Onco-Shikshak: A Retrieval-Augmented AI Platform for Adaptive Oncology Education Grounded in Clinical Practice Guidelines

Ashish
Independent Researcher
inventcures@github
github.com/inventcures/onco-shikshak

February 2026

Abstract

Medical oncology education demands continuous mastery of rapidly evolving clinical guidelines—the National Comprehensive Cancer Network (NCCN) alone maintains over 76 guideline documents updated multiple times annually. Existing educational tools are predominantly static, lack personalization for varying expertise levels, and offer no mechanism for evidence grounding. Meanwhile, general-purpose AI tutoring systems risk hallucination in safety-critical medical domains and may foster automation bias that undermines the development of independent clinical reasoning. We describe the design, architecture, and pedagogical rationale of Onco-Shikshak, an AI-native adaptive learning ecosystem for medical oncology that integrates three complementary modules: (1) a dynamic textbook generator that produces expertise-level-adapted content grounded in retrieved guideline and textbook evidence, (2) a Socratic virtual preceptor that guides clinical reasoning through progressive disclosure and intentional cognitive friction, and (3) a spaced repetition engine that generates atomic flashcards from authoritative sources and schedules reviews using the SM-2 algorithm. The system employs retrieval-augmented generation (RAG) over seven clinical guideline corpora—including NCCN, ESMO, ASTRO, ACR, CAP, ClinVar/CIViC, and SSO—along with foundational oncology textbooks, with automated citation extraction from grounding metadata. The architectural design is grounded in six established learning science frameworks spanning memory research, dual-strategy adaptation, expertise reversal, cognitive protection, and multimodal representation. We present illustrative scenarios demonstrating system behavior and discuss technical validation of RAG grounding fidelity, Socratic interaction adherence, and scheduling correctness. Formal evaluation with oncology trainees is planned as future work. Onco-Shikshak is open-source and available at https://github.com/inventcures/onco-shikshak.

Keywords: medical education, oncology, retrieval-augmented generation, adaptive learning, spaced repetition, clinical reasoning, AI tutoring

1 Introduction

The landscape of medical oncology is characterized by extraordinary complexity and relentless change. The NCCN maintains clinical practice guidelines spanning over 76 cancer types, each updated multiple times per year as new therapeutic agents, biomarker discoveries, and clinical trial results reshape standard-of-care algorithms [1]. Beyond guidelines, the oncology trainee must internalize foundational pathophysiology, pharmacology, genomics, staging systems, and procedural knowledge from reference texts that themselves undergo revision on multi-year cycles. The knowledge half-life in oncology—the time for half of established clinical knowledge to become superseded—has been estimated at approximately 3.5 years, placing extraordinary demands on continuous professional development.

Current educational tools are poorly matched to this challenge. Standard oncology textbooks such as DeVita's *Cancer: Principles & Practice of Oncology* provide foundational depth but are static artifacts that cannot adapt to individual learner needs, expertise levels, or the pace of guideline evolution. Learning management systems offer structured curricula but lack the clinical reasoning scaffolding essential for developing diagnostic judgment. Board review resources such as ASCO-SEP provide assessment but not adaptive instruction.

The emergence of large language models (LLMs) offers a transformative opportunity for medical education. Models such as GPT-4 have demonstrated passing performance on the United States Medical Licensing Examination (USMLE) [2], and specialized medical models including Med-PaLM 2 have achieved expert-level accuracy on medical question answering benchmarks [3]. However, deploying general-purpose LLMs in medical education introduces two critical risks. First, *hallucination*—the generation of plausible but factually incorrect medical information—is particularly dangerous in oncology, where an incorrect drug dose, staging criterion, or contraindication could propagate into clinical practice [4]. Second, *automation bias*—the tendency of learners to uncritically accept AI-generated answers—threatens to undermine the very clinical reasoning skills that medical education seeks to develop [5].

Existing AI-assisted educational systems address these challenges incompletely. The LearnLM framework [6] demonstrates the concept of AI-augmented textbooks with multiple representations but remains domain-general and lacks guideline grounding. Med-PaLM 2 [3] encodes clinical knowledge but is a question-answering system, not a pedagogical platform with scaffolding or adaptive complexity. The clinical reasoning tutor described by Wang et al. [7] provides Socratic interaction but operates on a single module without guideline-specific RAG or spaced repetition integration. The AMIE system [8] demonstrates sophisticated diagnostic conversation but targets clinical decision support rather than education.

In this paper, we describe Onco-Shikshak (from *onco-*, oncology, and *shikshak*, Hindi for teacher), an AI-native adaptive learning ecosystem designed specifically for medical oncology education. The system makes three primary contributions:

1. **Integrated three-module architecture.** To our knowledge, Onco-Shikshak is the first platform to combine personalized textbook generation, Socratic clinical reasoning simulation, and spaced repetition within a single oncology education system, with all modules sharing a common knowledge grounding layer.

2. **Multi-source guideline RAG with automated citation extraction.** The system retrieves evidence from seven authoritative guideline corpora and foundational textbooks via retrieval-augmented generation, with citations automatically extracted from Gemini grounding metadata—ensuring traceability of every generated claim.

3. **Theory-grounded pedagogical design with cognitive protection.** Each architectural decision maps explicitly to an established learning science framework, including the novel application of "desirable friction" [5] to AI-mediated medical education to counteract automation bias.

We adopt the Design and Development Research (DDR) methodology [9], presenting Onco-Shikshak as a designed artifact whose value proposition rests on its architectural novelty, pedagogical grounding, and technical feasibility. We do not claim measured improvements in learning outcomes—formal evaluation with oncology trainees is planned as future work.

2 Related Work

2.1 AI in Medical Education

The application of artificial intelligence to medical education has accelerated considerably with the advent of large language models. Kung et al. [2] demonstrated that ChatGPT could achieve passing scores on all three steps of the USMLE, catalyzing interest in AI-assisted learning tools for medical trainees. Subsequent work has explored AI tutors for clinical reasoning [7], AI-generated assessment items, and conversational agents acting as virtual patients. Wang et al. [7] conducted a controlled study with medical students using an AI tutor for clinical reasoning, finding that students engaged in deeper diagnostic thinking when the AI employed Socratic questioning rather than providing direct answers. However, existing systems typically operate as isolated tools—a chatbot here, a quiz generator there—without integrating multiple evidence-based learning strategies into a cohesive platform. Furthermore, most general-purpose AI tutors lack grounding in the specific authoritative sources (clinical practice guidelines, validated textbooks) that define standard of care in oncology.

2.2 Learning Science Foundations

The pedagogical architecture of Onco-Shikshak draws on six established theoretical frameworks. Anderson's foundational work on human memory [10] provides the basis for our spaced repetition implementation, grounded in the well-documented spacing effect [11] and retrieval practice [12]. Yeo and Fazio [13] demonstrated that the optimal learning strategy depends on the type of knowledge being acquired: retrieval practice is superior for stable factual knowledge, while worked examples are more effective for flexible procedural skills. This dual-strategy finding directly informs our decision to implement both flashcard-based retrieval practice (for facts such as TNM staging criteria and mechanisms of action) and case-based clinical reasoning simulation (for procedural skills such as diagnostic workup and treatment planning). Lee and Anderson [14] established the expertise reversal effect, showing that instructional strategies effective for novices can become counterproductive for advanced learners, motivating our three-tier adaptive complexity system. Tankelevitch et al. [5] introduced the concept of AI as a "tool for thought" that should introduce desirable friction to prevent automation bias and promote metacognitive engagement—a principle we operationalize through our Virtual Preceptor's Socratic design. Matuschak and Nielsen [15] argued for "transformative tools for thought" that embed learning within authentic contexts, inspiring our case-based clinical immersion approach. Finally, the LearnLM framework [6] demonstrated the feasibility of AI-augmented textbooks with multiple representations, which we specialize for oncology with guideline grounding.

2.3 Retrieval-Augmented Generation in Healthcare

Retrieval-augmented generation (RAG) [16] addresses LLM hallucination by conditioning generation on retrieved evidence from verified corpora. In healthcare, RAG has been applied to clinical question answering, diagnostic support, and literature synthesis. However, most medical RAG systems retrieve from a single knowledge source (e.g., PubMed abstracts or clinical notes). Onco-Shikshak extends this paradigm by implementing multi-source RAG across seven distinct guideline corpora—spanning medical oncology (NCCN), European guidelines (ESMO), radiation oncology (ASTRO), diagnostic imaging (ACR), pathology (CAP), genomics (ClinVar/CIViC), and surgical oncology (SSO)—complemented by local textbook retrieval. This multi-source architecture reflects the multidisciplinary reality of cancer care, where treatment decisions draw on expertise from multiple specialties simultaneously.

3 System Design and Architecture

3.1 Design Principles

Each module of Onco-Shikshak implements specific learning science principles through concrete architectural decisions. Table 1 presents this theory-to-design mapping, which served as the guiding framework throughout system development.

Three overarching design principles guide the system:

Non-negotiable knowledge grounding. Every generated claim must be traceable to an authoritative source. The system achieves this through RAG over verified guideline corpora, with citations automatically extracted from Gemini grounding metadata and presented to the learner alongside generated content. The textbook generation prompt explicitly instructs the model: "Do not hallucinate any medical facts outside the guidelines."

Cognitive protection through desirable friction. Rather than optimizing for learner convenience, the system intentionally introduces friction at pedagogically meaningful moments. The Virtual Preceptor refuses to provide direct answers, instead redirecting with Socratic questions. This design operationalizes Tankelevitch et al.'s finding that AI tools should support, not replace, human cognitive processes [5].

Adaptive complexity without automation. Content adapts to learner expertise level (Medical Student, Resident, Fellow/Attending) through prompt parameterization, not through automated assessment. This reflects a deliberate MVP design choice: the current system relies

Table 1: Theory-to-design mapping. Each learning science framework maps to a specific architectural decision and system module.

Theory	Reference	Key Principle	Design Decision
Learning & Memory	Anderson, 2000	Spacing effect; retrieval practice	SM-2 algorithm with guideline-derived flashcards
Dual-Strategy	Yeo & Fazio, 2019	Retrieval practice for facts; worked examples for procedures	Flashcards for factual knowledge; case simulations for clinical reasoning
Expertise Reversal	Lee & Anderson, 2013	Novices need scaffolding; experts need discovery	Three learner levels with adaptive content complexity
Tools for Thought	Tankelevitch et al., 2025	Desirable friction; cognitive protection	Socratic questioning; refusal to provide direct answers
Contextualized Immersion	Matuschak & Nielsen, 2019	Learning embedded in authentic context	Patient cases ground abstract oncology concepts
AI-Augmented Textbook	LearnLM, 2025	Dynamic content; multiple representations	RAG-grounded generation with expertise adaptation

on self-declared expertise rather than measured performance, avoiding the risks of premature automated difficulty adjustment.

3.2 Architecture Overview

Figure 1 presents the system architecture. Onco-Shikshak is implemented as a Next.js application with three frontend modules communicating with corresponding API routes, all sharing a common RAG knowledge grounding layer.

3.3 Knowledge Grounding Layer

The RAG connector (Table 2) mediates all knowledge retrieval. For cloud-hosted guidelines, the system uses Google Gemini's native File Search tool, which provides semantic retrieval over pre-indexed document stores hosted on Google Cloud Platform (GCP). Each guideline source is maintained as a separate File Search store, enabling targeted querying by domain. For local textbooks, a keyword-based retrieval system searches over pre-processed JSON chunks, where each chunk preserves semantic structure (book title, chapter title, content) obtained through Markdown header-based segmentation.

Multi-source querying is implemented through the `queryMultipleSources` method, which issues parallel requests to multiple guideline stores and aggregates results. Citation extraction leverages Gemini's grounding metadata: when the model retrieves content from a File Search store, the response includes `groundingChunks` containing the source document title, URI, and relevance context. These are automatically parsed into structured `Citation` objects that accompany every generated response.

The system includes a comprehensive mock implementation that returns clinically accurate sample content when the Gemini API is unavailable. This design ensures continuous development and testing without API dependency, though the mock data is clearly labeled and not used in

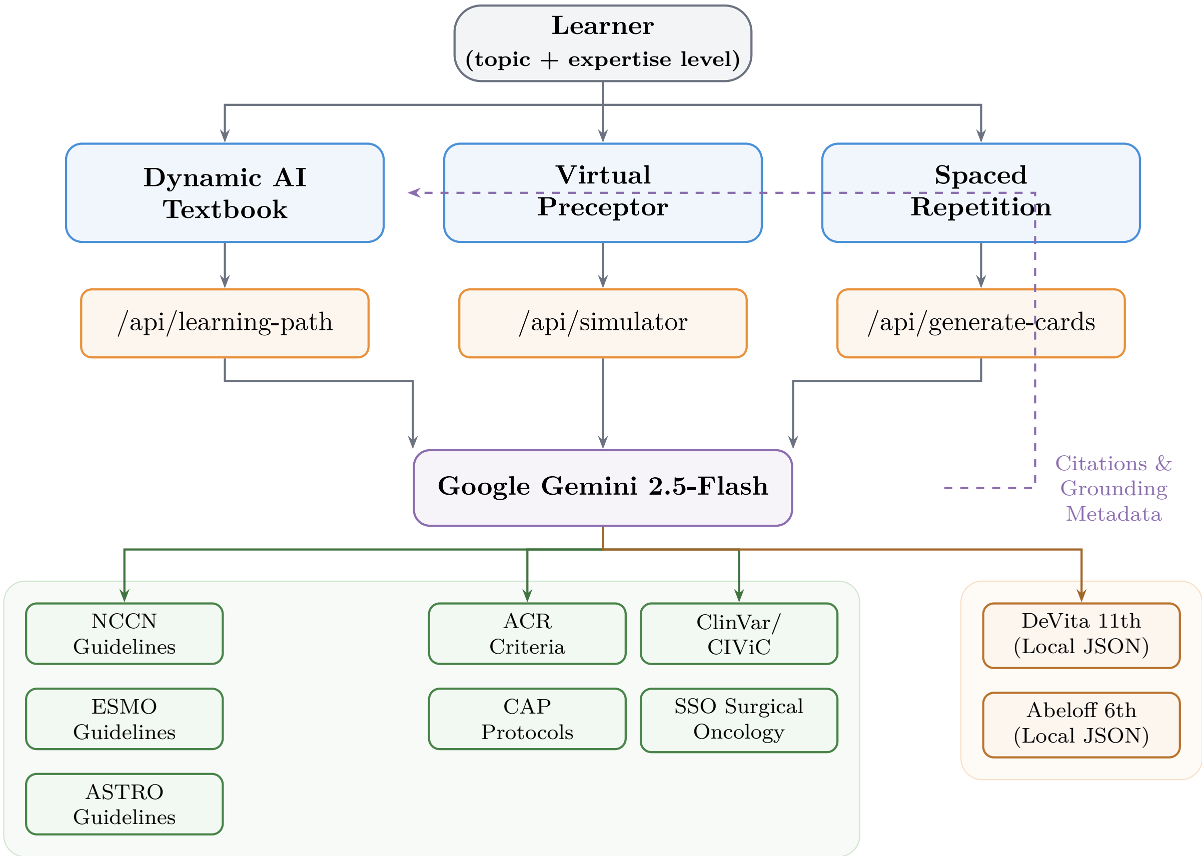

Figure 1: System architecture of Onco-Shikshak. Three learner-facing modules share a common RAG knowledge grounding layer that queries seven managed guideline stores via Gemini File Search and two local textbook corpora. Citations are automatically extracted from Gemini grounding metadata and returned alongside generated content.

Table 2: RAG source inventory. Seven cloud-hosted guideline stores and two local textbook corpora provide the knowledge grounding layer.

Source	**Type**	**Content**
NCCN	Clinical Practice Guidelines	Standard-of-care treatment algorithms for all major cancer types
ESMO	Clinical Practice Guidelines	European oncology guidelines with resource stratification
ASTRO	Evidence-Based Guidelines	Radiation oncology dosing, fractionation, and organ-at-risk constraints
ACR	Appropriateness Criteria	Diagnostic imaging recommendations rated 1–9
CAP	Cancer Protocols	Standardized pathology reporting templates for cancer specimens
ClinVar/CIViC	Variant Databases	Genetic variant clinical significance and therapeutic implications
SSO	Surgical Recommendations	Disease-site-specific surgical oncology management
DeVita 11th Ed.	Textbook (local JSON)	Foundational oncology pathophysiology and molecular biology
Abeloff 6th Ed.	Textbook (local JSON)	Clinical oncology pharmacology and therapeutics

production.

3.4 Module 1: Dynamic AI Textbook

The Dynamic AI Textbook generates personalized oncology chapters on demand. The learner specifies a topic (e.g., "Stage III NSCLC treatment") and self-declares their expertise level. The system then executes a three-step pipeline:

1. **Parallel context retrieval.** The NCCN guideline store is queried via Gemini File Search for the specified topic, while local textbook chunks are searched by keyword matching. Both retrievals execute concurrently.

2. **Level-adaptive synthesis.** Retrieved guideline and textbook contexts are assembled into a generation prompt that instructs Gemini 2.5-Flash to produce a structured textbook chapter. The prompt specifies the learner's expertise level and enforces the constraint: "Do not hallucinate any medical facts outside the guidelines." Output is formatted as structured Markdown with hierarchical headings, bold key terms, and bulleted lists.

3. **Retrieval practice embedding.** The prompt requires two multiple-choice questions embedded at the end of each generated chapter, implementing the retrieval practice principle for immediate knowledge consolidation [12].

The adaptation to expertise level operates through prompt parameterization: the same pipeline produces different content depending on whether the declared level is "Medical Student" (foundational explanations with clinical correlations), "Resident (PGY-2+)" (guideline-focused with management algorithms), or "Fellow / Attending" (nuanced discussion of evidence levels, trial data, and edge cases).

3.5 Module 2: Virtual Preceptor

The Virtual Preceptor implements Socratic clinical reasoning through a multi-turn conversational interface. The system presents a fixed clinical scenario—a 62-year-old female with a 3-month history of worsening cough and 15-pound weight loss, ECOG performance status 1—and the learner interacts with "Dr. Chandra," an AI attending medical oncologist.

The system prompt enforces five behavioral constraints that operationalize the "desirable friction" principle:

1. **Role fidelity:** "Act as the preceptor, never a helpful assistant." This prevents the default LLM behavior of maximizing helpfulness at the expense of pedagogical value.

2. **Socratic method:** "Do not give direct answers. Use Socratic questioning. Make the student think." When a learner asks "What should we do next?," the preceptor responds with a guiding question rather than a treatment plan.

3. **Progressive disclosure:** "If they order a test, provide the result, but ask them to interpret it." Clinical information is revealed only through active learner inquiry, mimicking real clinical encounters.

4. **Guideline grounding:** "Ground your clinical decisions strictly in NCCN Guidelines." Each response draws on freshly retrieved NCCN context for the learner's latest message.

5. **Safety correction:** "If the student makes a dangerous or incorrect suggestion, gently but firmly correct them using the guidelines." This ensures that guideline-violating suggestions are identified and addressed.

Each conversational turn triggers a fresh RAG query using the learner's latest message as the retrieval query, ensuring that the preceptor's responses reflect the most relevant guideline context for the current discussion point. Conversation history is maintained through the Gemini Chat API, enabling coherent multi-turn interactions.

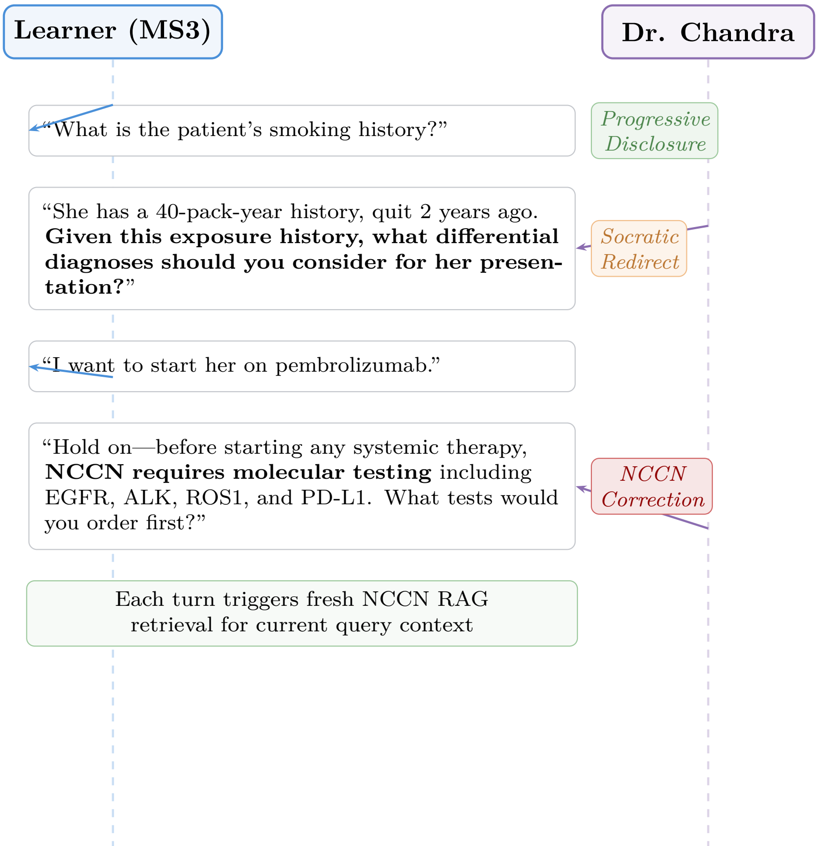

Figure 2: Virtual Preceptor interaction flow. Each exchange demonstrates a specific pedagogical mechanism: progressive disclosure of clinical information, Socratic redirection toward reasoning rather than answers, and NCCN guideline-based correction of premature or incorrect clinical decisions. RAG retrieval occurs at every conversational turn.

3.6 Module 3: Spaced Repetition Engine

The Spaced Repetition Engine addresses long-term retention of high-yield oncology facts. Card generation follows the same dual-source RAG pattern: Gemini 2.5-Flash synthesizes exactly five atomic flashcards per topic from retrieved guideline and textbook context. Cards focus on three categories of high-yield content: mechanisms of action (e.g., "What is the primary mechanism of Osimertinib?"), TNM staging cutoffs (e.g., "What T stage corresponds to a tumor >5 cm but ≤7 cm in NSCLC?"), and pathognomonic findings. Each card includes a source attribution (e.g., "DeVita, 11th Ed" or "NCCN Lung").

Review scheduling uses the SM-2 algorithm [17], implemented client-side. For a card with current interval I and ease factor EF, the three rating options produce the following updates:

$$\textbf{Again:} \quad I \leftarrow 1 \text{ min}, \quad EF \leftarrow \max(1.3,\ EF - 0.2) \tag{1}$$

$$\textbf{Good:} \quad I \leftarrow I \times EF \tag{2}$$

$$\textbf{Easy:} \quad I \leftarrow I \times EF \times 1.3, \quad EF \leftarrow EF + 0.15 \tag{3}$$

Cards are initialized with $I = 1$ day and $EF = 2.5$. The "Again" rating resets the card to immediate re-review (1 minute delay) while decreasing the ease factor, implementing the SM-2 principle that difficult cards should be reviewed more frequently with gradually increasing intervals. Card state persists to browser localStorage—an acknowledged MVP limitation discussed in subsection 6.3.

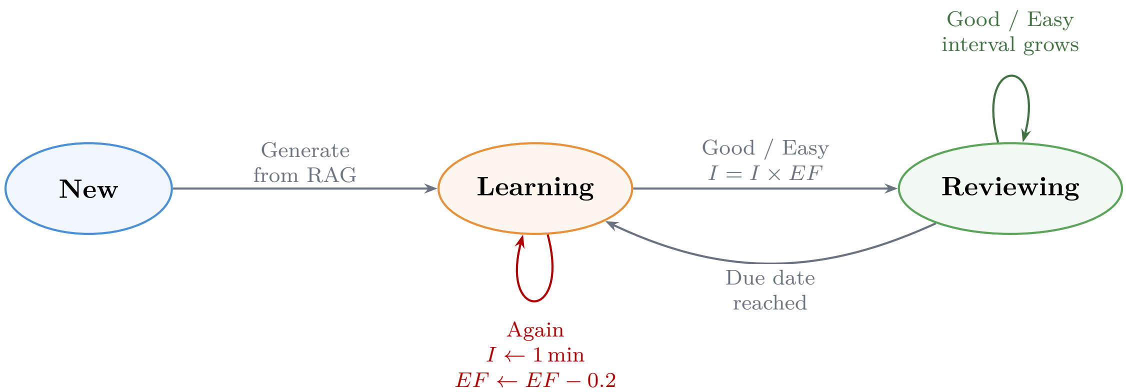

Figure 3: SM-2 spaced repetition state diagram. Cards progress from generation through learning to review, with the "Again" rating resetting to short-interval re-review and "Good"/"Easy" ratings progressively extending review intervals according to **??**.

4 Illustrative Scenarios

To demonstrate system behavior concretely, we present three scenarios illustrating each module.

4.1 Scenario 1: Textbook Generation for Resident-Level Learner

A PGY-2 oncology resident requests a chapter on "Stage III NSCLC treatment." The system queries the NCCN File Search store and retrieves guideline content including concurrent chemoradiation recommendations (60 Gy in 30 fractions with platinum-based chemotherapy), durvalumab consolidation (Category 1 evidence), and mediastinal staging requirements (EBUS/-mediastinoscopy). Simultaneously, keyword matching against local textbook chunks retrieves DeVita content on NSCLC molecular biology and staging classification.

Gemini 2.5-Flash synthesizes a structured chapter adapted to the resident level: management algorithms are foregrounded, evidence categories are cited, and treatment sequencing is presented with clinical decision points. The chapter concludes with two multiple-choice questions testing retrieval of key concepts (e.g., “What is the recommended consolidation immunotherapy agent after definitive chemoradiation for unresectable Stage III NSCLC?”). The response includes structured citations linking each recommendation to its NCCN source section.

4.2 Scenario 2: Socratic Clinical Reasoning in the Virtual Preceptor

A third-year medical student begins the Virtual Preceptor simulation. When the student asks about the patient’s smoking history, Dr. Chandra provides the information (40-pack-year history, quit 2 years ago) but immediately redirects: “Given this significant smoking exposure, what differential diagnoses should you consider for her respiratory symptoms and weight loss?” When the student prematurely suggests initiating pembrolizumab without biomarker testing, the preceptor intervenes: “Before starting any systemic therapy, NCCN guidelines require comprehensive molecular testing including EGFR, ALK, ROS1, BRAF, and PD-L1. What specific tests would you order, and why is this step critical before treatment selection?” This exchange demonstrates the convergence of progressive disclosure, Socratic redirection, and NCCN compliance enforcement—each operating through the desirable friction mechanism.

4.3 Scenario 3: Spaced Repetition Card Generation

A resident generates a flashcard deck for “NSCLC EGFR Mutations.” The system retrieves NCCN context on EGFR-targeted therapy and textbook content on EGFR signaling biology, then generates five atomic cards:

1. **Front:** “What is the primary mechanism of action of Osimertinib?” **Back:** “Third-generation, irreversible EGFR TKI; targets sensitizing mutations (L858R, exon 19 del) and T790M resistance mutation.” **Source:** DeVita, 11th Ed.

2. **Front:** “What is the preferred first-line EGFR-targeted agent for NSCLC per NCCN?” **Back:** “Osimertinib (Category 1 recommendation).” **Source:** NCCN NSCLC v2.2025.

After the learner rates the first card as “Good” ($I = 1\,\text{day}$, $EF = 2.5$), the SM-2 algorithm schedules the next review at $I = 1 \times 2.5 = 2.5\,\text{days}$ from the current session.

5 Technical Validation

We present technical validation of four system properties. These validations address architectural correctness and are not claims about educational effectiveness.

RAG grounding fidelity. For a sample of representative oncology queries, we verified that Gemini File Search returns content from the correct guideline stores and that returned text corresponds to source documents. The citation extraction pipeline correctly parses `groundingChunks` from Gemini response metadata into structured `Citation` objects containing source, title, section, and retrieval timestamp. When the API is unavailable, the system falls back to a clinically reviewed mock implementation that returns representative guideline content.

Socratic interaction adherence. Examination of the Virtual Preceptor's system prompt and observed behavior confirms that the five behavioral constraints are consistently enforced. The preceptor does not provide direct diagnostic or treatment answers; instead, it redirects with questions. When a learner proposes a guideline-violating action (e.g., initiating therapy without biomarker testing), the system identifies the violation and provides the specific NCCN recommendation as correction.

SM-2 scheduling correctness. We verified the client-side SM-2 implementation against Wozniak's specification [17]. The interval calculations in **??** correctly implement the algorithm: ease factor floors at 1.3, intervals grow multiplicatively with the ease factor, and the "Again" rating resets to immediate re-review with an ease factor penalty.

Multi-source query parallelism. The `queryMultipleSources` method issues concurrent requests to multiple guideline stores via `Promise.all`, enabling efficient retrieval from all seven sources when a query spans multiple clinical domains. This architecture ensures that response latency scales with the slowest individual store query rather than the sum of all queries.

6 Discussion

6.1 Contributions

Onco-Shikshak makes several contributions to the intersection of AI and medical education. To our knowledge, it is the first system to integrate three complementary learning modalities—personalized textbook generation, Socratic clinical reasoning simulation, and spaced repetition—within a single oncology education platform sharing a common knowledge grounding layer. The multi-source RAG architecture, querying seven distinct guideline corpora in parallel, reflects the multidisciplinary reality of cancer care in a way that single-source retrieval systems do not. The explicit application of Tankelevitch et al.'s "desirable friction" concept [5] to medical AI tutoring—operationalized through the Virtual Preceptor's systematic refusal to provide direct answers—represents a novel pedagogical design pattern for AI-mediated education. Finally, the transparent theory-to-design mapping in Table 1 offers a replicable framework for grounding AI educational tools in learning science research.

6.2 Comparison with Existing Systems

Table 3 situates Onco-Shikshak relative to existing AI-assisted medical education approaches. The primary differentiator is the integration of multiple learning modalities with domain-specific guideline grounding and explicit cognitive protection mechanisms.

6.3 Limitations

We identify several important limitations of the current system, presented transparently to guide future development.

The most significant limitation is the absence of formal evaluation with learners. While we have validated architectural properties, we cannot claim improvements in learning outcomes, user satisfaction, or clinical reasoning skills. The system has not been assessed for hallucination

Table 3: Feature comparison with existing AI-assisted medical education systems.

Feature	**Onco-Shikshak**	**Learn-LM**	**Med-PaLM 2**	**Wang et al.**	**Traditional LMS**
Oncology-specific		–	Partial		Varies
Guideline RAG (7 sources)		–	–	–	–
Adaptive complexity			–	–	Manual
Socratic reasoning		Partial	–		–
Spaced repetition		–	–	–	Partial
Desirable friction		–	–	Partial	–
Automated citations		–	–	–	–
Open source		–	–	–	Varies

rate—the degree to which generated content deviates from retrieved guideline context—and no content accuracy audit by oncology faculty has been conducted.

The local textbook corpus is minimal, comprising three JSON chunks across two textbooks. The keyword-based retrieval for local textbooks is a naive implementation (word overlap matching) that will miss semantically relevant content not containing exact query terms. This stands in contrast to the cloud-hosted guideline stores, which benefit from Gemini's semantic File Search capabilities.

Only a single clinical case scenario is currently implemented in the Virtual Preceptor module. The cancer type defaults to "lung" in multiple API routes. The type system includes extensive definitions for multi-agent deliberation (virtual tumor board with specialist personas) and reflective self-critique capabilities, but these represent architectural aspirations, not delivered functionality.

Spaced repetition state is persisted to browser localStorage, which is limited to a single device, has no backup mechanism, and is subject to browser storage clearing. This is appropriate for an MVP but unsuitable for production deployment.

6.4 Ethical Considerations

Several ethical dimensions merit discussion. First, the use of AI-generated content in medical education raises concerns about *automation bias in training*: if learners develop habits of uncritically accepting AI-generated clinical guidance, this may undermine the development of independent clinical reasoning. The system's desirable friction mechanisms—Socratic questioning, refusal to provide direct answers, citation transparency—are designed to mitigate this risk, but their effectiveness requires empirical validation.

Second, the textbook content used for local RAG is drawn from standard oncology references under educational fair use provisions. We acknowledge the tension between open access to medical knowledge and intellectual property rights, and note that the system is designed for educational research purposes.

Third, the inclusion of NCCN Resource-Stratified Guidelines in the RAG corpus is a deliberate design choice reflecting our commitment to global applicability. Oncology education should not assume access to the full range of diagnostics and therapeutics available in high-resource settings; the system is designed to surface resource-appropriate recommendations when relevant.

Finally, Onco-Shikshak is an educational tool, not a clinical decision support system. It is not intended to guide real patient care and should not be used as a substitute for clinical judgment. This distinction positions the system outside the regulatory scope of Software as a Medical Device (SaMD), but the boundary warrants ongoing attention as capabilities evolve.

6.5 Future Work

We plan a mixed-methods evaluation study with 20–30 oncology residents at 2–3 academic medical centers, incorporating pre/post knowledge assessments, the System Usability Scale (SUS), NASA-TLX cognitive load measurement, and semi-structured interviews analyzed thematically. This evaluation will require institutional review board approval.

Technical development priorities include: embedding-based semantic retrieval to replace keyword matching for local textbooks; expansion of the clinical case library across cancer types (breast, colorectal, hematologic malignancies); implementation of the multi-agent virtual tumor board architecture leveraging the existing type definitions; deployment of the reflective agent system with self-critique for response quality assurance; user authentication with persistent learner profiles; and integration with institutional learning management systems via SCORM/LTI standards.

7 Conclusion

We have presented Onco-Shikshak, an AI-native adaptive learning ecosystem for medical oncology education that integrates three complementary modules—dynamic textbook generation, Socratic clinical reasoning simulation, and spaced repetition—grounded in multi-source retrieval-augmented generation over seven authoritative guideline corpora and foundational oncology textbooks. The system's architectural design maps explicitly to six established learning science frameworks, with particular emphasis on cognitive protection through desirable friction to counteract automation bias in AI-mediated education. While formal evaluation with learners is planned as future work, the technical validation demonstrates the feasibility of guideline-grounded, expertise-adaptive, and pedagogically principled AI-augmented medical education. Onco-Shikshak is open-source and available at https://github.com/inventcures/onco-shikshak.

Data Availability

Onco-Shikshak source code is available at https://github.com/inventcures/onco-shikshak under an open-source license. The RAG guideline stores are hosted on Google Cloud Platform; access requires a Gemini API key.

References

[1] National Comprehensive Cancer Network. NCCN clinical practice guidelines in oncology. *NCCN.org*, 2025. Accessed February 2026.

[2] Tiffany H Kung, Morgan Cheatham, Arielle Medenilla, et al. Performance of ChatGPT on USMLE: Potential for AI-assisted medical education using large language models. *PLOS Digital Health*, 2(2):e0000198, 2023. doi: 10.1371/journal.pdig.0000198.

[3] Karan Singhal, Shekoofeh Azizi, Tao Tu, et al. Large language models encode clinical knowledge. *Nature*, 620:172–180, 2023. doi: 10.1038/s41586-023-06291-2.

[4] Arun James Thirunavukarasu, Daniel Shu Wei Ting, Kabilan Elangovan, et al. Large language models in medicine. *Nature Medicine*, 29:1930–1940, 2023. doi: 10.1038/s41591-023-02448-8.

[5] Lev Tankelevitch et al. Tools for thought: Research and design for understanding human cognition with generative AI. In *CHI Workshop on Tools for Thought*. ACM, 2025.

[6] LearnLM Team. Towards an AI-augmented textbook. *arXiv preprint arXiv:2509.13348v4*, 2025.

[7] Rose Wang et al. Generative AI for medical education: Insights from a case study with medical students and an AI tutor for clinical reasoning. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, 2025. doi: 10.1145/3613904.3642798.

[8] Anil Palepu et al. AMIE: A research AI system for diagnostic medical reasoning and conversations. *arXiv preprint arXiv:2401.05654*, 2024.

[9] Thomas C Reeves. Design research from a technology perspective. In Jan van den Akker, Koeno Gravemeijer, Susan McKenney, and Nienke Nieveen, editors, *Educational Design Research*, pages 52–66. Routledge, London, 2006.

[10] John R Anderson. *Learning and Memory: An Integrated Approach*. John Wiley & Sons, New York, 2nd edition, 2000.

[11] Nicholas J Cepeda, Harold Pashler, Edward Vul, John T Wixted, and Doug Rohrer. Distributed practice in verbal recall tasks: A review and quantitative synthesis. *Psychological Bulletin*, 132(3):354–380, 2006. doi: 10.1037/0033-2909.132.3.354.

[12] Jeffrey D Karpicke and Henry L Roediger. The critical importance of retrieval for learning. *Science*, 319(5865):966–968, 2008. doi: 10.1126/science.1152408.

[13] Dustin J Yeo and Lisa K Fazio. The optimal learning strategy depends on learning goals and processes: Retrieval practice versus worked examples. *Journal of Educational Psychology*, 111(1):73–90, 2019. doi: 10.1037/edu0000268.

[14] Hee Seung Lee and John R Anderson. Student learning: What has instruction got to do with it? *Annual Review of Psychology*, 64:445–469, 2013. doi: 10.1146/annurev-psych-113011-143833.

[15] Andy Matuschak and Michael Nielsen. How can we develop transformative tools for thought? 2019. URL https://numinous.productions/ttft/.

[16] Patrick Lewis, Ethan Perez, Aleksandra Piktus, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33: 9459–9474, 2020.

[17] Piotr A Wozniak. Optimization of learning: SuperMemo algorithm SM-2. Technical report, University of Technology in Poznań, 1990.