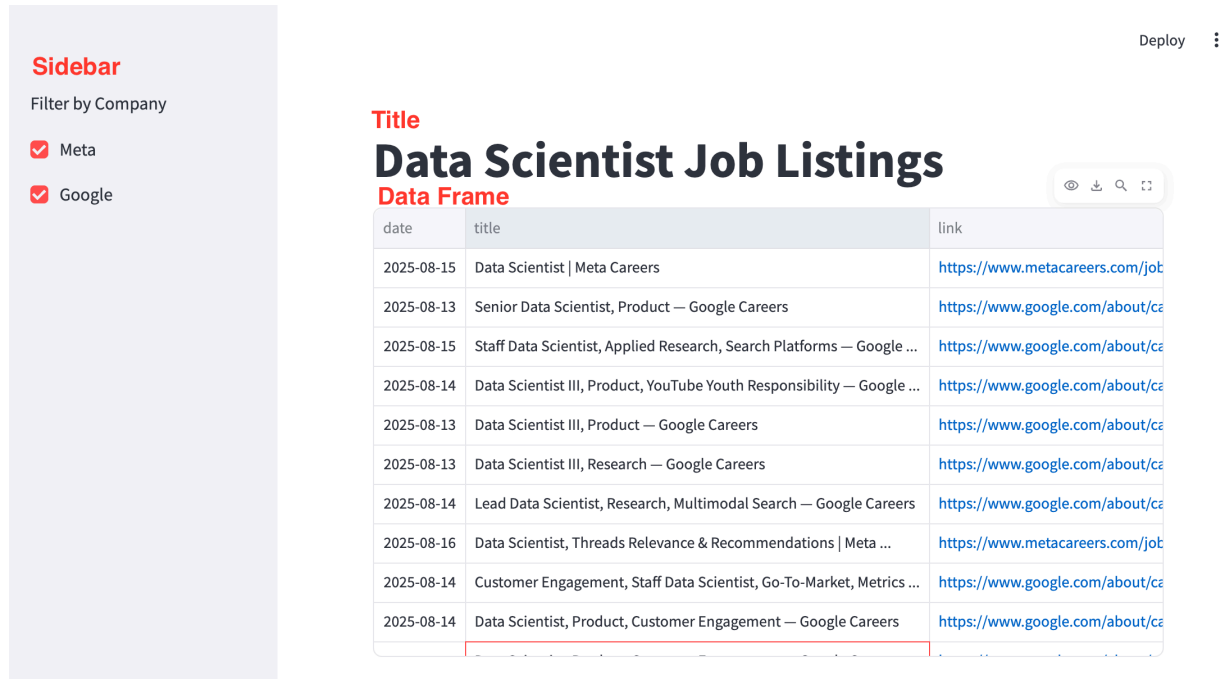


# Individual Assignment 1

## Goal :

Build a Streamlit dashboard that retrieves job postings from pickled data sources, filters them by company, and displays the results in an interactive table.



## Details & Requirements

### • Data Retrieval (0.75 pt)

- Implement `retrieve_data_from_urls(url_list)` to:
  - Get pickle files from the given URLs.
  - Each pickle file contains a list of dictionaries with job posting details:

```
{
  "title": str,
  "link": str,
  "snippet": str,
  "date": datetime.date
}
```

- Return a combined list of unique dictionaries (deduplicate entries).
- Example return:

```
[ {"title": "Data Scientist | Meta Careers", "link": "...", "snippet": "...", "date": datetime.date(2025, 8, 15)},
  {"title": "Senior Data Scientist, Product — Google Careers", "link": "...", "snippet": "...", "date": datetime.date(2025, 8, 13)},... ]
```

- **Dashboard Title (0.25 pt)**
  - The main dashboard title should be: {`role_name`} Job Listings, where `role_name` is imported from `user_definition.py`.
- **Sidebar Filtering (1.5 pt total)**
  - Sidebar must include:
    - A label: “Filter by Company”.
    - A checkbox for each key in `company_dictionary`.
  - Implement `filter_by_company(data, company_dictionary)` to:
    - Filter `data` so only job postings from checked companies are displayed.
    - If no checkboxes are selected, return an empty dataframe.
    - Validation: company names are determined by checking if the job’s link contains any substring from `company_dictionary`.
- **Dataframe Display (1 pt)**
  - Display a dataframe with the following columns only date, title, and link.
  - The link column should display as a clickable hyperlink using Streamlit’s LinkColumn.
  - If no companies are selected, the table should be empty.
- **Code Quality (0.5 pt)**
  - Do not hardcode variables. Use `role_name`, `company_dictionary`, and `url_list` from `user_definition.py`.
  - Your code must pass style checks with fewer than 5 PEP8 issues using `$ pycodestyle hw1.py`