

Individual Assignment 4

Deliverables (upload these 4 .py files only):

- google_parser.py
- meta_parser.py
- gemini_summarizer.py
- hw4.py

Goal

Extend Individual Assignment 4 so that for each job posting (the **link** column in your DataFrame) you scrape the page to extract the qualifications sections (e.g., “Minimum qualifications”, “Preferred qualifications”, etc.), summarize those qualifications with Google GenAI (Gemini), and add a **skills** column to the Streamlit app table showing the summary..

Testing Setup

Local test : Run with Docker Compose.

```
$ docker compose build
```

```
$ docker compose up
```

Make sure that your code passes pytest from test_hw4_streamlit.py and test_hw4_fastapi.py.

Details & Requirements

1. google_parser.py (1.5 pt)

Implement:

```
def retrieve_google_career_qualification(url: str) -> dict
```

Behavior:

- Given a job-posting URL, fetch the page and parse all qualification sections.
- Return a dictionary where each key is the qualification text (e.g., 'Minimum qualifications:') and the value is a list of paragraph strings under that heading.
- If no qualification sections are found, return None.
- Example return:

```
{
    'Minimum qualifications:': [
        'Currently enrolled in or graduated from a degree program within
Product Management, Computer Science, ...',
        'Internship or Teaching Assistant experience in product
management, ...'
    ],
    'Preferred qualifications:': [
        'Experience with methodologies aimed to drive product development
and delivery.',
        ...
    ]
}
```

2. meta_parser.py (1.5 pt)

Implement:

```
def retrieve_meta_career_qualification(url: str) -> dict:
```

Behavior:

- Same contract as google_parser.py but tuned for Meta-style job pages.
- Return same dictionary structure as above.

3. gemini_summarizer.py (0.7 pt)

Implement:

```
def return_gemini_summary(qualification: dict) -> list
```

Behavior:

- Input: the qualification dictionary (output from one of the parsers).
- Call Google GenAI with an appropriate system instruction so the model returns a concise list of nouns or short noun-phrases summarizing the skills/areas required by the job.
 - <https://ai.google.dev/gemini-api/docs/text-generation>
- Return a list of strings (e.g., ['product management', 'software development', 'machine learning', 'presentations']).

4. hw4.py (0.3 pt)

- Update the Streamlit app so that the table displayed contains the following columns from your filtered_df: date, title, skills, link.

Please see the provided URLs as a reference.

Streamlit : <https://dwoodbridge-hw3-streamlit-477009951698.europe-west1.run.app>

FastAPI: <https://dwoodbridg-hw3-fastapi-477009951698.europe-west1.run.app>