

# Approach to Building the SHL Assessment Recommender System

---

## Problem Overview:

The goal of this project was to create a recommendation system that efficiently suggests relevant SHL assessments based on a job description or a natural language query. The system was designed to provide detailed information about the assessments, such as test types, remote testing support, adaptive testing, duration, and a link to the SHL catalog.

## Solution Overview:

1. **Web Scraping:** I utilized **Playwright** to scrape the SHL product catalog pages. This allowed me to extract relevant information such as:
  - Assessment Name and URL , Remote Testing Support (Yes/No) , Adaptive/IRT Support (Yes/No) , Duration (in minutes) , Test Types (e.g., Cognitive, Personality, etc.)
  - The data was stored in a CSV file for later analysis and used as a database for matching job descriptions to assessments.
2. **Embedding Generation & Similarity Computation:**
  - To compare job descriptions and assessments, I employed **Sentence-BERT** from the **Sentence-Transformers** library. This model was used to generate embeddings for both the input job descriptions and SHL assessment descriptions.
  - **Cosine Similarity** was calculated between the job description embedding and each assessment's description, which helped to score the assessments based on their relevance.
3. **LLM Enhancement with Gemini API:**
  - To enhance the job descriptions and improve the accuracy of the recommendations, I integrated the **Google Gemini API**. This API helped me: Generate a structured summary of the job description. , Extract relevant **metadata** (e.g., Job Family, Job Level, Industry). Recommend suitable **test types** for the job.
  - The enriched job description and metadata were used to refine the recommendations by boosting the relevance score of assessments that matched the extracted metadata.
4. **Filtering and Recommendation:**
  - The recommendations were filtered based on the job description's **duration** requirement and **test types**.
  - The top **K** assessments (min 1, max 20) were selected based on their similarity to the job description and metadata match, ensuring that they met the user's specifications (e.g., duration, test types).

## 5. User Interface and Deployment:

- The final application was deployed using **Streamlit** to provide an easy-to-use web interface. The user can enter a job description, set a duration filter, and view a list of relevant assessments.
  - Additionally, an API endpoint was created that accepts a job description or query as input and returns the recommended assessments in **JSON format**.
- 

## Tools and Libraries Used:

1. **Playwright**: For web scraping to collect assessment details from SHL's product catalog.
  2. **Sentence-Transformers**: To generate embeddings for job descriptions and SHL assessments for similarity-based ranking.
  3. **Google Gemini API**: To enhance job descriptions by extracting structured metadata and test type recommendations.
  4. **Pandas**: For data manipulation and filtering recommendations based on user input.
  5. **Streamlit**: To create a simple web interface for users to input job descriptions, view recommendations, and apply filters.
  6. **Scikit-learn**: For calculating cosine similarity between job description and assessment embeddings.
- 

## Conclusion:

This approach uses a combination of web scraping, semantic embedding generation, LLM enhancements, and similarity scoring to build an intelligent recommendation system. The system provides accurate and relevant SHL assessment recommendations based on user input, improving the efficiency of the hiring process. The application is deployed with **Streamlit** for ease of use and provides a backend API for querying assessment recommendations