Dynamic Design Education – Daily Dose of Engineering

Prof. Deepika Walanjkar
Department of Information Technology
International Institute of Information Technology
Pune , India
deepikaw@isquareit.edu.in

Atharva Hegade
Department of Information Technology
International Institute of Information Technology
Pune , India
atharvahegade007@gmail.com

Swapnil Fuse
Department of Information Technology
International Institute of Information Technology
Pune,India
swapnilfuse85@gmail.com

Raghav Daga
Department of Information Technology
International Institute of Information Technology
Pune , India
raghavdaga1104@gmail.com

Sanket Dongre
Department of Information Technology
International Institute of Information Technology
Pune , India
sanketdongre33@gmail.com

1.Abstract—

The Dynamic Design Education (DDE) project is a strategic initiative aimed at creating a comprehensive and dynamic web application for engineering students. The core mission is to bridge the significant gap between theoretical academic knowledge and the practical demands of the modern technology industry. 1 By targeting this critical audience, the DDE seeks to foster technical proficiency, enhance interview readiness, and develop problem-solving capabilities that are directly relevant to professional engineering roles. The platform's unique value proposition is its holistic and integrated ecosystem. Unlike existing solutions that address only a single aspect of the learning journey, such as course platforms (e.g., Coursera), coding practice sites (e.g., LeetCode), or community forums (e.g., Reddit), the DDE is designed as a single, "one-stop-shop" solution. It integrates daily, bite-sized microlearning content, adaptive learning paths powered by sophisticated algorithms, advanced career-readiness tools such as an ATS resume checker, and a vibrant, interactive community platform. This comprehensive approach streamlines the user's educational and career development journey.

2. Introduction

2.1 Background and Motivation

Contemporary education is undergoing a profound transformation driven by the exponential growth of digitized knowledge and the widespread availability of open educational resources (OERs). The global disruption caused by COVID-19 dramatically accelerated the shift toward remote instruction, compelling institutions to rapidly adopt online delivery mechanisms that integrate learning management systems with OER repositories across platforms such as YouTube. Despite this rapid transition, conventional pedagogical models—which uniformly apply a single instructional strategy regardless of individual differences—consistently fail to account for variations in learner proficiency, pace of acquisition, or pre-existing knowledge. The foundational motivation of this project lies in reconceiving digital learning platforms not merely as repositories for instructional content, but as intelligent environments capable of fostering deep, lasting knowledge mastery, analytical reasoning, and applied problem-solving competencies. Achieving genuinely effective personalized learning, however, introduces significant engineering challenges. The most critical among these is the data cold-start problem, wherein a newly deployed or recently updated system lacks the historical interaction data required to generate contextually appropriate and accurate recommendations for incoming users or newly introduced content. This challenge is particularly pronounced within micro-open learning environments, where learners interact with content in brief, fragmented sessions, rendering the system's initial understanding of each learner extremely limited. The absence of sufficient early interaction data prevents the immediate construction of comprehensive learner models—a capability that is indispensable for offline adaptive computation systems such as Microlearning as a Service (MLaaS). In the absence of an effective mitigation strategy, this data scarcity leads to poor recommendation quality, diminished user experience, and ultimately, elevated platform abandonment rates. To address these shortcomings, this work proposes a novel solution grounded in real-time online computation, coupled with a hybrid algorithmic framework designed for fine-grained, personalized learning trajectory construction and long-term memory reinforcement. The DDE platform and the MLaaS paradigm collectively target the delivery of adaptive learning sequences and assessments that meaningfully bridge the gap between academic theory and the practical skill demands of the engineering industry.

2.2 The Challenge of Adaptive Learning and Cold Start

Building truly effective personalized learning systems entails overcoming substantial technical obstacles. The most pressing of these is the cold-start problem, which manifests when a platform lacks adequate prior interaction history to generate reliable and contextually relevant recommendations—whether for newly registered users or for recently published learning content. This issue becomes especially critical in micro-open learning environments, where learners engage with educational material in short, non-contiguous sessions, leaving the system with very limited behavioral signals to characterize each learner's needs.

This scarcity of early behavioral data prevents the immediate construction of comprehensive, high-fidelity learner models—a prerequisite for the effective operation of offline adaptive computation frameworks such as Microlearning as a Service (MLaaS). In the absence of a robust mitigation strategy, the resulting recommendation inaccuracies degrade the overall user experience and risk learner disengagement from the platform.

2.3 Project Goal and Novel Approach

To address these technical challenges, we introduce an approach grounded in real-time online computation combined with a hybrid algorithmic architecture specifically engineered for fine-grained learner personalization and long-term memory reinforcement. Both the DDE platform and the MLaaS framework are designed to produce adaptive learning sequences and targeted assessments that effectively connect theoretical academic content with the practical competencies demanded by the modern technology industry.

3.Related Work

3.1 Adaptive Learning and Personalized Systems

The rapid expansion of Open Educational Resources (OERs) and the accelerating global shift toward remote instructional delivery—catalyzed by events such as the COVID-19 pandemic—have underscored the critical demand for flexible, robust educational systems that can operate beyond the constraints of traditional classroom delivery. Conventional static instructional models, which present fixed content regardless of individual background knowledge or learning velocity, are poorly equipped to serve the diverse and fragmented learning contexts characteristic of today's digital learners.

Adaptive Learning Components: Research into building effective adaptive systems focuses heavily on modelling both the user and the content:

A. Learner Profiling: Effective adaptation relies on accurate learner models. Prior research has established comprehensive learner models, sometimes annotated using semantic approaches, incorporating intellectual, non-intellectual, and external contextual factors relevant to micro-open learning experiences and outcomes. Other works focus on predicting learning styles and automatic learner classification. However, deploying such comprehensive models is challenging in new environments due to initial data scarcity. This challenge motivates the use of simplified, lightweight learner profiles for swift decision-making during the platform's initialization phase.

B. Content Representation: Content organization, specifically for OERs, benefits significantly from semantic Web technologies and ontological modelling. To support fine-grained adaptation, OER ontologies can be augmented to represent micro-OERs as self-describing nodes annotated with metadata, including parameters like the didactic model, interaction type, and functional attributes such as Quality of Service (QoS).

3.2 Recommendation Systems and the Cold Start Problem

The construction of personalized microlearning pathways is fundamentally constrained by the cold-start problem—a consequence of insufficient historical interaction data for newly onboarded users or continuously published OER content. This inherent data sparsity undermines the effectiveness of offline, data-intensive computation systems, such as the proposed MLaaS framework, by rendering immediate and accurate learner model construction computationally infeasible.

Earlier approaches to cold-start mitigation have typically centered on user profiling techniques, frequently drawing upon well-established benchmark datasets such as MovieLens. Nevertheless, researchers have observed that recommendation models pre-trained on entertainment-domain data exhibit poor transferability to educational settings. This performance gap arises from the structural and semantic isolation of the educational domain, compounded by the limited availability of suitable large-scale annotated learning datasets. As a result, the field has increasingly oriented itself toward recommendation strategies that minimize reliance on explicit user preference ratings—information that is typically unavailable at the outset of a learning session.

To address these cold-start limitations, the literature proposes coupling real-time online computation with heuristic-driven interventions during the platform's initialization phase. These heuristic mechanisms leverage lightweight learner profiles alongside domain-specific rules to construct preliminary recommendation lists, while simultaneously integrating newly published micro-OERs into pre-existing learning pathways to enhance their discoverability and uptake.

3.3 Memory Reinforcement and Adaptive Assessment

Effective adaptive systems must move beyond content delivery to focus on long-term knowledge retention, primarily supported by cognitive science principles:

A. Spaced Repetition: The concept of scheduling repetition at progressively increasing intervals is rooted in the work of Ebbinghaus, who hypothesized that the human forgetting rate increases exponentially with time. The goal is to review items precisely as they are about to be forgotten, ensuring maximum retention. Foundational algorithms include the Pimsleur Method (fixed schedule) and the Leitner System (more adaptive spacing based on performance). More recently, Half Life Regression (HLR) has been proposed as a successful generalization of these models for language learning platforms.

B. Adaptive Algorithms for Retention: To address the varying needs of diverse learners, models like the Spaced Repetition for Slow Learners (SRSL) algorithm have been developed. SRSL computes a learner's score based on fine-grained factors such as response time, question difficulty, and prerequisite dependency. It integrates prerequisite relationships via a directed acyclic graph, allowing positive reinforcement (rewarding prerequisite questions when a subsequent question is answered correctly) and penalty application for closely associated topics when mistakes occur.

C Adaptive Quizzing: In adaptive testing, two models are frequently combined for robustness: Bayesian Knowledge Tracing (BKT), which probabilistically tracks a learner's mastery of specific skills over time, and Item Response Theory (IRT), which estimates the intrinsic characteristics of test questions, such as difficulty and discrimination. Combining these models is critical, as relying solely on IRT can present a significant "cold-start" problem due to the need for large amounts of response data for accurate item calibration.

3.4 Algorithmic Intelligence for Path Planning

The generation of dynamic, optimal learning sequences aligns with complex decision-making problems traditionally addressed by game Artificial Intelligence (AI) and planning algorithms.

3.4.1 Monte Carlo Algorithm

Monte Carlo Tree Search (MCTS) is a stochastic, simulation-based decision-making algorithm that has become a cornerstone technique in artificial intelligence, particularly for constructing autonomous agents capable of navigating complex, sequential decision environments. At its core, the method employs an asymmetric tree construction strategy governed by a principled trade-off between exploiting well-performing actions discovered in prior iterations and systematically exploring unvisited regions of the decision space. Through repeated stochastic simulations—commonly termed rollouts or playouts—the algorithm accumulates performance statistics, iteratively refining its action preferences. Initially conceived for developing competitive game-playing agents in the domain of Go, MCTS operates as a heuristic search procedure requiring only knowledge of state-transition rules, while optionally incorporating domain-specific guidance to accelerate solution convergence.

Each iteration of the classical MCTS algorithm consists of four well-defined, sequentially executed phases:

A. Selection: Beginning at the root node, the algorithm traverses the portion of the search tree that has already been constructed in memory. At each intermediate node, the subsequent node is selected according to a tree policy until either a leaf node or a terminal state is encountered.

B. Expansion: When the selection phase terminates at a non-terminal leaf node, the algorithm proceeds to the expansion phase, during which one or more new child nodes are appended to the tree. Each new node represents a distinct state reachable via an action not yet explored from that position.

C. Simulation (Playout): From the newly expanded node, a complete probabilistic simulation of the problem is conducted—typically guided by a default stochastic rollout policy—until the simulation reaches a terminal state and a final outcome or payoff value is obtained.

D. Backpropagation: The terminal outcome derived from the simulation is propagated backward through the traversal path, from the terminal state up to the root. This update step refines the stored statistical estimates—including visit frequencies and cumulative reward values—for each node and agent encountered along the path.

The exploration-exploitation balance that underpins MCTS effectiveness is most widely realized through the Upper Confidence Bounds applied for Trees (UCT) selection criterion, giving rise to the commonly referenced MCTS/UCT variant. UCT directs computational effort toward the most promising branches of the search tree by assigning each candidate action a score composed of two terms: an exploitation component reflecting the historical average reward for that action, and an exploration term proportional to the square root of the logarithm of the parent node's visit count divided by the child node's visit count, scaled by an exploration constant. Alternative selection strategies also exist, including Thompson Sampling and the Exponential-weight algorithm for Exploration and Exploitation (EXP3). Additionally, the Rapid Action Value Estimation (RAVE) enhancement has been introduced to mitigate initialization bias by maintaining a supplementary value estimate () that is updated for *all* actions taken during the simulation phase, effectively sharing action values across related subtreesApplications of MCTS extend widely beyond the realm of games into complex planning and decision-making systems.

• Automated Planning: MCTS is frequently applied in automated planning problems, which are typically formulated as a Markov Decision Process (MDP) or a Partially Observable MDP (POMDP). Specific industrial and academic challenges where MCTS has been utilized include scheduling problems (such as forest harvest scheduling) and various formulation of Vehical Routing Problem (VRP).

Chemical Synthesis: MCTS has achieved novel results when combined with deep neural networks for computer-aided retrosynthesis planning (planning chemical syntheses), in an approach termed 3N-MCTS.

Adaptive Learning Paths: In modern personalized learning platforms, MCTS is proposed for dynamically constructing optimal learning sequences for users. This process frames learning as a search problem on a knowledge graph where learning units are represented as nodes and prerequisites as edges. By intelligently managing the balance between exploring new content and exploiting (reinforcing) mastered topics, MCTS moves beyond static curricula to truly personalize the learning journey.

MCTS offers a distinctive set of technical advantages that make it well-suited for integration into research-driven adaptive systems. Its asymmetric tree construction naturally concentrates computational effort on the most decision-relevant branches of the search space. As an anytime algorithm, MCTS can return the best solution identified up to any point of interruption, making it suitable for time-constrained applications. The algorithm is also inherently amenable to parallelization, with established strategies including Leaf, Root, and Tree Parallelization. When combined with Machine Learning (ML)—as demonstrated in game-playing systems such as AlphaGo—MCTS benefits substantially from the approximation and generalization power of neural network-based value and policy functions, enabling evaluation of states that have not been previously encountered. A notable limitation, however, is MCTS's susceptibility to combinatorial search space explosion. This challenge is commonly addressed through action space reduction, problem model simplification, or by running MCTS in an offline precomputation mode to generate high-quality training datasets, which are subsequently used to train a more computationally efficient surrogate model for deployment.

3.4.2 MCTS in Personalized Learning (DDE Platform)

In the context of personalized education, MCTS is used to generate adaptive learning paths.

Role in Learning Engine: The DDE Learning Engine leverages MCTS to dynamically construct optimal learning sequences for users

Framing the Problem: The learning process is explicitly framed as a search problem on a knowledge graph.
Graph Components: In this graph, learning units are represented as nodes, and prerequisites are represented as edges.
Balancing Learning: MCTS intelligently explores this graph, ensuring a balance between "exploration" (introducing new, less-tested topics) and "exploitation" (reinforcing mastered topics) to create a personalized learning journey.
Differentiation: This novel application of a game-playing algorithm is a significant differentiator for the platform, allowing the system to move beyond static curricula and truly personalize the experience.
Modifications and Challenges
The vanilla MCTS algorithm often fails to deliver excepted performance in complex settings with other techniques.
Hybridisation: MCTS is frequently hybridised with other techniques, such as Machine Learning (ML) and Evolutionary Methods. For instance, ML models (like neural networks) can be used to model the value function (approximating the game outcome) and the policy function (informing which action to choose).
Performance Improvements: Methods exist to enhance the base algorithm, such as Rapid Value Action Estimation (RAVE), which minimizes the "cold start" effect by using action values gathered from all actions during simulation, not just selected ones.
Addressing Search Space Explosion: MCTS is prone to search space explosion (combinatorial complexity). Modifications often focus on Action Reduction (eliminating poor actions to prevent the tree from growing sideways) or simplifying the problem model.
Parallelization: MCTS is inherently easy to parallelize, which is a common strategy to increase the overall number of iterations performed within a computational budget. The three major approaches to parallel MCTS are Leaf Parallelization, Root Parallelization, and Tree Parallelization.

3.4.3.Bayesian Knowledge Tracing (BKT) and Item Response Theory (IRT)

The integration of Bayesian Knowledge Tracing (BKT) and Item Response Theory (IRT) models forms a robust hybrid approach specifically designed for adaptive quizzing and personalized assessment. This synergistic strategy addresses the limitations of each model when used in isolation, leading to a more accurate and scalable assessment engine, particularly suitable for platforms like the Daily Dose of Engineering (DDE).
Below is a detailed breakdown of this integration, focusing on the distinct roles, mechanisms, and benefits of each model:

A.Bayesian Knowledge Tracing (BKT)

BKT is the core model for tracking a learner's skill mastery over time.
Function: BKT is a probabilistic model that is specialized as a Hidden Markov Model. It infers and tracks whether a learner has mastered a specific skill, representing the learner's knowledge state as a binary variable (they either know the skill or they do not).
Skill Tracking: The system continuously updates its belief (the mastery probability,) regarding a user's skill mastery after every question is answered, regardless of whether the answer was correct or incorrect.
Key Parameters: BKT relies on four skill-specific parameters that must be calibrated for each knowledge component:
(p-init): The initial probability that a student knows the skill.(p-transit): The probability that a student learns a skill after an opportunity to apply it. (p-slip): The probability that a student makes an error even when they genuinely know the skill (accounting for human error). (p-guess): The probability that a student correctly answers a question without actually knowing the underlying skill.

Limitation in Isolation

Standard BKT models typically do not account for the difficulty of the questions presented. A correct response to a simple question is treated the same as a correct response to a difficult one in terms of updating the mastery probability, which can restrict the model's accuracy.

B. Item Response Theory (IRT)

IRT focuses on the inherent characteristics of the questions (items) themselves, providing context for the difficulty of the assessment.
Function: IRT provides a probabilistic characterization of the likelihood that a given examinee correctly responds to a specific item, conditioned on both the individual's latent ability level and the item's intrinsic psychometric properties. This model complements BKT by enabling the system to identify and prioritize the most diagnostically informative item to present at each assessment step.
Key Parameters: IRT models rely on item-specific parameters, commonly including:
Difficulty (): The ability level () at which a test-taker has a 50% chance of answering the item correctly.
Discrimination (): How well an item differentiates between examinees with ability levels just below and just above the item's difficulty.
A guessing factor () may also be included to account for random correct guesses on multiple-choice questions.
Limitation in Isolation: The major challenge of implementing IRT, especially in new platforms, is the "cold-start" problem. IRT requires a large quantity of pre-collected data (potentially hundreds of responses per item) to accurately calibrate and refine the difficulty and discrimination parameters of each question.

C.The Hybrid BKT/IRT Integration Approach

The combined model maximizes accuracy and addresses the cold-start data challenge by ensuring a continuous feedback loop between the question quality (IRT) and the learner state (BKT).
Mechanism of Integration: The core integration involves using the IRT-derived difficulty parameter (the 'b' value) of a question to refine the BKT update process. This means that if a user correctly answers a particularly challenging question (high IRT difficulty), their estimated mastery probability () for the associated skill should increase more significantly than if they answer an easy question correctly.

Addressing Cold-Start: This hybrid approach is particularly effective for new platforms (like DDE) with a nascent user base:

1. Initial Setup: Initially, subject matter experts can manually assign an estimated difficulty level to each question. This preliminary difficulty data informs the BKT model from the start, mitigating the immediate data scarcity.
2. Calibration and Refinement: As the system collects more user response data, IRT is continuously used to calibrate and refine the actual difficulty and discrimination parameters of each question.
3. Enhancement: These refined IRT parameters are then used to enhance the accuracy of the BKT mastery model, making its mastery probability updates more sensitive to item difficulty.

Adaptive Assessment: The hybrid model provides a robust foundation for real-time adaptive assessment. By tracking individual skill mastery (BKT) and knowing the precise characteristics of the available questions (IRT), the system can select the optimal question to maximize the efficiency and accuracy of assessment.

This comprehensive, hybrid modelling approach ensures the system provides precise and meaningful personalization while simultaneously gathering the necessary data for long-term algorithmic sophistication.

4.Methodology

4.1 System Architecture

The system architecture is detailed across several systems described in the sources, most notably the Dynamic Design Education (DDE) platform and the Microlearning as a Service (MLaaS) framework, as well as general e-learning cloud structural designs. Design Education (DDE) Platform Architecture Core Technology Stack Architectural Style: Microservices architecture, breaking the application into smaller, independent services.

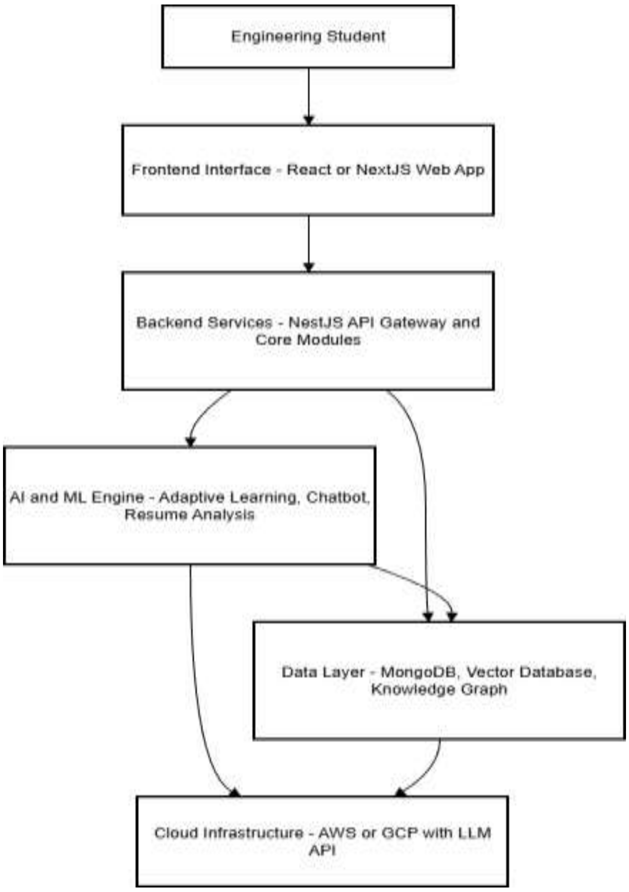

Fig. 1. Proposed System Architecture

Backend Core: NestJS, a progressive Node.js framework, is used for core business logic, providing a robust foundation and native support for the microservices architectural style. Specialized Logic: Dedicated Python microservices are employed for advanced machine learning (ML) functionalities, leveraging Python’s rich data science ecosystem.

Frontend: The user interfaces are built using React.

Data Layer: MongoDB (a NoSQL document database) is the primary data layer due to its flexible schema, suited for

Service	Responsibility	Service
User/Auth Service	Manages user profiles, authentication, and authorization.	User/Auth Service
Content Service	Handles all learning materials, including micro-notes, quizzes, and dynamic learning paths.	Content Service
Community Service	Manages the social-media-style feed and real-time interactions.	Community Service
AI/ML Service	A collection of Python microservices responsible for the AI chatbot and the ATS resume checker.	AI/ML Service

diverse data types like user profiles, content, and community posts. Prisma is used as the Object Data Model (ODM) layer for consistency and developer productivity. System Components and Data Flow All client requests interact with the system through a single API Gateway, which handles security, authentication (using JWT or OAuth 2.0), and routing to the appropriate backend service. The architecture is composed of several autonomous microservices:

Key Algorithmic Subsystems AI/ML Service The DDE learning engine integrates sophisticated algorithms, with the complex calculations routed to the Python microservice:

1. Adaptive Quizzing: Uses a hybrid Bayesian Knowledge Tracing (BKT) and Item Response Theory (IRT) model to track skill mastery and select optimal questions. **2.Personalized Flashcards**: Uses an Anki-inspired adaptation of the Super Memo SM-2 algorithm for dynamically scheduling review intervals based on user performance, ensuring long-term memory reinforcement. **3. Personalized Learning Paths**: Utilizes a Monte Carlo Tree Search (MCTS) algorithm operating on a foundational knowledge graph to dynamically construct optimal learning sequences.

4.AI Chatbot: Implemented using a Retrieval-Augmented Generation (RAG) architecture. This design grounds the Large Language Model (LLM) in DDE's proprietary knowledge, which is pre-processed into vectors and stored in a vector database (or MongoDB Atlas Vector Search). Real-Time Communication. The Community Service utilizes a hybrid real-time communication strategy:

5. Server-Sent Events (SSE): Used for unidirectional news feed updates (a "server push" or "fan-out on write" model).

6. WebSockets: Employed for bidirectional, low-latency communication, suitable for features like live chat.

7. Message Broker: Services are underpinned by a message broker like Kafka for high-throughput, durable event streams.

4.2 Deployment

The integration of e-learning systems with cloud infrastructure introduces a distinct set of operational and institutional challenges. Chief among these are cybersecurity vulnerabilities, constraints imposed by limited network bandwidth in resource-constrained settings, and organizational resistance arising from the reluctance of institutional stakeholders to commit to new technological paradigms. Furthermore, cloud-based and traditional learning management systems diverge significantly in their instructional delivery models, content governance practices, and assessment mechanisms, necessitating comprehensive adaptation efforts from both instructors and learners.

The literature consistently affirms that while cloud adoption substantially reduces infrastructure expenditure and enables dynamic scalability, institutional transitions must be managed with deliberate planning and stakeholder alignment. The disruptions caused by the COVID-19 pandemic further underscored the strategic value of cloud-hosted learning platforms in sustaining educational continuity and enabling collaborative, technology-mediated pedagogy across geographically distributed learner populations.

Empirical investigations into cloud adoption within higher education reveal that successful deployment is contingent upon multiple interacting factors: government policy support, institutional readiness, the maturity of existing IT infrastructure, and the depth of engagement with cloud service providers. Systematic literature reviews further highlight persistent barriers to broad adoption, including a lack of cross-platform interoperability, usability challenges in interface design, and a notable deficit of rigorous, large-scale empirical evaluations in this research domain.

In aggregate, cloud-enabled e-learning represents a transformative opportunity to broaden educational access and foster collaborative knowledge construction; however, realizing this potential demands targeted investment in research, evidence-based deployment frameworks, and robust security and interoperability standards to support sustainable, equitable adoption at scale.

The MLaaS framework presents a structured architectural model for delivering adaptive microlearning experiences through the systematic orchestration of Open Educational Resources (OERs).

Service Model: MLaaS operates as a cloud-hosted Software as a Service (SaaS) platform, abstracting the underlying computational infrastructure and delivering adaptive learning capabilities on demand.

Resource Integration: A centralized Master Service continuously crawls, identifies, and indexes all available OERs and their associated container environments, ensuring comprehensive resource coverage and up-to-date content availability.

Computation Integration: The framework employs a dual computation strategy, combining real-time online processing to address cold-start scenarios with computationally intensive offline adaptive modeling based on accumulated learner profiles, together governing the selection and sequencing of learning resources.

Semantic Integration: To achieve robust cross-system interoperability, MLaaS is architecturally grounded in an enriched micro-OER ontology. A Semantic Web-based representational scheme ensures that heterogeneous content originating from disparate OER repositories can be uniformly described, interlinked, and retrieved with high fidelity. The resulting ontological layer serves as a shared knowledge substrate that powers both the Adaptive Engine's personalization logic and the Cold Start Engine's initialization routines.

5.Results

Figure 5.1: The dashboard interface of the DDE platform displays the user's overall learning progress, including XP level, day streak, and skill proficiency heatmap across core engineering domains. It also provides quick access to spaced-repetition flashcards, quizzes, and the ATS resume checker. Weekly learning activity is visualized using a time-series graph to help learners monitor their engagement patterns

.

Fig.5.1

Figure 5.2: The Daily Missions and AI-Driven Recommendations section provides personalized learning tasks such as concept learning, flashcard revision, and quiz attempts. The system uses adaptive intelligence to identify high-priority topics (e.g., Binary Trees) and guide the learner toward areas requiring improvement. This supports a structured, goal-oriented learning experience.

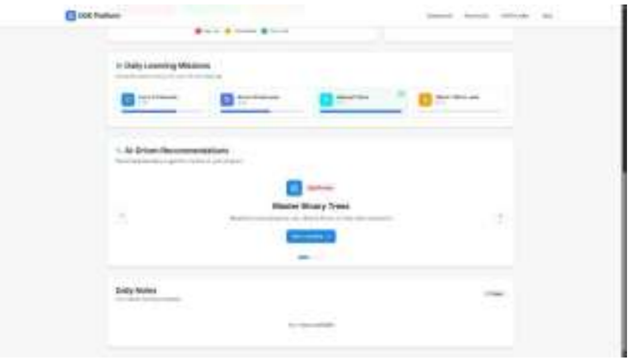

Fig.5.2

Figure 5.3:Dynamic flashcards containing information of various topic for revision

Fig.5.3

Figure 4 : ATS resume checker to check ats friendly resume

Fig.4

6 References

[1]*Dialogue Management in Conversational Systems: A Review of Approaches, Challenges, and Opportunities Hayet Brabra , Marcos Báez, Boualem Benatallah, Senior Member, IEEE, Walid Gaaloul , Sara Bouguelia, and Shayan Zamanirad*

[2]*Adapting New Learners and New Resources to Micro Open Learning via Online Computation Geng Sun , Member, IEEE, Wei Wei , Senior Member, IEEE, Tingru Cui, Dongming Xu, Shiping Chen , Senior Member, IEEE, Alex Shvonski, Li Li, Member, IEEE, Jun Shen , Senior Member, IEEE, and Soheila Garshasbi*

[3]*E-Learning-Based Cloud Computing Environment: A Systematic Review, Challenges, and Opportunities HANA ELJAK 1 , ASHRAF OSMAN IBRAHIM 2,3, (Senior Member, IEEE), FAKHRELDIN SAEED4 , IBRAHIM*

[4] *Spaced Repetition for Slow Learners Devansh P. Shah, Nikhil M. Jagtap, Shloka S. Shah and Anant V. Nimkar Department of Computer Engineering Sardar Patel Institute of Technology Mumbai, India*

[5]*O. K. J. Mohammad, ''Recent trends of cloud computing applications and services in medical, educational, financial, library and agricultural disciplines,'' in Proc. 4th Int. Conf. Frontiers Educ. Technol., 2018*

[6]*G. Riahi, ''E-learning systems based on cloud computing: A review,'' Proc. Comput. Sci., vol. 62, pp. 352–359, Jan. 2015, doi: 10.1016/j.procs.2015.08.415.*

[7] *S. Azouzi, S. A. Ghannouchi, and Z. Brahmi, ''Study of e learning system based on cloud computing: A survey,'' in Proc. Int. Conf. Intell. Syst. Design Appl. Cham,*

Switzerland: Springer, 2018, pp. 532–544, doi: 10.1007/978-3-030-16660-1_52.

[8] B. S. Shukur, M. K. A. Ghani, and M. Burhasnuddin, ''A cloud computing framework for higher education institutes in developing coun tries (CCF_HEI_DC),'' in Intelligent and Interactive Computing. Cham, Switzerland: Springer, 2019, pp. 397–409, doi: 10.1007/978-981-13 6031-2_24.

[9] E. Leloglu, T. Ayav, and B. G. Aslan, ''A review of cloud deploy ment models for e-learning systems,'' in Proc. 43rd Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN), Jun. 2013, pp. 1–2, doi: 10.1109/DSN.2013.6575331.

[10] M. A. H. Masud and X. Huang, ''An e-learning system architecture based on cloud computing,'' World Academy Sci., Eng., Technol., vol. 62, pp. 74–78, Jan. 2012.

[11] A. El Mhouti, M. Erradi, and A. Nasseh, ''Using cloud computing ser vices in e-learning process: Benefits and challenges,'' Educ. Inf. Technol., vol. 23, no. 2, pp. 893–909, Mar. 2018, doi: 10.1007/s10639-017-9642-x.

[12] M. Ketel, ''E-learning in a cloud computing environment,'' in Proc. IEEE SOUTHEASTCON, Mar. 2014, pp. 1–2, doi: 10.1109/ SECON.2014.6950728.

[13] J.Hylen,D.V.Damme,F.Mulder,and S. D' Antoni, "Open educational resources: Analysis of responses to the OECD country questionnaire," OECD, Paris, France, OECD Educ. Working Papers 76, Jun. 2012. [14] G. Sun, T. Cui, S. Chen, W. Guo, and J. Shen, "MLaaS: A cloud system for mobile micro learning in MOOC," in Proc. IEEE Int. Conf. Mobile Services, New York, NY, UAS, Jun. 2015, pp. 120–127.

[15] G. Sun, T. Cui, D. Xu, H. Chen, S. Chen, and J. Shen, "Assisting open education resource providers and instructors to deal with cold start problem in adaptive micro learning: A service oriented solution," in Proc. IEEE Int. Conf. Services Comput. (SCC), Honolulu, HI, USA, Jun. 2017, pp. 196–203.

[16] G. Sun, T. Cui, G. Beydoun, S. Chen, D. Xu, and J. Shen, "Organizing online computation for adaptive micro open education resource recom mendation," in Proc. 16th Int. Conf. Web Learn. (ICWL),CapeTown, South Africa, Sep. 2017, pp. 177–182.

[17] A. Corbi and D. Burgos, "OERaaS: Open educational resources as a service with the help of virtual containers," IEEE Latin Amer. Trans., vol. 14, no. 6, pp. 1–7, Aug. 2016.

[18] R. Hernandez, C. Gutl, and H. R. Amado-Salvatierra, "Cloud learning activities orchestration for MOOC environments," in Proc. 3rd Int. Workshop Learn. Technol. Educ. Cloud, Santiago, Chile, Sep. 2014, pp. 25–36.