SYLLABIQ: AI-POWERED SMART SYLLABUS, ASSESSMENT & PERSONALIZED LEARNING PLATFORM

[1]Dr. S. R. Khare, [2]Mrs. V. O. Katiwal, [3]Jitendra Gaherwar, [4]Twinkal Dhekwar, [5]Nikhil Hans, [6]Manav Biranware

[1]Project Guide, [2]Project Co-Guide, [3]B.Tech Student, [4]B.Tech Student, [5]B.Tech Student, [6]B.Tech Student
[1-6]Department of Information Technology,
[1-6]Priyadarshini College of Engineering, Nagpur, Maharashtra, India

Abstract : *The growing need for organized digital upskilling and knowledge management has created demand for platforms that can connect learners with suitable training content and suitable trainers in a structured, auditable manner. This paper presents a competency-based framework for digital capacity building, implemented as a multi-role learning management platform named SyllabIQ. The proposed framework supports three roles — trainee, trainer, and admin — and considers trainer proficiency level, years of experience, historical course rating, and prior teaching frequency to support effective trainer-to-course matching. A weighted multi-factor scoring mechanism is used to rank available trainers for a given subject competency, while an approval-gated onboarding process and role-based access control govern platform-wide security. The framework continuously supports course authoring, enrollment, timed multiple-choice assessment, certificate issuance, and organizational analytics. The proposed approach provides a structured alternative to manual or purely seniority-based trainer assignment. Its effectiveness can be evaluated using metrics such as approval turnaround time, enrollment conversion rate, assessment pass rate, matching acceptance rate, and trainer utilization. The framework provides a practical, explainable approach to competency-based capacity building that can be extended toward learned matching models as historical data accumulates.*

IndexTerms *- Digital Capacity Building, Learning Management System, Competency-Based Matching, Trainer-Course Matching, Role-Based Access Control, Assessment Engine*

I. INTRODUCTION

The increasing emphasis on continuous organizational learning and structured skill development has raised the need for centralized platforms that can manage training content, track learner progress, and connect learners with appropriately qualified trainers. This need becomes more pronounced in specialized technical domains, such as earth-science and government-sector training programs, where subject-matter expertise is unevenly distributed across an organization and must be matched carefully to the training being delivered.

Conventional learning management systems (LMS) largely function as static content repositories: courses are authored and manually assigned to instructors and learners, generally without a structured mechanism to verify that the assigned trainer's competencies genuinely match the subject being taught. While such approaches are simple to implement, they do not scale well to organizations with a large trainer pool and a wide subject taxonomy, and they may under-utilize trainer expertise that is not immediately visible to administrators.

This paper presents SyllabIQ, a competency-based framework for digital capacity building that combines a role-gated learning management system with a multi-factor trainer-matching engine. The framework tags every trainer and every course against a single shared skill taxonomy, allowing trainer suitability for a course to be computed directly through relational queries rather than manual review or free-text search. Trainer proficiency, experience, historical trainee feedback, and teaching frequency are combined into a single composite score that ranks candidate trainers for a course, while the final assignment decision remains with the administrator.

The main contributions of this work are:

- A role-based digital capacity-building platform integrating trainee, trainer, and admin workflows under a single shared competency taxonomy.
- An approval-gated onboarding mechanism that keeps new trainee and trainer accounts in a pending state until administrator verification, supporting auditable access control.
- A multi-factor, weighted trainer-matching algorithm that ranks trainers for a course using proficiency level, normalized experience, historical course rating, and prior teaching count.
- A framework for evaluating platform effectiveness using metrics such as enrollment conversion, assessment pass rate, matching acceptance rate, and trainer utilization.

II. PROBLEM STATEMENT

Organizations undertaking large-scale digital capacity building need to efficiently connect a changing pool of learners with suitable courses and connect course subjects with suitably competent trainers. Manual trainer assignment, or assignment based only on trainer seniority or availability, may not always produce a suitable pairing because it does not adequately account for demonstrated subject proficiency, teaching history, or trainee-reported outcomes for that specific competency.

The problem addressed in this work is to design a framework that can (i) gate platform access through an auditable approval workflow, (ii) allow trainees and trainers to build structured profiles against a shared skill taxonomy, and (iii) rank available trainers for a course using multiple weighted factors rather than a single selection criterion, while leaving the final decision to an administrator. The objective is to improve the transparency and suitability of trainer allocation in organizational learning platforms while supporting measurable analytics on participation, certification, and competency growth.

III. LITERATURE REVIEW

Recent research has examined competency-based methods for guiding, recommending, and matching learners and instructors within e-learning platforms.

Hsu and Li [1] proposed a competency-based guided-learning algorithm (CBGLA) for adaptively guiding e-learners, using computational analysis of competency-based learning to steer learners toward suitable content. The approach demonstrates that competency modeling can be used to adapt content sequencing rather than only to describe learner ability.

Ghadirli and Rastgarpour [2] proposed an adaptive and intelligent tutoring approach for mobile devices that integrates expert-system reasoning with pre-test, learning, and post-test stages, showing that structured evaluation stages can personalize m-learning delivery at low infrastructure cost.

Ait Haddouchane et al. [3] reviewed the application of the competency-based approach to the training-employment matching problem, highlighting referential frameworks that align curricula and labor-market skill requirements, which motivates the use of a shared, structured skill taxonomy rather than free-text descriptions.

Sölch et al. [4] integrated competency-based education into the Artemis interactive learning system, allowing instructors to define competency relation graphs and visualize learner progress toward mastering each competency, demonstrating the practicality of embedding competency tracking directly inside an existing LMS architecture.

Alfonso et al. [5] proposed a semi-automatic method for generating competency maps from repositories of multiple-choice question responses using Bayesian structural learning, showing that assessment data itself can be mined to validate or refine a competency taxonomy over time.

Dutta et al. [6] presented an AI-based personalized course recommender that constructs learner competency profiles against the DigComp framework and ranks course recommendations using a hybrid scoring model combining semantic similarity and competency alignment, illustrating that multi-factor composite scoring is an effective and explainable strategy for competency-driven recommendation.

Zhang et al. [7] developed a comprehensive framework, grounded in the CC2020 competency model, for assessing and presenting student competencies in computer-science coursework, reinforcing that structured, multi-dimensional competency assessment remains an active and evolving research area in technical education.

The reviewed studies show that competency-based e-learning platforms increasingly rely on structured taxonomies, multi-factor scoring, and graph- or data-driven competency modeling. However, several of these approaches depend on advanced learning models, fuzzy computation, or bespoke recommender pipelines that require substantial historical or behavioral data before they can be trained reliably. In contrast, this work focuses on a practical, explainable, weighted multi-factor trainer-matching mechanism suitable for immediate deployment in a role-gated capacity-building platform, with a defined upgrade path toward a learned model once sufficient assignment-outcome data has been collected.

IV. PROPOSED METHODOLOGY

The proposed methodology implements a role-based digital capacity-building platform that connects trainee learning journeys, trainer content authoring, and administrator oversight through a single shared competency taxonomy. The system consists of trainee, trainer, and admin applications, a REST API layer with role-based guards, a competency-matching engine, an assessment engine, and a relational persistence layer.

A. Overall Proposed Model

The proposed system architecture, shown in Fig. 1, connects the client applications with a REST API gateway that enforces JWT authentication and role guards before dispatching requests to feature-isolated backend modules (Auth, Users, Courses, Assessments, Competency, Admin, and Announcements). PostgreSQL serves as the primary relational store, Redis is used for session caching and rate limiting, and an object storage layer holds uploaded learning resources and issued certificates.

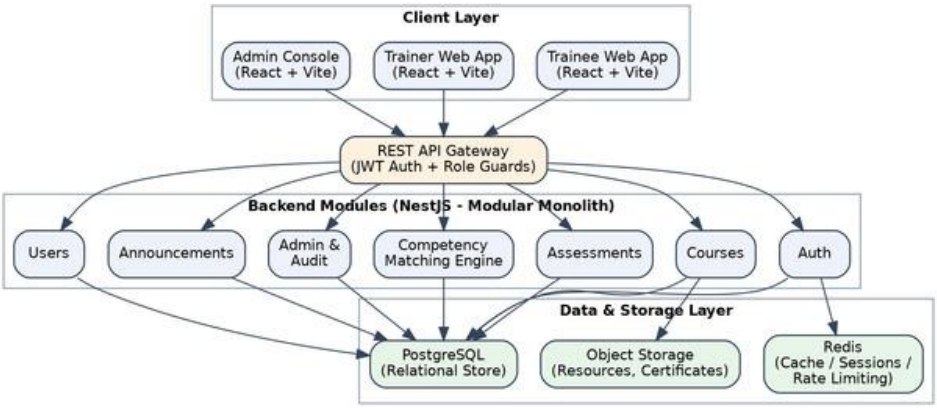

Fig. 1: Overall proposed system architecture

B. Trainee Module

The trainee application allows learners to build a profile of qualifications, work experience, and tagged skills; browse and search a course catalog by subject, difficulty, and trainer; self-enroll in open courses; consume uploaded learning resources; attempt timed multiple-choice assessments with auto-grading; download auto-generated certificates on successful completion; and submit post-completion feedback and ratings (Fig. 2).

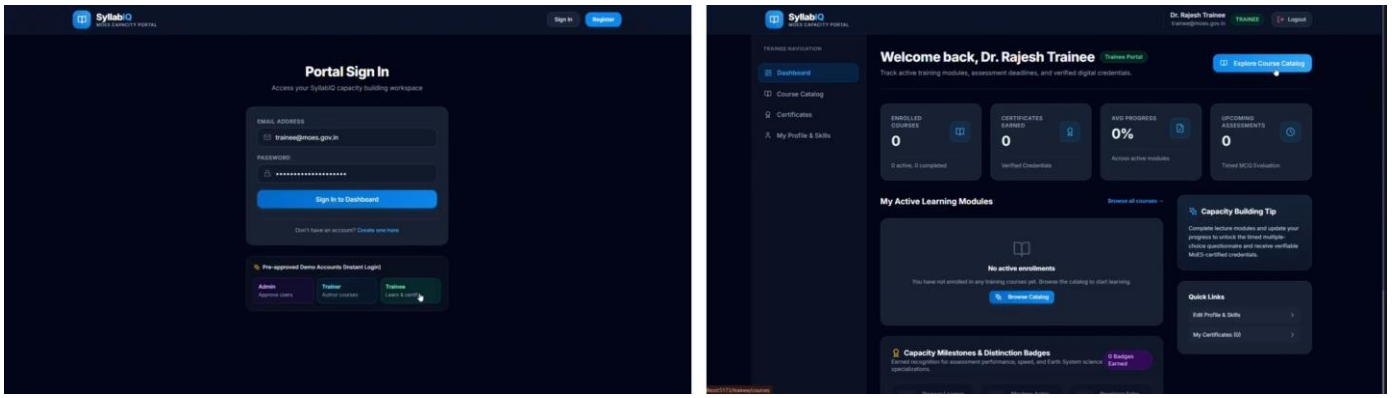

Fig. 2: Trainee-facing interface — portal sign-in (left) and trainee dashboard (right)

C. Trainer Module

The trainer application allows trainers to manage a teaching profile and competency tags, author courses and tag each course against the shared skill taxonomy, upload learning resources to a trainer library visible to enrolled trainees, build subject-wise question banks, and monitor trainee enrollment, progress, and quiz performance for their own courses (Fig. 3).

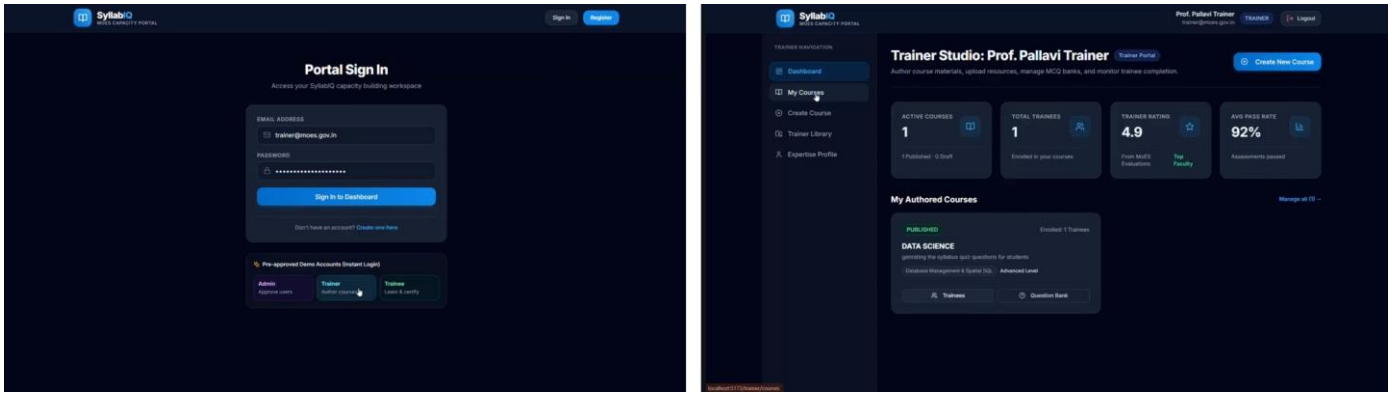

Fig. 3: Trainer-facing interface — portal sign-in (left) and trainer dashboard (right)

D. Admin Module and Competency-Matching Engine

The admin application gates account activation: newly registered trainee and trainer accounts remain in a pending state until explicitly approved, and every approval, role change, and administrative action is written to an audit log. The admin can additionally invoke the competency-matching engine for a given course, which queries the shared skill taxonomy and ranks all trainers holding the required competency using the weighted scoring function described in Section V-A, before presenting a ranked, overridable shortlist for final trainer assignment. The admin module also aggregates enrollment, certification, assessment pass-rate, and trainer-utilization analytics for organization-wide monitoring (Fig. 4).

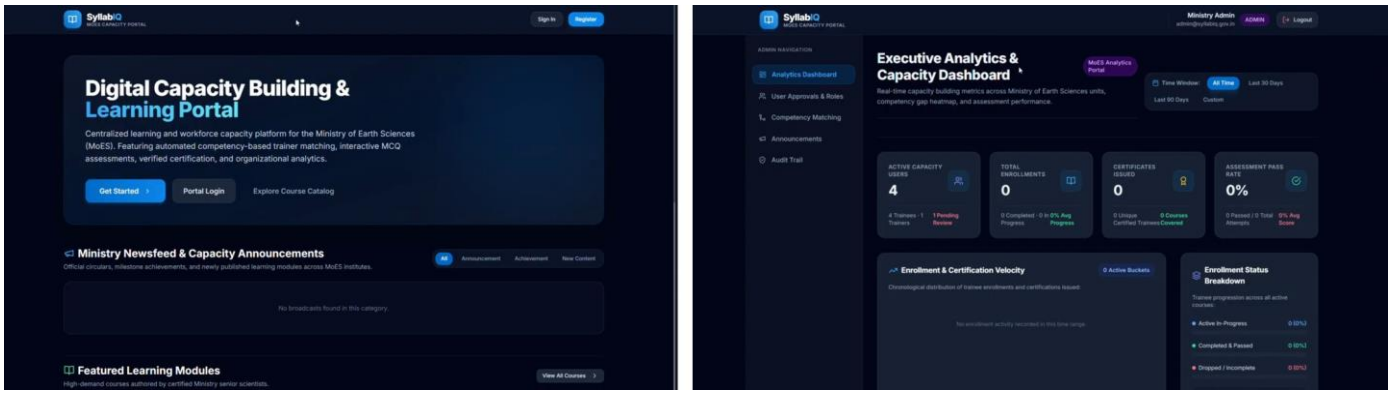

Fig. 4: Public landing page (left) and admin analytics dashboard (right)

V. IMPLEMENTATION

The platform is implemented as a modular monolith. The frontend uses React 18 with Vite and TypeScript, styled with Tailwind CSS, and relies on TanStack React Query for server-state caching, Zustand for client authentication/UI state, React Hook Form with Zod for validated forms, and React Router v6 for role-guarded routing. The backend uses Node.js with the NestJS framework, organized into feature modules with PostgreSQL as the primary data store accessed through parameterized SQL queries, and Redis (via ioredis) for caching and session-related operations. Authentication uses short-lived JWT access tokens together with password hashing (bcrypt/Argon2), and every protected route is guarded by role-based access control decorators.

A. Competency-Matching Algorithm

Given a course tagged against a subject-skill identifier, the framework ranks all trainers holding that skill using a weighted composite score:

$$score(\text{trainer, course}) = w_p \times 0.5 + e_n \times 0.2 + r_n \times 0.2 + c_n \times 0.1 \quad (1)$$

where w_p is the trainer's proficiency-level weight for the required skill (beginner = 0.25, intermediate = 0.5, advanced = 0.75, expert = 1.0); e_n is years of experience normalized as min(years/10, 1.0); r_n is the trainer's average past course rating for that skill, normalized to the [0, 1] range from trainee feedback; and c_n is the number of prior courses taught in that skill, normalized against the maximum count observed across all trainers. The factors and weights of Eq. (1) are summarized in Table I. Trainers are sorted in descending order of score, with years of experience and prior course count used as tie-breakers.

Table I: Factors and weights of the trainer-matching score

Factor	Symbol	Weight	Normalization
Proficiency level	w_p	0.5	Beginner 0.25, intermediate 0.5, advanced 0.75, expert 1.0
Years of experience	e_n	0.2	min(years/10, 1.0)
Average course rating	r_n	0.2	Trainee feedback scaled to [0, 1]
Prior courses taught	c_n	0.1	Count / maximum count across all trainers

This weighted, explainable scheme avoids the cold-start problem of a purely learned model and provides administrators with an interpretable score breakdown per trainer, while remaining structured so that the static weights in Eq. (1) can later be replaced by a learned model (e.g., logistic regression or gradient boosting) trained on historical assignment outcomes.

VI. RESULTS AND DISCUSSION

A. System Workflow

The overall workflow of the implemented platform is shown in Fig. 5. A new user registers as a trainee or trainer and is held in a pending state until an administrator approves the account. Once approved, the trainee path supports catalog browsing, enrollment, resource consumption, timed assessment, and certificate issuance with feedback; the trainer path supports course authoring with competency tagging, resource upload, question-bank creation, and trainee monitoring; and the admin path supports running the competency-matching engine, assigning or overriding trainer recommendations, and reviewing organization-wide analytics.

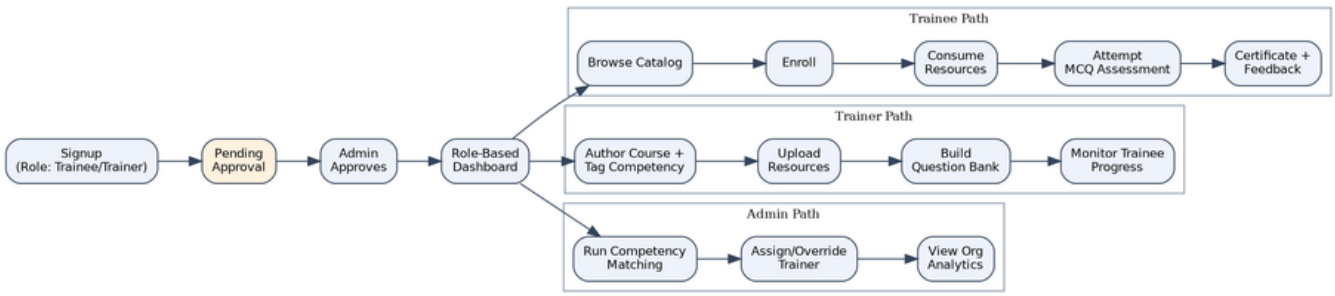

Fig. 5: System workflow of the proposed capacity-building platform

B. Discussion

The implemented platform provides an integrated environment for learner onboarding, course delivery, assessment, certification, and administrative oversight. It supports approval-gated authentication, role-based dashboards, course catalog search and enrollment, timed multiple-choice assessment with auto-grading, auto-generated certificates, post-completion feedback, and a competency-matching engine that produces a ranked, overridable list of trainers per course.

C. Evaluation Metrics

The effectiveness of the proposed framework can be evaluated using metrics such as approval turnaround time, enrollment conversion rate, assessment pass/fail rate by subject, matching acceptance rate (the proportion of admin-confirmed assignments that match the engine's top-ranked suggestion), and trainer utilization across the tagged skill taxonomy. Actual performance values should be reported after conducting experiments with real organizational usage data; therefore, no numerical performance claims are made in this paper without experimental measurement.

VII. Advantages

- Provides a centralized platform for managing training content, enrollment, and trainer allocation.
- Supports approval-gated onboarding for auditable, government-appropriate access control.
- Improves trainer selection by considering proficiency, experience, historical rating, and teaching frequency instead of seniority or availability alone.
- Uses a single shared skill taxonomy so trainer-course matching is a direct relational join rather than fuzzy text matching.
- Supports dynamic, overridable trainer recommendations that keep a human administrator in the final decision loop [9].
- Provides auto-graded assessment, certificate issuance, and feedback collection within one workflow.
- Reduces manual effort in learner and trainer administration through role-based dashboards and audit logging.
- Offers a modular monolith architecture that can be split into independently scalable services as usage grows.

VIII. Future Work

Future work can extend the static weighted-scoring model in Eq. (1) into a learned model (e.g., logistic regression or gradient boosting) trained on historical assignment outcomes such as completion rate and trainee satisfaction, once sufficient data accumulates. Planned directions also include AI-assisted content authoring, where trainer-uploaded material is used to draft candidate assessment questions for trainer review [8]; rule-based, and later model-based, at-risk-learner detection using login frequency and quiz performance; gamification through badges, streaks, and leaderboards to improve completion rates; hash-verifiable or digitally signed certificates with a public verification link; multilingual interface support and an offline-first, low-bandwidth mode for field or remote deployment contexts; and organization-wide skill-gap heatmaps built from the same shared competency taxonomy used for trainer matching.

IX. Conclusion

This paper presents a competency-based framework for digital capacity building, implemented as SyllabIQ, a role-gated learning management platform for trainees, trainers, and administrators. The proposed framework provides a structured, auditable approach for onboarding learners and trainers, delivering and assessing course content, and matching trainers to courses using a multi-factor weighted scoring mechanism instead of manual or seniority-based assignment. The system architecture integrates role-based client applications, a guarded REST API, feature-isolated backend modules, and a relational persistence layer with caching support. The implemented platform supports course authoring, enrollment, timed assessment, certification, feedback, and organizational analytics, providing the technical foundation required for competency-driven capacity building. The modular architecture and explainable matching mechanism can support future improvements, including learned matching models and large-scale organizational deployment.

Acknowledgment

The authors thank the Department of Information Technology, Priyadarshini College of Engineering, Nagpur, for the guidance and facilities provided during this work.

References

[1] W.-C. Hsu and C.-H. Li, "A competency-based guided-learning algorithm applied on adaptively guiding e-learning," Interactive Learning Environments, vol. 23, no. 1, pp. 106–125, 2015.

[2] H. Movafegh Ghadirli and M. Rastgarpour, "An adaptive and intelligent tutor by expert systems for mobile devices," International Journal of Managing Public Sector Information and Communication Technologies, vol. 3, no. 1, pp. 21–29, 2012.

[3] Z. Ait Haddouchane, S. Bakkali, S. Ajana, and K. Gassemi, "The application of the competency-based approach to assess the training and employment adequacy problem," arXiv:1704.04985, 2017.

[4] M. Sölch, M. Aberle, and S. Krusche, "Integrating competency-based education in interactive learning systems," in Companion Proc. 13th Int. Conf. Learning Analytics & Knowledge (LAK23), 2023.

[5] D. Alfonso, A. Manjarrés, and S. Pickin, "Semi-automatic generation of competency maps based on educational data mining," International Journal of Computational Intelligence Systems, vol. 12, no. 2, pp. 744–760, 2019, doi: 10.2991/ijcis.d.190627.001.

[6] S. Dutta, F. Beier, and D. Werth, "AI-based personalized multilingual course recommender system using large language models," in Proc. 17th Int. Conf. Agents and Artificial Intelligence (ICAART), vol. 3, 2025, pp. 1069–1076, doi: 10.5220/0013260100003890.

[7] Y. Zhang, N. Zheng, S. Li, and W. Liu, "Student competency assessment and presentation methods based on algorithm courses," arXiv:2606.00200, 2026.

[8] "MCQGen: A Large Language Model-Driven MCQ Generator for Personalized Learning," IEEE Access, vol. 12, pp. 102261–102273, 2024, doi: 10.1109/ACCESS.2024.3420709.

[9] D. C. Fajardo-Ramos and A. Chiappe, "Human-in-the-loop assessment with AI: implications for teacher education in Ibero-American universities," Frontiers in Education, vol. 10, 1710992, 2025, doi: 10.3389/feduc.2025.1710992.

Copyright & License:
© Authors retain the copyright of this article. This work is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.