SHL Assessment Recommendation System – Approach Overview

Problem Statement

Hiring managers often struggle to select suitable SHL assessments based on job descriptions or unstructured natural language queries. The challenge lies in understanding and mapping the intent behind a job description to relevant assessments within the **SHL Product Catalog**, which contains hundreds of diverse assessments.

Objective:
Develop an intelligent recommendation system that automatically suggests the **top 1–10 most relevant SHL assessments** based on a user-provided input (e.g., job description or role-based query). The system should support real-time interactivity, ensure relevance, and maintain traceability to original source data.

Tools & Technologies Used

Category	Tools & Libraries
Scripting & Logic	Python
Web Scraping	Selenium
Data Handling	Pandas, LangChain CSVLoader
Vector Embeddings	Sentence-Transformers (`all-MiniLM-L6-v2`)
Vector Indexing	FAISS (Facebook AI Similarity Search)
LLM & Embeddings	Hugging Face Transformers, `mistralai/Mistral-7B-Instruct-v0.3`
UI Development	Streamlit

End-to-End System Workflow

1. Data Collection – Web Scraping SHL Catalog

- **Tool Used:** Selenium

- **Source:** SHL Product Catalog
- **Extracted Fields:**
    - Product Name
    - Product URL
    - Remote Testing Availability
    - Adaptive/IRT Support
    - Test Duration
    - Assessment Type
    - Target Job Role
    - Detailed Description
- **Output File:** `shl_product_catalog.csv`
- **Purpose:** Creates a structured dataset for downstream retrieval and processing.

---

2. Embedding & Vector Store Creation

- **Data Loader:** LangChain `CSVLoader` to convert CSV into retrievable document format.
- **Embedding Model:** `sentence-transformers/all-MiniLM-L6-v2`
- **Indexing:** FAISS used for building a local vector store (`vectorstore/db_faiss`) enabling fast similarity search.

---

3. Retrieval-Augmented Generation (RAG) Pipeline

- **Retriever:**

    - Fetches top-3 most semantically similar assessment entries from FAISS based on the query embedding.

- **LLM Component:**
    - Hugging Face-hosted `mistralai/Mistral-7B-Instruct-v0.3`
    - Custom prompt engineering ensures:
        - Relevance
        - Factual accuracy
        - Consistent tabular output format
- **Query Processor:**
    - LangChain's `RetrievalQA` module combines retrieved results with the user query to generate contextual and structured answers.

---

4. Interactive Web Interface

- **Frontend Tool:** Streamlit
- **Features:**
    - Simple input box for users to paste a job description or enter a free-text query.
    - Real-time recommendations (top 1–10) with:
        - Assessment Name
        - URL
        - Duration
        - Remote Testing Support
        - Adaptive/IRT Support

        - Description

📈 Evaluation & Accuracy Metrics

- **Relevance Optimization:**
    - RAG ensures that recommendations are grounded in the actual SHL catalog.
- **Prompt Engineering:**
    - Designed to reduce hallucinations and enforce output structure.
- **Quantitative Metrics:**
    - **Mean Recall@3**: Measures how often the true relevant assessments are retrieved in the top-3.
    - **MAP@3 (Mean Average Precision)**: Measures overall ranking quality of returned assessments.

Deployment & Deliverables

- **Live Demo:**
  https://shlrecommendersys.streamlit.app/
- **API Endpoint:**
  https://shl-recommender-byqr.onrender.com
- **GitHub Repository:**
  https://github.com/SharmaKanishkaa/SHL-Assessment