

Using Claude Projects as a Pedagogical Thinking and Course Design Partner

Instructor Name	Tomás Galguera, Professor of Education, College of Professional Studies and Director of AI Curricular Transformation, Mills College
Course Subject	EDUT 6305: Human Resource Management in Education; Faculty Development (MCNU)
Student Level	Graduate / Faculty
Developed by	Tomás Galguera, Special Advisor to the Dean on AI, MCNU; with support from Priyank Bagad, Student

What the Instructor Did

The instructor built a Claude Project to support the development and revision of a course he was teaching. He added course documents as well as a statement of his pedagogical goals and values as Resources in the Project. He leveraged the Project as a thought partner by asking it where his materials were in tension with his goals. He then used the Project to redesign assignments and other course elements in a way that reflected his goals and values. He also used the Project to generate pages for his Canvas course and facilitation notes for class sessions. His work within the Project involved a series of interactive chats over time that contributed to the tool's knowledgebase and capacity to provide context-sensitive suggestions.

Purpose

The purpose of the Project is to serve as a thought partner that can generate prompts and guidance that reflect and build upon previous chats and provided knowledge. Using the Project in this way facilitates critical reflection and self-exploration related to teaching, and thus contributes to course improvements and student learning.

Faculty Reflections

I discovered how to use a Claude project as a thinking partner almost by accident. I didn't start out with a deliberate decision to redesign my teaching, but rather through trying to solve the tedious problem of formatting Canvas pages. I discovered that AI is remarkably good at translating a vision for the student learning experience into an actual design and materials that reflect that vision, all the way down to the working HTML code.

To my surprise, AI did not save me time, but rather it raised my standards. The hours I used to spend typing and formatting are now spent on conversations that help me clarify my thinking and teaching. My courses are the best they have been. I am also thinking harder.

Step-by-Step Directions

A. Create the Project

Step 1	Before creating, have a conversation. Open Claude and describe the course you have in mind, in its current form: what you aim to accomplish, where the prior iteration fell short, what you are uncertain about. Let that conversation run until something clarifies.
Step 2a	Generate your project instructions. In the same conversation, ask Claude to draft project instructions — a description of the course, your role, and the pedagogical principles, curricular commitments, and learning outcomes that should shape every response. Review and revise until the instructions reflect what you actually want.

Step 2b	When you're satisfied with the project instructions, ask Claude to provide them in a markdown (.md) format, since this is the easiest for the LLM to read. The .md file will be available in the artifact window. Save the artifact as an .md file. Upload the .md file into the chat you were previously using.
Step 3a	Create context document. Still in the same conversation, ask Claude to generate a "context document." This is a synthesis of everything the founding conversation surfaced—a working portrait of the course as you currently understand it. It will grow. For now, this is the starting point.
Step 3b	When you're satisfied with the context document, ask Claude to provide them in a markdown (.md) format, since this is the easiest for the LLM to read. The .md file will be available in the artifact window. Save the artifact as a .md file. Upload the .md file into the chat you were previously using.
Step 4	Build the project and upload your files. Create and name your project, add a brief description, paste your instructions into the Instructions field. Then upload in "Files" the context document alongside your syllabus, rubrics, key assignment sheets, and any theoretical frameworks central to the course. If notes from a prior iteration exist, add those too.
Step 5	Start a conversation Open a new conversation within the project. Claude will have access to everything you uploaded and will apply your custom instructions automatically. Five example tasks that can be performed with the project follow.

B. Use Project for Course Planning, Evaluating, and Revising

Step 1 Evaluate and refine your syllabus.

Your syllabus is already in the project. Open a conversation (within the project) and ask Claude to evaluate it not for surface corrections, but for alignment: do the assessments actually reflect learning goals? Is anything redundant? Does the pacing hold across a semester for typical students?

Ask Claude to respond to the syllabus in the role of one of your typical students — provide specific traits, based on experience. What works? What confuses? Does the pacing hold across a semester for students carrying full academic/professional loads — is the workload distributed with a rhythm that is sustainable, or does difficulty cluster in ways that will cost you in week ten? The shift in frame helps uncover what we often miss.

This conversation will be long. Work through workload, sequencing, and reading currency iteratively, so Claude's responses accumulate specificity rather than starting fresh each time.

Step 2 Map the course onto Canvas.

Canvas provides a set of built-in components (assignments, discussions, pages, syllabus, rubrics, modules, etc.). Before building anything, decide which you will use. Then have a conversation with Claude about how your syllabus translates into that structure.

Canvas “Pages” is a flexible open space worth noting: I use it to house individual class sessions. Ask Claude for a bird's-eye view of the entire course before building any individual component. This is where syllabus logic becomes site architecture — and where decisions that the syllabus left implicit get made explicit.

Step 3 Build Canvas pages

With your course architecture mapped, ask Claude to generate structured HTML for each page — module outlines, assignment descriptions, reading guides, discussion framing, or individual class sessions. Because your syllabus and course goals are already in the project, the pages Claude generates are drawn from your actual course rather than generic templates. Paste the output into Canvas's HTML (</>) editor.

Where content should be linked, Claude will indicate it with underlined red text. Complete the links yourself in the standard Canvas editing view, not the HTML editor. Do not skip this step — manually linking every page forces a close read of the entire site, which is when you will catch what is missing, redundant, or inconsistent. It is as much a review process as a technical one.

NOTE: If you are unfamiliar with Canvas's HTML (</>) editor, have a colleague or instructional designer walk you through the paste-and-preview process before your first attempt.

Step 4 Plan individual class sessions.

For each class meeting, describe what you want to accomplish, the relevant readings or prior sessions, and any constraints — time, student energy, competing deadlines. Ask Claude to produce a structured session plan with time markers and facilitation notes. If your project includes reflective notes from prior semesters, Claude will draw on those to flag potential problems before you encounter them.

Check that the session plan is coherent with the corresponding Canvas page — what students see before class should align with what happens during it. Use the Claude project as a sounding board for pedagogical decisions, not just a plan generator.

Step 5	Adapt as the course unfolds As the course progresses, bring real situations to the project: an assignment that is not working as intended, a class session that ran differently than planned, a student question that revealed a gap in the materials. Because the project holds your course history, Claude can respond with specificity rather than generic advice. Ask it to flag when a proposed adjustment runs counter to your stated pedagogical values or course learning outcomes (Step 2). This is where the project functions most distinctively, not as a tool for producing content, but as a thinking partner that knows your course and can hold you to what you said you believed about teaching it.
Step 6	Refine as you go and wrap it up. Add files, update your instructions, and revise your context document as the course evolves. At semester's end, upload student evaluation data and your own reflective notes. Ask Claude to help you identify patterns — what worked, what confused students, where the course underserved its goals — and generate specific revision priorities for the next iteration. The project improves as it accumulates context. What begins as a portrait of the course you planned becomes, over time, a record of the course you actually taught.
Step 7	Course/Project assessment At semester's end, return to the project and ask Claude to evaluate the course from within the accumulated context, including the original intentions, the adjustments made along the way, and the outcomes. Because the project has been a thinking partner throughout, it can identify patterns, tensions, and revision priorities that a fresh-start review would miss. This closes the loop and seeds the next iteration. The course project that documented what you taught becomes the foundation for designing what will be taught next.

Related Materials

- Prompt Bank : Getting Started with Claude Projects for Course Work: [link to Claude_Projects_Prompt_Bank.pdf]

- Getting Started with Claude Projects: A Reference Guide for Faculty: [link to] [Claude_Projects_Handout.pdf](#)
- Claude Projects Faculty Workshop Facilitator Guide: [link to] [Claude_Projects_Workshop_Facilitator_Guide.docx](#)
- Claude Projects Faculty Workshop Slides: [link to] [Salon_Workshop1.pptx](#)