SHL Assessment Recommendation System – Approach Document

Problem

Hiring managers face difficulty identifying the most suitable SHL assessments for specific roles using traditional keyword filters. The goal is to build an intelligent recommendation system that returns relevant SHL assessments based on a natural language job description.

Tools & Libraries Used

Data Scraping: Selenium

Embedding Model: Sentence Transformers (`all-MiniLM-L6-v2`)

Similarity Metric: Cosine Similarity

Backend: FastAPI

Frontend: Streamlit

Storage: CSV + Numpy (for embeddings)

Deployment: Streamlit Cloud (frontend), Render (API)

Approach

1) **Data Collection**

- ✓ Scraped the SHL product catalog using Selenium.
- ✓ Collected fields: Assessment Name, URL, Duration, Test Type, Remote/Adaptive Support.
- ✓ Stored structured data as a CSV file.

2) **Text Embedding & Preprocessing**

- ✓ Combined each assessment's metadata into a full-text representation.
- ✓ Used Sentence Transformers (`all-MiniLM-L6-v2`) to embed the text into dense vectors.
- ✓ Stored the resulting embeddings for efficient retrieval.

3) **Query Handling & Recommendations**

- ✓ The user query is embedded using the same model.
- ✓ Cosine similarity is used to retrieve the top-K most relevant assessments.
- ✓ Filters such as duration or test type are applied (if inferred from query).

4) **Evaluation**

- ✓ Accuracy is measured via `Recall@3` and `MAP@3` on benchmark queries.
- ✓ Initial experiments showed strong alignment between job roles and retrieved tests.

5) **Frontend & API**

- ✓ Developed an interactive Streamlit frontend for demo.
- ✓ Provided a `/recommend` API endpoint using FastAPI, accepting a query and returning JSON.

Demo Links

Working Demo: https://shl-assignment.streamlit.app/

API Endpoint: https://shl-assignment-b4f9.onrender.com/recommend

GitHub Repo: https://github.com/Manas-07-24/SHL-Assignment