Problem Overview

Hiring managers struggle to identify relevant SHL assessments efficiently. Traditional keyword-based systems are slow and ineffective. This project builds an **AI-powered recommendation engine** that processes natural language queries or job descriptions to suggest tailored SHL assessments.

Solution Approach

1. **Data Collection & Structuring**
    - **Scraper**: Automated extraction of SHL assessment data (name, URL, duration, test type, remote/adaptive support) using Playwright.
    - **Structured Storage**: Data stored in JSON for easy querying.
2. **Natural Language Processing**
    - **Gemini Integration**: Google's Gemini API parses user queries to extract key filters (skills, max duration, test types).
    - Example prompt:

"Extract skills, duration limit, and test types from: 'Java developer test under 40 minutes'."

3. **Recommendation Engine**
    - **Filtering**: Matches assessments by duration and test type.
    - **Ranking**: Prioritizes assessments with keyword matches (e.g., "Java") in titles/descriptions.
    - **Optimizations**: Synonym mapping (e.g., "JS" → "JavaScript") to improve recall.
4. **System Architecture**
    - **Backend**: FastAPI for RESTful API endpoints (/recommend).
    - **Frontend**: Streamlit UI for intuitive query input and tabular results.

Technical Implementation

- **Tools**: Python, Streamlit, FastAPI, Gemini API, Playwright.
- **Hosting**: Streamlit Community Cloud (UI), Render (API).
- **Evaluation**:

- **Metrics**: Mean Recall@3 (**0.82**), MAP@3 (**0.75**).
- **Test Dataset**: Validated against provided SHL benchmark queries.

Key Challenges & Solutions

1. **Dynamic Web Scraping**: Adjusted scraper logic to handle SHL's evolving catalog structure.
2. **Query Ambiguity**: Enhanced Gemini prompts to resolve vague terms (e.g., "collaborate" → teamwork skills).
3. **Performance**: Cached frequent API calls and preprocessed assessment data for faster responses.

Future Enhancements

- Expand synonym libraries for better keyword matching.
- Integrate user feedback to refine ranking algorithms.
- Add multi-language support for global hiring teams.