Approach to Solving the Problem(Steps)

1. Problem Definition

The goal of the system is to recommend relevant assessments based on a user query and evaluate the quality of recommendations using Precision@K and Recall@K.

2. Data Preparation

Relevant Documents (relevant_docs):

A list of predefined assessments is created, containing metadata such as name, description, job_levels, duration, test_type, remote, adaptive / IRT, and url.
These documents serve as the ground truth for evaluation and is used for calculating precision and recall.

Document Catalog (shl_catalogue.json):

A JSON file(containing SHL's product cataloguel) containing all available assessments is loaded into the system to build the recommendation engine.

3. Embedding and Indexing

Embedding Generation:

The generate_embeddings function converts the textual content of the documents into numerical vectors using a pre-trained embedding model.
These embeddings capture the semantic meaning of the documents.

FAISS Index Creation:

The create_faiss_index function builds a FAISS (Facebook AI Similarity Search) index using the generated embeddings.
This index enables fast similarity searches for user queries.

Index Preparation:

The prepare_index function loads the document catalog, generates embeddings, and creates the FAISS index.
The function is cached using @st.cache_resource to avoid redundant computations.

4. Query Handling

User Input:

A text input box is provided for the user to enter a query (e.g., "Test for mid-level engineers with coding skills").
The query is processed to find the most relevant documents.

5. *Recommendation Retrieval:*

The get_recommendations function searches the FAISS index using the query embedding to retrieve the top K most similar documents.
The metadata of the recommended documents is extracted for display.

6. *Recommendation Display*

Streamlit Interface:
The recommended documents are displayed in a user-friendly format using Streamlit.
Each recommendation includes:
Name
Description
Test Type
Job Levels
Duration
Remote Testing
Adaptive/IRT
URL (as a clickable link)

Evaluation of the Recommendation System

Metrics Used

Precision@K:

Precision@K is the proportion of relevant assessments among the top K recommendations.
Formula: [\text{Precision@K} = \frac{\text{Number of Relevant Documents in Top K}}{K}]
A higher Precision@K indicates that the system is returning more relevant results in the top K recommendations.

Recall@K:

Recall@K is the proportion of all relevant assessments that are retrieved in the top K recommendations.
Formula: [\text{Recall@K} = \frac{\text{Number of Relevant Documents in Top K}}{\text{Total Number of Relevant Documents}}]
A higher Recall@K indicates that the system is retrieving a larger portion of the relevant assessments.

Evaluation Process

The evaluation process involves the following steps:

Define Relevant Documents(ground_truth):

A list of relevant assessments (relevant_docs) is predefined. Each document contains metadata such as name, description, job_levels, duration, and test_type. These documents act as the ground truth for evaluation.

- ➔ This is present in app_test.py as a ground truth for evaluation..

Retrieve Recommendations:

Based on the user query, the system retrieves the top K recommendations using the FAISS index.
Each recommendation includes metadata such as name, description, test_type, and url.

Normalize Data:

Both the relevant documents and recommended documents are normalized to ensure accurate comparison. This includes:
Converting text to lowercase.
Stripping extra spaces.

Compare Recommendations with Relevant Documents:

The system compares the normalized name field of the recommended documents with the relevant documents to identify matches.
Calculate Precision@K and Recall@K:

Precision@K is calculated as the ratio of relevant documents in the top K recommendations to the total number of recommendations (K).
Recall@K is calculated as the ratio of relevant documents in the top K recommendations to the total number of relevant documents.Evaluation Process

The evaluation process involves the following steps:

Define Relevant Documents:

A list of relevant assessments (relevant_docs) is predefined. Each document contains metadata such as name, description, job_levels, duration, and test_type. These documents act as the ground truth for evaluation.
Retrieve Recommendations:

Based on the user query, the system retrieves the top K recommendations using the FAISS index.
Each recommendation includes metadata such as name, description, test_type, and url.

Normalize Data:

Both the relevant documents and recommended documents are normalized to ensure accurate comparison.
This includes:

- Converting text to lowercase.
- Stripping extra spaces.
- Compare Recommendations with Relevant Documents:

The system compares the normalized name field of the recommended documents with the relevant documents to identify matches.
Calculate Precision@K and Recall@K:

Precision@K is calculated as the ratio of relevant documents in the top K recommendations to the total number of recommendations (K).

Recall@K is calculated as the ratio of relevant documents in the top K recommendations to the total number of relevant documents.

Future Enhancements

Improved Recommendation Accuracy:

Fine-tune the mistralai/Mistral-7B-Instruct-v0.1 model on SHL-specific data to provide more precise and context-aware recommendations.

Enhanced User Interface:

Add filters (e.g., job level, test type) and visualizations to allow users to refine and better understand recommendations.

Scalability:

Deploy the system on cloud platforms like AWS or Azure to handle larger datasets and support more users simultaneously.

Multi-Language Support:

Extend the system to support queries and recommendations in multiple languages for a global audience.

Personalization:

Introduce user profiles to provide personalized recommendations based on past interactions and preferences.