Approach Overview:

1.Web Scraping & Data Extraction

- **Tools:** `requests`, `BeautifulSoup`
- Scraped SHL's product catalog: `https://www.shl.com/solutions/products/product-catalog/`, which contains **two paginated tables**.
- For each test, navigated to its individual page and extracted:
  - Test Name, URL
  - Remote Testing / Adaptive IRT
  - Test Type (Assessment Types), Duration
  - Description, Job Levels, and Languages
- Saved extracted information in **JSON format**.

2. Chunking & Embedding

- **Embedding Model:** `'all-MiniLM-L6-v2'`
- **Vector Store:** `FAISS`
- Chunked each test's content to optimize retrieval quality.
- Stored:
  - `FAISS` index (`.faiss`) for embedding documents, used in similarity checking.
  - Documents/tests (`.pkl`) for sending prompt to LLM in backend, data to frontend.

3. RAG Pipeline & LLM Integration

- **Frameworks/Libraries:** `LangChain`, `FAISS`, `Onnx`, `Google Generative AI (Gemini)`
- Used **LangChain** to:
  - Load `FAISS` index and docstore/`.pkl` file contents
  - Retrieve **Top 20 most relevant documents** with a criteria which is a minimum of 40% similarity.

Transformed retrieved data(**names**) into the following format before prompting Gemini:

```
name: <value>
remote_testing: <value>
adaptive_irt: <value>
assessment_types: <value>
description: <value>
job_levels: <value>
languages: <value>
assessment_length: <value>
```

- Prompted Gemini (`gemini-flash`) to recommend **Top assessments** , based on this structured context.

4. Frontend (Streamlit Web App)

- **Tool:** `Streamlit`
- Displayed recommended assessments in **ranked order** without explicitly showing ranks.
- Allowed user to enter query

5. Backend API

- **Framework:** `Flask`
- `/recommend` endpoint accepts JSON input of form {“query”:”your query here”} and returns JSON response with top recommended assessments .
- Deployed on **Render** for hosting.

🧰 Tools & Libraries Used

- `requests`, `BeautifulSoup` – Scraping
- `Onnx`, `transformers`, `faiss-cpu`, `numpy`, `pickle` – Embeddings & Indexing
- `langchain`, `InMemoryDocstore`, `FAISS` – Retrieval
- `google-generativeai` – LLM (Gemini-1.5-flash)
- `Flask`, `Cors` – Backend API
- `Streamlit` – Frontend UI
- `Render` – Hosting & Deployment

Try to search (Sales manager)/(Personality & Behaviour remote) to view results.Do **not search** more than 10 queries per minute as Google API has a limit. As idle app sleeps **wait for 2 minutes** for first response.