miya and Irene Govender. Ux/ui design of online learning platforms and their impact on learning: A review. *International Journal of Research in Business and Social Science (2147- 4478)*, 11(10):316–327, Dec. 2022.

[34] Jaime Govea, Ernesto Ocampo Edye, Solange Revelo-Tapia, and William Villegas-Ch. Optimization and scalability of educational platforms: Integration of artificial intelligence and cloud computing. *Computers*, 12(11), 2023.

[35] Bernhard Ertl, Katharina Ebner, and Kathy Kikis-Papadakis. Evaluation of e-learning. *International Journal of Knowledge Society Research (IJKSR)*, 1(3):33–43, 2010.

[36] Docker Inc. Docker overview. `https://docs.docker.com/get-started/overview/`, 2020. Accessed: 2020-11-07.

[37] E. N. Preeth, F. J. P. Mulerickal, B. Paul, and Y. Sastri. Evaluation of docker containers based on hardware utilization. In *2015 International Conference on Control Communication & Computing India (ICCC)*, pages 697–700, Trivandrum, 2015.

[38] Docker Inc. What is a container? a standardized unit of software. `https://www.docker.com/resources/what-container`, 2021. Accessed: 2020-11-07.

[39] Docker. Docker overview. `https://docs.docker.com/compose/`, 2021. Accessed: 2020-11-07.

[40] Laravel Developers. Laravel documentation. `https://laravel.com/docs`, 2020. Accessed: 2020-12-04.

[41] Mark Smallcombe. SQL vs NoSQL: 5 critical differences. `https://www.integrate.io/blog/the-sql-vs-nosql-difference/`, jun 2024. Integrate.io Blog. Accessed August 19, 2024.

[42] U. Singh. Top 5 database for web applications. `https://bestinterviewquestion.medium.com/top-5-database-for-web-applications-d71a4229fa37`, 2019. Accessed: 2020-10-12.

[43] Stephen J. Bigelow. What is REST API (RESTful API)? `https://www.techtarget.com/searchapparchitecture/definition/RESTful-API`. Accessed: 2020-11-12.

[44] R. Carvalho. Single page applications: When and why you should use them. `https://www.scalablepath.com/blog/single-page-applications/`, 2017. Accessed: 2020-11-12.

(This page intentionally left blank.)

A User consent form

UNIVERSIDADE da MADEIRA

Consentimento Informado, Esclarecido e Livre para Participação em estudos de Investigação

Identificação do Investigador: Bruno Rodrigo Faria Gonçalves Rocha

Título do estudo: Development of an E-learning platform: Application to sleep analysis

Enquadramento: Atualmente deparamo-nos com o crescimento do interesse na área do sono, com especial atenção na qualidade do sono, tal como na área do E-learning, pelo que o seu cruzamento é inevitável apesar de ainda existirem preocupações quanto à eficácia da educação online na área da medicina. Esta proposta tem como objetivo maximizar a aprendizagem dos profissionais clínicos e facilitar o acesso e a troca de informações sobre o tema.

Explicação do estudo: O objetivo do estudo é avaliar a usabilidade de uma plataforma de e-learning desenvolvida para a aprendizagem sobre o sono. Pretende-se identificar oportunidades de melhoria na interface e na experiência de utilização, de forma a otimizar o processo de aprendizagem. Não serão efetuadas quaisquer gravações de audio / video, sendo que os dados recolhidos através dos questionarios serão guardados até um período máximo de 6 meses sendo que após esse prazo todos os dados serão eliminados das bases de dados e backups associados.

Condições e financiamento: Não temos financiamento ao estudo a ser realizado e que a participação no estudo por parte dos participantes é voluntária e não tem direito a qualquer pagamento.

Anonimato e confidencialidade: A confidencialidade dos dados é assegurada através da recolha dos mesmos de forma anónima, sem qualquer mecanismo de registo que permita identificar os participantes. Além disso, todos os dados serão armazenados de forma segura e apenas acessíveis à equipa de investigação, sendo utilizados exclusivamente para a avaliação da usabilidade da plataforma. Não ha registos de quaisquer dados de identificação ou outros dados pessoais que possam identificar os participantes diretamente / indiretamente. O participante pode agora ou a qualquer momento, questionar ou esclarecer dúvidas sobre o projeto e a sua participação.

Investigador: Bruno Rodrigo Faria Gonçalves Rocha, Consultor IT, Funchal.
Telefone: +351 967069324.
Email: 2041316@student.uma.pt

Por favor, leia com atenção esta informação. Se achar que algo está incorreto ou que não está claro, não hesite em solicitar mais informações.

Se concorda com a proposta que lhe foi feita, queira assinar este documento.

Assinatura de quem pede consentimento: _____

Fig. 50. Usability consent form.

UNIVERSIDADE da MADEIRA

Declaração de Consentimento do Participante

Eu, _____ declaro ter lido e compreendido este documento, bem como as informações verbais que me foram fornecidas pela/s pessoa/s que acima assina/m. Foi-me garantida a possibilidade de, em qualquer altura, recusar participar neste estudo sem qualquer tipo de consequências. Desta forma, aceito participar neste estudo e permito a utilização dos dados, que de forma voluntária forneço, confiando em que apenas serão utilizados para fins científicos e publicações que delas decorram e com as garantias de confidencialidade e anonimato que me são dadas pelo/a investigador/a.

Assinatura legível e manuscrita: _____

Data: ___ / ___ / _____

Fig. 51. Participant consent declaration.

B Pre-test Questionnaire

The pre-test questionnaire was designed to assess participants' familiarity and comfort with digital learning environments and interest in the topic of sleep prior to using the e-learning platform. Each statement was rated on a 5-point Likert scale:

1 = Strongly Disagree 2 = Disagree 3 = Neutral 4 = Agree 5 = Strongly Agree

1. I am familiar with using e-learning platforms.
2. I feel comfortable using digital devices (computers, tablets, smartphones).
3. I feel comfortable with the format of usability testing.
4. I trust my ability to identify technical problems while using online platforms.
5. I am interested in or curious to learn more about sleep.

C Post-Test Questionnaire

The post-test questionnaire evaluated the usability, visual design, structure, and educational effectiveness of the e-learning platform after user interaction. All items were rated on the same 5-point Likert scale:

1 = Strongly Disagree 2 = Disagree 3 = Neutral 4 = Agree 5 = Strongly Agree

1. The platform's design makes it easy to navigate between chapters and resources.
2. The platform's structure allows quick and intuitive access to information about sleep.
3. The content about sleep is clear, objective, and easy to understand.
4. Page and resource loading times are appropriate and do not hinder the user experience.
5. The site's features meet the needs of an effective learning environment on sleep.
6. The layout and organization of content encourage deeper exploration of the topic.
7. The visual and structural design of the site is consistent across all pages.
8. The site is responsive and adapts well to different devices (desktop, tablet, smartphone).
9. Help options, FAQ, or technical support are easily accessible and effective.
10. I believe this platform would be a valuable tool for individual or group learning.
11. Overall, the platform is intuitive and facilitates learning about sleep.

D Pre-Test Questionnaire (All participants)

Participant	P1	P2	P3	P4	P5
1	2	5	3	3	4
2	4	4	5	4	1
3	1	1	5	4	3
4	4	5	4	3	4
5	3	4	5	4	3
6	4	5	5	5	5
7	5	4	5	5	4
8	1	5	5	4	4
9	5	5	5	5	4
10	5	5	5	4	4
11	3	4	3	4	4
12	2	4	4	4	4
13	4	4	4	4	3
14	4	5	5	4	4
15	5	5	5	5	5
16	5	5	5	5	5
17	4	5	5	5	4
18	4	4	4	4	4
19	4	5	5	4	4
20	5	5	5	5	4
21	5	5	5	4	4
22	2	5	5	3	4
23	4	3	4	2	3
24	4	5	4	4	4

E Post-Test Responses (All Participants)

Participant	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
1	5	5	5	4	4	4	4	5	5	5	5
2	3	2	3	4	2	3	4	2	3	3	3
3	4	3	2	2	3	4	3	3	4	3	4
4	3	4	4	5	5	5	5	5	5	5	5
5	4	4	4	5	5	4	4	5	5	4	4
6	4	4	4	4	4	4	5	5	5	5	5
7	5	5	4	5	4	4	4	5	5	5	5
8	5	5	4	4	5	5	5	5	5	5	5
9	5	5	5	5	5	5	5	5	5	5	5
10	4	4	4	4	5	4	5	5	5	5	5
11	4	4	5	4	4	5	5	4	4	5	4
12	4	4	5	4	5	4	4	5	5	4	5
13	5	5	5	5	5	5	5	5	5	5	5
14	4	4	4	5	4	4	4	4	5	5	5
15	5	5	5	5	5	5	5	5	5	5	5
16	5	5	5	5	5	5	5	5	5	5	5
17	5	5	5	5	5	5	5	5	5	5	5
18	5	5	4	4	4	4	4	4	4	4	4
19	4	4	4	4	4	4	4	4	4	4	4
20	4	5	5	5	5	5	5	5	5	5	5
21	4	5	5	5	5	5	5	5	5	5	5
22	4	5	4	4	4	4	4	4	4	4	4
23	4	4	4	4	4	4	4	4	4	4	4
24	5	5	5	5	5	5	5	5	5	5	5

F Full Database Schema

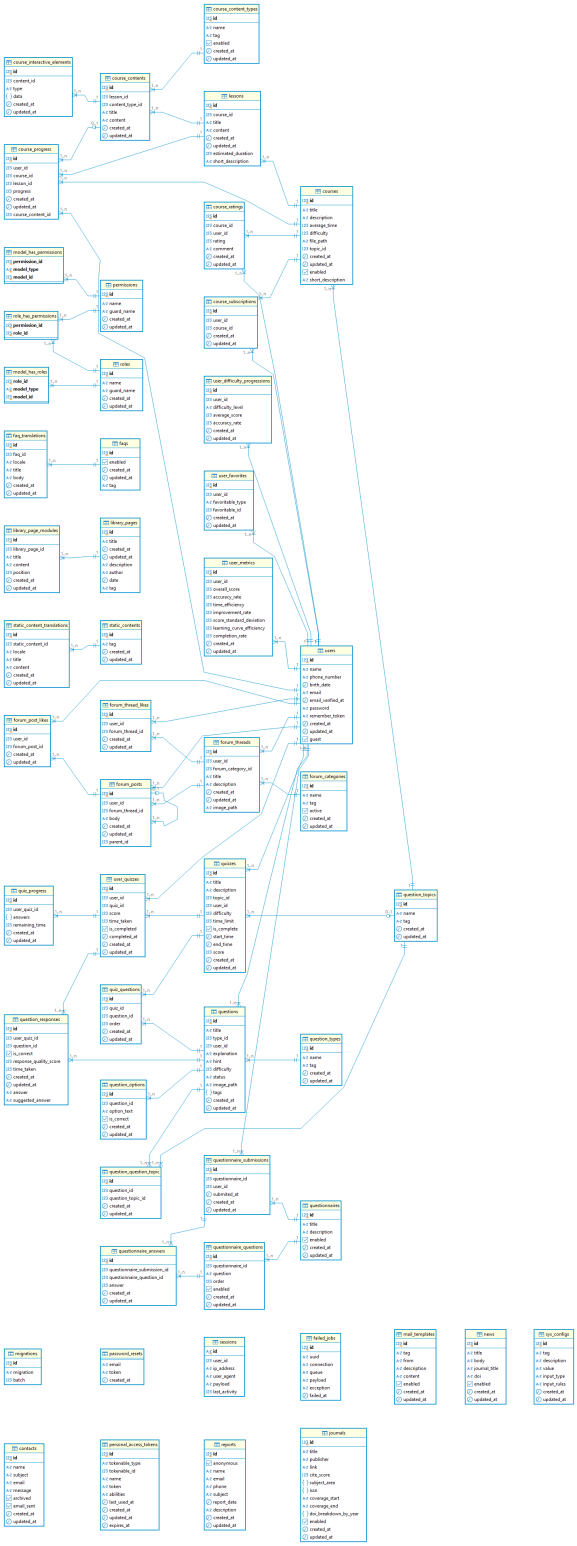

Fig. 52. Complete Entity-Relationship Diagram for the Platform Database.