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

Anonymous ACL submission

Abstract

We present an Interactive AI Tutor designed to make textbook learning more accessible, adaptive, and engaging. Addressing the limitations of static educational resources, the system transforms textbook chapters into dynamic learning experiences using retrieval-augmented generation, interactive challenges, and narrative-based instruction. Initially developed using DeepSeek Coder-6.7B for question-answer generation, later optimized with Mistral-7B, and further adapted for deployment using Falcon-RW-1B on resource-constrained platforms such as Google Colab, the system integrates LangChain pipelines, FAISS retrieval, and Google APIs. It supports modular learning modes including storytelling, business simulations, and quizzes, with real-time progress tracking. Our findings demonstrate the feasibility of deploying lightweight yet interactive AI systems to personalize learning from open educational resources (OER) at scale.

1 Introduction

Traditional textbook-based learning often struggles to meet the evolving needs of today's learners. Long, static readings can lead to disengagement, while lack of personalization limits deeper conceptual understanding. Inspired by these challenges, this work proposes an Interactive AI Tutor that transforms static textbook content into adaptive, learner-centered experiences. Many learners find traditional textbook learning overwhelming and disengaging. Despite strong interest in the subject, reading dense chapters without interaction can reduce motivation and hinder long-term retention. This insight inspired the design of an AI Tutor that moves beyond static text to offer multimodal, scenario-based learning. Using large language models and retrieval-augmented generation techniques (Lewis et al., 2021), the AI Tutor personalizes learning pathways through storytelling, business case simulations, interactive challenges, and real-time quizzes. Each learning mode improves active participation, contextual understanding, and long-term retention. Current AI-enhanced education tools often focus narrowly on generating question-answer pairs or summarization, but lack support for diverse learning modes such as storytelling, timed challenges, or case-based reasoning. Many systems are also not modular, difficult to personalize, and often require significant computational resources for deployment. These limitations create a technical gap in accessible, adaptable, and interactive AI-driven learning systems, especially those that can support free and open educational content. Recent advances in intelligent tutoring systems have demonstrated the potential of AI to create personalized educational experiences (Woo, 2009).

However, many existing systems focus narrowly on one learning style or lack integration with open educational resources (OER). Our approach builds on these foundations by using openly available textbooks (OpenStax, 2023) and deploying modular and flexible learning modes that align with diverse cognitive strategies. In addition, retrieval-augmented architectures allow the system to dynamically retrieve, summarize and contextualize information, reducing the risk of hallucination and improving factual grounding (Qin et al., 2023). To further enhance contextual reasoning, the system supports multimodal enrichment through integration of external text and visual resources, supporting decision-based learning paths and narrative-driven modules (Zhang et al., 2024).

This study is guided by the following research questions:

1. How effectively can retrieval-augmented generation with DeepSeek Coder transform textbook content into coherent question–answer pairs and summaries?

2. How do learners perceive and engage with interactive learning modes, such as storytelling, business case simulations, and gamified challenges within the proposed AI Tutor system?

3. Is it feasible to deploy an AI-powered tutoring system on resource-constrained platforms such as Google Colab, while maintaining usability, responsiveness, and adaptability for diverse users?

To address these questions, we introduce an Interactive AI Tutor designed to convert static textbook content into adaptive, learner-centered experiences using retrieval-augmented generation and multimodal enrichment. The system incorporates lightweight open-source models, LangChain pipelines, and interactive learning modes. A detailed description of the architecture, deployment, and learning modules is provided in the following sections.

This work makes four key contributions. First, we present a modular AI Tutor system that personalizes learning through retrieval-augmented content and interactive formats. Second, we curate a refined dataset of question–answer pairs and summaries aligned with OpenStax textbooks, based on manually validated outputs from DeepSeek Coder. Third, we offer an open-source deployment framework suitable for accessible use on cloud-hosted notebooks. Finally, we provide an empirical evaluation based on structured user feedback to inform future development and application.

Overall, this work contributes to Human-Centered NLP and NLP Applications by demonstrating a scalable, accessible tutoring framework that enhances open educational content through multimodal and retrieval-based learning strategies.

2 Related Work

The application of AI-driven techniques in education has gained momentum as researchers seek to enhance engagement, personalization, and adaptability in learning systems. However, challenges remain in dynamically adapting large-scale educational resources such as textbooks into interactive, learner-centered experiences.

2.1 AI-Powered Learning Systems and Personalization

Several studies have explored the use of large language models and retrieval-augmented generation for educational applications. (Lewis et al., 2021) introduced Retrieval-Augmented Generation (RAG), demonstrating how combining retrieval and generation improves factual accuracy and contextual relevance. More recently, (Qin et al., 2023) highlighted the persistent challenge of hallucinations in language models, emphasizing the importance of retrieval-grounded approaches for educational settings.

Interactive learning platforms have also evolved, with initiatives like Khanmigo (Khan Academy Labs, 2023) leveraging GPT-based systems for tutoring (Khan Academy Labs, 2023). While these systems offer personalized guidance, they primarily focus on conversational assistance rather than modular, chapter-wise adaptation delivered through lightweight deployment modes such as browser-based interfaces and Colab notebooks.

Despite these advancements, current models often emphasize one-to-one dialogue rather than modular content delivery aligned with diverse learning modes such as storytelling, business cases, and gamified challenges. Moreover, the integration of open educational resources (OER) into adaptive and multimodal learning environments, particularly those designed for rapid feedback and accessible deployment, remains insufficiently explored.

2.2 Retrieval-Augmented Architectures in Education

Recent work by (Mialon et al., 2023) proposed Retrieval-Augmented Multimodal Models (RAMM) for combining textual, visual, and structured data in generation tasks. Although promising, their research primarily addressed general multimodal question answering rather than adaptive educational content transformation informed by retrieval and user interaction data.

Similarly, (Ma et al., 2025) evaluated large language models' performance on machine reading comprehension tasks, identifying gaps in maintaining factual consistency over long contexts. However, they did not extend their methods to interactive learning settings or adaptive textbook restructuring.

2.3 Our Approach

While prior work has focused on retrieval-augmented QA (Lewis et al., 2021), interactive tutoring (Khan Academy Labs, 2023), or multimodal educational content (Zhang et al., 2024), few systems combine these approaches into a unified, lightweight pipeline tailored for OER transformation. Our work builds on this gap by integrating retrieval grounding, modular instruction modes, and lightweight deployment strategies. Compared to systems like Khanmigo, which focus on conversational tutoring using large language models, the proposed AI Tutor emphasizes modular, multimodal learning. Unlike Khanmigo, which primarily supports general-purpose conversational tutoring, our system enables chapter-specific retrieval, structured gamified challenges, and multimodal enrichment using curated datasets, all within a reproducible open-source pipeline. It supports six distinct challenge types, storytelling, and business simulations, all designed to promote contextual and active learning.

3 Methodology

We designed the AI Tutor as a modular system to deliver personalized, chapter-specific learning using the "Workplace Software and Skills" textbook (OpenStax, 2023). Deployed in environments like Google Colab, the pipeline includes content extraction, question–answer generation, manual review, and integration into learning modes such as flashcards, storytelling, and simulations. Figure 1 outlines the end-to-end process, from raw textbook input to interactive delivery. The modular structure allows easy updates to individual components without affecting the full system.

3.1 Data Acquisition and Structuring

We extracted textbook chapters from the OpenStax "Workplace Software and Skills" textbook (2023) using PyMuPDF (Team, 2023). Each chapter was segmented based on section headers, and unrelated instructional metadata was removed to retain only meaningful content blocks. These segments were processed using DeepSeek Coder-6.7B (AI, 2024) with constrained prompt templates to generate a consistent set of question–answer (QA) pairs and summaries. All generated outputs were manually reviewed for hallucinations and formatting issues, referencing mitigation strategies in (Ji et al., 2023). Only high-quality QA pairs (up to five per chapter) and a concise summary were retained. The final dataset, *MergedChapterDataset.csv*, was compiled in structured CSV format and is publicly available [1] to support reproducibility.

[1]Anonymized dataset link provided in supplementary material.

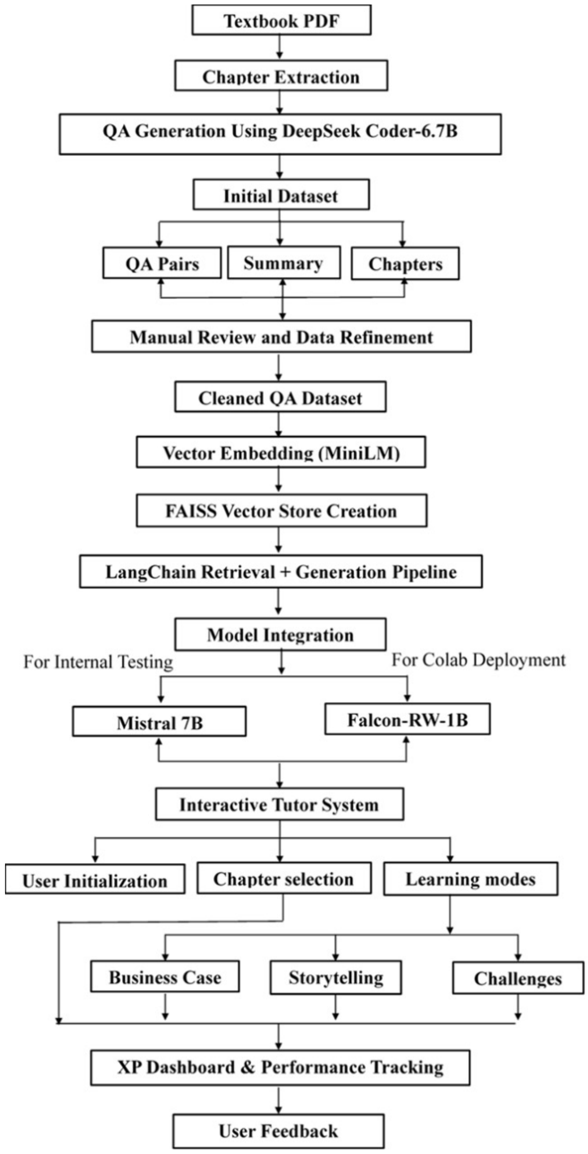

Figure 1: Pipeline overview of the AI Tutor system.

The dataset, though not directly visible to users, played a central role in shaping the AI Tutor's instructional flow. It enabled precise content alignment across learning modes such as storytelling, guided walkthroughs, and interactive challenges. To ensure quality, we spent two to three days manually reviewing and cleaning the outputs generated by DeepSeek Coder. Attempts to automate this step using Python libraries like NumPy and pandas were unsuccessful due to inconsistent formatting, extraneous content, and lack of structural patterns in the QA and summary fields. Manual inspection allowed us to extract coherent and relevant content, which was then validated for factual accuracy, clarity, and alignment with instructional goals. The cleaned dataset also supported keyword extraction for use with the Google Search API, allowing retrieval of supplementary resources like diagrams and external articles to enrich the learning experience. Although this implementation used a single OpenStax textbook, the pipeline can be adapted to other structured PDFs with consistent chapter formatting, and we plan to test this adaptability in future work.

3.2 Interactive Learning Mode Design

Learners can engage with content through three instructional formats, each supporting different cognitive goals. Business case generation promotes applied reasoning by placing learners in real-world decision-making scenarios grounded in the chapter's context (wri, 2023). Storytelling mode enhances comprehension through character-driven narratives that illustrate workplace problems and solutions (sma, 2023). Interactive challenges reinforce knowledge using task-based activities such as flashcards, multiple-choice questions, fill-in-the-blanks, matching, timed responses, and scenario-based simulations (Dicheva et al., 2015). These modes collectively support both recall and higher-order thinking.

Each learning mode incorporates feedback and an XP-based progression system to support engagement and track progress. We selected storytelling, business case simulations, and interactive challenges to promote deeper learning beyond passive reading. While students often study textbooks to pass exams, they may struggle to retain or apply the content afterward. Storytelling and business cases help bridge this gap by connecting concepts to real-world scenarios, improving comprehension and memory. These approaches align with constructivist learning theory, which emphasizes learning through experience, and Bloom's taxonomy, which encourages moving from basic recall to applied understanding. Interactive challenges offer hands-on opportunities to reinforce knowledge. Together, these modes aim to make textbook content more engaging, memorable, and transferable.

3.3 User Onboarding and Guided Walkthrough

The system supports user onboarding through two documentation formats: a detailed full guide[2] and a concise mini guide[3] These documents outline interaction modes, challenge types, and system walkthroughs. In addition, embedded prompts within the notebook offer step-by-step guidance during runtime.

[2]Anonymized documentation link provided in supplementary material.

[3]Anonymized mini-guide link provided in supplementary material.

3.4 Progress Tracking

Learner progress is tracked via a gamified XP system. Each completed challenge awards XP, which contributes to level advancement, badge collection, and a visual dashboard of user activity. Feedback mechanisms are integrated into each mode to promote adaptive learning without requiring external supervision.

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, 2023) 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 Clemson Palmetto Supercomputing Cluster, where we launched JupyterLab sessions with 64 CPU cores, 256 GB of memory, and 8 GPUs 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 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

A summary of the challenge types, their instructional structure, and the cognitive skills they aim to develop is provided in Table 1. The XP allocation for each challenge type is summarized in Table 2.

4.5 Gamification Engine

The XP system uses nonlinear thresholds (Landers, 2014) to drive engagement. Table 3 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.

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 1: Summary of Interactive Challenge Types and Targeted Learning Skills

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 2: XP reward schedule for interactive challenges.

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

Table 3: XP thresholds for each level in the gamification engine.

5 Evaluation Design

This evaluation was conducted under IRB-approved protocols (IRB2025-0182) at a U.S.-based university. All procedures adhered to institutional ethical guidelines, with voluntary participation and electronic informed consent.

We adopted a mixed-method approach to assess the AI Tutor's usability, engagement, and instructional effectiveness. Participants first completed a structured pre-survey capturing demographics, prior experience with AI learning tools, and platform familiarity. They then engaged in a guided interaction session with the AI Tutor on Google Colab, followed by a post-survey evaluating satisfaction, content quality, and perceived learning outcomes. Optional open-ended responses were collected to gather feedback on system strengths, limitations, and improvement suggestions.

The evaluation framework drew on established HCI and educational assessment practices. Survey instruments were developed in reference to ISO 9241-11 usability standards and refined through pilot testing. Anonymized interaction logs captured metrics such as response times and activity completion rates to complement survey responses. The co-investigator completed all required human-subjects research training to ensure eligibility for conducting this study.[4]

6 Results

6.1 Participant Overview

We received responses from 30 participants (Table 4). Most identified as students (46.7%) or working professionals (30.0%), followed by professional learners, researchers, and alumni. The majority accessed the AI Tutor through Google Colab (66.7%), while others used GitHub links, browsers, or social platforms.

6.2 Usability and Engagement

User feedback indicated high satisfaction with the system's usability and instructional design. As shown in Table 5, the average rating for overall experience was 4.57 out of 5, with 83.3% of participants giving it the highest score. The interface's ease of use (avg. 4.43) and the clarity of guided walkthroughs (avg. 4.57) were also positively rated. Learning-mode engagement had the highest aver-

[4] Anonymized verification link provided in supplementary material.

age (4.69), with many participants noting the interactivity and contextual relevance of the content.

6.3 Learning Mode Usage

Participants explored multiple learning modes (Table 6). MCQs and fill-in-the-blank tasks were the most frequently used (each by over 70%), followed by flashcards (50%), matching exercises, timed questions, and scenario-based challenges. Flashcards were cited as the most effective by 14 participants, who appreciated their concise format and immediate feedback. Scenario-based and MCQ challenges were also noted for fostering practical understanding and application. It is important to note that these responses reflect users subjective impressions and perceived usefulness, not objective measures of learning effectiveness or knowledge gain.

6.4 System Performance and Feedback

Most participants (60%) described the system performance in Google Colab as "fast and smooth," while 26.7% reported it as "acceptable." No critical technical issues were noted. Feedback mechanisms—such as XP points, correctness indicators, and progress tracking—were considered helpful by 96.6% of respondents. Suggested improvements included enhanced UI design, mobile responsiveness, a voice/chat interface, and better guidance for initial setup and model loading.

6.5 User Willingness to Reuse

A strong majority (96.4%) indicated they would use the AI Tutor again if it were made publicly available as a web-based application. This level of endorsement highlights the system's potential for broader deployment and further refinement.

6.6 Summary Insights

The results demonstrate high usability, perceived learning value, and engagement across diverse user backgrounds. Interactive formats like flashcards and scenario-based challenges were especially effective. Performance on Google Colab was stable for most users, though several noted the need for more intuitive design and platform flexibility. Participants shared positive feedback highlighting specific strengths. One user noted, "I liked the scenario-based module because it felt like a real workplace task." Another commented, "Flashcards were helpful for quick review and helped me retain key definitions." Such responses indicate that users appreciated the contextual engagement and modular variety offered by the system.

Metric	**n**	**%**
Total respondents	30	100.0
Students	14	46.7
Working professionals	9	30.0
Professional learners	4	13.3
Researchers	2	6.7
Alumni	1	3.3
Google Colab	20	66.7
GitHub link	7	23.3
Browser (direct)	1	3.3
Instagram	1	3.3

Table 4: Participant demographics and access patterns.

Metric	**Mean**	**SD**	**%≥4**
Overall experience	4.57	0.78	86.7
Interface ease of use	4.43	0.77	90.0
Walkthrough clarity	4.57	0.72	90.0
Learning-mode engagement	4.69	0.53	96.7
Content quality/clarity	4.63	0.55	96.7

Table 5: Self-reported usability and engagement ratings (5-point scale).

7 Discussion

This section reflects on the study findings in relation to the three research questions.

First, regarding the effectiveness of generating question–answer pairs and summaries from open educational textbooks using a retrieval-augmented generation (RAG) approach, our system combined FAISS-based semantic retrieval with DeepSeek Coder and LangChain. Manually curated outputs showed high factual accuracy and coherence. Participants confirmed the content was chapter-aligned and contextually appropriate, with minimal hallucinations. The strong average content quality score (Mean = 4.63; Table 5) supports the effectiveness of the RAG pipeline in producing meaningful educational content.

Second, learners responded positively to the interactive modes. Flashcards aided concept reinforcement, while MCQs and fill-in-the-blanks were most frequently used. Scenario-based tasks and storytelling encouraged contextual reasoning. These modes, grounded in structured content (Section 4.3), promoted a balance of recall and applied learning. The experience point system further supported engagement, as reflected in a high average engagement score (Mean = 4.69).

Third, on deployment feasibility, the AI Tutor functioned reliably across user types—including students, professionals, and researchers—via

Challenge type	Tried (n)	Tried (%)	Votes[†]
MCQ quiz	21	70.0	4
Fill-in-the-Blank	16	53.3	0
Flashcards	15	50.0	6
Match-the-Answers	12	40.0	1
Timed questions	11	36.7	0
Scenario-based	11	36.7	3

Table 6: Challenge adoption and perceived effectiveness (14 respondents)

Google Colab. Most users reported smooth performance without major issues. The usability rating (Mean = 4.45) and a strong reuse intent (96.3%) suggest the system is practical for lightweight, accessible environments.

In sum, the AI Tutor demonstrated that retrieval-augmented, modular instruction using open educational resources can enable effective, scalable, and user-friendly learning experiences.

8 Limitations and Future Work

The AI Tutor is currently hosted on Google Colab to support access in low-resource environments. This choice enables free, open access but introduces several limitations. Sessions are time-limited, lack multi-user access, and do not support user authentication or progress tracking. System startup times varied, with some participants experiencing delays of up to 30–40 minutes. While we guided users through the setup, these issues may have discouraged broader participation.

Model deployment posed additional challenges. Larger models like mistralai/Mistral-7B-v0.1 and deepseek-ai/deepseek-coder-6.7b-instruct were used during development but proved unreliable in Colab due to memory constraints. As a result, we used tiiuae/falcon-rw-1b for public testing. These infrastructure limitations partly explain the modest sample size of 30 participants. We also did not include a direct comparison with baseline tutoring systems or existing RAG-based learning tools. Future evaluations will include such comparisons to better contextualize the AI Tutor's relative strengths and learning impact.

This work presents a proof-of-concept system that demonstrates the feasibility of an interactive AI tutor. While participants responded positively to the interface and content, we did not include a pre/post learning assessment. Future studies will integrate quiz-based evaluations to measure actual learning gains. We also plan to migrate to a web-based platform with persistent storage, user tracking, and support for additional textbooks, videos, and real-time interaction. The current system provides a reusable foundation for future educational tools grounded in retrieval-augmented generation.

9 Conclusion

This paper presents an Interactive AI Tutor designed to enhance textbook-based learning through retrieval-augmented generation and modular instructional features. By integrating DeepSeek Coder, FAISS retrieval, and LangChain pipelines, the system generates review questions and summaries aligned with OpenStax content. Although manual refinement was required to ensure accuracy and coherence, the resulting dataset guided the design of interactive features such as storytelling, business scenarios, and structured challenges. Deployment using Falcon-RW-1B on Google Colab demonstrates feasibility in low-resource settings, while earlier testing with Mistral-7B validated system performance on higher-end infrastructure. The tutor encourages learner engagement through varied modes that support recall, reasoning, and contextual application, illustrating the value of combining AI capabilities with open educational resources. While automation accelerates content generation, human oversight remains essential for maintaining instructional quality and relevance. This work contributes a reusable dataset and a replicable design framework for building AI-powered educational tools that support personalized, interactive, and accessible 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. 2024. Deepseek coder-6.7b release. `https://github.com/deepseek-ai/DeepSeek-Coder`. Accessed: 2025-04-26.

Mistral AI. 2023. 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.

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.

Shengkun Ma, Hao Peng, Lei Hou, and Juanzi Li. 2025. Mrceval: A comprehensive, challenging and accessible machine reading comprehension benchmark. *Preprint*, arXiv:2503.07144.

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.

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.

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

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.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2024. Multimodal chain-of-thought reasoning in language models. *Preprint*, arXiv:2302.00923.