

Jul 19, 2026

AskBeevs - How to build your own Online Class System

Summary

The session detailed agent-driven software development workflows and strategies for building custom educational applications with cost-efficient model selection.

Agentic Development Strategy

The team established a workflow using advanced models for planning to optimize expenses before switching to cost-effective models for implementation. This approach successfully produced a functional learning management system using autonomous agent tasks.

Infrastructure and Deployment

Participants examined the deployment of local applications to cloud hosting and discussed hardware trends alongside corporate security limitations. They concluded that sandboxed environments are necessary for secure AI experimentation within corporate settings.

Application Evolution

The project transitioned into refining user interfaces and administrative features, demonstrating that AI can rapidly update visual designs and functionality. The system now supports cohort management and personalized learning through modular, AI-generated curriculum tools.

Next steps

- ☐ [Timothy Beevor] Update Color Mode: Add light and dark mode functionality to the application. This update will ensure the interface is easier to read.

- [Timothy Beevor] Research Readable Fonts: Research the best fonts for readability. Apply the findings to improve the visual design of the application.

Details

- **Introduction to Prompt Engineering Learning Management System:** Timothy Beevor introduces the objective of creating a learning management system from scratch to teach prompt engineering, utilizing AI agents to perform the development work. The project aims to provide an interactive platform where students can log in, access lessons, and complete mini-tests to assess their understanding of prompt engineering skills.
- **Model Strategy and Cost Optimization:** Timothy details a strategic workflow for model selection, noting that while advanced models like Claude Fable 5 are highly capable for complex planning, they are costly at approximately \$50 US per million output tokens. To manage expenses, the team uses these sophisticated models to generate the project plan and then switches to more cost-effective options, such as Grok 4.5 in Cursor, for the implementation phase, which costs roughly \$6 per million tokens.
- **Defining Agent Tasks and Goals:** Timothy outlines the specific instructions provided to the AI agent to initiate the project, which include reviewing a provided document titled "prompt engineering.pdf," suggesting content improvements, and creating a markdown file containing the revised content. The agent is also tasked with designing a learning program that includes clear descriptions of techniques, working examples, and the overall flow of the application to support the development of the software solution.
- **Execution of the Build Plan:** Upon executing the prompt in Claude Code, the agent successfully reads the document and initiates the planning mode. The agent proceeds to generate the four required deliverables: a content review, suggested improvements, a technical approach, and a comprehensive build plan for the application.
- **Evolving AI Landscape and Competition:** The discussion touches on the rapid improvements in model quality, user experience, and speed across various platforms, including OpenAI, Grok, and the emergence of Chinese models like Kimmy K3. Timothy suggests that the competitive nature of the AI industry, where open-source models are increasingly available, is reshaping global technology access and market acceptance.

- **Infrastructure Development and Historical Context:** Timothy and Bruce discuss the deployment strategy, noting that the application is built to run locally before being deployed to Railway. The conversation shifts to the broader implications of data center expansion in Australia, with Timothy drawing a comparison between modern resistance to such technology and the historical "Luddite" movement, while also identifying the potential for sodium batteries to improve energy management and grid efficiency.
- **Application Build Progress:** Timothy demonstrates a prototype of the learning management system, showcasing features such as user authentication, module tracking, and light/dark mode settings. As the AI agent continues working on the current build, it systematically writes the HTML code required to embed content modules and establishes necessary backend structures.
- **Efficiency of Agent-Based Workflows:** Timothy analyzes the resource usage of the project, noting that only 14% of the weekly token limit was consumed, primarily by the planning model. Timothy emphasizes that the industry has shifted from simple prompting to complex, agent-based workflows where the agent acts as a harness for multiple processes, allowing for more autonomous and effective problem-solving.
- **Complex Project Completion with Cursor:** To finalize the remaining features, Timothy migrates the half-built project to Cursor, where the agent quickly identifies missing components and generates the necessary code for complex functionality. The agent efficiently processes the existing files and outlines the remaining work, such as building the instructor cohort features, without manual intervention.
- **Testing and Administrative Functionality:** The session concludes with the system using browser tools to perform local testing of the application, including validating the backend for the administrator to manage cohorts and invite students. Timothy confirms that the system is functional, enabling administrators to oversee user access and manage training cohorts effectively.
- **Environment Configuration and API Setup:** Timothy Beevor demonstrates how to set up an Application Programming Interface (API) key within the Cursor development environment to activate AI functionality. They create a new key specifically for July 19 and update the local environment files to ensure the application can properly interact with the AI model.

- **Prompt Engineering Fundamentals and Quiz:** Timothy Beevor guides the participants through a series of prompt engineering exercises, emphasizing that effective prompting requires treating the AI as a capable system and focusing on clear intent rather than memorized tricks. They discuss the concept of context windows as short-term memory, noting that specificity can compound errors and that exceeding the context window leads to the loss of earlier information.
- **AI Tool Demonstration and Visual Design:** Timothy Beevor showcases the process of using Claude to update an application's interface, requesting a visual style similar to SpaceX to move away from generic aesthetics. They demonstrate that the AI can modify the visual design and feel of the application on the fly without needing to rewrite the underlying functionality.
- **UI Enhancements and Strategic Tool Selection:** Bruce and Timothy Beevor discuss the necessity of adding light and dark mode toggles to improve font readability and overall user experience. They conclude that because different AI tools offer similar capabilities, users do not need to commit to a single platform; instead, they can switch between tools based on temporary deals, costs, or specific feature availability.
- **Corporate AI Training and Coaching:** Timothy Beevor discusses the potential for coaching senior executives on AI by teaching them fundamental building blocks rather than just the tools themselves. They argue that exposing people to multiple tools builds confidence and enables practical application, as many corporate environments currently limit exposure to AI by providing only restricted versions, such as basic versions of Copilot.
- **Corporate IT Security and Infrastructure:** Timothy Beevor addresses the challenge that many corporate IT departments have locked down computers for security, which limits the ability of employees to experiment with creative AI building. They suggest that organizations may need to adopt alternative approaches, such as providing sandboxed environments or secondary machines, to allow staff to explore AI tools without compromising internal data security.
- **Hardware Costs and Compute Capacity:** Timothy Beevor notes that the landscape of AI compute capacity is shifting, with platforms like Chat GPT now having significant spare capacity compared to other providers. They also comment on the rising cost of hardware, noting that MacBook Pro prices have increased from 2,200 to 3,200 for comparable models over the past year.

- **Educational Application and Personalized Tutoring:** Timothy Beevor proposes that students can use AI as a personal tutor to accelerate their learning. They demonstrate how a student, such as Harper Kerr, could prompt an AI to create a custom HTML application designed to teach specific curriculum concepts, such as trigonometry at a year 7 level, allowing the student to move at their own pace and deconstruct educational material effectively.

You should review Gemini's notes to make sure they're accurate. [Get tips and learn how Gemini takes notes](#)

How is the quality of **these specific notes**? [Take a short survey](#) to let us know your feedback, including how helpful the notes were for your needs.