SHL Assessment Recommendation System

1. Data Collection and Crawling Strategy:

The SHL Product Catalog was crawled programmatically to extract assessment-level metadata.

After crawling, the final dataset contained over 377 individual assessments, satisfying the dataset requirement. Each assessment was stored as a structured JSON object and later used as the foundation for embedding generation.

2. Data Cleaning and Normalization:

Before embedding, I normalized the textual content to reduce noise and improve semantic consistency.

This preprocessing ensured that embeddings reflected semantic meaning rather than formatting artifacts.

3. Embedding Model Selection:

I chose Sentence Transformers model all-MiniLM-L6-v2 for embedding generation.

Each assessment description was converted into a dense vector embedding, producing a numerical representation suitable for similarity search.

4. Vector Indexing with FAISS:

- I used FAISS (Facebook AI Similarity Search) for indexing and nearest-neighbor search.
- Stored all assessment embeddings in a FAISS index
- Inner-product was used as the similarity metric
- This allowed sub-millisecond retrieval even as the dataset size grows
- FAISS provided both speed and accuracy, making it suitable for production-grade recommendation systems.

5. Final Evaluation and Performance Reasoning

After iterative improvements, the system consistently returned relevant assessments within the top-5 to top-10 results across both labeled and unlabeled queries. Dense semantic embeddings combined with FAISS indexing proved effective in bridging vocabulary gaps between job descriptions and assessment descriptions.

6. System Deployment and Usability

The solution was exposed via a RESTful API built using FastAPI, supporting both raw text input and URL-based job descriptions. A lightweight web frontend was implemented to allow interactive testing of the system. The modular design ensures that components such as the embedding model or indexing strategy can be upgraded independently.

7. Efforts put in to refine the product:

I began with a simple semantic retrieval baseline using dense embeddings and FAISS. Initially, I experimented with larger sentence embedding models to maximize accuracy, but they were slower and did not provide meaningful improvements in retrieval quality for this task. I then switched to the all-MiniLM-L6-v2 model, which offered a better balance between semantic accuracy and inference speed, resulting in more stable top-K recommendations.

In early iterations, the system embedded raw scraped assessment text, which included boilerplate and non-informative content. I introduced a text normalization step and applied the same preprocessing to both assessment descriptions and query inputs. This reduced noise and improved semantic alignment between queries and indexed assessments.

I also evaluated different similarity measures and observed that cosine similarity produced more consistent rankings than Euclidean distance, particularly for abstract or skill-based job descriptions. Finally, instead of excluding assessments with incomplete metadata, I retained all individual test solutions and handled missing fields at the output stage, which improved recall.

Each change was validated using the labeled training dataset, and only optimizations that increased Recall@K were retained. These iterations collectively improved both relevance and robustness while keeping the system efficient.