

BÀI THỰC HÀNH: Xây dựng Deep Research Agent và gửi kết quả qua Email

Mục tiêu

Xây dựng một hệ thống gồm nhiều agents có thể thực hiện song song các tìm kiếm sâu (deep research), tổng hợp kết quả, và gửi email HTML chứa bản báo cáo chuyên sâu.

Công nghệ sử dụng

- Python 3.10+
- `asyncio` (chạy agents song song)
- OpenAI API (hoặc giả lập)
- SMTP (gửi email)
- `rich` (in log ra console đẹp)
- (Tuỳ chọn) LangChain / Guidance framework

Các thành phần

Thành phần	Mô tả
Planner Agent	Nhận đầu vào là chủ đề nghiên cứu và chia thành nhiều truy vấn nhỏ
Search Agents (n)	Thực hiện các tìm kiếm song song theo truy vấn được giao
Writer Agent	Tổng hợp kết quả từ các search agents và viết báo cáo HTML
Email Agent	Gửi báo cáo qua email HTML đến người nhận

Bước 1: Cài đặt cấu trúc project

```
deep_research_agent/  
├── agents/  
│   └── planner.py
```

```
| └─ search_agent.py
| └─ writer.py
| └─ email_agent.py
└─ main.py
└─ utils.py
└─ .env
└─ requirements.txt
```

Bước 2: Cấu hình file `.env`

```
OPENAI_API_KEY=sk-...
EMAIL_HOST=smt.gmail.com
EMAIL_PORT=587
EMAIL_HOST_USER=your_email@gmail.com
EMAIL_HOST_PASSWORD=your_app_password
RECEIVER_EMAIL=receiver@example.com
```

Bước 3: Viết từng agent

planner.py

```
async def planner(topic: str, num_tasks: int = 5) -> list[str]:
    return [f"{topic} - góc nhìn {i+1}" for i in range(num_tasks)]
```

search_agent.py

```
import asyncio

async def search(query: str) -> str:
    await asyncio.sleep(1) # giả lập thời gian tìm kiếm
    return f"Kết quả cho: {query}"
```

writer.py

```
def write_report(results: list[str], topic: str) -> str:
    html = f"<h1>Báo cáo về: {topic}</h1><ul>"
    for r in results:
        html += f"<li>{r}</li>"
    html += "</ul><p><strong>Kết luận:</strong> Báo cáo này dựa trên {len(results)} truy vấn.</p>"
    return html
```

email_agent.py

```

import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
import os

def send_email(subject: str, html: str):
    msg = MIMEMultipart()
    msg['From'] = os.getenv("EMAIL_HOST_USER")
    msg['To'] = os.getenv("RECEIVER_EMAIL")
    msg['Subject'] = subject

    msg.attach(MIMEText(html, 'html'))

    server = smtplib.SMTP(os.getenv("EMAIL_HOST"), int(os.getenv("EMAIL_PORT")))
    server.starttls()
    server.login(os.getenv("EMAIL_HOST_USER"), os.getenv("EMAIL_HOST_PASSWORD"))
    server.send_message(msg)
    server.quit()

```

Bước 4: `main.py`

```

import asyncio
from agents.planner import planner
from agents.search_agent import search
from agents.writer import write_report
from agents.email_agent import send_email

async def main():
    topic = "Ứng dụng AI trong marketing"
    queries = await planner(topic, num_tasks=10)

    tasks = [search(q) for q in queries]
    results = await asyncio.gather(*tasks)

    report_html = write_report(results, topic)
    send_email(subject=f"[Báo cáo] {topic}", html=report_html)
    print("✉ Email đã được gửi!")

if __name__ == "__main__":
    asyncio.run(main())

```

Mục tiêu học viên đạt được

- Hiểu cách tổ chức một hệ thống nhiều agents (planner → search → writer → email)
- Thực hành `asyncio.gather()` để chạy nhiều agents song song
- Tạo báo cáo HTML tự động
- Gửi email chuyên sâu theo mô hình tự động

Tác giả: Đỗ Ngọc Tú

Công Ty Phần Mềm VHTSoft

Phiên bản #1

Được tạo 2 tháng 5 2025 08:55:56 bởi Đỗ Ngọc Tú

Được cập nhật 2 tháng 5 2025 09:00:55 bởi Đỗ Ngọc Tú