Name: Student ID:

Agentic AI for Business and FinTech (FTEC5660)
Individual Homework 02, Due date: 27 February, midnight.

Instructions

- **Modality:** This is an individual homework. Students are encouraged to discuss with other students, but each student should submit their own work.
- **Input Data:** We provide a 5 sample CVs at this Google Drive link:
`https://drive.google.com/drive/folders/1adYKq7gSSczFP3iikfA8Er-HSZP6VM7D?usp=sharing`
- **Resources:** This assignment requires you to build an AI agent system that connects to MCP (Model Context Protocol) servers for social media verification. We have provided:
  1. **Starter Code (Notebook):**
  `https://colab.research.google.com/drive/1pEQ7KaHTRVsmDvoHg8NMRuXJO3Jr5avD?usp=sharing`
  2. **MCP Server Documentation:** Detailed tool descriptions are provided in Section 2.
  3. **Use Case Reference:** See how AI agents work in KYC (Know Your Customer) verification:
  `https://www.workfusion.com/videos/kayla-ai-agent-for-kyc/`
- **Submission Requirements:** All students are required to create a public GitHub repository for this course and upload their submission files to the GitHub folder. The GitHub folder should contain at least the following:
  - A report file `report.pdf`. The report file should be maximally 5 pages, documenting:
    - Your system architecture and design decisions
    - Agent workflow and tool usage strategy
    - Sample verification results (just final result on the sample CV)
  - The Python implementation of your solution (e.g., `.ipynb` or `.py` files)

  For students unfamiliar with GitHub, we recommend the following introductory tutorial:
  `https://www.youtube.com/watch?v=-RZ03WHqkaY`

1 Practical Problem: CV Verification System

1.1 Background

In recruitment and compliance processes (e.g., KYC), companies need to verify candidate information from CVs against public social media profiles (LinkedIn, Facebook). Manual verification is time-consuming and error-prone. Your task is to build an **agentic AI system** that automates this process.

1.2 Evaluation

Your submission will be graded as follows:

- **System Design (70%):** A reasonable, functional CV verification system that:
  - Connects to the MCP server and uses the provided tools

  - Extracts information from CV files
  - Searches and matches LinkedIn/Facebook profiles
  - Detects discrepancies between CV claims and social media data
  - Generates a verification report

- **Testing Performance (30%):** We will test your system with 5 new CVs containing potential discrepancies. Your score depends on whether your system can correctly flag the problematic points in these CVs.

2 MCP Server Tools

Your agent has access to the following tools via the SocialGraph MCP server:

2.1 Facebook Tools

2.1.1 Tool 1: `search_facebook_users`

Purpose: Find Facebook users by display name (supports fuzzy matching)

Input Parameters:

```
{
    "q": str,           # Search query (name)
    "limit": int,       # Max results (default: 20, max: 20)
    "fuzzy": bool       # Enable fuzzy search (default: True)
}
```

Output:

```
[
    {
        "id": int,              # User ID for get_facebook_profile()
        "display_name": str,    # May differ from legal name
        "city": str,
        "country": str,
        "match_type": str       # "exact" or "fuzzy"
    }
]
```

Example:

```
# Exact match
search_facebook_users("Alex Chan", limit=5)

# Fuzzy match (handles typos)
search_facebook_users("Alx Chn", limit=5)
```

2.1.2 Tool 2: `get_facebook_profile`

Purpose: Retrieve complete Facebook profile with personal info and activity

Input Parameters:

```
{
    "user_id": int  # From search_facebook_users()
}
```

Output:

```
{
    "id": int,
    "display_name": str,
    "original_name": str,     # Legal name from LinkedIn
    "city": str,
    "country": str,
    "hometown": str | None,
    "bio": str,
    "status": str | None,     # Relationship status
    "education": str | None,
    "current_job": str | None,
    "current_company": str | None,
    "interests": str,         # Comma-separated
    "friend_count": int,
    "friends": List[int],
    "posts": List[dict]
}
```

2.1.3 Tool 3: `get_facebook_mutual_friends`

Purpose: Find mutual friends between two users (verify connections)

Input Parameters:

```
{
    "user_id_1": int,
    "user_id_2": int
}
```

Output:

```
{
    "user_1_id": int,
    "user_2_id": int,
    "mutual_friends": List[int],
    "mutual_count": int
}
```

2.2 LinkedIn Tools

2.2.1 Tool 4: `search_linkedin_people`

Purpose: Search LinkedIn by name, skills, or keywords with filters

Input Parameters:

```
{
    "q": str,                # Name, skill, or job title
    "location": str | None, # City or country filter
    "industry": str | None, # Industry filter
    "limit": int,           # Max results (default: 20, max: 20)
    "fuzzy": bool           # Enable fuzzy search (default: True)
}
```

Output:

```
[
    {
        "id": int,
        "name": str,
        "headline": str,
        "industry": str,
        "location": str,        # "City, Country" format
        "years_experience": int,
        "match_type": str       # "exact" or "fuzzy"
    }
]
```

Example:

```
# Search by name and location
search_linkedin_people("Alex Chan", location="Hong Kong")

# Search by skill
search_linkedin_people("Python", industry="Software")
```

2.2.2 Tool 5: `get_linkedin_profile`

Purpose: Retrieve complete professional profile (primary verification tool)

Input Parameters:

```
{
    "person_id": int  # From search_linkedin_people()
}
```

Output:

```
{
    "id": int,
    "name": str,
    "headline": str,
    "city": str,
    "country": str,
    "industry": str,
    "status": str,              # employed, open_to_work, etc.
    "years_experience": int,
    "summary": str,
    "skills": [
        {
            "name": str,
            "proficiency": int  # 1-5
        }
    ],
    "experience": [
        {
            "company": str,
            "title": str,
            "seniority": str,   # junior, mid, senior
            "start_year": int,
            "end_year": int | None,
            "is_current": bool
        }
    ],
    "education": [
        {
            "school": str,
            "degree": str,      # BSc, MSc, MBA, PhD
            "field": str,
            "start_year": int,
            "end_year": int
        }
    ]
}
```

2.2.3 Tool 6: get_linkedin_interactions

Purpose: Assess network strength and engagement

Input Parameters:

```
{
    "person_id": int
}
```

Output:

```
{
    "profile_id": int,
    "post_count": int,
    "total_likes": int,
    "liked_by": List[int],      # Profile IDs
    "engagement_score": float    # Likes per post
}
```