Gen AI Task: Build an SHL Assessment Recommendation System

Problem Overview

Hiring managers often struggle to find the right assessments for the roles that they are hiring for. The current system relies on keyword searches and filters, making the process time-consuming and inefficient. Your task is to build an **intelligent recommendation system** that simplifies this process. Given a **natural language query or a job description text or URL**, your application should return a list of relevant SHL assessments. You can take a look at the data sources that you are going to work with here, https://www.shl.com/solutions/products/product-catalog/

Your Task

Design and develop a web application that:

1. Takes a given natural language query or job description text URL
2. Recommends at most 10 (min 1) most relevant Individual test solutions from *here* in the tabular format
3. Each recommendation needs to have at least the following attributes
   - Assessment name and URL (linked to SHL's catalog)
   - Remote Testing Support (Yes/No) and Adaptive/IRT Support (Yes/No)
   - Duration and Test type

Submission Materials

You need to submit the following items using this form.

- Three URLs, 1) First URL where you can host the working demo (we type in queries and see the result) 2) API end point which can be queried using a query or piece of text and returns result in JSON (API Configurations mentioned below in Appendix 2) 3) URL of the code on GitHub which we can see

- 1-page document outlining your approach on how you solved this problem. Write this document as concisely as possible with appropriate information. Highlight the tools, libraries that you have used.

Evaluation Criteria

We will be using the following criteria to evaluate your solution.

- Approach: The approach that you took to solve this problem. How you have crawled, represented, stored, and searched the data? How much of emerging LLM stack that you leveraged? We love evals and tracing -- would like to see how you have leveraged those.

- Accuracy: Accuracy on a benchmark set. Measured using Mean Recall@3 and MAP@3.

- Demo quality: The quality of the end-to-end demo and attention to the details. It is OK if you do not have front-end skills. You can use low-code frameworks like Streamlit, Gradio.

Resources

You are not restricted to use these – feel free to use from anywhere else. Number of cloud platforms allow you to host applications and APIs for free for some time. Feel free to leverage those.

1. LLMs/Gemini Free APIs: *https://ai.google.dev/gemini-api/docs/pricing*

Index: Metrics to compute accuracy

Your solution will be assessed using the following **ranking evaluation metrics**:

1. Mean Recall@K

This metric measures how many of the **relevant assessments** were retrieved in the **top K recommendations**, averaged across all test queries.

$$Recall@K = \frac{\text{Number of relevant assessments in top K}}{\text{Total relevant assessments for the query}}$$

$$MeanRecall@K = \frac{1}{N} \sum_{i=1}^{N} Recall@K_i$$

where **N** is the total number of test queries.

2. Mean Average Precision @K (MAP@K)

MAP@K evaluates both the **relevance** and **ranking order** of retrieved assessments by calculating **Precision@k** at each relevant result and averaging it over all queries.

$$AP@K = \frac{1}{\min(K, R)} \sum_{k=1}^{K} P(k) \cdot rel(k)$$

$$MAP@K = \frac{1}{N} \sum_{i=1}^{N} AP@K_i$$

where:

- **R** = total relevant assessments for the query
- **P(k)** = precision at position **k**
- **rel(k)** = 1 if the result at position **k** is relevant, otherwise 0
- **N** = total number of test queries

A higher **Mean Recall@K** and **MAP@K** indicate a better-performing recommendation system.

Appendix 1: Metrics to compute accuracy

Here are some of the queries that you can use to test your application.

- I am hiring for Java developers who can also collaborate effectively with my business teams. Looking for an assessment(s) that can be completed in 40 minutes.

- Looking to hire mid-level professionals who are proficient in Python, SQL and Java Script. Need an assessment package that can test all skills with max duration of 60 minutes.

- Here is a JD text, can you recommend some assessment that can help me screen applications. Time limit is less than 30 minutes.

- I am hiring for an analyst and wants applications to screen using Cognitive and personality tests, what options are available within 45 mins.

Appendix 2: API Structure & Endpoints

This section outlines the required API configuration for the take-home assignment. Your API must implement the endpoints described below exactly as specified to ensure proper evaluation of your submission.

Base requirements:

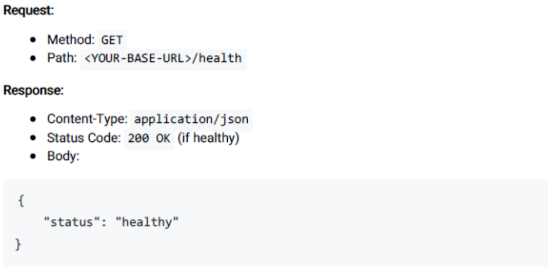
- Your API should be accessible via HTTP/HTTPS

- All responses should use proper HTTP status codes

- All data exchanges must be in JSON format

Required Endpoints:

1. **Health Check Endpoint:** This endpoint provides a simple status check to verify the API is running.



Request:

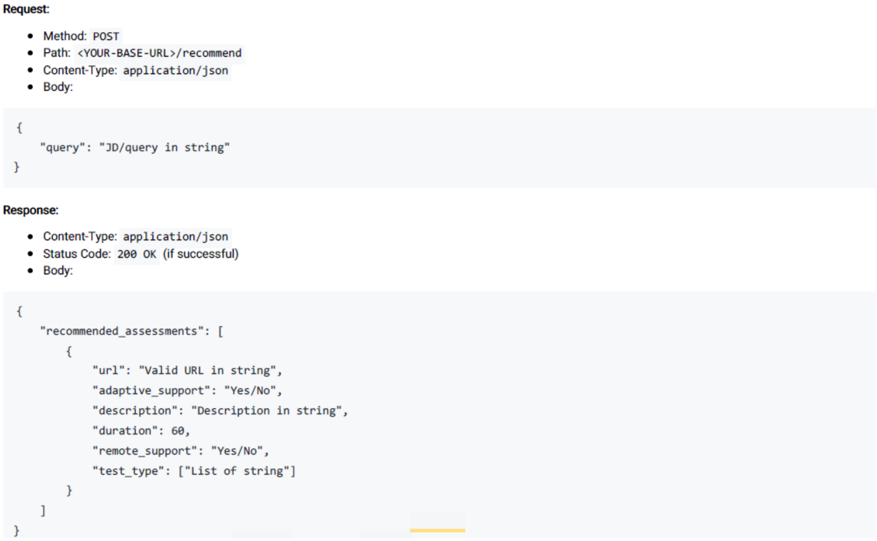
- Method: `GET`
- Path: `<YOUR-BASE-URL>/health`

Response:

- Content-Type: `application/json`
- Status Code: `200 OK` (if healthy)
- Body:

```
{
    "status": "healthy"
}
```

2. **Assessment Recommendation Endpoint:** This endpoint accepts a job description or Natural language query and returns recommended relevant assessments (At most 10, minimum 1) based on the input.



Request:

- Method: `POST`
- Path: `<YOUR-BASE-URL>/recommend`
- Content-Type: `application/json`
- Body:

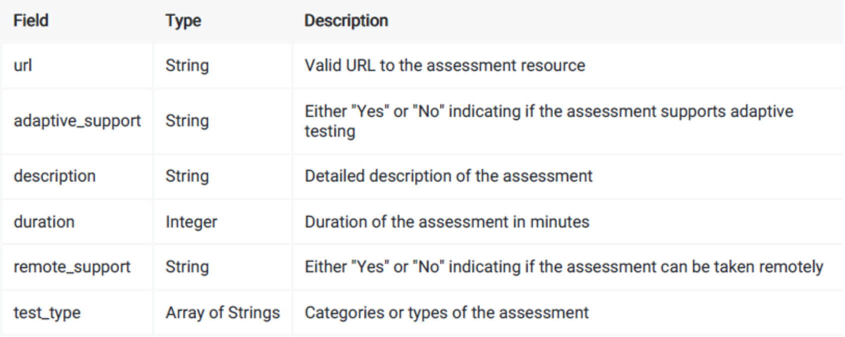
```
{
    "query": "JD/query in string"
}
```

Response:

- Content-Type: `application/json`
- Status Code: `200 OK` (if successful)
- Body:

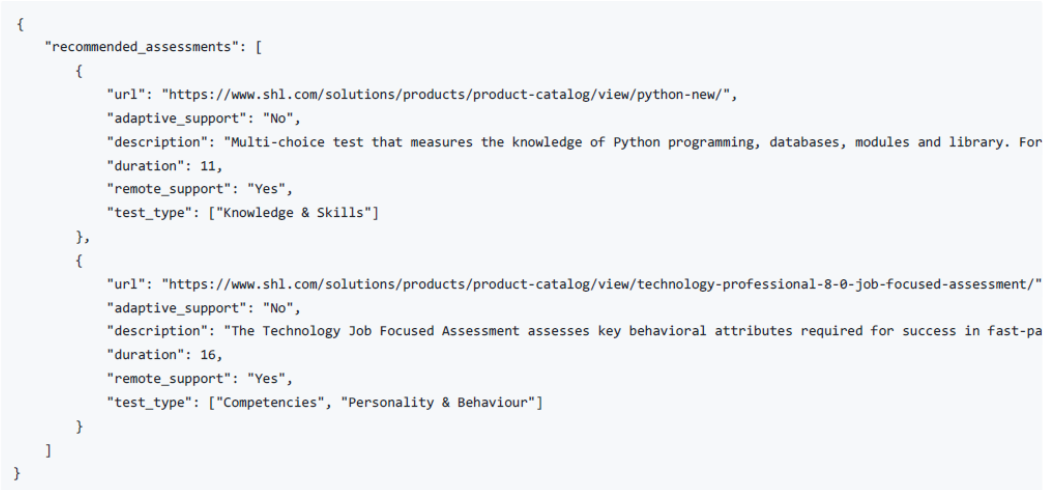
```
{
    "recommended_assessments": [
        {
            "url": "Valid URL in string",
            "adaptive_support": "Yes/No",
            "description": "Description in string",
            "duration": 60,
            "remote_support": "Yes/No",
            "test_type": ["List of string"]
        }
    ]
}
```

Response Fields Explanation: For the /recommend endpoint response ->



Field	Type	Description
url	String	Valid URL to the assessment resource
adaptive_support	String	Either "Yes" or "No" indicating if the assessment supports adaptive testing
description	String	Detailed description of the assessment
duration	Integer	Duration of the assessment in minutes
remote_support	String	Either "Yes" or "No" indicating if the assessment can be taken remotely
test_type	Array of Strings	Categories or types of the assessment

Example Response:



```
{
    "recommended_assessments": [
        {
            "url": "https://www.shl.com/solutions/products/product-catalog/view/python-new/",
            "adaptive_support": "No",
            "description": "Multi-choice test that measures the knowledge of Python programming, databases, modules and library. For
            "duration": 11,
            "remote_support": "Yes",
            "test_type": ["Knowledge & Skills"]
        },
        {
            "url": "https://www.shl.com/solutions/products/product-catalog/view/technology-professional-8-0-job-focused-assessment/"
            "adaptive_support": "No",
            "description": "The Technology Job Focused Assessment assesses key behavioral attributes required for success in fast-pa
            "duration": 16,
            "remote_support": "Yes",
            "test_type": ["Competencies", "Personality & Behaviour"]
        }
    ]
}
```

Testing your API: Before submitting your API link, please verify that ->

1. Both endpoints are functioning correctly
2. The response formats match exactly what is