SHL-GenAI-Assessment-Recommender

1. Solution Overview

This system recommends relevant **SHL Individual Test Solutions** from natural-language hiring queries or job descriptions.

It follows a **retrieval-augmented generation (RAG)** approach and combines:

- structured catalog data,
- hybrid retrieval (semantic + keyword),
- and GenAI-based ranking.

2. Data Ingestion and Catalog Preparation

Catalog Collection

- The SHL product catalog was collected using a **Python-based crawler built with BeautifulSoup**.
- Only **Individual Test Solutions** were retained.
- Extracted fields include:
    - assessment name and URL
    - test type codes
    - remote testing and adaptive (IRT) indicators

Metadata Enrichment

- Each assessment's detail page was processed to extract:
    - description
    - job level suitability
    - supported languages
    - assessment duration
- Retries and timeouts were used to ensure reliable data collection.

Dataset Normalization

- Durations were converted to numeric minute values.
- Test type codes were mapped to semantic categories.
- Boolean and list fields were standardized.
  The final dataset was stored as a **canonical JSON artifact** and reused across all stages of the pipeline.

3. Retrieval-Augmented Recommendation Pipeline

3.1 Embedding and Hybrid Retrieval

- All assessments were embedded **offline** using **Gemini** `gemini-embedding-001`.

- Embeddings were generated from combined metadata:
    - name, description, test types
    - skills, job level, duration
- Embeddings were indexed using **FAISS (cosine similarity)**.

To complement semantic retrieval:

- A **BM25-based lexical retriever** was used to capture exact keyword matches (e.g., skills like SQL, Java, or duration constraints).

Results from FAISS and BM25 are merged and deduplicated to form a candidate set.

3.2 Query Understanding with GenAI

Query intent is extracted using **Gemini** `gemini-2.5-flash-lite` with schema-constrained JSON outputs.
For each query, the system extracts:

- required technical and soft skills
- maximum assessment duration
- job level
- required assessment test types

Using a lightweight model keeps intent extraction fast and cost-efficient.

3.3 Ranking and Selection

- Candidate assessments are reranked using **Gemini** `gemini-2.5-flash`.
- Ranking prioritizes:
    - test type alignment
    - role relevance
    - duration and job-level constraints

4. Evaluation Methodology

- The labeled training dataset was used during development to tune:
    - retrieval depth
    - intent extraction prompts
    - reranking behavior
- For final submission, the system generated **top-10 recommendations** for each unlabeled test query.
- Outputs were exported in the **exact CSV format** required by SHL's evaluation pipeline.

5. API and Deployment

- The system is exposed via a **FastAPI service**:
  `/health` – service status
  `/recommend` – assessment recommendations from natural-language queries
- **FAISS and BM25 artifacts** are loaded at startup for low-latency inference.
- A React frontend UI is created for interactive evaluation.