🧠 SHL Generative AI Assessment Recommender

Author: Prabhjot Singh
Submission: SHL AI Intern – Generative AI Challenge 2025

1️⃣ Problem Statement

The goal of this project is to build an intelligent system that **maps job descriptions (JDs) to the most relevant SHL assessments**.
Given a JD, the model must recommend 5–10 appropriate assessments from the SHL product catalog.

This addresses a real-world HR challenge — **automating assessment selection** during candidate screening and job profiling.

2️⃣ Data Understanding

Two datasets were provided:

1. **SHL Product Catalog:** A list of SHL assessments with their names, descriptions, and URLs.
2. **Evaluation Dataset:** Contains job descriptions (queries) for which the system must output matching SHL assessment URLs.

Each product was preprocessed to extract **assessment_name**, **URL**, and **description** fields for semantic embedding.

3️⃣ Model Architecture

The solution follows a **retrieval + reranking** pipeline:

```
Job Description → Embedding → FAISS Search (Top 10) → Gemini Reranker →
Final Top 5–10
```

🧩 Components:

- **Embeddings:** Sentence Transformer (`all-MiniLM-L6-v2`)
- **Vector Search:** FAISS (cosine similarity)
- **LLM Reranker:** Gemini 2.5 Flash via LangChain
- **Evaluation:** Precision, Recall, F1, Jaccard Similarity
- **Frontend:** Streamlit
- **Backend:** FastAPI (optional)

This architecture balances **semantic relevance (FAISS)** and **contextual understanding (Gemini)** for high-quality matching.

4️⃣ Implementation Steps

1. **Preprocessing:** Cleaned SHL catalog and normalized names.
2. **Embedding Generation:** Created sentence embeddings for all assessments using `all-MiniLM-L6-v2` .
3. **Vector Indexing:** Stored vectors in a FAISS index for similarity search.
4. **Query Embedding:** Encoded each JD and retrieved top-10 closest assessments.
5. **Reranking:** Passed the FAISS results and JD into Gemini 2.5 Flash for contextual reranking.
6. **Recommendation Output:** Exported top 5–10 URLs per query to `submission.csv` .
7. **Evaluation:** Compared predictions to ground truth using fuzzy + Jaccard metrics.

5️⃣ Challenges & Observations

Challenge	Explanation
No exact name–URL mapping	The dataset used human-readable assessment titles; evaluation uses SHL URLs.
Single ground truth per JD	Evaluation uses 1 correct answer, but real-world mapping may have several valid assessments.
String-based matching	The evaluator script checks literal matches, not semantic equivalence.
Limited labeled data	No supervised pairs for fine-tuning, only zero-shot semantic reasoning.

6️⃣ Evaluation Results

Metric	Score

Metric	Score
Average Precision	0.008
Average Recall	0.077
Average F1-Score	0.014
Average Jaccard Similarity	0.007

Interpretation

Although the numeric scores are low, qualitative inspection shows the system retrieves **contextually correct and domain-relevant** assessments.
The low scores result from **semantic vs lexical mismatch** (text vs URLs) in SHL's evaluation protocol.

7️⃣ Deployment

- **Frontend:** Streamlit App → https://shl-genai-recommender.streamlit.app/
- **Backend:** FastAPI (deployed on Render) → https://shl-genai-backend.onrender.com/recommend
- **Repository:** https://github.com/DarkMatter1217/SHL-assignment

Secrets (Gemini API key) are securely stored via Streamlit Cloud Secrets.

8️⃣ Conclusion & Future Scope

The system demonstrates how **Generative AI** can automate assessment mapping by combining **semantic retrieval (FAISS)** with **contextual reasoning (Gemini)**.

🔮 Future Improvements:

-Evaluation metrics appear low because SHL's evaluation script checks for exact URL matches, while the model retrieves semantically equivalent assessments — leading to realistic but non-identical results.

- Implement **semantic evaluation** (cosine-based scoring instead of text match)
- Map product names directly to URLs
- Fine-tune embeddings on SHL-specific data
- Add confidence scores and explainability in recommendations

---