

Solution Approach Document – SHL Assessment Recommendation System

Objective

To build an AI-driven recommendation system that suggests SHL assessments based on job descriptions using a REST API and a user-friendly web interface.

Tools & Libraries Used

Tools/Library	Purpose
Flask	Backend REST API development
Streamlit	Frontend web app
Google Gemini	AI model for generating recommendations
Requests	Fetch job descriptions from URLs
BeautifulSoup4	HTML parsing from job pages
Regex	Extract structured JSON from AI output
Pandas	Display tabular data in Streamlit
Gunicorn	Production WSGI server

Implementation Overview

Streamlit Web App


 <https://muskansangwan2003.streamlit.app/>

- Option to paste job description or fetch via URL
- Displays AI-generated SHL recommendations in a table
- Raw Gemini response viewable for debugging
- Includes example input for quick testing

Flask REST API

 Base URL: <https://shl-recommender-dha7.onrender.com>

Endpoints:

- /health → <https://shl-recommender-dha7.onrender.com/health>
- /recommendations (POST) →  <https://shl-recommender-dha7.onrender.com/recommendations>
 - Method: POST
 - Body > raw > JSON
 - Use with Postman:

```
{
  "job_description": "Looking for a data analyst familiar with Python, SQL, and basic statistics."
}
```

GitHub Repository

 https://github.com/muku1009/SHL_RECOMMENDER

Contains all backend, frontend, and AI integration code.

Gemini Prompt Strategy

Prompt instructs Gemini to return up to 10 SHL assessments in a structured JSON format with fields like:

- "Assessment Name", "URL", "Remote Testing Support", "Adaptive/IRT Support", "Duration", "Test Type"

Regex is applied to extract valid JSON from the response.

Deployment

- Backend deployed on **Render** using **Gunicorn**
- Frontend hosted on **Streamlit Cloud**
- Supports real-time inference with error handling

Conclusion

An end-to-end solution using Flask, Streamlit, and Gemini to deliver SHL assessment recommendations from job descriptions—accessible via both web and API for ease of use and integration.