

Automatically Generating Interactive Learning Experiences with an LLM-Driven Agentic Pipeline

Gröner, F., Verma, A., & Bronowitz, J.

AI Pedagogy

Educational Technology

Instructional Design

Interactive Learning

Large Language Models

prompt engineering

Interactive and hands-on learning experiences encourage deeper engagement and produce better learning outcomes than passive consumption of media. We see an opportunity to improve learning by leveraging the ability of large language models (LLMs) to generate text and code for the automated production of interactive learning experiences. We present a chain of LLM agents that plan, generate, review, and refine a static interactive tutorial website for any topic. Their prompts were engineered to follow best prompting practices and employ pedagogic theory. This system is motivated by the effectiveness of interactive learning and the prospect of making education more accessible.

Introduction

Large language models (LLMs) are becoming increasingly integrated into educational contexts, particularly for the rapid generation of instructional content such as explanations, exercises, and summaries. However, much of AI-generated educational material remains pedagogically underdeveloped because it prioritizes fluency and surface-level coherence over meaningful learner engagement (Zawacki-Richter et al., 2019; Sharma et al., 2025). Many AI-supported learning resources reproduce passive formats that ultimately mirror textbooks or lecture notes and offer limited opportunities for learners to actively construct knowledge.

This limitation stands in contrast to decades of empirical research in the learning sciences which demonstrate that interactive and hands-on learning environments consistently outperform passive instructional formats. Prior work found that learners achieve stronger conceptual understanding, motivation, and retention when they are required to manipulate variables and explore systems dynamically (Koç & Kanadlı, 2025; Li et al., 2023).

In the context of the Learning Engineering Process cycle (Goodell & Kolodner, 2023), we take this challenge and report the results of the creation phase (designing and building a solution). We leverage the ability of LLMs to not only generate pedagogically-grounded instructional content but also automate the development of web pages that offer interactive and individually tailored learning experiences. Our iterative pipeline plans, evaluates, and refines the content quickly and cheaply, promising to make education more easily accessible.

Theoretical Foundations

In the spirit of Learning Engineering, we are not just developing technology but following the guidance of empirically backed science. Our design is anchored in complementary frameworks that articulate how learning occurs, how instruction should be structured, and how technology should be integrated into pedagogy.

The ICAP framework (Chi & Wylie, 2014) characterizes learning activities according to levels of cognitive engagement: passive, active, constructive, and interactive. It demonstrates that learning outcomes systematically improve as learners move toward higher levels of engagement. The distinction between mere activity and genuine interactivity is particularly relevant: Learning is most robust when learners are required to generate, test, and revise ideas through interaction.

Mayer's multimedia learning theory provides further constraints on instructional design by specifying how information should be presented to align with human cognitive architecture (Mayer, 2021; Clark & Mayer, 2016). Our design follows the principles such as coherence, signaling, and segmentation to reduce extraneous cognitive load and support meaningful processing.

The Technological Pedagogical Content Knowledge (TPACK) framework (Mishra & Koehler, 2006) situates these cognitive principles within a broader systems perspective, emphasizing that effective instruction emerges from the integration of content knowledge, pedagogical knowledge, and technological design.

System Design

We use LangGraph to chain together several LLM invocations, create iteration loops, and run processes in parallel. We use a combination of OpenAI's GPT-5 mini and Google's Gemini 3 Flash as they proved most practical because they are fast, cheap, and easily accessible. We found that other models did not produce results of sufficient quality. OpenAI's and Google's models were practically interchangeable for most agents with the exception of the initial website generation for which OpenAI's results were unsatisfactory.

In a first step, the user-given topic and additional info are provided to a Tutorial Planner to formulate learning goals, compose an outline for the tutorial, and suggest interactive elements.

This plan is then passed to the Website Generator which generates a first draft of the static web page. We decided to separate these two tasks as specialized prompts generally produce better outputs, even though today's LLMs have sufficiently large context windows for much longer prompts. While it is generally best practice to provide the LLM with examples, we found that a complete finished page amounted to more tokens than the rest of the prompt, threatening to overpower our instructions. Instead, we insert an empty code template that includes predefined styles to ensure stylistic consistency.

We then iterate over this draft in a loop where first, the website is checked for syntax errors by ESLint and for runtime errors by Playwright. The latter also takes a screenshot of the page. These are then fed into three parallel critics: a Code Reviewer, a Visual Critic, and Pedagogy Expert. We found that separating this review into three specialized critics drastically improved the ratio of identified issues. The LLMs are prompted to produce a report and also state whether critical issues need to be resolved before publishing the page. If any reviewers identify critical issues, the reports and the code are fed into a Code Fixer and the loop starts over.

Findings, Implications, and Limitations

Preliminary results suggest that the proposed pipeline is capable of producing functional and conceptually coherent interactive tutorials across a range of topics. This novel technology could produce low-cost, individually tailored teaching applications at the learner's convenience, making effective learning more accessible than ever.

Our development experience aligns with the common finding that narrowly specialized agents outperform general-purpose prompts, particularly when reviewing different aspects such as code quality, aesthetics, usability, and pedagogical value. We also observed meaningful variation across 5 OpenAI and 4 Gemini LLMs we tested, with some models excelling at web design and others performing better in visual critique.

We assume that there are many domains for which our interactive learning experiences are not the best method. Because our pipeline aims at increasing understanding, skills that need to be practiced are not suited for our tutorials. We also found that limitations of the browser caused issues when extensive calculations are required (e.g., Fourier transform). Another limitation of our approach is the qualitative testing and evaluation during the iterative prompt engineering process. The low number of test outputs usually exhibited a high within-model variance, making it difficult to objectively and robustly assess the performance of prompts.

Future work should move this project to the next phase of the Learning Engineering Process cycle by implementing and investigating such generated tutorials with real learners.

References

- Chi, M. T. H., & Wylie, R. (2014). The ICAP framework: Linking cognitive engagement to active learning outcomes. *Educational Psychologist*, 49(4), 219–243. <https://doi.org/10.1080/00461520.2014.965823>
- Clark, R. C., & Mayer, R. E. (2016). E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning (4th ed.). Wiley.
- Goodell, J., & Kolodner, J. (Eds.). (2022). Learning Engineering Toolkit: Evidence-Based Practices from the Learning Sciences, Instructional Design, and Beyond. Routledge. <https://doi.org/10.4324/9781003276579>

Koç, A., & Kanadlı, S. (2025). Effect of interactive learning environments on learning outcomes in science education: A network meta-analysis. *Journal of Science Education and Technology*, 34, 681–703. <https://doi.org/10.1007/s10956-025-10015-z>

Li, M., Ma, S., & Shi, Y. (2023). Examining the effectiveness of gamification as a tool promoting teaching and learning in educational settings: A meta-analysis. *Media Psychology*, 26(4), Article 1253549. <https://doi.org/10.1080/15213269.2023.1253549>

Mayer, R. E. (2021). *Multimedia learning* (3rd ed.). Cambridge University Press.

Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record*, 108(6), 1017–1054.

Sharma, S., Mittal, P., Kumar, M., & Bhardwaj, V. (2025). The role of large language models in personalized learning: A systematic review of educational impact. *Discover Sustainability*, 6, Article 243. <https://doi.org/10.1007/s43621-025-00243-7>

Zawacki-Richter, O., Marín, V. I., Bond, M., & Gouverneur, F. (2019). Systematic review of research on artificial intelligence applications in higher education: Where are the educators? *International Journal of Educational Technology in Higher Education*, 16, Article 39. <https://doi.org/10.1186/s41239-019-0171-0>

