

SHL GenAI Assessment Recommender – Approach Summary

Objective

The goal of this project was to develop a Generative AI-powered web tool that helps recommend the most suitable SHL assessments based on any job description. This was achieved using a Retrieval-Augmented Generation (RAG) technique that combines semantic search with intelligent ranking.

Tools & Technologies

- **Frontend:** Streamlit (for interactive web UI)
 - **Backend API:** FastAPI (for recommendation endpoint)
 - **Language:** Python
 - **Embedding Models:**
 - all-MiniLM-L6-v2 (via Sentence Transformers – local model)
 - Optional: OpenAI's text-embedding-ada-002 (via API key)
 - **Similarity Method:** Cosine Similarity
 - **Dataset:** shl_catalog.csv – a curated list of SHL assessments and their details
-

Architecture Overview

1. **User Input:** Job description is entered via the web app or API.
2. **Embedding:**
 - If selected, OpenAI API generates the embedding for the input.
 - Otherwise, local SentenceTransformer model generates it.
3. **Similarity Calculation:** Computes cosine similarity between the input and each assessment description.
4. **Top Recommendations:** The system ranks and returns the top 3 matching assessments (or up to 10 via API).
5. **Output Delivery:**
 - The Streamlit app displays the results clearly to the user.
 - The FastAPI endpoint returns the response in a clean, SHL-compliant JSON format.

Highlights

- Switch easily between local model and OpenAI embeddings
- Output format follows SHL's official JSON guidelines
- Both web app and API are designed to be scalable and easy to integrate

Key Links

- **Web App:** <https://shl-genai-assessment-recommendation-tool-iqgrsw5dgsijvlqg4kbvv.streamlit.app/>
 - **GitHub Repository:** <https://github.com/iris3014/shl-genai-assessment-recommendation-tool>
-

Output



