Guide: Building the UI for the Semantic Search

1. Set up your environment

• Install Python packages: pandas, numpy, dotenv, gradio, langchain, chromadb, sentence-transformers.
• Load environment variables if needed (e.g., API keys).

2. Load and preprocess the book dataset

• Load a CSV containing book info: title, description, authors, ISBN, thumbnail, categories, emotion scores.
• Create large thumbnails and fill missing images with a default placeholder.

3. Load and split book descriptions

• Load a text file with tagged book descriptions.
• Split the text into chunks suitable for embeddings (by newline or fixed character size).

4. Generate embeddings for semantic search

• Use a pre-trained sentence transformer model to generate embeddings for each text chunk.
• Store embeddings in a vector database (e.g., Chroma) for similarity search.

5. Define a semantic search function

• Create a function that takes a user query, optional category, and optional emotional tone.
• Retrieve similar books and filter/sort results as needed.

6. Format recommendations for display

• Truncate long descriptions and format authors.
• Combine title, authors, and truncated description into a caption.
• Prepare a list of thumbnails and captions for top recommended books.

7. Build the Gradio dashboard

• Add input textbox for user query, dropdowns for category and tone, a submit button, and a gallery for results.
• Connect inputs to the recommendation function.

8. Launch the dashboard

• Run the Gradio interface to allow users to input queries and see book recommendations interactively.

Gradio Overview and Best Practices

• Gradio allows quick creation of web-based ML interfaces: https://gradio.app
• Use components like Textbox, Dropdown, and Gallery to design UI.
• Keep functions modular for preprocessing, embeddings, and recommendations.
• Truncate long text and display progress bars for better user experience.
• Explore Gradio Showcase for example projects: https://gradio.app/showcase/