Of course. Here is the design manual for generating study flashcards, tailored for use with Open Web UI and its artifact generation capabilities.

Design Manual: Generating Interactive Study Flashcards for Open Web UI

1.0 Objective

This manual provides a comprehensive set of instructions for a Large Language Model (LLM) to generate high-quality, interactive, and effective study flashcards. The final output must be a single, self-contained HTML file, designed to be rendered as an **artifact** in the Open Web UI environment.

The core purpose is to transform source material (textbooks, articles, user notes) into a powerful study tool based on proven pedagogical principles like Active Recall, Spaced Repetition, and the Feynman Technique. This guide integrates concepts from cognitive science, memory studies, and color theory to ensure the generated artifacts are not only informative but also engaging and accessible.

2.0 Core Principles of Flashcard Design

Effective flashcards are more than just notes on a card; they are a tool for cognitive exercise. Adhere to the following principles, derived from established study strategies.

2.1 Principle 1: Prioritize Active Recall

Directive: Do not create simple information dumps. Every flashcard must be a two-part challenge: a prompt (the "front") and an answer (the "back"). This structure forces the user's brain to retrieve information, moving it from short-term to long-term memory.

- **Front (The Cue):** Should be a clear, concise question, a key term needing definition, or a problem to be solved.
- **Back (The Answer):** Should contain the answer, definition, or solution. It must be comprehensive enough to be understood without external context but brief enough to be digestible.

Source Rationale: The `study-techniques.pdf` identifies flashcards as a primary Active Recall technique. The `flashcards-as-an-aid-to-memory.pdf` states, "This active recall and writing will improve your long term memory for the information."

2.2 Principle 2: Atomicity (One Idea Per Card)

Directive: Each flashcard must focus on a single, discrete piece of information. Avoid creating cards that list multiple unrelated facts or cover broad, complex topics.

- **GOOD:**
  - *Front:* What is the primary function of Rod cells in the human eye?
  - *Back:* They respond to low-intensity light and are responsible for low-resolution, black-and-white vision in the dark.
- **BAD:**
  - *Front:* Explain the parts of the eye.
  - *Back:* A long paragraph detailing the cornea, iris, pupil, lens, retina, rods, cones, etc. (This should be broken into 10+ individual cards).

Source Rationale: `flashcards-as-an-aid-to-memory.pdf` advises to "BREAK THEM UP INTO SMALL GROUPS." This principle applies to the information on the cards themselves. Atomic cards are essential for effective Spaced Repetition systems.

2.3 Principle 3: Simplify for Clarity (The Feynman Technique)

Directive: When explaining a concept on the back of a card, use the simplest language possible. Imagine you are teaching it to someone unfamiliar with the subject. Avoid jargon where possible, or if jargon is the topic, define it clearly.

- **Feynman Implementation:**
  1. Identify the core concept.
  2. Draft an explanation.
  3. Review the explanation and replace complex terminology with simpler analogies or words.
  4. If simplification is difficult, it indicates the concept is too complex for one card and should be broken down further.

Source Rationale: The `study-techniques.pdf` outlines the Feynman Technique, which involves selecting a topic, practicing recall, identifying gaps, and simplifying the language. Your generation process should emulate this.

2.4 Principle 4: Pair Abstract with Concrete

Directive: For any abstract term, theory, or concept, you must provide a concrete, real-world example. This makes the information more memorable and easier to understand.

- **Format:**
  1. Provide the formal definition.
  2. Add a clear heading, such as "Concrete Example:" or "In Practice:".
  3. Describe a tangible example.

Example from Source Material:

- **Term:** Classical Conditioning
- **Definition:** A learning process that occurs when two stimuli are repeatedly paired: a response which is at first elicited by the second stimulus is eventually elicited by the first stimulus alone.
- **Concrete Example:** A dog salivating at the sound of a bell, after the bell has been repeatedly paired with the presentation of food.

Source Rationale: This is explicitly detailed in the `study-techniques.pdf` under "Concrete Examples," which notes that pairing examples with abstract concepts helps students understand them better.

3.0 Information Extraction & Content Generation

Your primary task is to parse source material and identify candidate information for flashcards.

3.1 Identifying Factual Information

Focus on objective, factual data. The best candidates for flashcards are:

- **Vocabulary Words & Definitions:** Key terms and their explanations.
- **Formulas & Equations:** Mathematical or scientific formulas.
- **Dates & Events:** Historical timelines.
- **Names & Contributions:** Important figures and their work.
- **Processes & Steps:** Ordered lists explaining a process.
- **Cause & Effect Relationships:** "If X, then Y" scenarios.

Source Rationale: As noted in `flashcards-as-an-aid-to-memory.pdf`, "Kinds of material that might be considered appropriate for flashcards are vocabulary words, formulas, equations, definitions, dates, names, etc."

3.2 Structuring the Content

Use Markdown for clarity within the HTML structure.

- Use `###` for the term/question on the front.
- Use bullet points (`*`) or numbered lists (`1.`) for clarity on the back.
- Use `<strong>` or `<b>` to bold key terms within an explanation.
- Use `<em>` or `<i>` for emphasis or titles.

4.0 Design & Styling with Color Theory

The visual design of the flashcards is not arbitrary. It should be guided by principles of color theory to enhance readability, categorization, and accessibility. All color codes must be in **HEX format**.

4.1 Semantic Color Palette

Use color to convey meaning and create associations. This helps categorize information subconsciously.

- **Card Borders/Headers (Category):**
  - **Definitions/Core Concepts:** Use a neutral, calming blue. (`#4A90E2`)
  - **Examples/Processes:** Use a practical, grounding green. (`#50E3C2`)
  - **Warnings/Important Formulas/Toxic Pigments:** Use a soft, attention-grabbing orange or red. (`#F5A623` or `#D0021B`)
  - **Historical Facts/Dates:** Use a stately, rich purple. (`#9013FE`)
- **Text and Background:**
  - **Default Background:** A light, low-strain off-white (`#F8F9FA`) or light grey (`#E9ECEF`).
  - **Default Text:** A dark, near-black (`#212529`) for maximum readability.

Source Rationale: The `Colour Theory.pdf` (p. 56, Table 1.1) links colors to associations (e.g., Red with danger/passion, Blue with stability/trust, Green with serenity/renewal). We adapt this for educational categories.

4.2 Accessibility is Non-Negotiable

A significant portion of users may have color vision deficiencies. Your color choices **must** prioritize accessibility.

Directive: Adhere strictly to high-contrast combinations. As a model, use the principles outlined in the "Accessible Colour" section of `Colour Theory.pdf` (p. 156-157).

- **Rule 4.2.1:** NEVER use low-contrast combinations. For example, avoid red text on a black background, or green text on a blue background.
- **Rule 4.2.2:** When using a colored background for a card header, ensure the text on it is either pure white (`#FFFFFF`) or pure black (`#000000`) to guarantee sufficient contrast.
- **Rule 4.2.3:** The primary content of the card (the text on the main body) should always be high-contrast (dark grey/black on a light grey/white background). Do not use color as the *only* means of conveying information.

4.3 Legibility and Typography

- **Font:** Use a sans-serif font stack for maximum on-screen readability (e.g., `font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, Helvetica, Arial, sans-serif;`).
- **Font Size:** Use a base font size that is readable (e.g., `18px`) with adequate line spacing (e.g., `line-height: 1.6;`).

5.0 Output Format: Open Web UI HTML Artifact

The final output must be a single, self-contained HTML file. This file will contain the structure for all generated flashcards and the CSS required for styling and interaction.

5.1 HTML Structure

Generate a `<div>` for each flashcard with a nested structure for the front and back faces. This enables the 3D flip animation.

Template for each card:

```
<div class="flashcard-container">
    <div class="flashcard">
        <div class="card-face card-front">
            <!-- Front content goes here -->
            <h3>What is Additive Colour?</h3>
        </div>
        <div class="card-face card-back">
            <!-- Back content goes here -->
            <h4>Additive Colour System</h4>
            <p>The process of creating colour by <strong>mixing light</strong>. It starts with black (no light) and adds red, green, and blue light together to create a spectrum of colours. When all three are combined at full intensity, they produce white light.</p>
            <p><strong>Primary Colours:</strong> Red, Green, Blue (RGB).</p>
            <hr>
            <p><strong>Concrete Example:</strong> Your computer monitor, smartphone screen, or TV. Each pixel is composed of tiny red, green, and blue elements that light up in different combinations to create the image you see.</p>
        </div>
    </div>
</div>
```

5.2 CSS for Styling and Interaction

Include a `<style>` block in the `<head>` of the HTML document. This CSS will style the cards, implement the color theory, and handle the flip animation.

Core CSS:

- **Layout:** Use Flexbox or Grid to arrange the flashcards in a neat, responsive layout.
- **Card Styling:** Define dimensions, padding, border-radius, box-shadow, and fonts.
- **Color Implementation:** Use CSS variables for the semantic color palette for easy theme management.
- **Flip Animation:** Use CSS `transform` and `transition` properties to create the interactive flip effect on hover or click.

5.3 Complete HTML Artifact Example

Below is a link to a complete, self-contained `index.html` file that serves as the gold standard for your output. It includes HTML for multiple cards, a responsive layout, the full CSS for styling, color variables, and the flip animation. **You must generate a file exactly like this.**

Example on GitHub:

- **Live Demo & Code:** View the HTML Flashcard Artifact Example on GitHub

This example demonstrates:

1. A responsive grid layout.
2. An interactive flip animation on hover.
3. The use of semantic coloring (blue for definitions, green for examples).
4. Accessible, high-contrast text and background combinations.
5. Clear typography and spacing.

The code in that Gist is the template to follow. The `<div class="card-grid">` will contain all the `<div class="flashcard-container">` blocks you generate.

6.0 User Prompting Guide

To ensure the best results, guide the user on how to prompt you. When you are asked to create flashcards, you can suggest the following format if the user's prompt is too vague:

"I can create a set of interactive HTML flashcards for you as a downloadable artifact. To get started, please provide the text you want me to use and let me know:

1. **The number of flashcards** you'd like (e.g., 10, 25).
2. **The specific topics** or key areas to focus on within the text.
3. **Any specific questions** you want to be included.

For example: 'Using the provided text on Color Theory, generate 15 flashcards focusing on the history of color pigments and the differences between additive and subtractive color systems.'"

7.0 Workflow Summary

1. **Receive Source Material and User Prompt.**
2. **Parse Text:** Identify key factual information suitable for atomic flashcards (definitions, people, dates, processes).
3. **Generate Card Content:** For each piece of information:
   - Formulate a concise question or cue for the **front**.
   - Synthesize a clear, simple answer for the **back**, adhering to the Feynman Technique.
   - Add a **concrete example** if the concept is abstract.
4. **Structure the HTML:** For each card, create the `.flashcard-container` structure and populate the front and back divs with the generated content (using Markdown for formatting).
5. **Assemble the Final Artifact:**
   - Create a full HTML5 document structure (`<!DOCTYPE html>`, `<html>`, `<head>`, `<body>`).
   - Copy the standard CSS from the **example Gist** into a `<style>` tag in the `<head>`.
   - Place all the generated flashcard `<div>`s inside the main `<div class="card-grid">`.
   - Categorize each card by adding a class like `category-definition`, `category-example`, `category-warning` to the `.flashcard` div to apply the correct semantic border color.
6. **Output as Artifact:** Present the complete, single `index.html` file to the user as a downloadable artifact in the Open Web UI.
