



Come ThinkAgain

Certification based Education
Training System

Functional and Technical Specification Framework

The information and views set out in this report are those of the authors and do not necessarily reflect the official opinion of the Commission. Neither the Commission nor any person acting on the Commission's behalf may be held responsible for the use which may be made of the information contained therein.





TABLE OF CONTENTS

EXECUTIVE SUMMARY	7
2. INTRODUCTION	8
2.1 Project Overview	8
2.2 Purpose of the document	8
3. System Overview	9
3.1 Product Vision	9
3.2 Target Users and Stakeholders	9
3.3 Key Features and Benefits.....	10
4. Functional Requirements	11
4.1 User Roles and Permissions	11
4.2 Functional Modules.....	11
4.2.1 Learning and Knowledge Building	11
4.2.2 Collaboration and Interaction	12
4.2.3 Feedback and Evaluation	12
4.2.4 Skills Tracking and Reporting.....	12
4.3 Use Cases and Scenarios	12
4.4 User Stories	13
5. Technical Specifications	13
5.1 System Architecture Overview	13
5.2 Frontend Technologies	14
5.3 System Architecture and User-Centred Requirements	14
5.4 Database Schema.....	16
5.5 API Interfaces	16
5.6 Deployment Architecture	17
5.7 Security Measures.....	18
6. Technology Stack	19
6.1 Overview of Selected Technologies	19
6.2 Rationale for Technology Choices	20
6.3 Integration of Third-Party Tools	21
7. Integration Framework	21
7.1 System Interoperability.....	21



- 7.2 External Systems Integration** 22
- 7.3 Middleware and API Gateway** 22
- 7.4 Data Exchange Protocols** 22
- 8. UI/UX Design Principles** 23
 - 8.1 User Interface Guidelines** 23
 - 8.2 Accessibility Standards** 23
 - 8.3 Responsive Design Strategy** 23
 - 8.4 User Journey and Experience Flows** 24
- 9. Non-Functional Requirements** 24
 - 9.1 Performance Expectations** 24
 - 9.2 Scalability Requirements** 24
 - 9.3 Availability and Uptime** 25
 - 9.4 Security and Privacy** 25
 - 9.5 Maintainability and Support**..... 25
- 10. Compliance and Regulatory Requirements** 25
 - 10.1 Data Protection and GDPR** 25
 - 10.2 Accessibility Compliance** 26
 - 10.3 Industry Standards**..... 26
- 11. Testing Strategy** 27
 - 11.1 Test Types and Coverage**..... 27
 - 11.2 Test Automation Strategy** 27
 - 11.3 User Acceptance Testing (UAT)** 27
 - 11.4 Bug Tracking and Quality Assurance**..... 28
- 12. Deployment and Maintenance**..... 28
 - 12.1 CI/CD Pipeline**..... 28
 - 12.2 Monitoring and Logging**..... 28
 - 12.3 Maintenance Plan** 29
 - 12.4 Versioning and Change Management** 29
- 13. Appendix** 29
 - 13.1 Glossary**..... 29
- REFERENCES** 32



LIST OF ABBREVIATIONS

- ADL – Advanced Distributed Learning
- API – Application Programming Interface
- ARIA – Accessible Rich Internet Applications
- CI/CD – Continuous Integration / Continuous Deployment
- CoP – Community of Practice
- CSS – Cascading Style Sheets
- EQF – European Qualifications Framework
- GDPR – General Data Protection Regulation
- GUI – Graphical User Interface
- HTML – HyperText Markup Language
- HTTP/HTTPS – Hypertext Transfer Protocol / Secure
- ID – Identifier
- ISO – International Organization for Standardization
- ITIL – Information Technology Infrastructure Library
- JWT – JSON Web Token
- LMS – Learning Management System
- LOM – Learning Object Metadata
- LTI – Learning Tools Interoperability
- MDN – Mozilla Developer Network
- MOOC – Massive Open Online Course
- OER – Open Educational Resources
- OWASP – Open Worldwide Application Security Project
- RBAC – Role-Based Access Control
- RFC – Request for Comments
- SCORM – Sharable Content Object Reference Model
- SDK – Software Development Kit
- SEO – Search Engine Optimization



D4.1 – Functional and Technical Specification

SLA – Service Level Agreement

SLO – Service Level Objective

SRE – Site Reliability Engineering

SSO – Single Sign-On

UI/UX – User Interface / User Experience

UAT – User Acceptance Testing

URL – Uniform Resource Locator

VET – Vocational Education and Training

WCAG – Web Content Accessibility Guidelines

xAPI – Experience API (also known as Tin Can API)



EXECUTIVE SUMMARY

This deliverable (D4.1) presents the functional and technical specification framework for the ComeThinkAgain Competence Education and Training System (CETS), forming the foundation for its implementation in WP4. It builds upon the educational content developed in WP2 and the experimental insights from WP3, and aims to establish a scalable, secure, and learner-centered digital platform that supports micro-certification across diverse learning contexts.

The document details the selection process and evaluation criteria for the Learning Management System (LMS), ultimately confirming Moodle as the core platform based on its compatibility with the project's pedagogical and technical needs. It outlines critical functional requirements, such as role-based access control, multilingual and modular content delivery, certification workflows, and collaborative features. These are informed by a consolidated User Requirement Registry contributed to by all consortium partners.

From a technical perspective, the deliverable defines the system architecture, backend services, integration strategy, and compliance protocols. It emphasizes interoperability through SCORM and xAPI, GDPR-aligned data protection, and adherence to accessibility standards (WCAG 2.1 AA). Best practices from industry leaders like Coursera and edX, such as gamification features (e.g. badges, progress tracking) and analytics dashboards have been adopted to enhance learner engagement and instructional effectiveness.

This specification framework ensures that the CETS can be developed in alignment with both user needs and technical robustness, setting a clear foundation for the platform's next stages: course design (D4.2), assessment and certification (D4.3), and system deployment (D4.4–D4.5).



2. INTRODUCTION

2.1 *Project Overview*

Living in a rapidly changing world, there is a need to prepare future generations for dealing with many aspects concerning modern life (e.g., digitalisation or climate change) by equipping them with necessary skills to do so. With respect to this, interdisciplinarity and multifaceted competences are key in the 21st century also when addressing the challenges of tomorrow's professions. Thus, the ComeThinkAgain project aims to develop and implement a cross-sectoral, standardised training and certification system which builds on three competence pillars, interwoven with each other: computational thinking (CT), entrepreneurship education (EE), and green skills (GS). Given the fact that teachers and trainers are educating the future workforce, the main target groups are teachers trained at Higher Education level and Vocational Education Training (VET) trainers. The outcome of the project will be a Micro-Certification based Education Training System called "ComeThinkAgain CETS (CTA-CETS)", offering micro-modules for both vocational and higher education at a European level, ensuring a workforce which is ready for the future.

This deliverable is the result of Task 4.1 and builds upon the foundations established in the previous work packages. It provides a comprehensive evaluation and selection of the Learning Management System (LMS) that will host the online courses. The deliverable specifies essential network metrics and technical requirements, including accessibility, security, privacy, and inclusion. Additionally, it explores potential gamification elements, analytics dashboards, and incorporates insights from companies with extensive experience in online education. Furthermore, it outlines the technical infrastructure necessary for the secure issuance of digital credentials.

2.2 *Purpose of the document*

This document presents a comprehensive evaluation and selection process for the Learning Management System (LMS) to host the ComeThinkAgain online courses. It outlines the specification of essential network metrics and technical requirements, including accessibility, security, privacy, and inclusion, to ensure an optimal and secure learning environment. Furthermore, the document explores potential gamification elements and analytics dashboards aimed at enhancing learner engagement and providing actionable insights. Additionally, the document details the technical infrastructure necessary for the secure issuance of digital credentials within the ComeThinkAgain micro-certification framework.

The structure of D4.1 comprises the following chapters, briefly described below:

- Chapter 1 introduces the ComeThinkAgain project, the objectives of this deliverable, and the methodology used for LMS evaluation and selection.
- Chapter 2 outlines the functional requirements of the platform, including user roles, course management needs, certification processes, and integration features.
- Chapter 3 provides a system overview with the product vision, target users, and key functional benefits of the proposed ComeThinkAgain LMS.
- Chapter 4 describes the core functional modules and user scenarios, including learning, collaboration, evaluation, and progress tracking features.



- Chapter 5 details the technical specifications, such as backend architecture, frontend technologies, database schema, APIs, and deployment setup.
- Chapter 6 presents the selected technology stack, the rationale behind these choices, and the integration of third-party tools (e.g., H5P, Zoom).
- Chapter 7 defines the integration framework, including interoperability standards (SCORM, xAPI), external system integration, API gateways, and data protocols.
- Chapter 8 outlines UI/UX design principles, accessibility standards (WCAG 2.1 AA), and responsive design strategies tailored to learner needs.
- Chapter 9 describes the non-functional requirements including performance benchmarks, scalability, uptime expectations, and maintainability.
- Chapter 10 explains compliance and regulatory requirements, particularly around data protection (GDPR), accessibility, and adherence to industry standards.
- Chapter 11 presents the testing strategy, including test types, automation plans, user acceptance testing (UAT), and quality assurance workflows.
- Chapter 12 covers deployment and maintenance procedures, including the CI/CD pipeline, monitoring, versioning, and long-term system support.
- Chapter 13 provides appendices including a glossary of terms used throughout the document.

3. System Overview

3.1 Product Vision

ComeThinkAgain is envisioned as an integrated, learner-centric digital platform designed to enhance 21st-century cognitive and collaborative skills through structured educational experiences. Built upon the Moodle Learning Management System (LMS), the platform serves both as a robust course delivery environment and a skills development hub, empowering users to build competencies such as critical thinking, communication, self-regulation, and digital literacy. The system supports blended and hybrid learning models, integrating synchronous and asynchronous modalities.

Strategically, ComeThinkAgain seeks to:

- Support equitable access to learning and meaningful participation.
- Integrate with institutional learning ecosystems while offering modular flexibility.
- Scale across diverse learning contexts, from K12 to higher education and vocational training.

3.2 Target Users and Stakeholders

ComeThinkAgain is designed to serve a diverse ecosystem of users and stakeholders, each with distinct roles, expectations, and interaction patterns. At the core of the platform are the primary users, who directly engage with the learning environment on a regular basis. Learners are the central focus of the system; they are expected to participate in interactive tasks, engage in peer discussions, reflect through journals or prompts, and respond to feedback. For these users, the platform must provide an intuitive and motivating experience that supports various levels of digital literacy and accommodates individual learning styles.



Educators and instructors form the second layer of primary users. They are responsible for designing and delivering instructional content, facilitating collaborative activities, assessing student performance, and offering personalized feedback. The platform must support their pedagogical goals by providing flexible tools for course management, student tracking, and formative assessment. Administrators, on the other hand, are tasked with configuring the platform for institutional needs, managing user roles and permissions, monitoring platform-wide usage, and ensuring compliance with institutional policies and data regulations.

In addition to the primary users, ComeThinkAgain also supports a set of secondary stakeholders whose contributions are essential to the platform's long-term value and strategic alignment. Curriculum designers utilize the system to define learning objectives, map skill taxonomies, and align course content with institutional or national education standards. Institutional IT staff play a crucial role in supporting the deployment, integration, and ongoing technical maintenance of the platform, ensuring that it operates smoothly and securely within the broader IT infrastructure.

To accommodate this diverse stakeholder base, ComeThinkAgain employs a role-based architecture that tailors the user experience through customized dashboards, data views, and workflows. Each user type is provided with the tools and information most relevant to their functions, ensuring that all stakeholders can effectively engage with the system in ways that align with their responsibilities and goals.

3.3 Key Features and Benefits

ComeThinkAgain integrates both functional and pedagogical innovations to support a modern, skills-oriented learning experience. Among its key features are collaborative learning modules that enable group-based tasks, peer interaction boards, and real-time co-editing tools, fostering meaningful social learning. The platform also includes reflective journals and structured feedback systems, which are designed to promote metacognitive awareness and continuous improvement. To support transparency and motivation, skill progress trackers offer visualizations of both individual and group development over time, aligned with defined learning outcomes. ComeThinkAgain supports multi-modal content delivery, accommodating video, audio, text, and embedded interactive media to meet diverse learner needs. A role-based access system ensures that learners, instructors, and administrators each have tailored interfaces and toolsets appropriate to their responsibilities. Additionally, a comprehensive analytics dashboard provides real-time insights into engagement, participation, and skill mastery, empowering educators and institutional leaders with actionable data. The platform is built with interoperability in mind, using API-based integration to connect seamlessly with external learning management systems, authentication services, and third-party content repositories.

These features yield significant benefits. ComeThinkAgain promotes active learning and collaborative knowledge construction, creating dynamic educational environments that extend beyond passive content consumption. It enhances transparency in the learning process by making skill development visible and trackable for all users. Through its modular and customizable design, the platform adapts to a wide range of educational contexts and learner profiles. For educators, ComeThinkAgain improves efficiency by automating routine tasks such as feedback collection and performance monitoring. At an institutional level, the platform



enables stakeholders to track equity, inclusivity, and performance trends, supporting informed decision-making and continuous improvement in teaching and learning practices.

4. Functional Requirements

4.1 User Roles and Permissions

ComeThinkAgain is structured around a role-based access control system, ensuring that each user type interacts with the platform in ways that suit their functional needs and responsibilities. The primary user roles include Participants (learners), Trainers (educators or instructors). Participants, referred to as learners, have non-editorial rights and engage with pre-defined modules that match their educational level (primary, secondary, or vocational education and training). Their experience is designed to be intuitive, motivational, and adaptive, allowing access to only the modules they are registered for, such as Computational Thinking, Entrepreneurship, or Green Skills.

Trainers are granted editor-level permissions, allowing them to design and structure content, administer assessments, track learner progress, and issue certifications.

4.2 Functional Modules

The ComeThinkAgain platform is composed of several interconnected functional modules that deliver educational value, support collaboration, and ensure progress tracking and assessment. These modules are organized logically into courses, phases, and tasks, aligned with learning goals and often framed around ECTS credit structures.

4.2.1 Learning and Knowledge Building

The core of the platform lies in its robust learning delivery system, which must support various content types including videos, audio, interactive texts, simulations, quizzes, and downloadable documents. Learners should be able to personalize their learning path by selecting micro-modules and completing content in their preferred sequence, while also unlocking further lessons based on prerequisite completion. The system must allow for materials to be provided in English, German, Estonian, French and Spanish to support inclusivity and better comprehension. Accessibility standards (e.g., WCAG) are essential to ensure usability for all learners, including those with disabilities.

Trainers must be able to create and reuse course structures, schedule classes, and manage learning paths for different cohorts. Moodle has been selected as the primary Learning Management System (LMS) for the consortium therefore, ensuring compatibility with SCORM and xAPI is essential to fully leverage Moodle's capabilities and to maintain interoperability with other systems if needed.



4.2.2 Collaboration and Interaction

Collaborative learning is a core design principle of ComeThinkAgain. The platform supports group work, discussion forums, peer-to-peer activities, and live webinars, enabling both synchronous and asynchronous interaction. Trainers can facilitate classroom-like environments using integrated virtual tools such as BigBlueButton or Zoom, while learners benefit from structured opportunities for teamwork and reflection.

A dedicated Community of Practice (CoP) component allows learners and experts to exchange insights and experiences, fostering deeper knowledge construction, networking, and ongoing professional dialogue.

4.2.3 Feedback and Evaluation

A continuous feedback mechanism is required to support formative and summative assessment. Trainers should provide feedback through structured tools and automated assessments. Certificates may be issued to acknowledge participation or completion, depending on the course structure and objectives must be issued upon completion of defined milestones or full modules. Where possible, the certification process should align with the Europass digital credential framework and support micro-certifications that reflect granular learning achievements.

Surveys and questionnaires must be integrated into the system to allow trainers to gather feedback on content effectiveness, user satisfaction, and system usability. These instruments are vital for maintaining a cycle of quality assurance

4.2.4 Skills Tracking and Reporting

To promote metacognition and targeted growth, the platform must offer robust progress tracking tools. Learners should see visualizations of their learning journey, including completed modules, skill acquisition, and pending requirements. Trainers and administrators require reporting dashboards to monitor individual and cohort progress, engagement levels, and learning gaps.

Data analytics modules must support real-time and longitudinal analysis of learner behavior and performance, enabling institutions to assess learning outcomes, ensure equity, and make data-informed decisions.

4.3 Use Cases and Scenarios

ComeThinkAgain supports a range of use cases reflective of its diverse stakeholder base. For example, a vocational education participant might begin a course by watching an introductory video, engage in an interactive coding sandbox, collaborate on a group assignment, and complete a quiz to unlock the next module. Meanwhile, a trainer might design a new module using reusable content blocks, schedule synchronous webinars, and track participant engagement via analytics dashboards.

Another scenario could involve preparing a quarterly report using performance analytics and user feedback to optimize resource allocation and platform usage policies. These use cases reflect the system's flexibility and its alignment with modern digital pedagogy.



4.4 User Stories

The functional design of ComeThinkAgain is guided by a comprehensive set of user stories derived from stakeholder needs. Participants emphasize the importance of seamless navigation, multilingual access, and progress tracking. They also express the need for personalized learning journeys with optional content visibility based on their progression. Trainers highlight the necessity for reusable content structures, integrated assessment tools, and the ability to run surveys and certification workflows. Administrators and content designers focus on scalability, interoperability, and the modularity of content creation.

The platform addresses high-priority user stories such as providing collaborative tools, issuing digital certifications, ensuring system compatibility with international standards. Additionally, features like gamification, offline development, and hybrid navigation (linear with flexibility) further enrich the user experience. All these stories have been translated into functional specifications, many of which are already implemented or currently under development.

5. Technical Specifications

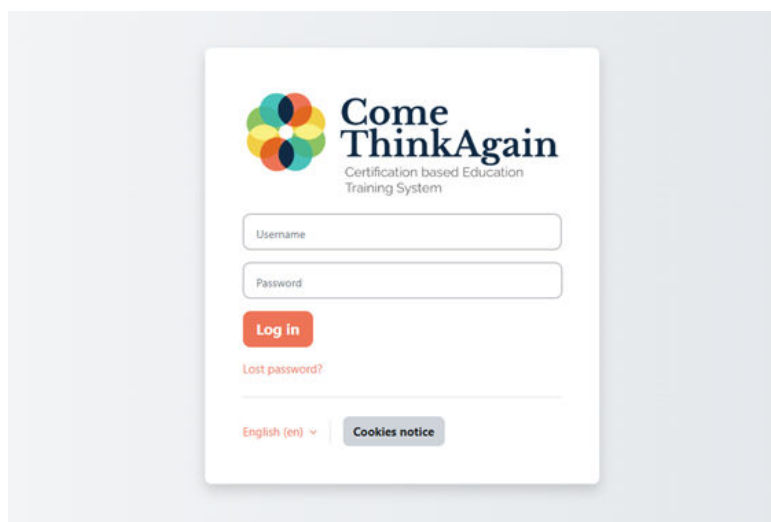
5.1 System Architecture Overview

The ComeThinkAgain platform will be built on a modular, service-oriented architecture (SOA) that ensures flexibility, scalability, and maintainability. The system will consist of a presentation layer (frontend), a logic layer (backend services), and a data layer (database and content repository). Integration layers will enable interoperability with external systems such as institutional LMSs and certification frameworks.

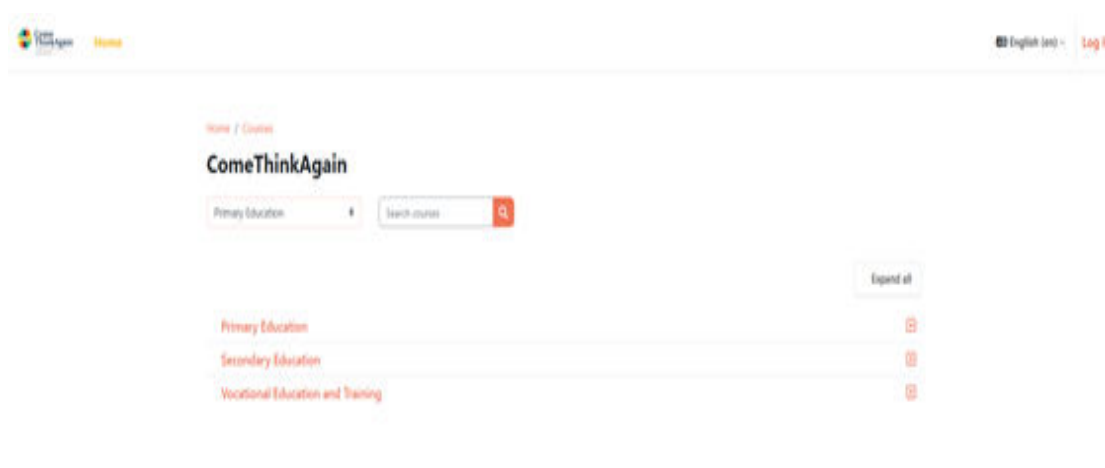
The architecture will support role-based access control (RBAC) to distinguish between Participants (learners) and Trainers (teachers with editorial privileges). Upon registration, users will self-identify their educational sector (Primary, Secondary, or VET), determining the visibility and access to modules like Computational Thinking, Entrepreneurship, and Green Skills.

The system will be cloud-ready and capable of horizontal scaling to accommodate simultaneous usage by multiple institutions and hundreds of concurrent users. Deployment will support both online and offline learning scenarios, including micro-module access, hybrid delivery, and integration with tools such as H5P and BigBlueButton for interactive engagement.

Moodle's login in page:



Options for educational levels:



5.2 Frontend Technologies

The frontend of the platform will be designed with a strong focus on usability, responsiveness, and accessibility, ensuring a seamless experience across desktops, tablets, and mobile devices. Learners will benefit from a personalized dashboard displaying their progress, earned certifications, and upcoming tasks. To enhance engagement, the platform will incorporate gamification elements like badges and progress bars. Navigation will follow a hybrid model, offering both linear module progression and flexible access based on prior completions. A multilingual interface will accommodate localized content delivery.

5.3 System Architecture and User-Centred Requirements

The core learning platform is being developed on Moodle, which was selected as the primary Learning Management System (LMS) by consortium consensus. This choice reflects both technical maturity and alignment with pedagogical flexibility, multilingual support, modularity, and strong community adoption.



This chapter summarizes how user requirements collected collaboratively from all project partners directly shape the backend architecture and platform functionalities. These requirements were categorized as Essential, Nice to Have, or Out of Scope, and cover multiple user types including learners, teachers, administrators, and content creators.

Requirement evaluation process

Each partner contributed to identifying user needs, which were captured in the User Requirement Registry. The structure included:

- Short description of the functionality
- User type (e.g., learner, teacher, admin)
- User tasks and expected results
- Priority level (Essential, Nice to Have)
- Associated technical components and risks

This structured evaluation helped us prioritize features that are pedagogically and technically feasible within the project timeline. For example:

- Essential Requirements: SCORM/xAPI compliance, role-based content access, multilingual support, micro-certification structures, performance tracking, collaborative learning tools.
- Nice to Have: Offline content authoring, collaboration via external tools, advanced analytics, content download for reuse.
- Out of Scope: Requirements lacking feasibility within current resources or not aligned with Moodle's extensibility model.

Moodle integration & interoperability

Moodle's modular design enables efficient integration of required functionalities. Key reasons for its selection include:

- Support for blended learning (online + in-person)
- Rich plugin ecosystem for collaboration (forums, group work)
- Built-in tools for progress tracking, quizzes, and feedback
- Multilingual content support
- Structured support for learning paths and conditional access

To ensure interoperability and future-proofing, the platform will comply with SCORM and xAPI (Tin Can API) standards. This enables:

- Reuse of content across systems (e.g. other Moodle instances)
- Seamless import/export of learning materials and user progress
- Analytics integration and external reporting

Certification and modularity

Certificates will be generated based on module completion and configured according to learning paths. Although certificate types are still being finalized, the system will support flexible configurations e.g. combining micro-modules into a full certification. Alignment with Europass standards is under consideration for compatibility with European frameworks.



5.4 Database Schema

The platform's database will be a relational system preferably PostgreSQL or MySQL designed to manage and scale educational data efficiently. It will be structured around core entities, including Users, which store account details, roles (Trainer/Participant), and authentication information, in line with standard LMS user models. Modules will represent structured learning units enriched with metadata such as titles, descriptions, and links to specific educational levels, following metadata standards like IEEE LOM (LTSC, 2002)

Each module will contain Courses and Tasks, which will be associated with learning objectives, media resources (e.g., videos, PDFs, H5P content), and quizzes (Dougiamas, 2003). A dedicated Progress Tracking table will store participant-specific data such as completion rates, quiz outcomes, and time logs, supporting personalized learning analytics (Romero, 2007). The schema will also include a Certificates table to manage the issuance of digital micro-credentials and full-module certifications, which may align with frameworks like Europass and MicroHE metadata (Aljawarneh, 2020).

To support peer interaction and knowledge sharing, Forum Threads will be linked to modules, allowing discussion and community exchange in a structure familiar from open-source LMS systems like Moodle (Dougiamas, 2003). Moreover, the design will include support for offline content development and import/export capabilities, implemented through staging tables and versioning mechanisms, ensuring interoperability and reusability (LTSC, 2002).

5.5 API Interfaces

The platform will utilize RESTful API architecture to ensure scalable, modular, and interoperable integration with both internal components and external systems. REST (Representational State Transfer) is chosen for its stateless nature, scalability, and widespread support across modern web technologies. These APIs will serve as the backbone for both user-facing and administrative functionalities.

To enable seamless integration with institutional Learning Management Systems (LMS) such as Moodle, Canvas, or Blackboard, the platform will support educational interoperability standards including SCORM (Sharable Content Object Reference Model), LTI (Learning Tools Interoperability), and xAPI (Experience API). These standards ensure that course content, learner activity data, and progress tracking can be easily shared, imported, and reused across partner systems. This aligns with the platform's modular and portable content goals, allowing content creators and institutions to maintain compatibility and consistency across learning environments.

Additionally, the system will offer API endpoints for connecting with external video conferencing platforms such as Zoom, Microsoft Teams, and BigBlueButton. These integrations will support real-time learning activities like webinars, virtual classrooms, and synchronous collaboration sessions functions that are essential for a hybrid or blended learning format. API-based integration will allow events such as attendance tracking, session recordings, and scheduling to be managed from within the platform.

Secure user authentication is another critical feature of the platform's API architecture. The platform will implement OAuth2 protocols for federated identity management, enabling users to authenticate using familiar accounts such as Google, Apple, or Microsoft. This approach simplifies onboarding, enhances user



experience, and reduces the overhead associated with account management, while also supporting single sign-on (SSO) use cases in institutional environments.

A unique API offering of the platform will be the generation and export of digital certifications. These will conform to Europass Digital Credentials standards and, where possible, include MicroHE-compliant metadata, enabling verifiable and machine-readable credentials that can be stored, shared, and recognized across European educational and employment platforms.

In addition to public-facing APIs, the system will expose internal APIs for handling real-time interactions and automation. These include support for dashboard updates, gamification mechanics (like leaderboard updates and badge awards), and feedback distribution systems. For example, when a user completes a quiz or task, an internal API call will immediately reflect the change in progress bars, update gamified elements, and trigger personalized feedback messages.

All API interactions will be standardized around JSON for lightweight, human-readable data exchange. To ensure secure communication, APIs will be protected using token-based authentication mechanisms, such as JWT (JSON Web Tokens), and adhere to HTTPS protocols to prevent data interception and tampering.

This API-driven architecture not only promotes interoperability and automation but also enhances the adaptability of the system for future expansions, third-party integrations, and customized institutional deployments.

5.6 Deployment Architecture

The deployment architecture of the ComeThinkAgain platform will be built upon a containerized microservices model, designed for high scalability, modularity, and resilience. Each major component of the system such as user management, content delivery, progress tracking, and communication will run as an isolated microservice packaged using Docker containers. This approach facilitates platform independence, ease of deployment, and simplifies the development and maintenance lifecycle.

For orchestration and service coordination, Kubernetes is the preferred platform, though Docker Swarm may be considered for simpler environments. Kubernetes provides robust tools for automatic scaling, health checks, service discovery, and rolling updates, ensuring continuous availability and performance even during version upgrades. Its declarative configuration and self-healing capabilities will help maintain a reliable and fault-tolerant learning platform, especially as user demand grows.

To accelerate development and improve delivery efficiency, a Continuous Integration and Continuous Deployment (CI/CD) pipeline will be implemented using platforms such as GitHub Actions or GitLab CI. This pipeline will automate testing, integration, and deployment of new code, ensuring that updates reach users faster with reduced manual effort and fewer errors. Automated testing stages will also contribute to system robustness and reduce the likelihood of regressions.

Cloud hosting will be employed to host the platform, with deployment options tailored to institutions' preferences (e.g., AWS, Azure, or Google Cloud Platform). These cloud services offer global scalability, high availability zones, and built-in compliance features necessary for hosting sensitive educational data. In addition, Content Delivery Networks (CDNs) will be leveraged to serve rich media content—including videos, interactive assets, and documents—closer to users' geographic locations, thereby reducing latency and ensuring fast content loading times.



A key architectural consideration is auto-scaling. The platform will be designed to dynamically allocate computing resources based on real-time user demand. This is especially important to handle X concurrent users (as specified in the use case data) during peak usage, such as live webinar sessions or assignment deadlines. Combined with load balancing strategies, auto-scaling ensures that no single server becomes a bottleneck, maintaining a seamless user experience.

Performance optimization will also be addressed through caching mechanisms such as Redis for session data and Varnish for content caching. These technologies help reduce database load and speed up content delivery, especially for frequently accessed resources.

Moreover, to support offline content development and transfer, the platform will provide local development kits for teachers and content creators. These toolkits will allow educators to create and edit content offline and later export/import courses in SCORM format, maintaining compatibility with the platform and other institutional LMS systems. This feature is especially useful for educators in regions with limited or inconsistent internet access, aligning with the platform's goal of accessibility and inclusivity.

Overall, this deployment strategy ensures that ComeThinkAgain is scalable, reliable, maintainable, and user-centered, capable of supporting a wide range of learners and educational institutions across Europe and beyond.

5.7 Security Measures

The ComeThinkAgain platform will be designed with security and data protection as foundational priorities, especially given the sensitive nature of user data and the inclusion of minors from primary and secondary educational institutions. All security measures will align with General Data Protection Regulation (GDPR) requirements and educational data privacy best practices, ensuring responsible handling of user information across all system layers.

A robust Role-Based Access Control (RBAC) model will form the backbone of the platform's internal security architecture. This model will ensure that users—whether administrators, trainers (teachers), or participants (students)—only access the specific resources and functionalities pertinent to their roles. For example, trainers will have editorial privileges within assigned modules, while participants will have restricted access limited to the modules they are enrolled in. This segmentation prevents unauthorized access to sensitive data or teaching content and reduces the attack surface from within the system.

Authentication will be managed using industry-standard protocols such as OAuth2 and OpenID Connect, supporting Single Sign-On (SSO) via trusted identity providers like Google, Microsoft, or institutional logins. This enhances both security and usability, minimizing the risk of password fatigue and enabling secure federated identity management. It also facilitates integration with external services like learning management systems (LMS) and video conferencing platforms while maintaining a strong identity trust chain.

All data exchanged between users and the platform will be encrypted in transit using Transport Layer Security (TLS) to prevent interception or tampering during communication. Additionally, sensitive information stored on the server will be encrypted at rest using algorithms such as AES-256, ensuring that data remains protected even in the unlikely event of a storage breach.

The system will undergo routine vulnerability scanning using automated tools (e.g., OWASP ZAP, Nessus) and regular penetration testing conducted by certified



security professionals. This proactive security posture will help identify and mitigate known vulnerabilities before they can be exploited. Updates to system components will be tightly controlled, and any third-party libraries or dependencies will be continuously monitored for Common Vulnerabilities and Exposures (CVEs).

To ensure operational transparency and enable forensic analysis in the event of an incident, the platform will implement comprehensive logging and activity monitoring. Anomaly detection systems will flag suspicious behavior patterns such as multiple failed login attempts, unusual access patterns, or privilege escalation attempts. These mechanisms will support incident response and help prevent potential misuse, especially in collaborative environments such as discussion boards and peer feedback modules.

Special attention will be paid to data privacy for minors, a particularly sensitive aspect of the platform given its use in primary and secondary education. Where applicable, parental or guardian consent mechanisms will be built into the registration process, along with opt-in protocols for participation in social or public-facing platform features. Data minimization principles will be applied—collecting only what is strictly necessary for learning purposes—and users will have the ability to control their personal information and visibility settings.

In summary, the security framework for ComeThinkAgain will be multi-layered, proactive, and aligned with best practices for educational platforms, ensuring that users can engage with the content and community safely and confidently.

6. Technology Stack

The ComeThinkAgain platform will be built upon a modern, modular, and scalable technology stack designed to support blended learning, high levels of user interaction, data interoperability, and accessibility across a wide user base, including both academic and vocational training environments.

6.1 Overview of Selected Technologies

The ComeThinkAgain platform will implement a decoupled system architecture, separating the frontend and backend to enhance scalability, maintainability, and flexibility. This design allows for independent updates of interface and business logic components, supporting faster deployment cycles and improved user experiences.

On the frontend, the platform will use either React.js or Vue.js, both of which are widely adopted JavaScript frameworks for building fast, component-driven interfaces. These frameworks are known for their strong community support and robust ecosystems, making them suitable for dynamic and interactive learning platforms (MDN Web Docs, 2024). To ensure a responsive design that adapts seamlessly across devices, Tailwind CSS or Bootstrap will be used. For multilingual support, libraries like i18next will enable localization and dynamic content translation, vital for a diverse European user base.

The backend will be developed using either Node.js with Express.js or Python with the Django REST Framework. Node.js offers high performance for I/O-heavy applications, while Django provides a mature, secure, and scalable environment ideal for educational platforms (Django Project, 2024). Authentication will be



managed using OAuth2 or JWT, allowing for secure session handling and integration with external identity providers like Google or Microsoft.

The database layer will rely on PostgreSQL (preferred) or MySQL, both being reliable relational database systems. PostgreSQL is known for advanced features such as JSON support, indexing, and extensibility, making it well-suited for storing structured data related to users, modules, tasks, and progress tracking (PostgreSQL Global Development Group, 2024).

For deployment, the platform will use Docker for containerization and Kubernetes for orchestration, ensuring high availability and scalability. CI/CD pipelines will be managed via GitHub Actions or GitLab CI, enabling continuous testing and delivery workflows. Hosting will be handled through major cloud providers such as AWS, Azure, or Google Cloud, with Cloudflare CDN employed for rapid content delivery.

To optimize performance, Redis will be used for in-memory caching and Varnish for HTTP caching, especially for high-traffic learning modules. Real-time collaboration and learning will be supported by integrating BigBlueButton, Zoom, or MS Teams, while H5P will be used for creating interactive educational content. The platform will also ensure interoperability with institutional learning management systems by supporting SCORM and xAPI standards (ADL Initiative, 2023).

6.2 Rationale for Technology Choices

The selection of technologies for the ComeThinkAgain platform is driven by the need to support critical functions such as blended learning, personalized learning paths, collaborative participation, and certification workflows. The technology stack has been carefully curated to ensure that each layer of the system, from the interface to the backend and deployment infrastructure, contributes to these goals with scalability, accessibility, and interoperability in mind.

For scalability and flexibility, backend technologies such as Node.js with Express.js or Django REST Framework were chosen due to their ability to handle RESTful APIs efficiently and support horizontal scaling. These frameworks are ideal for high-concurrency environments, particularly during peak activity such as live webinars, submission deadlines, or when issuing mass certifications. Their mature ecosystems and modular design enable rapid development while maintaining system performance.

The user experience (UX) is central to platform adoption and engagement. Frameworks like React.js and Vue.js enable the creation of dynamic, component-based user interfaces, which are critical for rendering dashboards, interactive progress tracking, and gamification elements (e.g., badges and leaderboards). For responsive and accessible design across devices, Tailwind CSS was selected for its utility-first approach, making it easier to implement mobile-first, consistent layouts.

From a security and authentication perspective, the use of OAuth2 and JSON Web Tokens (JWT) ensures secure, standards-based access management. These protocols allow for Single Sign-On (SSO) integration with major identity providers (e.g., Google, Microsoft), which enhances both security and user convenience, particularly in institutional contexts where centralized login is the norm.

To support data interoperability, particularly with external Learning Management Systems (LMS), the platform will implement SCORM and xAPI standards. This enables the export and import of course content while maintaining learning



analytics and outcomes, ensuring the platform is LMS-agnostic and compatible with institutional requirements (e.g., Moodle, Canvas).

The commitment to accessibility and localization is reflected in the use of WCAG 2.1 compliant frontend frameworks and multilingual libraries like i18next. These choices ensure the platform is inclusive of users with disabilities and adaptable to regional language needs, which is especially important for reaching VET and higher education audiences across Europe.

Lastly, the focus on maintainability and DevOps excellence is addressed through the use of Docker for containerization, ensuring consistent deployment environments across development, staging, and production. CI/CD pipelines (e.g., GitHub Actions, GitLab CI) support automated testing and deployment, while Kubernetes orchestration enables elastic scaling, self-healing services, and high availability critical for a reliable and always-on learning environment.

6.3 Integration of Third-Party Tools

To meet the extended functional requirements, particularly those related to collaboration, feedback, and real-time instruction, the platform will integrate a variety of third-party tools, including both open-source and commercial options.

For learning and collaboration, the platform will embed H5P directly into the LMS to enable interactive assessments. Synchronous learning and virtual classrooms will be supported through tools such as BigBlueButton, Zoom, or Microsoft Teams. To facilitate engagement within Communities of Practice (CoP), Discourse or integrated forum solutions will be implemented.

Regarding authentication and identity management, the platform will support OAuth2 integration with major providers like Google, Apple, Microsoft, as well as institutional identity providers. Additionally, it will offer optional anonymous access for open MOOC-style content, allowing users to participate without certification or tracking if they prefer.

For content standards and certification, the system will support SCORM and xAPI modules to enable the import and export of structured learning content. Certification issuance will follow standardized metadata frameworks using Europass Digital Credentials and MicroHE.

In terms of analytics and feedback, tools such as Matomo or Google Analytics (subject to GDPR-compliant consent) will be employed to track learning behavior. Furthermore, custom-built feedback forms and survey modules will be incorporated to capture user satisfaction and support ongoing quality assurance.

These integrations have been carefully selected not only for their technical compatibility but also for how well they align with pedagogical goals—fostering social learning, providing targeted feedback, and supporting lifelong learning through micro-credentials.

7. Integration Framework

7.1 System Interoperability

ComeThinkAgain embraces educational interoperability standards to ensure seamless interaction with a wide range of institutional systems and learning



platforms. By adopting SCORM, xAPI, and LTI, the platform enables the import and export of course content, learner progress tracking, and grade synchronization across different Learning Management Systems (LMS) (Advanced Distributed Learning, 2023; IMS Global, 2023). SCORM facilitates modular packaging of learning assets, while xAPI supports detailed activity logging, allowing institutions to merge learning data from multiple sources. LTI integration enables the embedding of the ComeThinkAgain interface within host LMS platforms, maintaining a unified user experience. Furthermore, compliance with the Europass Digital Credentials Infrastructure ensures that digital certificates produced by ComeThinkAgain are verifiable across EU systems, aligning with the European Qualifications Framework (EQF) and enabling learners to present these credentials seamlessly to employers or educational institutions (European Commission, 2021; Cedefop, 2020). This structured approach not only enhances portability and data reuse but also positions the platform as part of a cohesive ecosystem of interoperable educational tools within Europe.

7.2 External Systems Integration

To support blended and community-based learning, ComeThinkAgain integrates with a mix of synchronous and asynchronous collaboration tools. BigBlueButton, optimized for educational settings, offers core features like real-time whiteboarding and breakout rooms, whereas Zoom and Microsoft Teams provide additional scalability and security options (BigBlueButton, 2023; Al-Fraihat et al., 2020). For threaded, community-driven discussions, the platform embeds Discourse, which is widely recognized for supporting structured, persistent conversation aligned with Wenger’s concept of communities of practice (Wenger, 1998). Institutional authentication needs are met through SSO via OAuth2/OpenID Connect, enabling secure federated sign-in for users from federated identity systems such as eduGAIN (Hardt, 2012; GÉANT eduGAIN, 2023). Integrating these external applications requires robust backend connectors and authentication flows, while preserving the user’s learning context (e.g., linking a webinar to a specific module, capturing attendance data, merging forum contributions with user profiles).

7.3 Middleware and API Gateway

The ComeThinkAgain platform will implement an API gateway as an interface between internal services and external consumers (including LMS systems, third-party apps, and administrative tools). This gateway, built on frameworks like Kong or AWS API Gateway, enforces authentication, rates limits, and logging on all external RESTful endpoints. Internally, a middleware orchestrator funnels user-facing requests to services such as user management, content delivery, assessment engines, and analytics modules. The gateway also serves as a negotiation point for future integration with external learning analytics tools or artificial intelligence models for personalized learning. Using token-based security (OAuth2/JWT), along with middleware policies, ensures consistent governance, enhanced scalability, and resilience, while keeping services modular and replaceable.

7.4 Data Exchange Protocols

Safe, efficient, and compliant data transfer is key to ComeThinkAgain. All communication between clients, servers, and third-party systems will occur over



HTTPS/TLS, ensuring end-to-end encryption and data integrity. APIs will utilize JSON and XML serialization based on context and standards, with JSON being used for most JavaScript-driven frontends. Internally, the platform can leverage gRPC or message queues (e.g., RabbitMQ) for reliable asynchronous communication, such as logging analytics events or triggering background jobs. The adoption of xAPI ensures rich, interoperable learner data collection, while SCORM remains in use for LMS content packaging (ADL, 2023). For credential exchange, the platform will generate EDCI-compliant XML, enabling digital certificates to be shared across systems in line with GDPR and European digital education objectives (European Commission, 2021). This protocol-driven infrastructure ensures the platform is flexible, secure, and ready for national or pan-European educational integration.

8. UI/UX Design Principles

8.1 User Interface Guidelines

The user interface (UI) of ComeThinkAgain will prioritize simplicity, clarity, and modularity, following established Human-Computer Interaction (HCI) principles. Interfaces will be designed using a component-based approach via frameworks like React.js or Vue.js, supporting reusable UI elements such as progress dashboards, module cards, and quiz components.

UI components will follow Material Design or Fluent UI standards, ensuring consistency in visual hierarchy, iconography, and micro-interactions (Nielsen, 1995). Key user personas trainers (teachers), participants (students), and administrators will each have dedicated dashboards and tailored navigation paths to minimize cognitive load and maximize task efficiency (Krug, 2014).

8.2 Accessibility Standards

Accessibility is fundamental, particularly given the inclusion of participants from primary and secondary education levels, including minors and potentially neurodivergent learners. The platform will conform to WCAG 2.1 AA standards to ensure all learning materials are perceivable, operable, understandable, and robust for users with disabilities (W3C, 2018).

Specific accommodations will include keyboard navigability, alternative text for media, screen reader support (ARIA roles), and high-contrast visual modes. All interactive components (e.g., quizzes, drag-and-drop H5P content) will be tested for accessibility using tools like Axe DevTools and WAVE. Moreover, accessibility will be continuously evaluated through user feedback and usability testing with representative learners.

8.3 Responsive Design Strategy

The ComeThinkAgain platform will use a mobile-first responsive design strategy, ensuring full functionality and user satisfaction across devices, desktops, tablets, and smartphones. This is especially critical for asynchronous learning activities and lesson access in low-resource contexts where mobile may be the primary device.



UI layouts will adapt fluidly using CSS frameworks like Tailwind CSS or Bootstrap 5, and breakpoints will be defined for key screen sizes. Performance optimization through lazy loading, content compression, and adaptive image serving will further support a consistent experience even in limited bandwidth environments.

8.4 User Journey and Experience Flows

User journeys will be mapped for key user roles (trainer, participant, admin) and tasks such as enrolling in a module, completing a lesson, or issuing a certificate. These flows will be optimized for minimum friction and maximum feedback.

The learner flow will start with a dashboard overview, moving through suggested modules based on their selected education type (Primary, Secondary, VET). Progress indicators, gamification elements (badges, level-ups), and access to forums will support learner motivation and retention (Deterding et al., 2011).

Educator flows will focus on lesson authoring, scheduling, assessment review, and learner analytics. Key touchpoints will offer contextual help and guided walkthroughs to reduce onboarding time. Experience mapping and user testing (moderated and unmoderated) will be conducted using tools like Figma prototypes and Hotjar analytics.

9. Non-Functional Requirements

9.1 Performance Expectations

The platform is expected to deliver high responsiveness to ensure a seamless user experience across all educational levels. User actions such as navigating modules, submitting quizzes, or launching video sessions should complete in under one second for standard interactions, while resource-intensive tasks like rendering complex media or launching H5P-based content should load within two to three seconds under normal network conditions. Backend performance will be supported through the use of asynchronous processing queues, background task runners, and caching strategies to minimize delays in content delivery and system feedback. The system will be benchmarked regularly to monitor latency, throughput, and server response under varying load conditions, including concurrent webinar sessions and assignment submissions.

9.2 Scalability Requirements

Scalability is critical due to the expected variability in platform usage, particularly during scheduled training periods, webinars, and certification deadlines. The system will be built to scale horizontally, using microservices architecture that allows individual services such as user authentication, course delivery, and assessment tracking to be replicated independently based on demand. Load balancers will manage traffic distribution, ensuring that no single node becomes a bottleneck. Auto-scaling mechanisms will allow the infrastructure to dynamically respond to changes in user load, scaling up during peak hours and scaling down during periods of low activity, thereby optimizing cost-efficiency and performance. The system will be prepared to support thousands of concurrent users without degradation of performance, based on stress testing results.



9.3 Availability and Uptime

High availability is fundamental to supporting uninterrupted learning experiences. The system will be designed to maintain 99.9% uptime annually, with infrastructure redundancy built across multiple geographic regions or availability zones. Components will be deployed in a fault-tolerant configuration, including duplicate instances of application servers, replicated databases, and failover-enabled content delivery services. Monitoring systems will be in place to conduct continuous health checks and automatically restart failing components. Additionally, scheduled maintenance will be managed through deployment pipelines that allow updates without downtime, using rolling updates or blue-green deployments to maintain service continuity.

9.4 Security and Privacy

Security and data privacy are paramount, especially given the system's use in educational environments that may include minors. The platform will enforce role-based access control (RBAC) to ensure users can only access content appropriate to their roles (e.g., trainers vs. participants). All user data will be encrypted both in transit and at rest, and secure protocols such as HTTPS will be enforced across all endpoints. The authentication system will support federated identity providers using secure standards like OAuth2, allowing for single sign-on (SSO) while preventing unauthorized access. Audit logs and activity tracking will be implemented to support both operational monitoring and regulatory compliance. The platform will include features such as consent management, anonymized data handling, and opt-in mechanisms to ensure alignment with European data protection laws.

9.5 Maintainability and Support

The architecture will be designed for maintainability through the use of loosely coupled modules, clearly defined APIs, and standard development conventions. The use of containerized services will make it easier to test, deploy, and rollback updates with minimal risk. A continuous integration/continuous deployment (CI/CD) pipeline will automate testing and deployment processes, reducing the risk of regressions and ensuring faster delivery of improvements. Documentation for developers and administrators will be maintained in sync with feature updates, and automated testing suites will ensure system stability across iterations. For end users, a tiered support system will be offered including a self-service knowledge base, user guides, and issue reporting interfaces. Maintenance windows will be communicated in advance, and emergency updates will follow a predefined escalation and notification protocol.

10. Compliance and Regulatory Requirements

10.1 Data Protection and GDPR

Given that the platform will operate within the European Union (European Commission, 2016) and serve users from multiple educational levels, including



minors, compliance with the General Data Protection Regulation (GDPR) is non-negotiable. The system will implement data minimization principles by collecting only the information necessary for learning, progress tracking, and certification. Consent mechanisms will be integrated at account creation and before any data processing that extends beyond essential educational use.

Users will be granted full control over their personal data, including access, rectification, and deletion requests. For participants under legal age, optional parental consent workflows will be incorporated where appropriate. Data will be encrypted both in transit and at rest, and storage will be limited to EU-based servers or GDPR-compliant providers. Additionally, audit trails will be maintained for all data handling operations to ensure transparency and accountability. The platform will also implement automated data retention policies to remove or anonymize inactive accounts after a defined period.

10.2 Accessibility Compliance

To ensure equitable access for all users, including those with disabilities, the platform will comply with the Web Content Accessibility Guidelines (WCAG) 2.1 at the AA level. This includes the use of semantic HTML, proper ARIA attributes, keyboard navigation support, and screen reader compatibility. Color contrast, font sizes, and interactive elements will be designed for legibility and ease of use across a wide range of visual or motor impairments.

Multimedia content, such as videos and audio materials, will include alternative formats like subtitles, transcripts, and audio descriptions. Interactive elements like quizzes and drag-and-drop interfaces will be designed with fallback methods for users relying on assistive technologies. Testing for accessibility will be incorporated into the quality assurance process using both automated tools and manual evaluations by usability experts. Regular accessibility audits will be conducted, and user feedback will be used to improve compliance over time.

10.3 Industry Standards

The platform will align with widely accepted industry standards to ensure interoperability, data portability, and long-term maintainability. For e-learning interoperability, support for SCORM and xAPI standards will enable seamless integration with institutional Learning Management Systems (LMS) used by project partners. Additionally, learning outcomes and credentials will be structured according to frameworks such as the European Qualifications Framework (EQF) and Europass Digital Credentials, ensuring consistency across educational contexts.

Secure authentication will follow industry practices, including OAuth2 and SAML for federated identity and single sign-on (SSO). The platform's infrastructure and development practices will adhere to DevSecOps principles, including version control, automated testing, and container-based deployment using standardized formats like Docker and Kubernetes. Logging, monitoring, and data backup policies will follow ITIL-based service management guidelines, ensuring reliability and recoverability in case of incidents.



11. Testing Strategy

11.1 Test Types and Coverage

A multi-layered testing approach will be adopted to ensure platform robustness, scalability, and user satisfaction. The strategy includes:

- **Unit Testing:** Focused on individual functions, services, and UI components (e.g., React/Vue units, backend endpoints). This ensures that the smallest units of logic behave as expected.
- **Integration Testing:** Validates the communication between modules, such as the interaction between the frontend and RESTful APIs, or the LMS and SCORM/xAPI content engines.
- **End-to-End (E2E) Testing:** Simulates real-world user scenarios (e.g., course enrollment, certificate download, quiz submission) using tools like Cypress or Selenium.
- **Performance & Load Testing:** Assesses system behavior under stress to ensure response time, throughput, and concurrent user capacity targets are met. Tools such as JMeter or Gatling will be used for simulation.
- **Security Testing:** Includes vulnerability scanning (e.g., OWASP ZAP) and penetration testing to detect common exploits such as XSS, SQL injection, or authentication flaws (OWASP, 2023).

Coverage goals will target a minimum of 80% code coverage for unit and integration tests and 100% coverage for critical user flows, especially those involving authentication, progress tracking, and certification.

11.2 Test Automation Strategy

Automation will play a central role in continuous quality assurance. Automated test scripts will be developed and integrated into the CI/CD pipelines using platforms such as **GitHub Actions**, **GitLab CI**, or **Jenkins**. Key automation practices include:

- Automated **unit and integration tests** triggered on every code commit.
- Nightly **regression tests** to detect breakages from recent changes.
- Cross-browser and responsive layout tests for major viewports (desktop, tablet, mobile).
- API contract testing using tools like Postman or Pact.

This ensures rapid feedback cycles and minimizes human error during frequent releases, particularly important in agile development environments (Fowler, 2021).

11.3 User Acceptance Testing (UAT)

User Acceptance Testing will be conducted in collaboration with actual users trainers and participants from the target groups (Primary, Secondary, and VET sectors). A formal **UAT plan** will define test cases, acceptance criteria, and roles:

- **Pre-release demo environments** will be deployed for user testing.



- Feedback will be gathered via surveys, screen recordings, and feedback forms.
- Observed issues will be documented and fed into QA sprints.

The UAT process will follow best practices outlined in ISO/IEC/IEEE 29119 software testing standards, emphasizing usability, learning flow coherence, and satisfaction metrics.

11.4 Bug Tracking and Quality Assurance

All testing activities will be tightly integrated into a centralized **bug and issue tracking system**, such as **Jira**, **Redmine**, or **GitHub Issues**. Key features include:

- Issue categorization (critical, major, minor, enhancement)
- Reproducibility logs, test screenshots/videos, and stack traces
- Linked pull requests and resolution tracking

Each sprint will include a **QA review phase** and mandatory sign-off before feature promotion. Bug-fix verification will be validated through regression automation and manual retesting.

To maintain ongoing quality, the QA team will perform **regular audits** using **SonarQube** or similar tools for static code analysis, focusing on code smells, security vulnerabilities, and test coverage anomalies.

12. Deployment and Maintenance

12.1 CI/CD Pipeline

The platform will utilize a robust Continuous Integration/Continuous Deployment (CI/CD) pipeline to ensure rapid, safe, and consistent delivery of new features and updates. The CI process will automatically trigger builds and run tests on every code commit, ensuring code quality and early error detection. The CD process will deploy tested code to various environments development, staging, and production based on predefined workflows. Key components of the pipeline will include static code analysis, unit and integration testing, containerization with Docker, and environment-specific deployments managed by tools such as GitHub Actions, GitLab CI, or Jenkins. This setup promotes automation, reduces manual errors, and accelerates release cycles.

12.2 Monitoring and Logging

Real-time monitoring and centralized logging will be essential for maintaining system reliability and ensuring rapid incident resolution. The platform will implement monitoring tools to track performance metrics like server response time, error rates, and resource utilization. Dashboards will provide visual insights into system health, while alerts will notify the support team of abnormal behavior or downtime. Logging mechanisms will capture detailed application events, user activities, and system errors. These logs will be indexed and searchable, enabling



efficient root cause analysis and system audits. Monitoring and logging together will support both operational continuity and compliance needs.

12.3 Maintenance Plan

A well-defined maintenance plan will be adopted to ensure system stability, minimize disruptions, and maintain security. Scheduled maintenance windows will be designated for applying updates, performing database optimizations, and executing system diagnostics. Regular automated backups of data and configurations will be taken, with secure storage and recovery procedures in place. Dependencies and third-party libraries will be routinely audited for vulnerabilities. Maintenance efforts will also include routine testing of critical paths and regression checks after each update. A support framework will be established with clear procedures for incident response, troubleshooting, and user communication.

12.4 Versioning and Change Management

To ensure clarity and control over system evolution, a semantic versioning approach will be adopted, categorizing updates as major, minor, or patch releases. This method allows users and developers to anticipate the nature and impact of each release. All code changes will be tracked using a version control system such as Git, with structured workflows for reviewing and merging changes. Changelogs will be maintained and published with each release, highlighting new features, resolved issues, and known limitations. Change management will also involve staging deployments and user acceptance testing (UAT) before final production rollout. Internal documentation and team communication will accompany each update to ensure operational alignment.

13. Appendix

13.1 Glossary

Accessibility (WCAG)

Guidelines and technical standards ensuring that digital content is usable by people with disabilities, based on the Web Content Accessibility Guidelines 2.1 AA.

API (Application Programming Interface)



A set of rules and protocols that allow different software systems to communicate with each other. RESTful APIs are used in the platform for integration and data exchange.

Authentication

The process of verifying a user's identity, often using OAuth2 or Single Sign-On (SSO) technologies.

Authorization (RBAC)

Role-Based Access Control system used to assign specific permissions to different user roles, such as Participants, Trainers, and Administrators.

Blended Learning

A learning model that combines online digital media with traditional face-to-face teaching.

CI/CD (Continuous Integration / Continuous Deployment)

Software development practices that enable frequent code changes, automatic testing, and streamlined deployment into production environments.

CoP (Community of Practice)

A collaborative learning environment where users and experts share knowledge, feedback, and professional insights.

COTS (Commercial Off-The-Shelf Software)

Software products that are readily available for general use and not custom-built for a specific project.

Docker

A platform used to develop, ship, and run applications inside containers to ensure consistency across environments.

Gamification

The use of game design elements—such as badges, points, and leaderboards in non-game contexts to enhance learner motivation and engagement.

H5P

An open-source content collaboration framework used to create interactive HTML5 content like quizzes, presentations, and games.



Interoperability

The ability of different systems and applications to exchange data and use the information that has been exchanged. Often facilitated through SCORM, xAPI, and LTI standards.

LMS (Learning Management System)

A digital platform for delivering, tracking, and managing educational courses and training programs.

Micro-Certification

A modular credential awarded for the successful completion of short, focused units of learning that demonstrate specific competencies.

Middleware

Software that connects different components or systems, often handling communication between the frontend, backend, and APIs.

Modular Learning

An instructional design model that organizes content into self-contained units, or "modules," allowing flexibility and personalization in learning paths.

OAuth2

An open standard protocol for secure authorization that enables users to grant applications access to their information without revealing passwords.

Open Badges

Digital credentials that validate learning achievements using verifiable metadata and are portable across platforms.

Participant

A platform user role representing a learner or student engaging with learning content, tasks, and assessments.

Progress Tracker

A dashboard component that visualizes a learner's advancement through course modules, tasks, and certifications.

Responsive Design

A web development approach that ensures optimal display and usability across devices of various screen sizes, including desktops, tablets, and smartphones.



SCORM (Sharable Content Object Reference Model)

A set of technical standards for e-learning software products that allows for the packaging and sharing of content across different LMS platforms.

Single Sign-On (SSO)

An authentication process that allows users to access multiple applications with one set of login credentials.

Trainer

A user role representing educators, instructors, or course designers who have permissions to create content, assess learners, and issue certifications.

UAT (User Acceptance Testing)

The final phase of testing where real users validate the system against requirements before going live.

VET (Vocational Education and Training)

Education and training that focuses on the acquisition of practical skills for specific occupations.

xAPI (Experience API)

A specification for learning technology that enables tracking of learning experiences both inside and outside traditional LMS environments.

REFERENCES

- Advanced Distributed Learning (ADL). (2023). *SCORM & xAPI documentation*. <https://adlnet.gov>
- Advanced Distributed Learning (ADL) Initiative. (2023). *xAPI and SCORM Standards*. <https://adlnet.gov>
- Aljawarneh, S. A. 2020). A survey on learning management systems (LMS): Benefits, challenges, and future opportunities. *Journal of Theoretical and Applied Information Technology*, 98(23), 3809–3825.
- BigBlueButton. (2023). *Open source web conferencing*. <https://bigbluebutton.org>
- Cedefop. (2020). *Microcredentials and Europass digital credentials*. <https://www.cedefop.europa.eu/en>



- Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: Defining gamification. *Proceedings of MindTrek*.
- Dougiamas, M., & Taylor, P. (2003). Moodle: Using learning communities to create an open source course management system. In *Proceedings of EDMEDIA 2003*.
- Django Software Foundation. (2024). *Django documentation*. <https://www.djangoproject.com>
- European Commission. (2016). *Regulation (EU) 2016/679 (General Data Protection Regulation)*.
- European Commission. (2021). *Europass digital credentials infrastructure*. <https://europass.europa.eu/en/european-digital-credentials>
- Fowler, M. (2021). *Continuous integration*. <https://martinfowler.com/articles/continuousIntegration.html>
- GÉANT eduGAIN. (2023). *Federated identity management*. <https://edugain.org>
- Google. (2023). *Mobile-friendly test*. <https://search.google.com/test/mobile-friendly>
- Hardt, D. (2012). The OAuth 2.0 authorization framework. *IETF RFC 6749*.
- IMS Global Learning Consortium. (2023). *LTI & Open Badges specifications*. <https://www.imsglobal.org>
- ISO/IEC/IEEE 29119:2013. *Software and systems engineering — Software testing*.
- Krug, S. (2014). *Don't make me think, revisited: A common sense approach to web usability*. New Riders.
- LTSC, IEEE. (2002). *Standard for Learning Object Metadata (LOM)*. IEEE 1484.12.1.
- Mozilla Developer Network (MDN). (2024). *Frontend frameworks and libraries*. <https://developer.mozilla.org>
- Nielsen, J. (1995). *10 usability heuristics for user interface design*. Nielsen Norman Group.
- PostgreSQL Global Development Group. (2024). *PostgreSQL documentation*. <https://www.postgresql.org>
- Romero, C., & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, 33(1), 135–146.
- W3C. (2018). *Web Content Accessibility Guidelines (WCAG) 2.1*. <https://www.w3.org/TR/WCAG21/>
- Wenger, E. (1998). *Communities of practice: Learning, meaning, and identity*. Cambridge University Press.



CITATION

This report is open access. It can be republished and cited free of charge without asking the project for permission. However, credit is always appreciated. When referencing information from this report, please mention it and link back whenever possible.

Report title: Functional and Technical Specification Framework

Report author: ComeThinkAgain Project

Website: <https://comethinkagain.eu/>