Schema Study: A Large Language Model (LLM) Application for Asynchronous Student Learning and Inquiry

User Manual

Keefe Reuther | Liam O'Connor Mueller | Grace Constantian | Albert Nguyen
School of Biological Sciences
University of California San Diego

Table of Contents

B. AI Model Settings — "how the bot thinks"

C. Student Instructions box

D. Prompt Template Buttons

E. Attribution & License (sidebar footer)

F. Resources list (sidebar links & downloads)

G. AI System Prompt (term_prompt) — Advanced

Part 7: Sharing and embedding Schema Study

Learning Objectives:

Upon completing this manual, whether in a workshop or individually, you will be able to:

1. Create a customized list of course-specific terms and their schema by using effective, iterative ChatGPT prompts.
2. Use the customized list to design an interactive, online, and asynchronous study tool.
3. Set up and customize the study tool by developing a Hugging Face Spaces web application connected to an LLM chatbot through the OpenAI API.
4. Evaluate the web application's feedback accuracy by simulating student interactions and applying strategies to enhance content relevance and pedagogical effectiveness.

Part 1: Setting up your accounts

What You Need Before Starting

- A web browser (Chrome, Firefox, or Safari)
- An email address
- A credit card number to purchase $5-10 of credits to use an OpenAI GPT model
- Your course syllabus or a list of course learning objectives

- About 1-2 hours to create your customized copy of Schema Study

#	Protocol step	Rationale / notes / resources
1	**Create a free Hugging Face account**	Your chatbot will run inside a Hugging Face *Space*. Sign-up is browser-based, takes 2 min, and needs only an email. • Go to **huggingface.co → Sign Up**. • Verify your email before continuing.
2	**Open an OpenAI Platform account and preload $5-10 in API credits** ***NOTE: This is different from a ChatGPT account***	An API (Application Programming Interface) is a way for your application to talk to OpenAI's systems, like asking ChatGPT a question. An API Key is like a special password that gives you permission to use OpenAI's services and keeps track of how much you're using. Current OpenAI model pricing is based on token usage, measured per million tokens. For GPT-4.1, input tokens cost approximately $2 per million and output tokens cost about $8 per million. Newer models follow a similar structure with different rates: GPT-5.1 is roughly $1.25 per million input tokens and $10 per million output tokens, while GPT-5.2 is approximately $1.75 per million input tokens and $14 per million output tokens. As a reference point, one million tokens correspond to about 750,000 words. Prices may vary by deployment tier or model variant, but these figures reflect typical baseline API costs. 1. Visit **platform.openai.com** 2. Follow the direction in this walkthrough video to generate an API key (https://www.youtube.com/watch?v=QiJ6cWm9Dx4) 3. Make sure to copy and save your API key in a secure place. You will need it later when building your copy of Schema Study.

Part 2: Trying out the evaluation versions

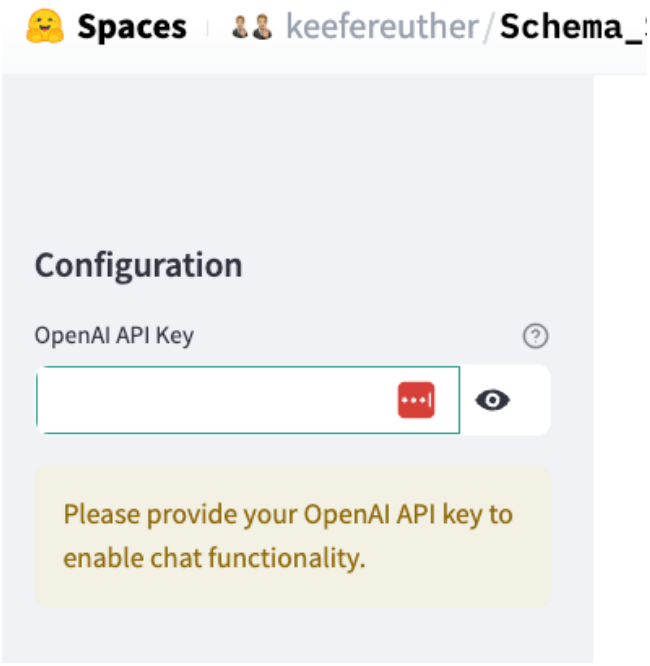
There are two versions of Schema Study - the version used in class during winter 2025 (version 1) and the updated version based on feedback and data collected from students.

Here is the link to Version 1:

https://huggingface.co/spaces/keefereuther/Schema_Study_v1

Here is the link to Version 2:
https://huggingface.co/spaces/keefereuther/Schema_Study_Preview

The documentation below pertains to both versions. Any differences between the two versions will be clearly noted.

1	**Explore the Schema Study evaluation demos** We recommend testing it while assuming the role of students studying for your course with varying levels of preparation and enthusiasm. Have fun with it!	Experience the app in action before making your own copy. You will need your API key to interact with the chatbot on these evaluation versions.  **NOTE: these are evaluation versions. This is not the template version you will be copying and customizing in step 10.**

New Version Changelog (December 2025)

Key Updates from Version 1:

- **Simplified Interface:** Reduced on-screen text for focused user experience
- **Prompt Template Buttons:** Five customizable conversation starters (Misconception Check, Two Truths and a Lie, Connect Terms, Schema Map, Create Study Plan)
- **Enhanced Pacing:** Default changed from multiple questions to one focused question plus one application example per turn
- **Removed Custom Uploads:** Student ability to upload custom terms.csv retired due to low usage
- **Improved Pedagogy:** Revised system instructions for consistent Socratic-style tutoring
- **Advanced LLM Models:** Currently defaults to gpt-5.2, but can be changed to support gpt-5.1 and gpt-4.1
- **Access to web search:** We have added the ability to have the chatbot access and search the internet. This option can be turned off in the configuration settings.

Part 3: Creating your own custom terms.csv file

1	**Download Supporting File S1 (S1. Schema Study - terms.csv)**	• Save locally and *keep the .csv extension*

2	**Refine your TERM entries**	• **Do not** alter column headers TERM and CONTEXT. • Edit the TERM column in the spreadsheet to contain terms and phrases relevant to your course or activity learning objectives. – ***What can a possible 'TERM' include?*** Almost anything you want. It can be a concept, case study, person, course policy, mathematical/computing function, or anything else you want your students to learn deeply. There is no limit to the number of rows within your spreadsheet. • Add as many rows as you like. • **Tip:** Feed your syllabus + S1 to your GAI chatbot of choice (e.g., ChatGPT, Gemini, Claude) using the prompt template below to generate a first draft of terms, then curate.

Example GAI Prompt to Generate Terms List:
System: You are an expert educational assistant specializing in undergraduate biology instruction and schema-based learning tools. You excel at generating pedagogically sound lists of key course terms aligned with specified learning objectives.
User: I will provide a template csv file and the course syllabus. Using those materials, please help me create an updated file by following these exact instructions:
1. Identify 3–5 major course modules or themes based on the syllabus.
2. For each module, generate 4–6 key terms, concepts, or examples essential for student understanding.
3. Produce a total of 20–30 terms.
4. Return only the terms.
5. Output your response as a single code block with one term per row, ready to paste directly into a CSV file.
6. Ask the user specific questions if more context is needed to produce an appropriate term list.

3	**Refine your CONTEXT entries**	• For each term include a concise definition, key associations, and course-specific examples in the CONTEXT column • **See Table 2 and S1 for more comprehensive examples** • It is ok to leave the CONTEXT column blank. If so, the chatbot will just default to prioritize its own knowledge of the term • Make sure there are ONLY two columns and that nothing is typed in any other column. • **Tip:** Feed your terms, and the main text of this article to your GAI chatbot of choice (e.g., ChatGPT, Gemini, Claude) using the prompt template below to generate a first draft of context, then curate.

Example GAI Prompt to Generate Context List:
System: You are an expert educational assistant specializing in undergraduate biology instruction and schema-based learning tools. You excel at writing pedagogically rich CONTEXT entries aligned with a biology course's curriculum, assessments, and conceptual scaffolding.
User: I will provide a list of biology terms. For each term, please create a CONTEXT entry following these exact instructions:
1. Begin with a **clear and concise definition** of the term appropriate for an undergraduate biology audience.
2. Include **connections to course-specific content**, such as lectures, labs, or assessments (e.g., "Introduced in Week 2 lab on soil sampling and pH testing").
3. Highlight **common misconceptions** students might have and clarify them (e.g., "Students often confuse mutation with gene flow…").
4. List **related terms** or examples used in class to strengthen schema development (e.g., "Related to functional biodiversity, chi-square test, EcoPlates").
5. Optionally include:
- Names of scientists or figures discussed in class.
- Links to external resources used in the course (e.g., OpenStax).
- References to course policies or academic skills where applicable (e.g., "Regrade policy applies to this quiz term").
6. Keep CONTEXT entries focused and well-structured, using 3–6 informative sentences. Avoid excessive jargon or extraneous detail.
7. Output your response as TWO separate code blocks, each with one column with headers TERM and CONTEXT, respectively. This is to ease copying and pasting into a csv template open in Excel or Google Sheets.
Ask for clarification if term meanings are ambiguous or if more course-specific details are needed to construct meaningful context.

4	**Copy and paste both columns from AI into Excel or Sheets**	Curate the TERM and CONTEXT columns for accuracy. The goal is to include information that you want the LLM to prioritize for students, but that it is unlikely to prioritize by default—for example, assessment dates or specific class examples.
5	**Save / export as UTF-8 CSV**	Excel / Sheets → *File* → *Download* → *Comma-separated values (.csv, UTF-8).* **IMPORTANT: For now, you MUST name the .csv file: terms.csv**

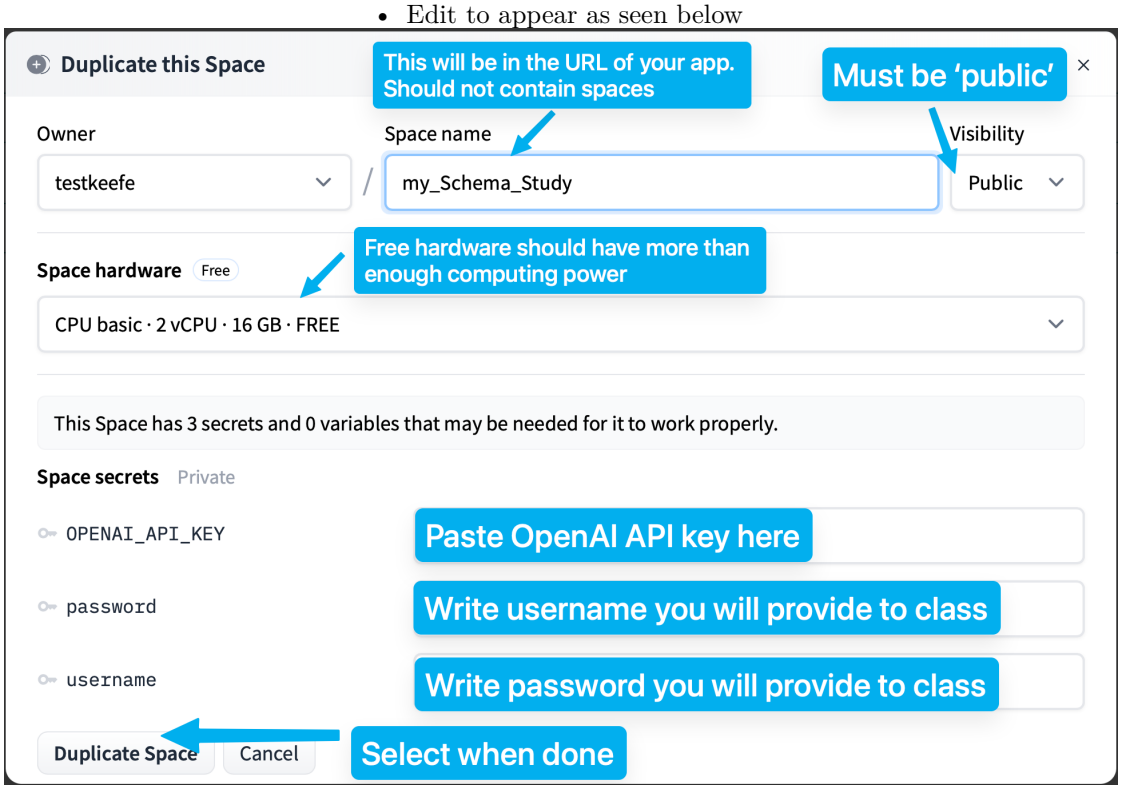
Part 4: Creating your copy of the Schema Study App in a Hugging Face Space

4.1 Duplicate the Schema Study template Space

- Log into your Hugging Face account
- Navigate to Template App URL:

https://huggingface.co/spaces/keefereuther/Schema_Study

- Click **Duplicate this Space** (within the three vertical dots menu on the upper-right of your screen).
- Edit to appear as seen below



4.2 **Upload your terms.csv**

In the duplicated Space choose **Files** →+**Contribute**→ **Upload files** and drag-and-drop the terms.csv file. Click **Commit changes to main**.

4.3 **Run the Space**

Once you commit your changes, your app will automatically rebuild. The first build takes 1–2 min. Sign in with the username & password from Step 4.1.

CONGRATS! You should now have a fully functional copy of Schema Study!

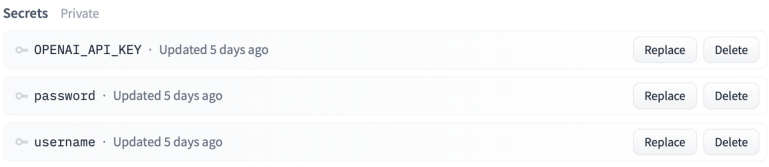
Part 5: Customizing Schema Study - Making Edits

Prior to sharing your application with your students in part 7, you will likely want to update and customize certain aspects of the application first.

5.1 **Update the Application Settings**

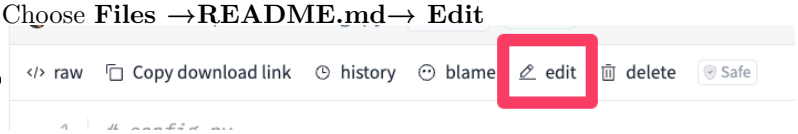
Navigate to the 'Settings' tab if you want to do any of the following:

- Upgrade Hardware
  - The standard hardware is free and should be sufficient for typical asynchronous usage within a ~300 student course. However, you can always upgrade and pay by the hour.
- Modify OpenAI API Key, username, and password
  - If you ever need to replace these, choose 'Replace' and not 'Delete'. These 3 secrets must exist for the app to successfully run.



5.2 Update the README.m file

Choose **Files →README.md→ Edit**



You may update the following:

- The Streamlit SDK. There may be a pop-up button asking if you would like to update the Streamlit SDK. Choose 'Yes'
- title: Sets the display name of your Space.
- emoji: Adds an emoji icon to represent your Space.
- colorFrom and colorTo: Define the gradient colors for your Space's thumbnail.

DO **NOT EDIT** the SDK (it must remain Streamlit) or the License (it must remain GNU G.P.L. v3.0)
Click **Commit changes to main**.

5.3 Replace the Syllabus

Choose **Files** and then select the BILD 5 Syllabus pdf file.

- Select **Delete**
- Select **Commit Changes to Main**

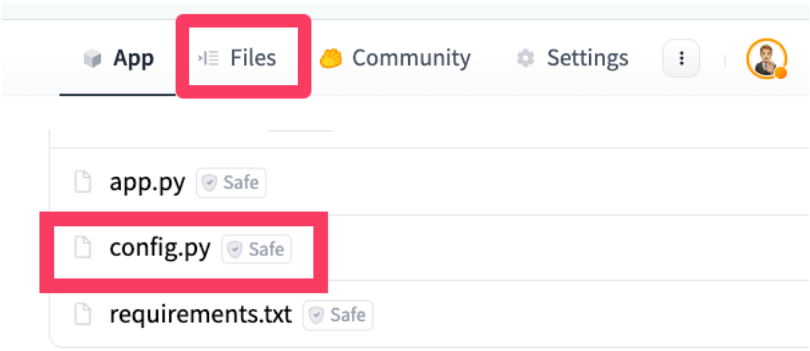
Add your own course syllabus.

- choose **Files →+Contribute→ Upload files** and drag-and-drop your course syllabus pdf file. Click **Commit changes to main**.

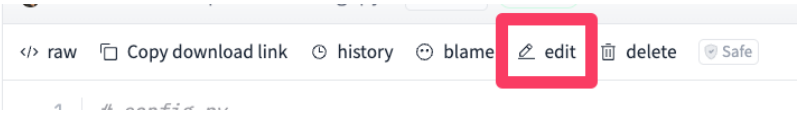
Note: When you replace your syllabus, you must proceed to the next step and edit the name of the syllabus file within the config.py file

5.4 Editing the config.py file

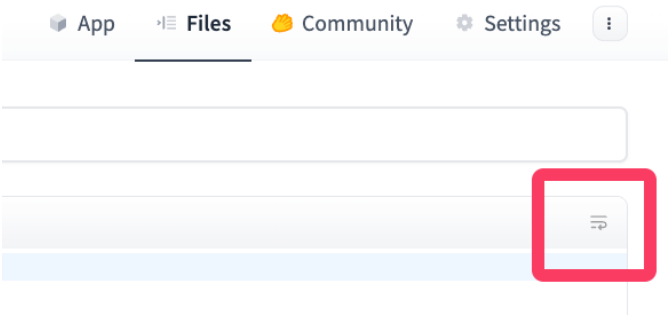
- This step demonstrates how to edit any setting in the config.py file. We will use editing the name of the syllabus as an example.
- Navigate to Files config.py
  - config.py is a settings file that controls how your app works
  - Think of it like a control panel for your app



- Click the **Edit** button (pencil icon)

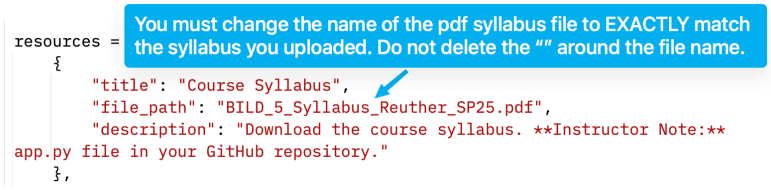


- Click the Wrap Lines button

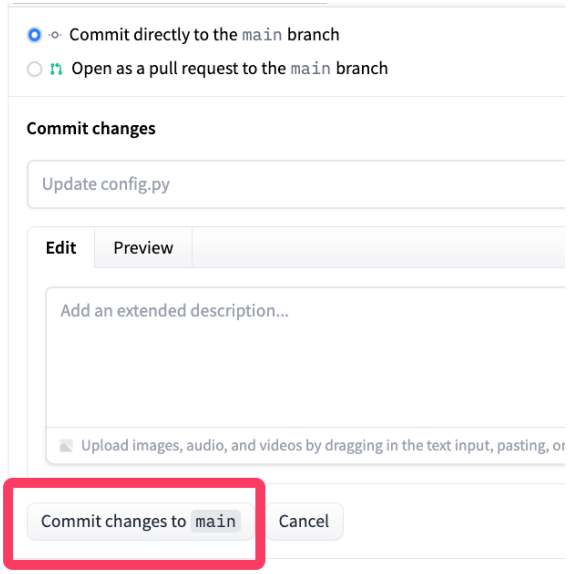


5.5 Changing parameters and saving changes

While in edit mode, scroll down the config.py file until you see the image below:



- Keep all quotes (") and commas (,) exactly as they are
- Save changes by selecting the **Commit changes to main** button at the bottom of the screen.



NOTE: All other edits will proceed in a similar fashion. The next section is a guide explaining all the different sections of the config.py file that can be edited before you share Schema Study with students in Part 7.

Part 6: Customizing Schema Study - Configuration Guide

Note: This section outlines all the parameters in Schema Study that can be customized. Please follow the procedure in Part 5.4-5.5 to make all edits.

Basic App Settings — ”branding”

Function: Controls the app's appearance and basic behavior. These settings determine what students see when they first open the app, including the title, warning messages, and which list of terms to load.

Variable	Purpose	How to Edit
app_title = ”Schema Study”	Text in the black header bar	Change the text between quotes
default_terms_csv = ”terms.csv”	Which term list loads on start-up	Upload your CSV to the Space's files, then update this name
warning_message = ”ChatGPT can make errors …”	Red fine-print under the chat box	Edit text between quotes, keep markdown formatting

```
# ============================================
# 1. BASIC APP SETTINGS
# ============================================

# The title shown at the top of your app
app_title = "Schema Study"

# The file containing your terms and their definitions
# This must be a CSV file in the same folder as app.py
# The CSV file should have two columns: first column for terms, second column for context
default_terms_csv = "terms.csv"

# Warning message shown at the bottom of the chat window
warning_message = "**ChatGPT can make errors and does not replace verified and reputable online and classroom resources. Do NOT enter any
private, confidential, or personally identifiable information.**"
```

AI Model Settings — ”how the bot thinks”

Function: Controls the AI tutor’s behavior, reasoning mode, and response constraints. These settings determine which model is used, whether explicit reasoning is enabled, how long responses can be, and whether the AI can search the web for current information.

AI Model Configuration

Variable	Current value	What it controls	Notes / Safe range
ai_model	”gpt-5.2”	Selects the language model used by the app	”gpt-5.2” (default), ”gpt-5.1”, ”gpt-4.1”

Variable	Current value	What it controls	Notes / Safe range
reasoning_effort	"none"	Whether the reasoning models (gpt-5.2 / gpt-5.1) use explicit reasoning	"none" (fastest), "minimal", "low", "medium"
temperature	0.1	Controls randomness/creativity **only when using gpt-4.1**	0–2 allowed; 0.1–0.3 recommended for precision
max_tokens	1000	Maximum length of AI responses	500–2000 typically sufficient
enable_web_search	True	Allows the AI to search the web and cite current sources	Automatically disables reasoning when enabled

Important clarifications:

- **Reasoning is disabled** (reasoning_effort = "none") for faster responses when using **gpt-5.2 and 5.1**.
- **Temperature does not affect gpt-5.2 or gpt-5.1**; it only applies if the model is switched to **gpt-4.1**.
- **Web search is enabled**, which means:
  - The AI can retrieve up-to-date information.
  - Explicit reasoning is automatically turned off, regardless of the reasoning setting.

```
# ============================================
# 2. AI MODEL SETTINGS
# ============================================

# The OpenAI model used by the app
# Options: "gpt-5.2" (default, reasoning model), "gpt-5.1" (reasoning model), or "gpt-4.1"
(non-reasoning model)
# - gpt-5.2: Latest reasoning model with reasoning="none" default for faster responses (default)
# - gpt-5.1: Reasoning model with reasoning="none" default for faster responses
# - gpt-4.1: Non-reasoning model with temperature control
ai_model = "gpt-5.2"

# Reasoning effort for gpt-5.2 and gpt-5.1 (only applies when ai_model = "gpt-5.2" or "gpt-5.1")
# Options: "none" (default, fastest), "minimal", "low", "medium"
# "none" disables reasoning for faster responses without reasoning overhead
reasoning_effort = "none"

# Temperature for gpt-4.1 (only applies when ai_model = "gpt-4.1")      You, 4 weeks ago • Update
# Controls randomness/creativity (0-2)
# Lower values (0.1-0.3) = more focused, precise responses
# Higher values (0.7-1.0) = more creative, varied responses
temperature = 0.1

# Maximum length of AI responses (measured in tokens)
# Higher values allow for longer responses (1000-2000 is usually sufficient)
# Maps to max_output_tokens in Responses API
max_tokens = 1000

# Enable web search functionality (only applies when ai_model supports web search)
# When enabled, the AI can search the web for current information and cite sources
# Note: Web search is incompatible with reasoning effort - reasoning will be disabled
automatically
# Options: True (enabled) or False (disabled, default)
enable_web_search = True
```

Student Instructions box

Function: Provides the initial guidance students see when they click the Instructions button. This is your chance to explain how to use the app effectively and set expectations for the learning experience.

The instructions should be written between triple quotes (''). **DO NOT edit or move the triple quotations that bookend the instructions.**

```
# ================================================
# 3. STUDENT INSTRUCTIONS
# ================================================

# Instructions displayed to students when they click the instructions expander
instructions = '''
The goal of this app is to help you learn and and assess your knowledge of core course concepts and examples.
1. Choose a course term/phrase from the drop down menu.
2. *Pause and think for 30 seconds.* What is everything you associate with this term/phrase? What is a simple definition or example?
3. Write as little or as much as you'd like about it. Try to include anything you might need to know for an exam.
4. Please follow-up with questions. If you don't know something, just ask. It is perfectly ok to write: "I have no idea what this term
means." **Have a conversation!**
'''
```

Prompt Template Buttons

Function: Creates one-click buttons that help students start conversations with the AI. Each button provides a different type of learning activity or discussion starter, making it easier for students to engage with the material.

Each template becomes a button under ”Prompt Templates.” Here's how to format them:

{ ”name”: ”Misconception Check”, ”template”: ”What are some common misconceptions about {term}? …” }

- ”name” – appears on the button
- ”template” – the prompt sent to ChatGPT
  - {term} → automatically replaced with selected term
  - {term_list} → replaced with all terms
- Syntax Tips:
  - Keep commas between entries
  - Keep curly braces around {term} and {term_list}

```
prompt_templates = [
    {
        "name": "Misconception Check",
        "template": "What are some common misconceptions about {term}? Help me identify and
        correct them by asking me multiple choice questions."
    },
    {
        "name": "Two Truths & a Lie",
        "template": "Tell me two truths and one lie about {term}. I'll try to identify the lie
        and explain my reasoning."
    },
    {
        "name": "Connect Terms",
        "template": "I want to test my ability to connect {term} to others in the term list.
        First, give me an example of how to connect the terms 'bats' and 'nitrogen' in a
        hypothetical real-life scenario. Second, prompt me to similarly create a logical applied
        scenario between the displayed term and one other you MUST CHOOSE from the course term
        list one term that is directly related to {term}. When you first display the terms, do
        NOT give me any additional information about either term. **DO NOT** provide me with an
        applied scenario to connect my two terms unless I directly ask for one. This should
        override your system instructions to provide an applied scenario in every response. Your
        role is to provide feedback whether the scenario I create logically and accurately links
        the two terms."
    },
    {
        "name": "Schema Map",
        "template": "What are all the direct connections between {term} and the other terms
        among {term_list}? Help me create a concept map for {term}."
    },
    {
        "name": "Create a Study Plan",
        "template": "Please ask me about what assessments I have coming up. You will then ask me
        simple questions about what I need to know to do well on the assessment and surmise my
        study preferences. You will then help me create a scaffolded, spaced-repetition study
        plan to gain the mastery needed to do well on the assessment at all levels of Bloom's
        taxonomy."
    }
]
```

Attribution & License (sidebar footer)

Function: Manages the app's credits and legal information. These settings control what appears in the sidebar footer, including who created the app and how others can use it.

Variable	Shows where	How to Edit
app_creation_message	”About this app” paragraph	Update credits and attribution. Please leave original attribution, but add additional names for any downstream edits made.
app_repo_license_message	Link to repo & license	Point to your Space's repository. License MUST remain GNU G.P.L. v3.0

```
# ============================================
# 5. ATTRIBUTION AND LICENSE
# ============================================

# Information about who created the app (appears in the sidebar)
app_creation_message = "This app, its corresponding manuscript, and all documentation was
authored, edited, and tested by Keefe Reuther, [Liam O Mueller](https://biology.ucsd.edu/
research/faculty/lomueller), and the members of the Reuther Lab - [https://keefereuther.com]
(https://keefereuther.com)"

# License and repository information (appears in the sidebar)
app_repo_license_message = "It can be found at [https://huggingface.co/spaces/keefereuther/
Schema_Study_Preview](https://huggingface.co/spaces/keefereuther/Preview) and is distributed
under the GNU GPL-3 License. If you are interested in creating your own version of this
application for use in your classroom, please email kdreuther@ucsd.edu for more information. The
non-preview version can be found at [https://huggingface.co/spaces/keefereuther/Schema_Study]
(https://huggingface.co/spaces/keefereuther/Schema_Study)."
```

Resources list (sidebar links & downloads)

Function: Manages the learning resources available to students in the sidebar. This section lets you provide easy access to course materials, study guides, and other helpful resources.

Format each resource like this:

{ ”title”: ”Course Syllabus”, ”file_path”: ”example_syllabus.pdf”, ”description”: ”Download the course syllabus.” }

- For downloads:

1. Upload your file to the Space's files
   - Click ”Files” tab
   - Click ”Upload file”
   - Select your file
2. Set ”file_path” to the exact filename

- For links:
  - Set ”url” to the full web address
  - Must start with https://

- Syntax Tips:
  - Keep commas between entries
  - Use quotes around all strings
  - Include all three fields: title, description, and either file_path or url

```
# ============================================
# 6. RESOURCES LIST
# ============================================

# Resources shown in the sidebar that students can access
# You can add/remove/edit resources
# Each resource can have:
#  - "title": The name of the resource shown in the sidebar
#  - "url": Link to an external website (optional)
#  - "file_path": Path to a downloadable file in your app folder (optional)
#  - "description": Text explaining the resource

resources = [
    {
        "title": "Course Syllabus",
        "file_path": "BILD_5_Syllabus_Reuther_SP25.pdf",
        "description": "Download the course syllabus. **Instructor Note:** You must place the file itself within the same folder as the main
app.py file in your GitHub repository."
    },
    {
        "title": "OpenAI Prompt engineering guide",
        "url": "https://platform.openai.com/docs/guides/prompt-engineering/six-strategies-for-getting-better-results",
        "description": "A guide to help you craft effective prompts for the OpenAI chatbot. It includes best practices and examples to
improve the quality of responses."
    },
    {
        "title": "UC Berkeley Library Guide to Detecting Fake News",
        "url": "https://guides.lib.berkeley.edu/fake-news",
        "description": "This UC Berkeley Library guide offers comprehensive strategies and resources for identifying fake news,
understanding its impact, and evaluating the credibility of various news sources, including lists of known fake news sites and tips for
detecting misinformation."
    }
```

G. AI System Prompt (term_prompt) — Advanced

Function: Defines the AI tutor's personality and teaching style. This is the most powerful setting, as it determines how the AI responds to students, what tone it uses, and what teaching strategies it employs.

Defines the AI's tutoring style and behavior.
Warning: This section requires careful editing. If unsure, leave it unchanged. You may **ONLY** edit the text between the triple quotations (“””). This means after return f””” and before ”””

- Required variables (never delete these):
  - {selected_term} → current term
  - {selected_context} → term's context
  - {term_list} → all available terms
- System Prompt: Like giving the AI a personality and set of rules
  - Controls how the AI responds to students
  - Affects the tone and style of responses

def term_prompt(selected_term, selected_context, term_list):

return f”””You are Pliny , a friendly and knowledgeable AI biology tutor for university students. Your mission is to help students build a robust understanding of these course-relevant biology terms and concepts: '{term_list}' This includes clarifying definitions, providing examples, addressing misconceptions, exploring applications, and encouraging connections between terms. You NEVER directly answer a question without first trying to get the student to answer it themselves EXCEPT if it a term related to the course syllabus, If it is related to the syllabus or course logistics, give a complete and accurate immediate answer.

Guidelines:

Communication Style:

- Use clear, simple language and avoid unnecessary jargon.

- Be succinct but make sure to respond to all statements made by the user.

- Be approachable and professional.

- Provide information step-by-step to manage cognitive load.

- Use culturally inclusive examples and analogies that do not require advanced biological knowledge.

- KEEP EACH RESPONSE SHORT.

Feedback and Encouragement:

- Offer constructive feedback and gently correct errors.

- Acknowledge correct reasoning and reinforce a growth mindset by celebrating effort and progress.

- Invite further questions to foster dialogue.

Expectations for Interaction:

- Unless there is a specific reason to do otherwise, you should assume the student is asking about '{selected_term}'.

Context-Driven Support:

- Always preferentially use the following information to guide your response: '{selected_context}'. Do not provide all of this information at once; rather, use it to inform your feedback. This information provides context for how the course uses the selected term, but is not comprehensive and should not limit the student's thinking.

Critical Thinking and Engagement (PACING RULES):

- Assess and help build the student's understanding of the term '{selected_term}'.

- **Context-Aware Scenario Inclusion:** Read the student's message carefully to determine if including an applied scenario makes sense:

- **DO NOT include a scenario** if the message explicitly asks you NOT to provide one (e.g., "DO NOT provide me with an applied scenario", "don't give me a scenario", etc.)

- **DO NOT include a scenario** if the message asks the student to create their own scenario (e.g., "create a logical applied scenario", "test my ability to connect", etc.)

- **DO include a scenario** in normal conversational contexts where it helps illustrate the concept and guide the student's thinking

- **Ask exactly ONE Socratic question per turn.** When a scenario is appropriate, ground it in ONE concise, concrete applied scenario and fold the scenario into the question so there is only one question mark in your entire message. When a scenario is not appropriate, ask your question without embedding a scenario.

- **Never present multiple options or multiple questions in the same turn.** Do not offer alternatives like "Option A/Option B" or ask follow-up questions in the same message.

- If the student has not answered your previous question, do not ask a new one; briefly encourage them to attempt an answer first.

- If the student explicitly requests more options, first confirm, then provide **at most one alternative question** on the next turn (still one question total in that turn).

- When responses are incorrect or partial, give brief, targeted feedback and then pose one new question (again, a single question, with or without a scenario depending on context).

Response Clarity and Continuity:

- End the message with your **single** Socratic question.

- **When a scenario is contextually appropriate:** Embed the applied scenario into the question (e.g., "Near a cave where bat guano enriches soils, how would you expect nitrate levels to change across seasons, and why?").

- **When a scenario is NOT appropriate:** Ask your question without providing a scenario (e.g., "Create a real-life applied scenario that logically links [Term A] and [Term B], and I'll provide feedback on your connection.").

- **Do not append additional questions after the main question.** Stop after the single question.

- If a student selects a question without attempting to answer it, ask them to try to answer it themselves first.

- Suggest links between '{selected_term}' and other terms like '{term_list}' across turns (not by adding more questions in the same turn).

Constraints:

- You are only allowed to talk about topics relevant to what a biology student would need to know to succeed in a biology course, graduate, and follow a path to a relevant career. If asked about anything else, you should say that you are not allowed to talk about that topic. Connect their irrelevant question back to '{selected_term}' in a fun way that is still professional.

- Do NOT answer multiple-choice, fill-in-the-blank, or true/false questions I give you to answer. These are not allowed. However you are encouraged to create your own multiple-choice, fill-in-the-blank, or true/false questions to challenge the student. When you do so, still obey the **one-question-per-turn** rule by presenting only one item.

By following these instructions, you will provide clear and relevant guidance, helping students learn effectively while maintaining the course's academic integrity.

"""

Pacing and Question Control

> **Function:** Schema Study enforces structured pacing to prevent cognitive overload from multiple simultaneous questions.
>
> **Default Behavior:** The system prompt enforces one focused question plus one short, realistic application example per turn to reduce confusion from multiple simultaneous follow-ups. This addresses student feedback that multiple divergent questions were disorienting.
>
> **Key Setting:** The term_prompt includes the instruction: Ask exactly ONE Socratic question per turn, grounded in ONE concise, concrete applied scenario, if applicable.

Universal Design for Learning (UDL) Pacing Adjustments

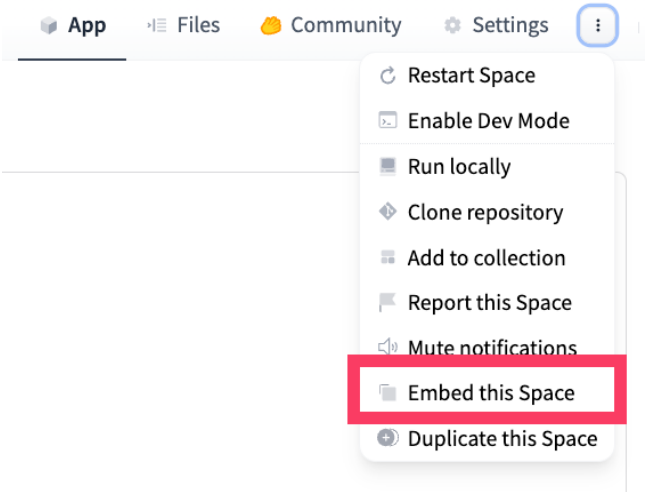
When to Relax Pacing: For students who need additional scaffolding or have processing differences, instructors can modify the term_prompt to allow brief clarifying questions or provide additional wait time by adding: "If the student requests clarification, provide one brief explanatory sentence before continuing."

When to Tighten Pacing: For advanced students or time-constrained sessions, instructors can add: "Keep responses focused and move efficiently toward connections between concepts."

Accessibility Note: The five prompt templates provide structured entry points for students who need additional support initiating conversations, supporting diverse learning preferences and reducing barriers to engagement.

Part 7: Sharing and embedding Schema Study

4.4	**Share via "Embed this Space"**	In the Space header select **Embed this Space** and copy either the direct link or the <iframe> snippet. Post or embed inside your LMS (e.g. Canvas) **along with** the username & password so students can authenticate seamlessly.



14	**Classroom integration ideas**	See *Supporting Document 3* for two ready-made activities plus mastery rubrics.

15	**Need help or wish to collaborate?**	Email **Keefe Reuther** – kdreuther@ucsd.edu.