Reimagining Textbook Learning: An Interactive AI Tutor Approach Using Retrieval-Augmented Generation

Anonymous ACL submission

Abstract

We present an Interactive AI Tutor that transforms static textbook chapters into adaptive and immersive learning experiences. Built around a reproducible Textbook to Interaction (T2I) pipeline, the system converts open educational resources into storytelling modules, business cases, and a diverse set of interactive challenges grounded in retrieval-augmented generation. A chapter-aligned dataset generated with DeepSeek and refined through manual validation underpins the tutor's content creation, while LangChain pipelines, FAISS retrieval, and Google APIs support retrieval and delivery. The system is designed for resource-constrained environments through a lightweight-first deployment strategy, combining four-bit quantization, retrieval caching, and a model cascade from Mistral 7B to Falcon RW 1B. In a mixed-methods user study with 62 participants (57 complete pre/post), we observe significant improvements on short-term self-report proxies (familiarity, comfort), along with high usability and engagement. These results demonstrate the feasibility of delivering retrieval-grounded and adaptive textbook learning experiences in widely accessible, low-resource settings, paving the way for scalable, personalized education built on open resources.

1 Introduction

Traditional textbook-based learning often fails to meet the needs of today's learners (Woo, 2009). Long, static readings can reduce motivation and limit conceptual understanding, particularly when they lack opportunities for interaction, feedback, and personalization. Even highly motivated students may struggle to engage deeply with dense chapters without guidance or adaptive support, slowing progress and hindering retention.

While artificial intelligence has shown promise in education (Woo, 2009), many existing tools focus narrowly on generating question–answer pairs or producing summaries. Such systems often overlook richer forms of engagement for example, narrative storytelling, case-based reasoning, or gamified challenges and rarely integrate seamlessly with Open Educational Resources (OER)(OpenStax, 2023). Moreover, they frequently depend on substantial computational budgets, limiting accessibility and scalability in real-world classrooms. Recent systems such as Newtbot (Lieb and Goel, 2024) and Ruffle&Riley (Schmucker et al., 2023) illustrate the potential of LLM-powered tutoring, but they typically emphasize a single interaction mode (e.g., conversational Q&A) and require substantial computational resources, limiting their scalability in classroom settings. This leaves a gap for modular, multi-mode systems that operate effectively under constrained environments while directly leveraging OER (Lewis et al., 2021; Qin et al., 2023).

This work introduces an *Interactive AI Tutor* that reimagines static textbook chapters as dynamic, learner-driven learning flows. At its core is a reproducible *Textbook-to-Interaction (T2I)* pipeline that transforms structured textbook content into storytelling modules, business cases, and a diverse set of interactive challenges grounded in retrieval-augmented generation (RAG) (Lewis et al., 2021). A curated, chapter-aligned dataset generated with DeepSeek (AI, 2023a) underpins content creation, while LangChain pipelines (Chase, 2022), FAISS retrieval (Johnson et al., 2021), and a lightweight model cascade enable retrieval fidelity and responsiveness. The system targets deployment in computationally limited environments via a *lightweight-first* strategy combining 4-bit quantization, retrieval caching, and a cascade from Mistral-7B (AI, 2023b) to Falcon-RW-1B (Institute, 2023). We preview key prompting patterns, retrieval flow, and generation safeguards (detailed in §3 and §3.4).

This study is guided by the following research questions:

1. How effectively can RAG with DeepSeek-generated drafts transform textbook content into coherent question-answer pairs and summaries?

2. How do interactive learning modes – storytelling, business case simulations, and gamified challenges – influence learner engagement, perception, and short-term outcomes within the tutor?

3. Is it feasible to deploy an AI-powered tutoring system in low-resource environments while maintaining usability, responsiveness, and adaptability?

To address these questions, we present an Interactive AI Tutor built with lightweight open-source models, modular learning modes, and retrieval grounding. Our mixed-methods evaluation (N=62; 57 complete pre/post) examines short-term learning proxies and user experience. The following sections describe the system architecture, dataset creation, deployment strategy, and interactive modules, followed by an evaluation of effectiveness and limitations.

Key Contributions and How We Differ from Prior Work Compared with existing LLM tutors that primarily rely on dialogue and require heavy infrastructure, our work advances the state of the art through the following contributions:

- **Textbook-to-Interaction (T2I) pipeline.** A reproducible pipeline that converts chapter-structured OER into three instructional formats such as storytelling, business cases, and six interactive challenge types, all aligned to the source text and checked for fidelity.

- **Adaptive learning foundation.** An initial framework that uses pre- and post-assessments to analyze short-term learning proxies and to inform future adaptive sequencing across modes.

- **Lightweight-first deployment.** A recipe for low-resource settings (e.g., Google Colab) combining 4-bit quantization, retrieval caching, and a Mistral-7B→Falcon-RW-1B cascade to meet latency/memory targets.

- **Released artifacts.** A validated, chapter-aligned dataset and IRB-approved instruments (surveys, protocol) to enable replication and extension.

Overall, this work contributes a transparent and accessible tutoring framework that demonstrates how static textbook content can be transformed into interactive learning experiences under realistic compute constraints. In a mixed-methods evaluation, we observe measurable improvements on short-term learning proxies and high usability. We conclude by discussing limitations and future directions toward adaptive, large-scale deployments.

2 Related Work

Intelligent tutoring systems (ITS) have long aimed to provide adaptive, personalized learning experiences (Woo, 2009). Recent advances in large language models (LLMs) have extended these capabilities, enabling conversational tutoring, adaptive feedback, and dynamic content generation. However, most existing systems focus on a single mode of interaction often dialogue and depend on large, proprietary models that are costly to deploy at scale. Moreover, many omit direct integration with open educational resources (OER) or fail to release datasets and instruments for replication.

LLM-powered tutoring systems. Early conversational tutors such as **Newtbot** (Lieb and Goel, 2024) and **Ruffle&Riley** (Schmucker et al., 2023) demonstrate how LLMs can support learning through dialogue and adaptive question answering. While they provide valuable design insights, their interaction modes remain narrow, and they require substantial compute resources. **Khanmigo** (Khan Academy Labs, 2023) illustrates the potential of large-scale proprietary deployments, but details on artifacts, datasets, or chapter-level alignment are limited. More recent work explores RAG integration: **Slade** (Slade et al., 2024) and **Nemeth** (Németh et al., 2024) combine retrieval grounding with tutoring support in coding and statistics courses, reporting usability gains but offering few details on multi-mode orchestration or resource-constrained deployment.

Interactive textbooks and retrieval-augmented models. Before the rise of LLMs, **Inquire Biology** (Chaudhri et al., 2013) illustrated the value of structured, interactive textbooks built on curated knowledge bases. Although not model-driven, it underscored the importance of aligning instruction with textbook content. More recent approaches extend retrieval beyond text: **RAMM** (Mialon et al., 2023) introduces

System / Paper	Modes	OER	Compute	Artifacts	Eval.	Notes
Newtbot (Lieb and Goel, 2024)	Dialogue	○	Heavy / Cloud	○	Usability	Secondary physics
Ruffle&Riley (Schmucker et al., 2023)	Dialogue	○	Heavy / Cloud	○	Mixed-methods	Design insights
Khanmigo (Khan Academy Labs, 2023)	Dialogue + tools	○	Heavy / Proprietary	○	Case studies	Large-scale platform
Slade et al. (Slade et al., 2024)	Dialogue + RAG	○	Mixed	Code	Usability	Coding courses
Németh et al. (Németh et al., 2024)	Dialogue + RAG	○	Mixed	○	Pilot	Statistics education
Inquire Biology (Chaudhri et al., 2013)	QA over KB	(non-LLM)	Lightweight	○	Classroom	Pre-LLM interactive textbook
RAMM (Mialon et al., 2023)	Multimodal gen.	○	Heavy / Research	Code	Benchmarks	Not education-specific
This work (T2I)	**Story, Cases, 6 Challenges + RAG**	**(OER chapters)**	**Lightweight (4-bit, cascade)**	**Dataset + Instr.**	**Pre/Post + UX**	**Colab-class deployment**

Table 1: Comparison to representative tutoring and RAG-in-education systems. "OER" indicates direct alignment with open educational resources. "Artifacts" include code, datasets, and/or instruments.

retrieval-augmented multimodal models and **Lewis et al.** (Lewis et al., 2021) show how retrieval-augmented generation (RAG) enhances factual accuracy. Yet, these works are not tailored to educational pipelines, nor do they integrate storytelling, challenge-based learning, or business scenarios tied to chapter-level OER.

Positioning and contributions. Prior work establishes the promise of LLM-based tutors and retrieval grounding but leaves key gaps in scope, reproducibility, and deployment feasibility. Most systems focus narrowly on dialogue-based tutoring, rely on heavy infrastructure, omit open datasets and instruments, or lack chapter-level alignment with OER. Few combine multiple instructional modes such as storytelling, scenario-based reasoning, and gamified challenges within a single reproducible pipeline.

How our work extends this landscape.

1. **Chapter-aligned multi-mode pipeline:** We present a reproducible *Textbook-to-Interaction (T2I)* pipeline that transforms OER chapters into storytelling, business scenarios, and six types of interactive challenges, going beyond dialogue-only interaction.

2. **Lightweight-first deployment:** We demonstrate deployment in constrained environments (e.g., Google Colab) using 4-bit quantization, retrieval caching, and a Mistral-7B→Falcon-RW-1B cascade to achieve low latency and memory efficiency.

3. **Released artifacts and validated study:** We release a chapter-aligned dataset and IRB-approved instruments, and we evaluate our system through a mixed-methods user study (N=62; 57 complete pre/post) under realistic resource conditions.

3 Methodology

We designed the tutor as a modular system that delivers **personalized, chapter-specific learning** from the OpenStax *Workplace Software and Skills* textbook. The system runs in Google Colab and covers content extraction, question–answer generation, human validation, semantic indexing, model integration, and adaptive interaction modes. Figure 1 illustrates the full Textbook-to-Interaction (T2I) pipeline, formalized in Algorithm B.

3.1 Data Preparation and Pipeline Overview

We extracted chapters from the OpenStax *Workplace Software and Skills* textbook (OpenStax, 2023) using PyMuPDF (Team, 2023), segmented them by section headers, and removed non-instructional text. Using DeepSeek Coder-6.7B (AI, 2024) with structured prompts, we generated up to five question–answer (QA) pairs and one summary per chapter, which were manually reviewed for accuracy and alignment (Ji et al., 2023). The curated dataset (`Merged_Chapter_Dataset.csv`) underpins all learning activities.

The Textbook-to-Interaction (T2I) pipeline integrates four key layers: (1) **Retrieval**, where QA pairs and summaries are embedded with MiniLM-L6-v2 (Wang et al., 2020)and stored in a FAISS vector index (Johnson et al., 2021) for semantic search; (2) **Modeling**, where the system uses Mistral-7B (AI, 2023b) on high-resource hardware or Falcon-RW-1B (Institute, 2023) with 4-bit quantization (Dettmers et al., 2022) in low-resource settings; (3) **Interaction**, which lets learners select chapters and learning modes; and (4) **Tracking**, which records XP, time, and correctness metrics and visualizes progress. An anonymized repository containing the curated dataset and IRB instruments is provided in the supplementary material; links are masked to preserve double-blind review.

3.2 Interactive Learning Modes

The curated dataset supports three instructional formats: (1) **Storytelling** (sma, 2023), which transforms summaries into narrative learning modules; (2) **Business Scenarios** (wri, 2023), which map key concepts to real-world case studies and industry roles; and (3) **Challenges**, which include six

activity types — flashcards, multiple choice, fill-in-the-blank, matching, timed questions, and hint-based scenarios — with distractors generated from validated content (Dicheva et al., 2015). All modes contribute to XP-based progression and log engagement metrics.

3.3 Deployment Environment

The public-facing prototype runs on Google Colab (Research, 2023) using CPU or entry-tier GPUs, and all datasets, templates, and utilities are released with pinned dependencies for reproducibility(see Appendix C).

3.4 Prompting and Retrieval Details

Models The public notebook uses tiiuae/falcon-rw-1b (Institute, 2023)via the Hugging Face `pipeline("text-generation")` API with default decoding parameters. Development runs used `mistralai/Mistral-7B-v0.1` (AI, 2023b) with 4-bit quantization via BitsAndBytes (Dettmers et al., 2022) on high-resource hardware. Full prompt templates are in Appendix C, Table 5; guardrails and decoding defaults are in Table 6.

Prompting & guardrails We use three templates (QA, summary, storytelling) with decoding defaults (Table 5). At runtime, a system instruction enforces grounding to retrieved spans; if retrieval is empty or below a similarity threshold, the assistant abstains and surfaces a hint (Table 6). We cap generations to 256 tokens and ban external web content unless explicitly whitelisted by the instructor mode.

Retrieval QA pairs and summaries are embedded using `MiniLM-L6-v2` Retrieval is handled via LangChain's (Chase, 2022) `ConversationalRetrievalChain` with conversation buffer memory. Default top-k search is used without re-ranking, and responses are drawn directly from retrieved content. If retrieval returns no relevant text, the system abstains from answering, consistent with the abstention rule in Table 6.

Overall, the T2I pipeline integrates lightweight retrieval-augmented generation, curated prompts, and modular learning modes to transform static textbook content into dynamic, chapter-specific experiences. The Algorithm B summarizes the complete process.

[0]See Table 2 for a summary of all challenge modes and their learning goals.

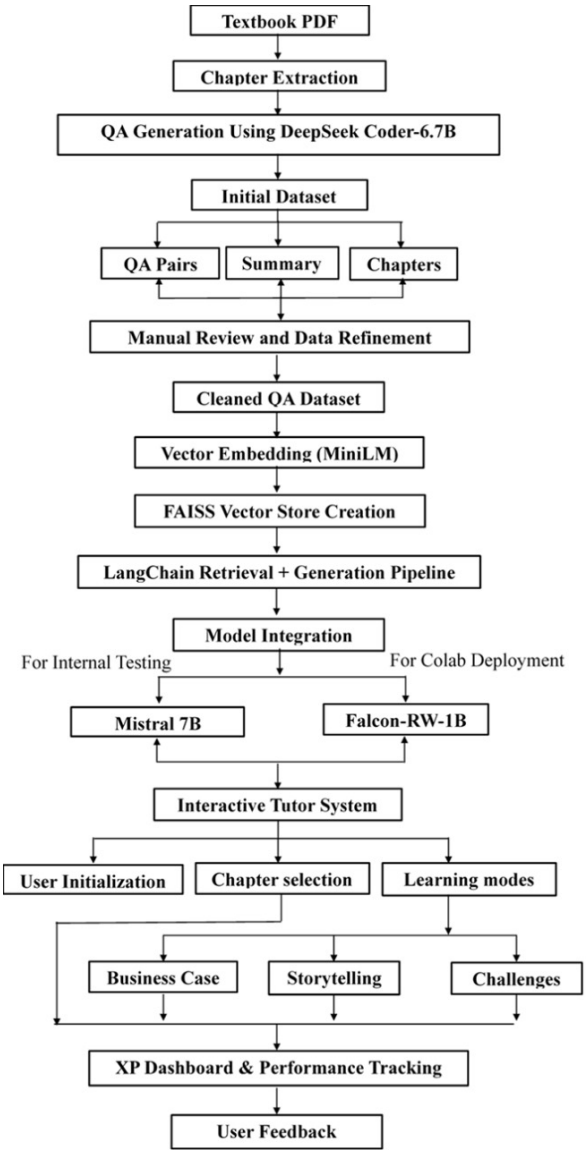

Figure 1: Pipeline overview of the AI Tutor system. Arrows indicate the sequential flow of data and control from textbook extraction through retrieval, generation, and delivery of interactive learning modes.

3.5 Baseline & Ablation Design (Summary)

To contextualize our system and address reviewer requests, we conducted three lightweight baseline and ablation studies (see Appendix D) sfor full protocols and tables). (1) **RAG ON vs. OFF:** For 40 textbook-aligned questions, we compared outputs with and without retrieval grounding, scored on factuality (0–2), grounding (0–2), and clarity (0–1) with adjudicated ratings and Cohen's κ. (2) **Single vs. multi-mode:** In a within-subjects design ($n \approx 20$), learners completed MCQ-only and MCQ+Story+Scenario conditions followed by a 6–8 item concept quiz; pre→post changes were analyzed with Wilcoxon tests, effect sizes (r), and

95% CIs. (3) **Model comparison:** On 30 held-out prompts, Falcon-RW-1B and Mistral-7B were compared for faithfulness to retrieved spans and rubric scores. These targeted baselines, while limited, situate the system against simpler configurations.

4 Model Development

The technical architecture combines quantized language models, semantic retrieval, dynamic challenge generators, and user-level tracking components. The system is optimized for deployment on environments with limited compute resources.

4.1 Model Loading and Inference Optimization

We used mistralai/Mistral-7B-v0.1 (AI, 2023b) and deepseek-ai/deepseek-coder-6.7b-instruct during the development phase because of their strong instruction-following and generative performance. These models were tested on the a university supercomputing cluster (64 CPU cores, 256 GB RAM, 8×GPU for 12-hour intervals). This high-performance environment supported the efficient processing of large models and enabled content generation and experimentation.

To make the system publicly accessible, we deployed it on Google Colab, which offers a free but memory-limited environment. For this setting, we used tiiuae/falcon-rw-1b (Institute, 2023), which loaded more reliably and performed smoothly under constrained resources. All models were accessed using Hugging Face Transformers (Wolf et al., 2020) and integrated with the HuggingFacePipeline. We applied 4-bit quantization using BitsAndBytes (Dettmers et al., 2022) with float16 precision to improve loading speed and reduce memory usage. While Falcon-RW-1B is less powerful than Mistral or DeepSeek, it allowed us to conduct user studies and system evaluations in a more accessible environment like Google Colab.

4.2 Semantic Retrieval Infrastructure

We embedded chapter-level QA pairs using the MiniLM model (Wang et al., 2020) and stored them in a FAISS index (Johnson et al., 2021). LangChain's ConversationalRetrievalChain (Chase, 2022) was used to connect this retrieval backend with generative components, enabling context-aware multi-turn responses.

Buffered memory and query caching supported fast response times, while consistent vector alignment enabled coherent feedback during dynamic learning interactions.

4.3 Dynamic Content Generation

As detailed in Section §3.1, we created a curated dataset of question–answer pairs and summaries using DeepSeek Coder, followed by extensive manual review and cleaning. This dataset served as the foundation for generating instructional content across various learning modes. We used the cleaned outputs to design storytelling scenarios, business case simulations, and interactive challenges that align with each chapter's key concepts.

To enrich these modules contextually, we extracted keywords from the QA content and summaries, and used the Google Search API to fetch relevant diagrams, definitions, and external references. This additional material supports multimodal enrichment across the AI Tutor's interface. We also integrated a scripting pipeline for content sequencing to automate parts of the instructional flow, enabling consistent logic in challenge construction and narrative design while maintaining adaptability across different textbook chapters.

- **Business Case Narratives**: Automatically constructed scenarios based on key concepts, illustrating tradeoffs and professional decision-making contexts (wri, 2023).
- **Storytelling Modules**: Generated using narrative templates populated with synthetic personas and real-world problem frameworks to align with learner context (sma, 2023).
- **Supplemental Content**: Retrieved through the Google Search API to enrich textual content with diagrams, infographics, or relevant external articles.
- **Walkthrough Instructions**: Step-by-step interface prompts generated from predefined templates to guide learners through selected challenges.

This content is selected and rendered at runtime based on the learner's chosen chapter and activity, ensuring that delivery remains contextually relevant.

4.4 Challenge Implementation

The AI Tutor integrates six types of interactive challenges to support varied learning strategies, built

Challenge Type	Description	Cognitive Skill Targeted
Flashcards (Flip)	Flip cards to reveal answers; mark as "Got it" or "Missed it" to track recall.	Active recall, spaced repetition
Fill in the Blanks	Complete sentences using dropdowns; incorrect answers reduce a limited life count (3).	Contextual understanding, error correction
Match the Answers	Match terms with definitions using dropdown menus; immediate feedback provided.	Relational mapping, concept reinforcement
Timed Questions	Answer within 15 seconds to promote fast recall and attention.	Rapid decision-making, memory retrieval
Scenario-Based (Hints)	Make decisions in real-world scenarios with optional hints and feedback.	Applied reasoning, narrative comprehension
Multiple Choice Questions (MCQs)	Adaptive quizzes with real-time feedback and XP rewards.	Concept clarification, reinforcement learning

Table 2: Summary of Interactive Challenge Types and Targeted Learning Skills

on the curated dataset described in Sections§3.1 and §4.3. This dataset, generated using DeepSeek Coder and manually refined, informed the design of flashcards, quizzes, fill-in-the-blanks, and scenario-based activities. To enhance contextual relevance, we extracted keywords and used the Google Search API to gather supplementary materials such as diagrams and articles.

The implemented challenges include flashcards for repetition-based recall with randomized shuffling and XP tracking; multiple-choice quizzes (MCQs) offering adaptive feedback and rewards (mcq, 2023); fill-in-the-blank tasks using dropdowns and life-based scoring (edu, 2023); matching. Summary of interactive challenge types and targeted learning skills. XP reward schedules and progression thresholds are provided in Table 2 and Appendix H .

4.5 Gamification Engine

The XP system uses nonlinear thresholds (Landers, 2014) to drive engagement. Table 13 outlines the XP required to progress between levels. Matplotlib-based dashboards visualize user performance metrics such as XP gains, challenge completion rates, and chapter history.

4.6 Conversational Assistant

A retrieval-augmented assistant (Lewis et al., 2021) allowed learners to engage in chapter-aware queries. The assistant used FAISS-indexed semantic chunks aligned with learner history and synthesized responses with Falcon-RW-1B. To maintain consistency, the assistant shared infrastructure with challenge generators and QA modules. Supportive mechanisms included timeout management, hint provisioning, and multi-turn buffer tracking. This ensured both relevance and responsiveness across long-running sessions. If retrieval returns no span above a cosine-similarity threshold ($\theta = 0.35$ by default), the assistant abstains from answering, surfaces a clarifying hint, and prompts the learner to refine or re-specify the query scope.

5 Evaluation and Results

Note An earlier ARR cycle included a 30-participant pilot; this revision reports the subsequent main study (N=62; 57 complete).

5.1 Study Design

We conducted a mixed-method evaluation to assess the tutor's usability, engagement, and instructional effectiveness. The study protocol was approved by the university IRB (**IRB2025-0182**), and all participants provided informed consent. A total of **62 participants** accessed the system, and **57 completed** both pre- and post-assessments. Participants were students or early-career professionals in computing-related fields, recruited through university mailing lists, social media (LinkedIn, WhatsApp), and faculty announcements. Participation was voluntary and unpaid.

Each participant completed a single **45-minute** session in **Google Colab**. A pilot study with **5 participants** helped refine instructions and survey items. Sessions followed a consistent flow: (1) a **pre-assessment** on AI familiarity and comfort, (2) interaction with the tutor across storytelling, case, and challenge modes, and (3) a **post-assessment** on perceived learning, usability, and experience. Participants completed fixed pre- and post-assessment instruments (full questionnaires in Appx. F and

Appx. G); both instruments were IRB-approved and administered immediately before/after the 45-minute session. To ensure a consistent experience, participants received both a full guide [1] and a concise mini guide[2], and embedded prompts within the notebook offered step-by-step guidance during runtime. Data collected included Likert-scale ratings, open-ended feedback, and system logs (session duration, number of activities completed, and interaction frequency). Pre/post items targeted two short-term learning proxies—*familiarity* and *comfort*—plus post-only usability/engagement items; exact wordings and scales are reproduced in Appx. F–G. We treat these as self-report proxies rather than objective learning gains. Quantitative analyses employed descriptive statistics, exploratory factor analysis, Wilcoxon signed-rank tests, and Mann–Whitney U tests.

5.2 User Perceptions

Post-assessment feedback (Table 3) indicated strong positive reception. Mean ratings across all items exceeded **4.5**, with participants reporting high ease of use (**4.60**), clarity (**4.76**), and satisfaction with guided walkthroughs (**4.70**). Engagement of learning modes (**4.70**) and overall experience (**4.70**) were similarly rated. **Ninety-six percent** reported improved understanding, and willingness to reuse the tutor scored **4.71**, highlighting strong perceived value.

Exploratory factor analysis revealed two latent factors: *Usability* and *Engagement and Content Quality*. The Kaiser–Meyer–Olkin value was **0.674**, and Bartlett's test was significant ($\chi^2 = 114.66$, $p < 0.001$), confirming data suitability. Together, the factors explained **50.27%** of the variance. Internal consistency was excellent for Usability ($\alpha = 0.895$), while Engagement and Content Quality showed moderate reliability ($\alpha = 0.449$) Given the heterogeneous nature of the "Engagement/Content Quality" construct and the small item set, internal consistency was modest ($\alpha = 0.45$). We therefore treat the two-factor solution as exploratory and report item loadings and communalities in (see Appendix E; future work will refine the item pool and validate the structure on a larger sample.

[1] Anonymized documentation link provided in supplementary material.

[2] Anonymized documentation links provided in supplementary material.

Table 3: Post-assessment descriptive statistics (N = 57). Items rated 1–5; "Understanding improved" coded as 1 = Yes, 0 = No/Neutral.

Feedback Item	Mean	Median	SD
Overall experience	4.70	5.0	0.82
Ease of using the interface	4.60	5.0	0.94
Helpfulness of walkthroughs	4.70	5.0	0.82
Engagement of learning modes	4.70	5.0	0.50
Content clarity and quality	4.76	5.0	0.47
Ease of session completion	4.78	5.0	0.46
Understanding improved (Yes=1)	0.96	1.0	0.16
Would use again (1–5)	4.71	5.0	0.50
Speed / performance rating	4.56	5.0	0.68

Table 4: Pre vs. post comparisons (Wilcoxon signed-rank tests). Significant p-values (< 0.05) indicate improvement.

Comparison	Pre	Post	Statistic	p-value
Familiarity to Understanding	2.39	2.77	88.0	0.00003
Comfort to Experience	2.38	2.82	102.0	0.00061

5.3 Learning Outcomes

Our analytical approach follows prior work on evaluating user perceptions of generative AI agents through exploratory factor analysis and non-parametric significance testing (Tolsdorf et al., 2025)

Wilcoxon signed-rank tests demonstrated significant learning gains between pre- and post-assessments (Table 4). Familiarity increased from **2.39** to **2.77** ($p < 0.001$), indicating improved conceptual understanding, while comfort increased from **2.38** to **2.82** ($p < 0.001$), reflecting enhanced confidence and experience.

No significant differences were observed across user subgroups based on initial familiarity or comfort (all $p > 0.42$), and effect sizes were negligible ($|\delta| < 0.15$), suggesting consistent improvements across participants regardless of prior experience.

Baseline & Ablation Results (Brief). RAG improved rubric scores versus generation-only (mean total 3.9→2.7/5; Δ = 1.2, $p < 0.001$, $r = 0.62$; $\kappa = 0.71$). Multi-mode exposure increased concept-quiz scores relative to MCQ-only (median +2/8 vs. +0/8; $p = 0.004$, $r = 0.49$; 95% CI [+0.5, +2.5]). Mistral-7B outperformed Falcon-RW-1B on faithfulness (exact-overlap 68% vs. 51%) and rubric totals (+0.6/5; $p = 0.02$), consistent with model capacity differences. Full tables, rubrics, and agreement statistics are provided in Appendix C.

6 Discussion

This study addressed three research questions on content generation, learner engagement, and deployment feasibility of the AI Tutor.

RQ1: Effectiveness of RAG-based content generation. Our pipeline combining FAISS retrieval, DeepSeek Coder, and LangChain effectively transformed textbook content into coherent question–answer pairs and summaries. Manual refinement ensured accuracy and contextual alignment, and participants rated content quality highly (**Mean = 4.63**). Minimal hallucinations and strong chapter relevance show that lightweight RAG methods can generate reliable educational content directly from open resources.

RQ2: Impact of interactive learning modes. Storytelling, business scenarios, and gamified challenges boosted engagement and supported deeper reasoning. Learners reported high engagement (**Mean = 4.69**) and usability (**Mean = 4.70**), with **96%** indicating improved understanding. Significant gains in familiarity (**2.39** $\rightarrow$ **2.77**) and comfort (**2.38** $\rightarrow$ **2.82**) highlight improved conceptual knowledge and confidence. These outcomes reflect short-term self-report improvements rather than objective learning gains; we therefore describe them as proxy measures and report them alongside usability/engagement.

RQ3: Feasibility in low-resource environments. Deployment on Google Colab showed that lightweight models like Falcon-RW-1B can deliver responsive, usable systems under resource constraints. Most participants reported smooth performance, and strong reuse intent (**4.71**) demonstrates the practicality of this approach.

Overall, combining retrieval-augmented generation with modular instructional modes enabled effective, engaging, and scalable learning. The AI Tutor improved short-term outcomes and demonstrated how AI can transform open educational resources into adaptive, interactive environments. Consistent with (Tolsdorf et al., 2025), who identified multiple latent user perception dimensions in generative AI, our work extends this perspective to education by directly linking usability and engagement to learning outcomes.

7 Limitations and Future Work

While the AI Tutor showed strong outcomes, several limitations remain. The current prototype is deployed on Google Colab to ensure free access in low-resource settings, but this introduces constraints such as time-limited sessions, lack of user accounts or progress tracking, and occasional startup delays of up to 30–40 minutes. Memory limitations also prevented deploying larger models (e.g., Mistral-7B), leading us to use a lighter model (Falcon-RW-1B) that may limit generative performance. We did not include a between-system baseline in the original submission; our added within-subjects ablations are small and exploratory and should be interpreted accordingly. Moreover, this study did not include direct comparisons with baseline tutoring systems or other RAG-based tools, which would provide stronger context for evaluating relative effectiveness. Future work will address these gaps by migrating to a dedicated web platform with persistent storage and user tracking, integrating richer learning assessments, and expanding support to additional subjects and multimodal resources. These steps aim to improve scalability, personalization, and educational impact in real-world classroom settings.

8 Conclusion

This paper presented an Interactive AI Tutor that enhances textbook-based learning through retrieval-augmented generation and modular instructional features. By integrating DeepSeek Coder, FAISS retrieval, and LangChain pipelines, the system generates aligned questions and summaries from OpenStax content and delivers them through storytelling, business scenarios, and structured challenges. Deployment with Falcon-RW-1B on Google Colab demonstrated feasibility in low-resource settings, while testing with Mistral-7B validated scalability on higher-end infrastructure. Evaluation with 57 participants showed consistently high usability and engagement ratings, significant improvements on short-term proxies (familiarity and comfort), and 96% reporting improved conceptual understanding. While automation accelerates content generation, human oversight remains essential for ensuring instructional quality and relevance. This work contributes a reusable dataset and a replicable design framework for AI-powered educational tools. Future directions include migrating to a dedicated web platform, incorporating richer assessments, and expanding multimodal capabilities to further support personalized, interactive, and scalable learning experiences.

Ethics Statement

The user study was conducted with voluntary participation under university-approved IRB protocol. No personally identifiable information was collected.

Broader Impact

This work aims to improve access to interactive learning using open educational resources and lightweight AI deployments. It can benefit learners in resource-constrained environments by enhancing personalization and engagement. Limitations include dependency on internet access and evolving AI model reliability.

References

2009. Copyright page. In Beverly Park Woolf, editor, *Building Intelligent Interactive Tutors*, page iii. Morgan Kaufmann, San Francisco.

2023. Fill in the blanks games. Accessed: 2025-04-28.

2023. How to write a business case. Accessed: 2025-04-28.

2023. Mcq quiz dataset. Accessed: 2025-04-28.

2023. Sell your product with powerful storytelling. Accessed: 2025-04-28.

DeepSeek AI. 2023a. Deepseek coder: Code large language model. https://huggingface.co/deepseek-ai/deepseek-coder-6.7b-instruct. Accessed: 2025-04-28.

DeepSeek AI. 2024. Deepseek coder-6.7b release. https://github.com/deepseek-ai/DeepSeek-Coder. Accessed: 2025-04-26.

Mistral AI. 2023b. Introducing mistral 7b. https://mistral.ai/news/introducing-mistral-7b/. Accessed: April 26, 2025.

Harrison Chase. 2022. Langchain: Building llm-driven applications. https://python.langchain.com/docs/introduction/. Accessed: 2025-04-26.

Vinay K. Chaudhri, David Gunning, Hannah Lane, Peter Clark, Ken Barker, Simon Chaw, Mark Greaves, Benjamin Grosof, Alex Leung, Suchi Mishra, Julio Pacheco, Bruce Porter, Aaron Spaulding, Dan Tecuci, and Peter Z. Yeh. 2013. Inquire biology: A textbook that answers questions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 27, pages 1624–1631. AAAI Press.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Gpt3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, volume 35, pages 30318–30332. Curran Associates, Inc.

Darina Dicheva, Christo Dichev, Gennady Agre, and Galia Angelova. 2015. Gamification in education: A systematic mapping study. *Educational Technology & Society*, 18(3):75–88.

Technology Innovation Institute. 2023. Falcon-rw-1b model card. https://huggingface.co/tiiuae/falcon-rw-1b. Accessed: 2025-04-28.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards mitigating hallucination in large language models via self-reflection. *Preprint*, arXiv:2310.06271.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Khan Academy Labs. 2023. Introducing Khanmigo: AI-Powered Tutor and Teaching Assistant. https://www.khanacademy.org/khan-labs. Accessed: 2025-04-25.

Richard N. Landers. 2014. Developing a theory of gamified learning: Linking serious games and gamification of learning. *Simulation & Gaming*, 45(6):752–768.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Anna Lieb and Toshali Goel. 2024. Student interaction with newtbot: An llm-as-tutor chatbot for secondary physics education. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, CHI EA '24, New York, NY, USA. Association for Computing Machinery.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented language models: a survey. *Preprint*, arXiv:2302.07842.

Renáta Németh, Annamária Tátrai, Miklós Szabó, and Árpád Tamási. 2024. Using a rag-enhanced large language model in a virtual teaching assistant role: Experiences from a pilot project in statistics education. *Hungarian Statistical Review*, 7(2):3–27.

OpenStax. 2023. *Workplace Software and Skills*. OpenStax, Rice University.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *Preprint*, arXiv:2302.06476.

Google Research. 2023. Google colaboratory. https://colab.research.google.com. Accessed: 2025-04-28.

Robin Schmucker, Meng Xia, Amos Azaria, and Tom Mitchell. 2023. Ruffle&riley: Towards the automated induction of conversational tutoring systems. In *NeurIPS 2023 Workshop on Generative AI for Education*.

Joseph J. Slade, Alina Hyk, and Regan A. R. Gurung. 2024. Transforming learning: Assessing the efficacy of a retrieval-augmented generation system as a tutor for introductory psychology. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 68, pages 1827–1830, Los Angeles, CA, USA. SAGE Publications.

MuPDF Team. 2023. Pymupdf: Lightweight pdf parsing and rendering. https://github.com/pymupdf/PyMuPDF. Accessed: 2025-04-26.

Jan Tolsdorf, Alan F. Luo, Monica Kodwani, Junho Eum, Mahmood Sharif, Michelle L. Mazurek, and Adam J. Aviv. 2025. Safety perceptions of generative ai conversational agents: uncovering perceptual differences in trust, risk, and fairness. In *Proceedings of the Twenty-First USENIX Conference on Usable Privacy and Security*, SOUPS '25, USA. USENIX Association.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, pages 5776–5788. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Appendix

B Core Textbook-to-Interaction Pipeline

Textbook PDF T Interactive AI Tutor Experience

Step 1: Chapter Extraction
Extract chapter-wise content $C = \{c_1, c_2, ..., c_n\}$ from T using PDF parsing tools.

Step 2: QA Dataset Generation
chapter $c_i \in C$ Generate question–answer pairs QA_i and summary S_i using DeepSeek Coder-6.7B Store initial dataset $D = \{(QA_i, S_i, c_i)\}$

Step 3: Manual Review and Cleaning
Filter for hallucinations, formatting issues, and factual errors to produce cleaned dataset D^*

Step 4: Semantic Indexing
Embed QA_i with MiniLM and store in FAISS vector index V

Step 5: Retrieval-Augmented Generation
Use LangChain with index V to enable context-aware content retrieval and generation

Step 6: Model Integration
Internal high-resource environment Load Mistral-7B Low-resource environment (e.g., Colab) Load Falcon-RW-1B with 4-bit quantization using BitsAndBytes

Step 7: User Interaction Layer
Initialize user profile and preferences
If user is new or walkthrough requested **then**
Launch guided walkthrough with:

- Step-by-step preview of all challenge types
- Navigation controls (next/back/skip)
- Visual prompts and tooltips

Else
Launch interactive learning modules:

- Storytelling modules
- Business case simulations
- Interactive challenges:
  - Flashcards
  - Multiple-choice quizzes (MCQs)
  - Fill-in-the-blank tasks
  - Matching exercises
  - Timed questions
  - Scenario-based challenges with hints

Step 8: Gamification and Feedback
Track XP, level progression, and badge collection
Provide real-time feedback and update learner dashboard

C Operational metrics

D Baseline and Ablation Details

D.1 Rubric (0–5 scale)

We evaluated generated responses on three criteria: (1) **Factuality** (0–2), measuring correctness relative to the textbook source; (2) **Grounding/Traceability** (0–2), measuring explicit linkage to retrieved spans; and (3) **Clarity** (0–1), assessing readability and coherence. Total possible score: 5. Table 7 details the rubric.

Template	Intent / Inputs	Full Prompt Text (Simplified)
QA Generation	*Intent:* Create up to *N* factual QA pairs from a chapter excerpt. *Inputs:* {chapter_title}, {chapter_excerpt}, {N} (default 5)	**System:** You are an educational content author. Produce concise, factual question–answer pairs strictly grounded in the provided chapter excerpt. Avoid speculation. **User:** Chapter: {chapter_title}. Excerpt: {chapter_excerpt}. Task: Generate up to {N} QA pairs. Each pair must be answerable solely from the excerpt. Use JSON schema: `[ { "question": "...", "answer": "...", "evidence_span": "..." }, ... ]`. If information is insufficient, omit that pair.
Summary Generation	*Intent:* Produce a short chapter summary. *Inputs:* {chapter_title}, {chapter_excerpt}, {max_words} (default 150)	**System:** You are summarizing textbook material for learners. Stay faithful to the source; no external facts. **User:** Chapter: {chapter_title}. Excerpt: {chapter_excerpt}. Task: Write a concise summary ({max_words} words) capturing key definitions, processes, and takeaways. Include no references beyond the excerpt.
Storytelling Scaffold	*Intent:* Transform a validated summary into a short workplace scenario. *Inputs:* {chapter_title}, {validated_summary}, {persona_A}, {persona_B}	**System:** You are an instructional designer. Create a realistic, workplace-relevant micro-story aligned with the provided summary. **User:** Summary: {validated_summary}. Task: Write a 120–180 word narrative featuring {persona_A} and {persona_B}. The scenario must exemplify 2–3 core ideas from the summary (**bold key terms on first mention**). End with a reflective question answerable from the narrative. Output JSON: `{"story": "...", "reflective_question": "..." }`.

Table 5: Prompt templates used in the Textbook-to-Interaction (T2I) pipeline. Braced fields are filled at runtime.

Setting / Rule	Public (Falcon-RW-1B)	Dev (Mistral-7B)	Notes
Decoding temperature	0.7	0.3–0.5	Higher for storytelling; lower for QA
Top-p (nucleus)	0.9	0.9	Paired with moderate temperature
Top-k	50	50	Conservative cap to reduce drift
Max new tokens	256	256 (QA), 180 (story)	Enforced at pipeline level
Repetition penalty	1.10	1.10	Mitigates loops
Stop sequences	`{"\n\n"}`	same	Prevent runaway JSON / paragraphs
Grounding mode	Required	Required	Answers must cite retrieved span IDs
Similarity threshold θ	0.35 cosine	0.35 cosine	If $\max \text{sim} < \theta$, abstain
Abstention response	"I don't have enough context to answer that."	same	Followed by one clarifying question
Evidence requirement	At least one quoted / cited span per key claim	same	Regex check before output
External web access	Disabled	Disabled	Prevent leakage beyond OER
Story length guard	120–180 words	120–180 words	Keeps mode parity
JSON schema checks	Strict (pydantic)	Strict	QA / story outputs validated

Table 6: Guardrails and decoding defaults. Values shown are the defaults used in our experiments unless otherwise noted.

Criterion	Scoring Description
Factuality (0–2)	0 = Incorrect, 1 = Partially correct, 2 = Fully correct
Grounding (0–2)	0 = No evidence cited, 1 = Partial citation, 2 = Explicit and correct citation
Clarity (0–1)	0 = Unclear or incoherent, 1 = Clear and coherent

Table 7: Rubric used to score generated outputs in baseline evaluations.

D.2 Rater Agreement

Two independent raters scored all outputs using the rubric above. Inter-rater reliability was $\kappa = 0.71$ (95% CI [0.62, 0.80]), indicating substantial agreement. Disagreements were resolved through adjudication.

D.3 RAG ON vs. OFF Results

D.4 Single vs. Multi-Mode Results

D.5 Model Comparison Results

D.6 Concept Quiz Items (Objective)

Below are representative items from the 8-question concept quiz used in the single- vs. multi-mode study.

- Q1: What is retrieval-augmented generation (RAG) and how does it improve factual grounding?
- Q2: Describe two benefits of 4-bit quantization in low-resource deployments.
- Q3: In the storytelling mode, what instructional purpose do reflective questions serve?
- Q4: What role does the FAISS index play in semantic retrieval?

Metric	RAG ON	RAG OFF
Mean total score (0–5)	3.9	2.7
Factuality	1.8	1.2
Grounding	1.7	0.8
Clarity	0.4	0.7

Table 8: Comparison of assistant performance with and without retrieval grounding (N=40 questions).

E Exploratory Factor Analysis Details

E.1 Factor Loadings and Communalities

Table 11 reports the factor loadings for post-assessment items based on principal axis factoring with varimax rotation. Two factors emerged, explaining a total of **50.27%** of the variance (**37.65%** for Factor 1 and **12.62%** for Factor 2). Factor 1 corresponds primarily to *Usability*, while Factor 2 captures *Engagement and Content Quality*.

Measure	MCQ Only	MCQ+ Story+ Scenario
Median concept quiz delta (0–8)	+0	+2
p-value (Wilcoxon)	0.004	
Effect size r	0.49	

Table 9: Learning gains with single- vs. multi-mode interaction (n20).

Metric	Falcon-RW-1B	Mistral-7B
Exact overlap with retrieved spans	51%	68%
Mean rubric total (0–5)	3.2	3.8
p-value	0.02	

Table 10: Model comparison on 30 held-out prompts.

Item	Factor1 (Usability)	Factor2 (Engagement /Content)
Overall experience	0.885	0.026
Ease of using the interface	0.881	0.224
Helpfulness of walkthroughs	0.833	-0.160
Engagement of learning modes	-0.011	0.599
Content clarity and quality	0.076	0.512
Ease of session completion	-0.009	0.246

Table 11: Exploratory factor loadings for post-assessment items ($N = 57$). Principal axis factoring with varimax rotation. Total variance explained: 50.27%.

E.2 Communalities and Reliability

Given the heterogeneous nature of the "Engagement/Content Quality" construct and the small item set, internal consistency was modest ($\alpha = 0.45$). We therefore treat the two-factor solution as exploratory. Future work will refine the item pool and validate the structure on a larger sample.

F Pre-Assessment Questionnaire

Participants completed the following pre-assessment form prior to using the AI Tutor. The form collected demographic information, prior experience with textbook learning and AI tools, learning preferences, and expectations from the system.

F.1 Section 1: Basic Information

- What best describes your current role?

- High school or less
- Associate's degree
- Bachelor's degree
- Master's degree
- PhD degree
- Instructor/Professor
- Working Professional
- Other

- How familiar are you with using AI-based educational tools? (Very familiar / Somewhat familiar / Not familiar at all / Other)

F.2 Section 2: Previous Experience with Textbook Learning

- How often do you use textbooks for learning? (Daily / Weekly / Occasionally / Rarely / Never)

- Have you used any interactive or digital learning platforms before? (Yes / No)

F.3 Section 3: Experience with Digital Learning Platforms

- Please list any digital learning platforms you've used.

F.4 Section 4: Learning Preferences & Expectations

- Which learning methods do you prefer? (Reading text / Watching videos / Interactive quizzes / Case-based learning / Storytelling / Group discussions)

- What are your main learning goals with the AI Tutor?

- What do you expect from an AI-powered educational tool?

F.5 Section 5: Technical Readiness

- How comfortable are you with using tools like Google Colab or Jupyter Notebooks? (Very comfortable / Somewhat comfortable / Not comfortable)

- Have you used any AI-driven tools before? (Yes / No / Maybe)

F.6 Section 6: Experience with AI Tools

- If yes, please describe the tools you've interacted with and your experience using them.

F.7 Section 7: Expectations from the AI Tutor

- What kind of content do you expect the AI Tutor to help you with? (Subject-specific knowledge / Study skills / Test preparation / Real-world applications / Problem-solving)

- Do you prefer a personalized or standardized learning experience? (Personalized / Standardized / No preference)

- What features would you want in an AI-powered learning tool?

G Post-Assessment Questionnaire

After completing their interaction with the AI Tutor, participants filled out the following post-assessment questionnaire. The form was designed to evaluate usability, user engagement, content quality, technical performance, and overall experience. Responses informed our analysis of short-term learning impact and user satisfaction.

G.1 Section 1: Basic Information

- What best describes your current role?
  - High school or less
  - Associate's degree
  - Bachelor's degree
  - Master's degree
  - PhD degree
  - Instructor / Professor
  - Working Professional
  - Other

- How did you access the AI Tutor?
  - Through Google Colab directly
  - Researcher sent me the link
  - Other

G.2 Section 2: Overall Experience

- How would you rate your overall experience with the AI Tutor?
  (1 = Very Poor . . . 5 = Excellent)

- How easy was it to use the AI Tutor interface?
  (1 = Very Difficult . . . 5 = Very Easy)

- How helpful were the guided walkthroughs in getting started?
  (1 = Not helpful . . . 5 = Very helpful)

- How engaging did you find the different learning modes (Business Case, Storytelling, Challenges)?
  (1 = Not Engaging . . . 5 = Very Engaging)

G.3 Section 3: Challenge Types

- Which challenge types did you try?
  - Flashcards
  - Multiple-Choice Questions (MCQ)
  - Fill in the Blanks
  - Match the Answers
  - Timed Questions
  - Scenario-Based Challenges
- Which challenge type was the most effective for your learning? Why?
- Was the feedback (XP points, correct/incorrect indicators, progress tracking) helpful in understanding your performance? (Yes / No / Maybe)

G.4 Section 4: Performance and Content

- How would you rate the quality and clarity of the content presented in the AI Tutor?
  (1 = Very Poor . . . 5 = Excellent)
- Overall, how easy was it to complete your learning session with the AI Tutor?
  (1 = Very Difficult . . . 5 = Very Easy)
- Did the AI Tutor help improve your understanding of the concepts?
  - Yes, significantly
  - Yes, somewhat
  - Neutral
  - No, not really
  - No
- How would you describe the speed and performance of the AI Tutor in Google Colab?
  - Fast and smooth
  - Acceptable
  - A bit slow
  - Very slow or laggy
  - Didn't work properly

G.5 Section 5: Suggestions and Future Use

- What features would you like to see added in the future?
- Would you use the AI Tutor again if it were available as a public website? (Yes / No / Maybe)
- Any other comments or suggestions to improve the AI Tutor?

H XP Schedules and Level Thresholds

H.1 XP Reward Schedule

Table 12 shows the XP reward structure used in the prototype implementation. Points are awarded for completing different interactive challenge types, and their share of total XP reflects relative contribution to learner progression.

Challenge Type	XP per Attempt	Share of Total XP (%)
Flashcards (Flip)	5	7.5
MCQ quiz	10	14.9
Fill-in-the-Blank	10	14.9
Match-the-Answers	12	17.9
Timed questions	15	22.4
Scenario-based (Hint)	15	22.4

Table 12: XP reward schedule for interactive challenges.

H.2 Level Progression Thresholds

Learners advance through levels as they accumulate XP. Table 13 lists the XP requirements for each level and the cumulative total needed for progression.

Level	XP to Next	Cumulative XP
$1 \rightarrow 2$	50	50
$2 \rightarrow 3$	75	125
$3 \rightarrow 4$	100	225
$4 \rightarrow 5$	125	350
$5 \rightarrow 6$	150	500
$6 \rightarrow 7$	175	675
$7 \rightarrow 8$	200	875
$8 \rightarrow 9$	225	1100
$9 \rightarrow 10$	250	1350
$10 \rightarrow 11$	275	1625

Table 13: XP thresholds for each level in the gamification engine.