로컬 LLM 완전 정복 가이드
실무자를 위한 단계별 설치 및 활용 매뉴얼

머리말

이 매뉴얼은 2025년 9월 현재, 비전문가도 스스로 AI 모델을 설치하고 활용할 수 있도록 제작된 실무 가이드입니다. 복잡한 용어와 기술을 배제하고, 실제 인턴십 과정에서 겪었던 시행착오와 문제 해결 과정을 상세하게 담았습니다. 이 문서를 통해 누구나 인터넷 연결 없이 안전하게 개인의 민감 정보를 다루는 로컬 AI 환경을 구축하고, 문서 기반 질의응답(RAG) 기능을 실무에 즉시 적용할 수 있게 될 것입니다.

목차

작성자: 이한솔

1. 로컬 AI 환경 구축 및 실행

1.1. 시스템 요구사항 및 준비사항

개요: 이 매뉴얼은 2025년 9월 현재, Windows 운영체제를 기준으로 작성되었으며, 다른 OS(Linux, macOS) 사용자는 각 OS에 맞는 설치 파일을 다운로드해야 합니다. 또한, 로컬 LLM 환경을 최초 구축하는 과정에서만 일시적인 인터넷 연결이 필요합니다. Ollama 모델 다운로드 및 Open WebUI 컨테이너 설치를 완료한 이후부터는 완전한 오프라인 상태에서 모든 기능을 사용할 수 있습니다. 더불어, 로컬 LLM 환경을 구축하기 위해 필요한 최소한의 시스템 사양과 준비 사항을 안내합니다.

- **운영체제:** Windows 11 (Ollama는 Windows 10 이상을 지원하지만, 11 버전을 권장합니다.)
- **메모리(RAM):** 최소 8GB, 권장 16GB 이상. 더 큰 모델을 구동하거나 여러 모델을 동시에 사용하려면 32GB 이상을 권장합니다.
- **저장 공간(SSD):** 최소 25GB 이상의 여유 공간이 필요합니다. 모델 크기에 따라 추가 공간이 필요할 수 있습니다.
- **그래픽 카드(선택):** GPU가 있으면 모델 추론 속도가 빨라지지만, 필수 사항은 아닙니다.

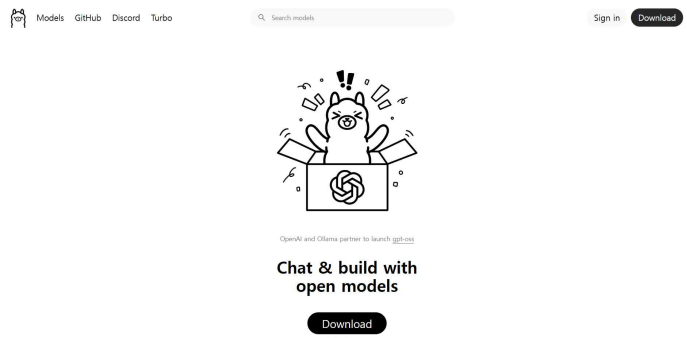
준비사항:

1. **Ollama 설치 파일**
2. **Docker Desktop 설치 파일**
3. **명령프롬프트(CMD) 또는 PowerShell**

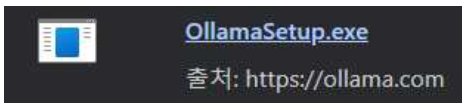
1.2. Ollama 설치 및 기본 설정

과정:

1. Ollama 공식 웹사이트(https://ollama.com/)에 접속하여 웹페이지 우측 상단의 Download 버튼을 눌러 해당 페이지로 접속한 후에, **사용자의 OS에 알맞은 설치 파일(OllamaSetup.exe)**을 다운로드합니다.
   (또는 다운로드 웹페이지(https://ollama.com/download)에 직접 접근하여 해당 설치 파일을 다운로드하도록 합니다.)

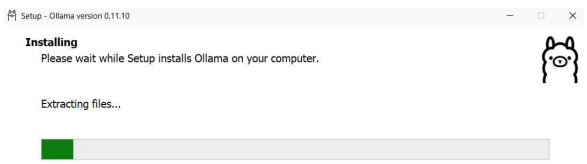


Ollama 공식 웹사이트 접속 화면



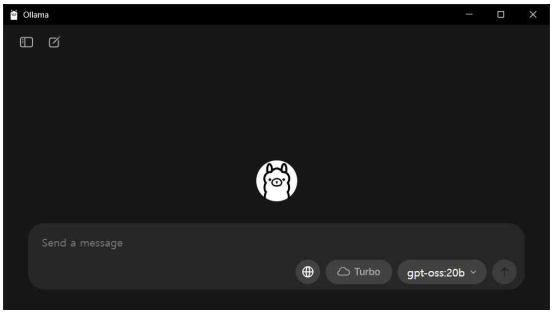
다운로드 완료된 설치 파일

2. 다운로드된 설치 파일을 실행하여 설치를 진행합니다.



3. 설치가 완료되면 Ollama 애플리케이션이 자동으로 실행됩니다.

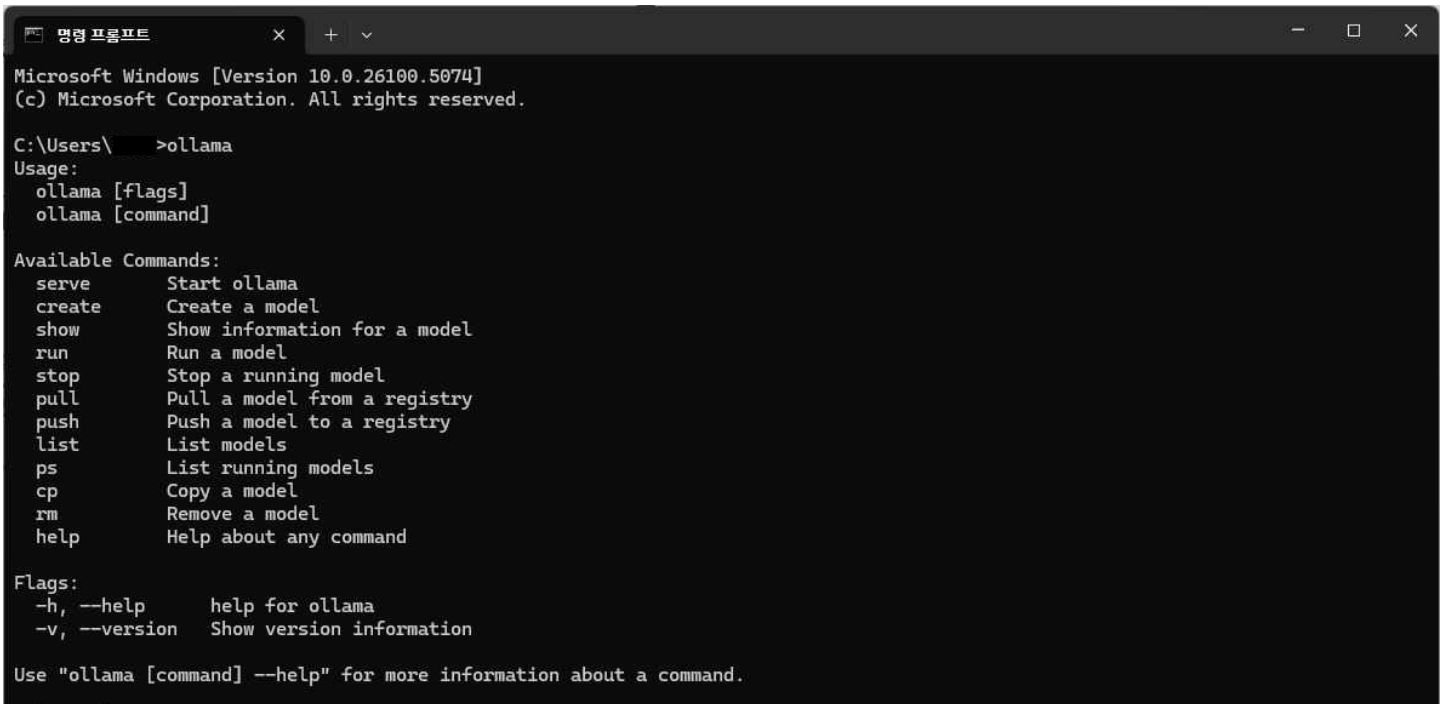
작성자: 이한솔



Ollama 데스크톱 앱 실행 화면

설치 확인:

1. 명령 프롬프트(CMD)를 열고 ollama를 입력한 후 Enter 키를 누릅니다.
2. Usage: ollama [command]와 함께 명령어 목록이 출력되면, Ollama가 올바르게 설치된 것입니다.



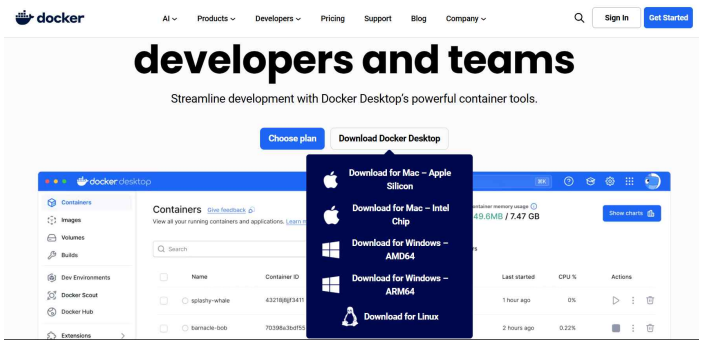
명령 프롬프트에서 ollama 명령어 실행 결과

1.3. Open WebUI 설치 및 접속

☐ Docker Desktop 설치

과정:

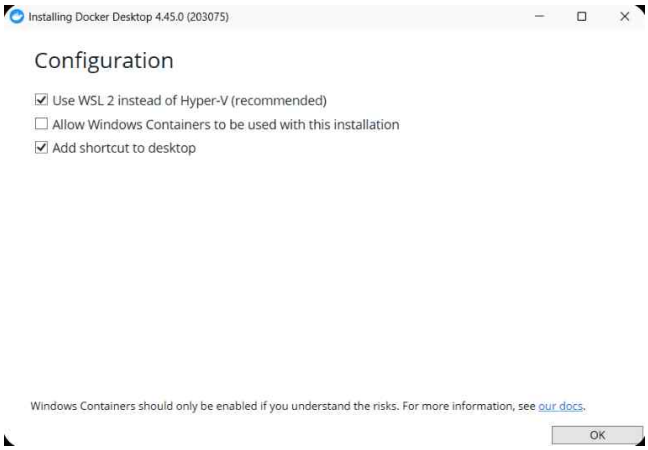
1. Docker 공식 웹사이트(https://www.docker.com/products/docker-desktop/)에 접속하여 사용자 컴퓨터의 운영체제에 맞는 파일을 다운로드합니다.



Docker Desktop 다운로드 페이지 화면

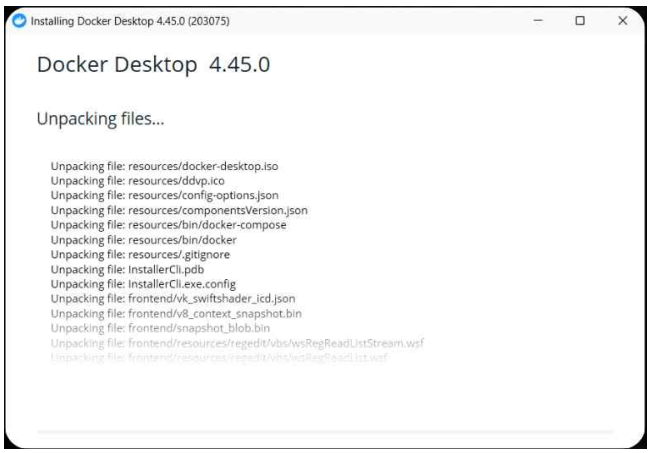
작성자: 이한솔

2. 다운로드한 설치 파일을 실행하여 설치를 진행합니다. 이 때, 기본 설정인 Use WSL 2 옵션을 그대로 두고 진행합니다.



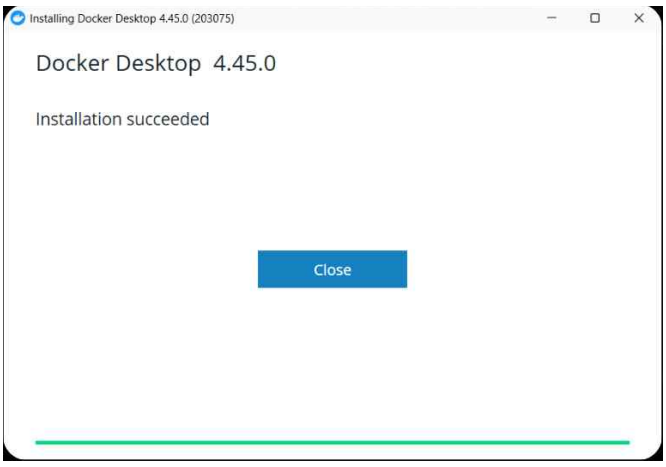
Docker Desktop 설치 Configuration 화면

3. 설치 과정이 진행되는 화면입니다.



Docker Desktop 설치 진행 중 화면

4. 설치가 완료되면 Installation succeeded 화면이 나타납니다.



Docker Desktop 설치 완료 화면

☐ Open WebUI 컨테이너 설치 및 실행

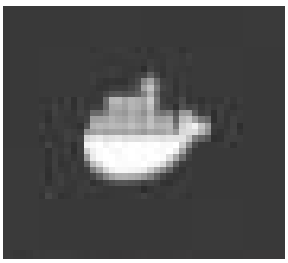
과정:

1. Docker Desktop 실행

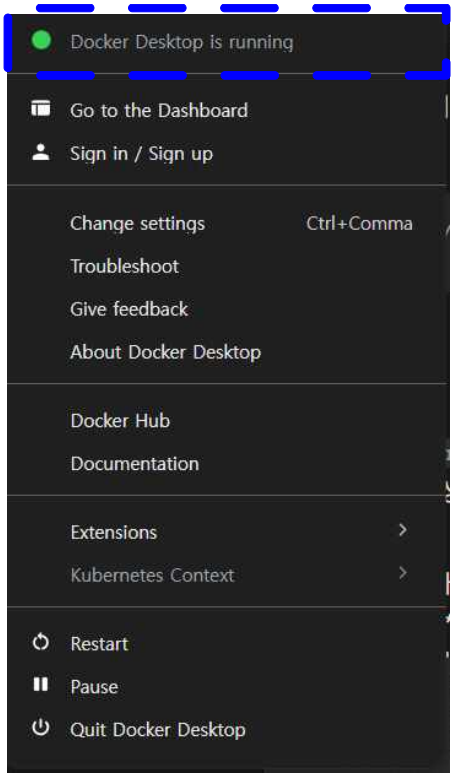
- Docker Desktop 애플리케이션을 실행하고, 모니터 화면 우측 하단 시스템 트레이 부분에 '고래 모양' 아이콘

에 마우스 우측 클릭하여 임시 메뉴가 뜨면 가장 위에 Docker Desktop is running 메시지가 뜨는지 확인합니다.

- 아래 이미지를 참고하여 정상적으로 실행되었는지 확인하십시오.



Docker Desktop 애플리케이션(시스템 트레이)



마우스 우클릭 후 임시 메뉴를 통한
'Docker Desktop is running' 메시지 확인

2. 명령 프롬프트(CMD) 실행

- Windows 11 사용자는 모니터 화면 하단의 시작 버튼 옆의 돋보기 아이콘을 클릭합니다. 또는 키보드 단축키 Windows + S를 누른 후, 검색창에 CMD 또는 명령 프롬프트를 입력하여 실행합니다.
- 이제 이 창에 아래 명령어들을 순서대로 입력하여 Open WebUI 환경을 구축합니다.

3. Ollama 네트워크 생성

- 명령 프롬프트에서 아래 명령어를 입력하여 ollama 네트워크를 생성합니다.

(Tip: 기존에 네트워크가 생성되었으면 Error response from daemon 메시지가 뜨지만 무시해도 됩니다.)

```
docker network create ollama
```

4. Open WebUI 컨테이너 설치 및 실행

- 아래 명령어를 입력하여 Open WebUI를 설치 및 실행합니다.

(Tip: 이 명령어는 Open WebUI 컨테이너를 설치하고, Ollama와 통신할 수 있도록 연결하며, PC 재부팅 시에도 자동으로 실행되도록 설정합니다.)

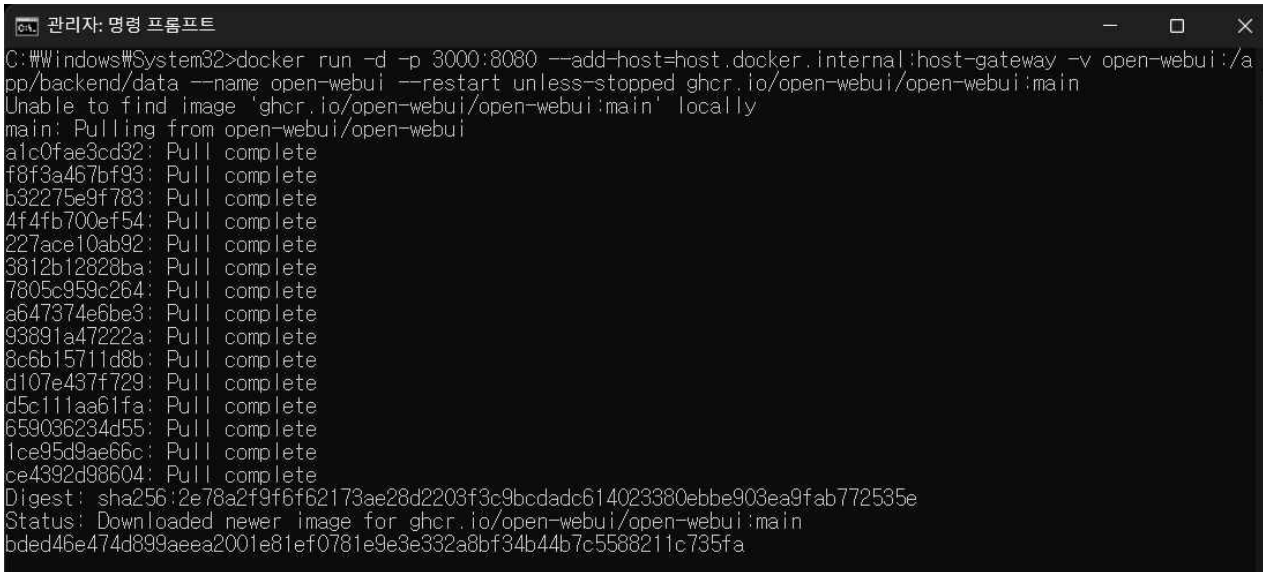
```
docker run -d --network=ollama --pull=always -v open-webui:/app/backend/data -p 8080:8080 --name open-webui --restart unless-stopped ghcr.io/open-webui/open-webui:main
```

※ 실습 내용:

아래 스크린샷 이미지는 실습 당시 docker run -d -p 3000:8080 --add-host... 명령어로 Open WebUI를 설치하고 실행했던 화면입니다.

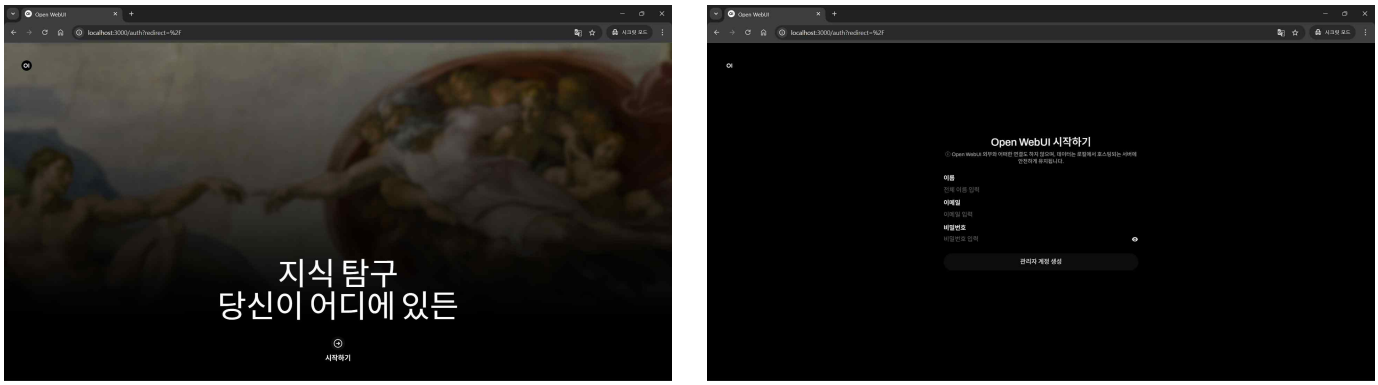
이는 컨테이너가 외부 네트워크를 통해 로컬 PC의 Ollama와 통신하는 방식이라 효율성과 안정성이 떨어집니다. 따

라서 최종 매뉴얼에는 Docker의 권장 사항인 **docker network**를 활용하는 명령어를 기재했습니다. 이 방식으로 진행하면 더 견고한 로컬 LLM 환경을 구축할 수 있습니다.



명령 프롬프트에서 docker run 명령어 실행 결과

5. **웹 브라우저 접속:** 웹 브라우저를 열고 주소창에 localhost:8080을 입력하여 Open WebUI에 접속합니다. 정상적으로 계정 생성 페이지가 나타나면, 입력란에 내용을 입력한 뒤, 계정 생성을 완료합니다.



웹 브라우저에서 Open WebUI 계정 생성 페이지 접속 화면

1.4. 설치 오류 및 문제 해결

1. network ollama not found 오류

- **오류 원인:** docker run 명령어를 실행하기 전에 docker network create ollama 명령어를 실행하지 않았기 때문입니다.
- **해결 방법:** docker network create ollama 명령어를 다시 실행하여 네트워크를 생성한 후, Open WebUI 설치 명령어를 재실행하면 문제가 해결됩니다.

2. localhost:8080 접속 오류

- **오류 원인**: Docker Desktop 애플리케이션이 정상적으로 실행 중이지 않기 때문입니다.
- **해결 방법**: 모니터 화면 우측 하단 시스템 트레이의 고래 모양 아이콘을 우클릭하여, Docker Desktop is running 메시지가 뜨는지 확인합니다.

작성자: 이한솔

2. LLM 모델 관리 및 활용

이 섹션에서는 Ollama를 통해 LLM 모델을 다운로드하고 관리하는 방법을 상세히 안내합니다.

2.1. LLM 모델 다운로드 및 연동

1. Ollama 모델 다운로드

모델을 다운로드하고 설치하기 위해서는 관리자 권한으로 명령 프롬프트를 실행해야 합니다. 아래 방법 중 하나를 선택하여 명령 프롬프트를 실행해 주세요.

- **방법 1: 윈도우 검색창**
  1. 윈도우 작업 표시줄의 검색창에 cmd 또는 명령 프롬프트를 입력합니다.
  2. 검색 결과로 나온 명령 프롬프트 앱을 마우스 오른쪽 버튼으로 클릭합니다.
  3. 나타나는 메뉴에서 **관리자 권한으로 실행**을 클릭합니다.
- **방법 2: 시작 메뉴**
  1. 윈도우 시작 버튼을 마우스 오른쪽 버튼으로 클릭합니다.
  2. 메뉴에서 명령 프롬프트(관리자) 또는 **Windows PowerShell(관리자)**를 클릭합니다.

Ollama 컨테이너가 실행된 상태에서, 관리자 권한으로 실행된 명령 프롬프트에 아래 명령어를 입력하여 llama2 모델을 다운로드합니다. 이 명령어는 명확하게 모델을 다운로드하는 역할만 수행합니다.

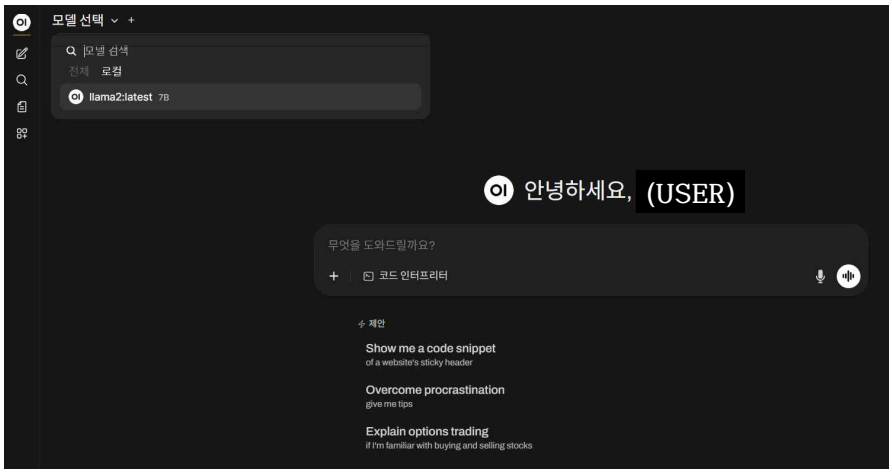
```
ollama pull llama2
```

ollama pull llama2 명령어 실행 및 다운로드 진행 화면

2. Open WebUI 연동 확인

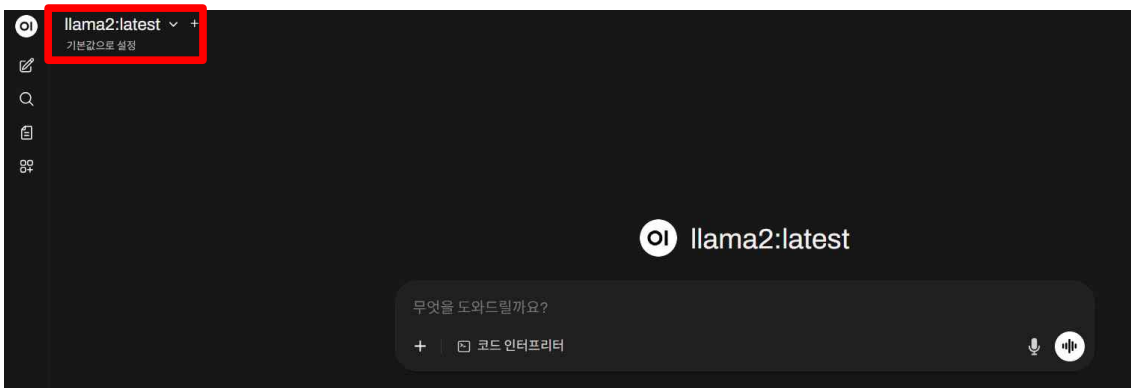
ollama pull 명령어를 통해 모델을 다운로드하면, Ollama 컨테이너가 자동으로 해당 모델을 Open WebUI 컨테이너와 연결해줍니다. 이것이 바로 '연동' 과정입니다.

다운로드가 완료되면 Open WebUI 웹페이지의 좌측 상단에 모델 선택 목록에서 llama2:latest 모델이 정상적으로 나타나는 것을 확인할 수 있습니다. 이제 해당 모델을 선택하면, LLM 모델과의 대화가 정상적으로 시작됩니다.

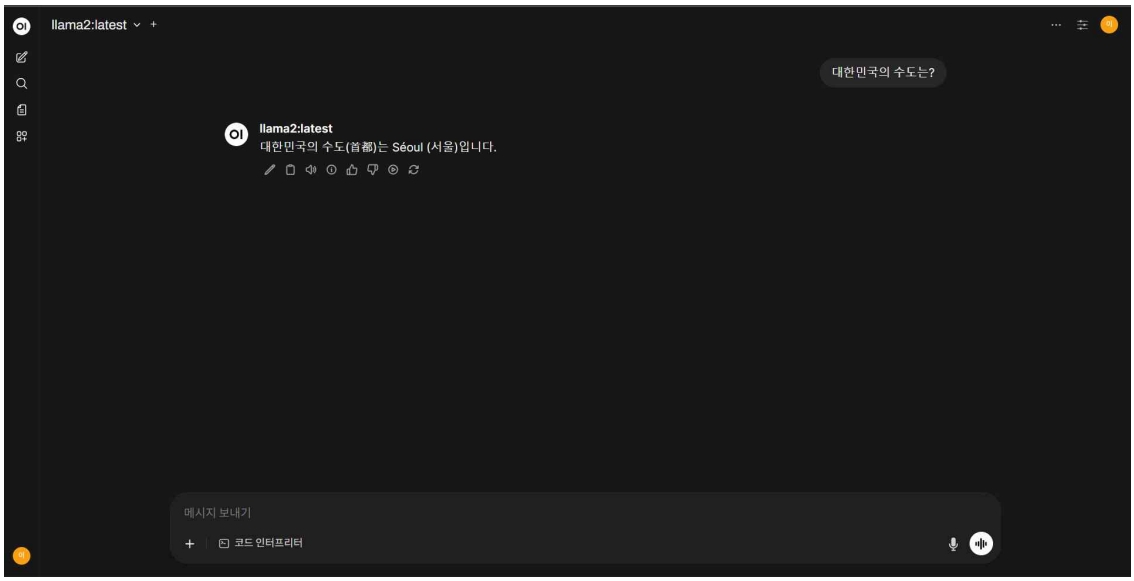


Open WebUI에서 llama2:latest 모델이 연동된 화면

작성자: 이한솔



llama2:latest 모델이 선택된 대화창 화면



llama2 모델과 대화하는 화면

2.2. 컨테이너 재부팅 후 모델이 사라지는 현상

- **문제 발생 원인:** Docker 컨테이너는 본래 휘발성(Volatile)의 특성을 가지고 있습니다. 이는 컨테이너를 재시작하거나 PC를 재부팅하면 컨테이너 내부에 저장된 데이터가 초기화될 수 있음을 의미합니다. 이 때문에 이전에 다운로드했던 LLM 모델이 사라진 것처럼 보일 수 있습니다.
- **해결 방법:** 모델을 다시 다운로드하는 것이 아니라, 명령 프롬프트(CMD)에서 ollama run [모델명] 명령어를 다시 입력하면 됩니다. Ollama는 해당 명령어를 입력했을 때, 컨테이너 내에 모델이 있는지 먼저 확인합니다. 만약 모델이 없다면 자동으로 다운로드하고, 이미 존재한다면 바로 실행하여 모델을 다시 사용할 수 있도록 합니다.
- **실행 예시:** 아래 명령어를 입력하면, 이전에 다운로드된 llama2 모델이 다시 활성화됩니다.

```
ollama run llama2
```

2.3. 모델별 특성 및 활용법

Ollama에서 다운로드할 수 있는 LLM 모델들은 각기 다른 특성을 가지고 있습니다. 매개변수(parameter)의 수가 모델의 크기와 성능에 영향을 미치며, 사용 목적에 맞는 모델을 선택하는 것이 중요합니다.

1. 범용 모델

- **llama2:** 가장 널리 사용되는 오픈소스 모델입니다. 7B, 13B, 70B 등 다양한 크기가 있으며, 일반적인 대화나 문서 요약 등 광범위한 작업에 적합합니다.
- **llama3:** llama2의 후속 모델로, 성능이 크게 향상되었습니다. 더 복잡한 추론과 코딩 작업에서 뛰어난 성능을 보입니다.

작성자: 이한솔

- **gemma:** 구글에서 개발한 경량 모델입니다. 7B와 2B 버전이 있으며, 상대적으로 작은 크기 덕분에 빠른 응답 속도가 중요한 환경에서 유용합니다.

2. 특화 모델

- **codellama:** 코딩 작업에 특화된 모델입니다. Python, C++, Java 등 다양한 프로그래밍 언어의 코드 생성, 디버깅, 주석 달기 등에 탁월한 성능을 발휘합니다.
- **phi3:** 마이크로소프트에서 개발한 소형 언어 모델입니다. 가벼운 크기에도 불구하고 준수한 성능을 보여주어, 저사양 기기나 교육용으로 사용하기에 좋습니다.

2.4. 성능 최적화 (메모리, 속도)

로컬 환경에서 LLM 모델을 효율적으로 사용하기 위해서는 하드웨어 자원(특히 메모리와 속도)을 최적화하는 것이 중요합니다. 아래 내용을 통해 모델 성능을 향상시키는 방법을 알아보겠습니다.

1. 메모리(VRAM) 최적화: 모델 양자화(Quantization)

대부분의 LLM 모델은 매우 큰 용량을 차지하며, 특히 그래픽카드(GPU)의 메모리(VRAM)를 많이 사용합니다. VRAM이 부족하면 모델이 실행되지 않거나 속도가 매우 느려질 수 있습니다. 이를 해결하기 위해 '양자화(Quantization)' 기술을 사용합니다.

- **양자화란?** 모델의 정밀도를 낮춰 모델의 크기를 줄이는 기술입니다. 용량과 메모리 사용량을 크게 줄여 저사양 PC에서도 원활하게 작동합니다.
- **활용법:** Ollama는 다양한 양자화 버전을 제공합니다. 모델 다운로드 시 용도에 맞는 버전을 선택하여 사용하면 됩니다. 예를 들어, ollama run llama2:7b-q4_0 명령어를 사용하면 4-bit로 양자화된 모델을 다운로드할 수 있습니다.

2. 속도 최적화: 모델 크기와 선택

모델의 크기는 응답 속도에 직접적인 영향을 미칩니다. 일반적으로 모델의 크기(매개변수 수)가 클수록 성능은 좋아지지만, 응답 속도는 느려집니다.

- **빠른 속도가 중요할 때:** gemma, phi, tinyllama 등 7B(70억 개) 이하의 경량 모델을 사용하는 것을 추천합니다. 이 모델들은 속도가 매우 빨라 실시간 대화나 빠른 응답이 필요한 작업에 적합합니다.
- **성능과 속도를 모두 고려할 때:** llama2:7b 또는 gemma:7b와 같이 7B 모델을 사용하면 균형 잡힌 성능을 기대할 수 있습니다.
- **정확도와 성능이 가장 중요할 때:** llama2:13b 또는 codellama:34b와 같이 더 큰 모델을 사용합니다. GPU 메모리가 충분하다면 더 정확하고 풍부한 답변을 제공합니다.

2.5. 업데이트 및 유지보수

로컬 AI 환경을 안정적으로 운영하기 위해서는 Ollama와 LLM 모델을 주기적으로 업데이트하고 관리하는 것이 중요합니다.

1. Ollama 업데이트

Ollama 개발팀은 버그 수정 및 성능 개선을 위해 정기적으로 업데이트를 배포합니다. Ollama를 업데이트하려면, 아래 명령어를 사용해 현재 실행 중인 Ollama 컨테이너를 제거하고 최신 버전의 컨테이너를 다시 다운로드해야 합니다.

- **Ollama 컨테이너 제거:**

```
docker stop ollama
docker rm ollama
```

- **최신 Ollama 컨테이너 재설치:**

```
docker run -d --name ollama -v ollama:/root/.ollama -p 127.0.0.1:11434:11434 --restart always ollama/ollama
```

작성자: 이한솔

2. LLM 모델 업데이트

다운로드한 LLM 모델의 최신 버전이 나왔을 경우, 기존 모델을 삭제하고 새 버전을 다운로드하거나 ollama pull 명령어를 통해 업데이트할 수 있습니다. 아래의 명령어는 모델의 최신 버전을 확인하고, 변경 사항이 있으면 업데이트를 진행합니다.

- **모델 업데이트:**

```
ollama pull [모델명]
```

예시) ollama pull llama2

작성자: 이한솔

3. 문서 기반 질의응답 (RAG) 기능 심화 탐구

3.1. RAG 개념 및 필요성

개요: RAG는 '검색 증강 생성' 기술로, LLM이 외부 문서(PDF, 웹페이지 등)에서 정보를 **'검색(Retrieval)'**하여 답변을 **'생성(Generation)'**하는 방식입니다. 이를 통해 모델의 기본 지식이 아닌, 특정 문서의 내용을 바탕으로 정확한 답변을 얻을 수 있습니다.

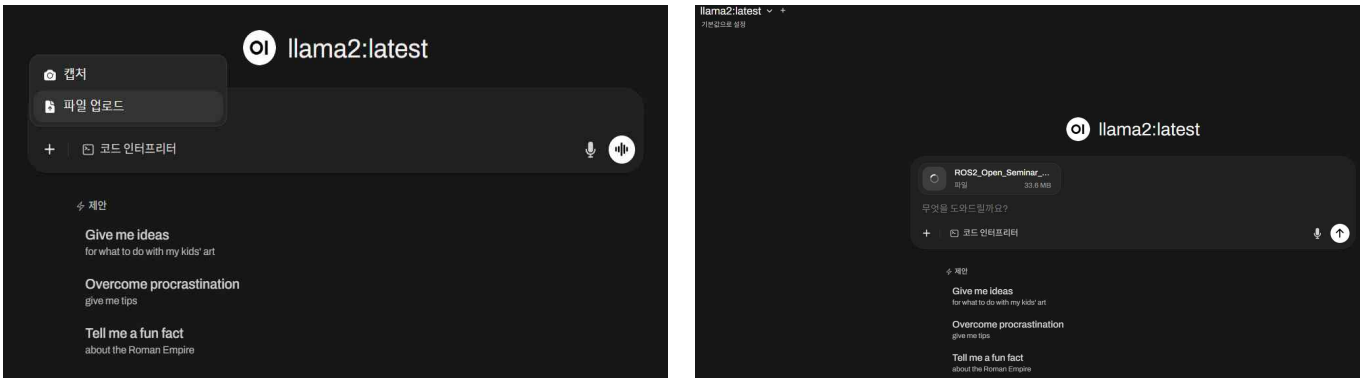
3.2. 단일 문서 활용 RAG 테스트

과정:

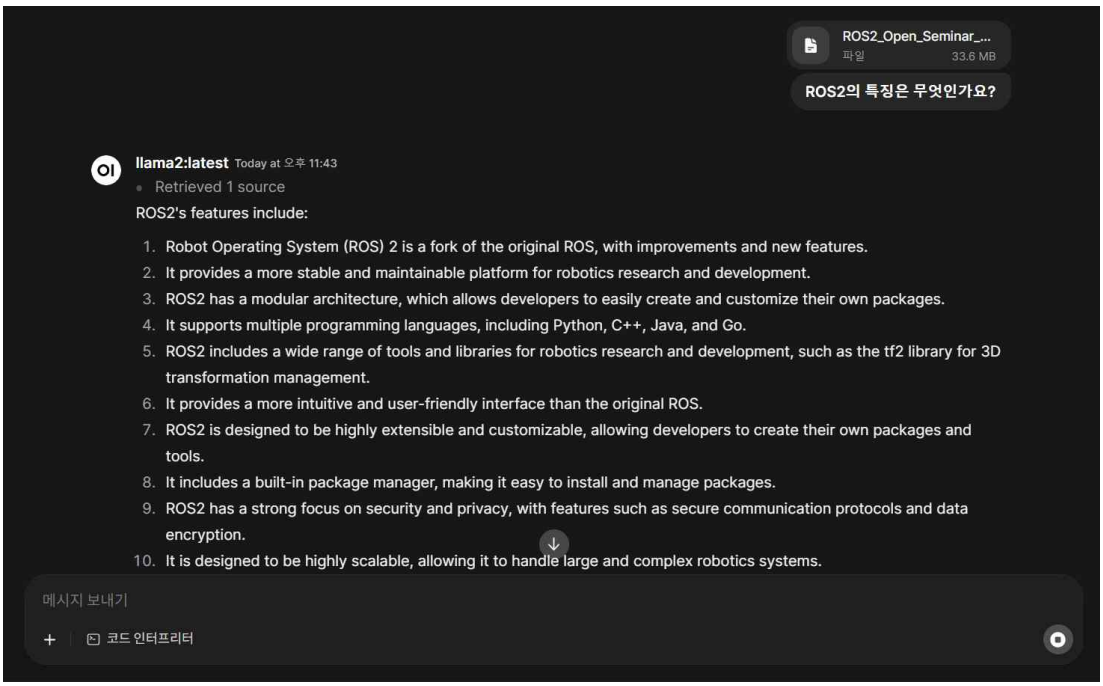
Open WebUI 대화창에서 우측 상단의 종이 아이콘을 클릭합니다.
'파일 업로드' 버튼을 선택하여 가지고 있는 PDF 파일을 업로드합니다.
업로드된 문서를 llama2 모델이 자동으로 학습합니다.

실습 내용:

아래 실습 이미지는 ROS2_Open_Seminar_교재.pdf 파일을 업로드하고 **"ROS2의 특징은 무엇인가요?"**라고 질문하여 답변을 받는 것을 확인한 모습입니다. 해당 사항은 참고하시어 개별적으로 각자 예시와는 다른 내용이 구성되어있는 pdf 파일을 직접 실습의 방법과 같이 순서대로 업로드 진행하신 후, 질문 사항을 입력해보시면 됩니다.



Open WebUI에 PDF 문서를 업로드하는 화면



문서의 내용에 대해 질문하고 답변을 받은 화면

3.3. 다수 문서 활용 RAG 테스트

개요: 하나의 PDF 문서만 활용했던 실습에서 나아가, 여러 개의 문서를 동시에 학습시켜 로컬 LLM의 RAG 기능이

다양한 지식을 통합하고 활용할 수 있는지 테스트했습니다.

과정:

테스트용 PDF 파일 준비: Ollama 모델의 다중 문서 처리 능력을 검증하기 위해 서로 다른 주제의 PDF 파일 5개를 준비했습니다.

반려동물 건강 관리 가이드

인공지능 기초 용어 설명

로마 시대 건축물 소개

커피 원두의 종류와 특징

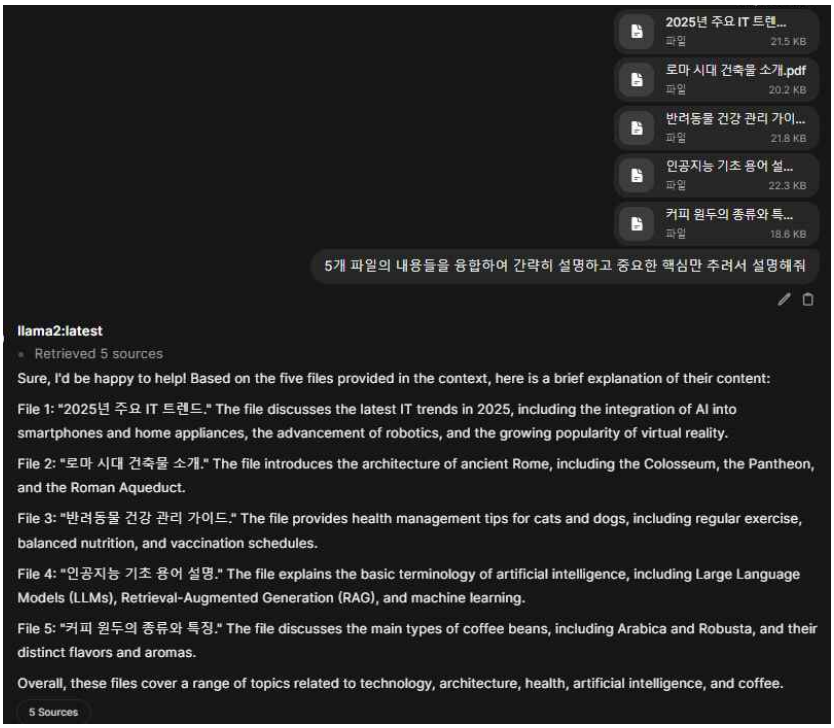
2025년 주요 IT 트렌드

PDF 동시 업로드: Open WebUI 대화창에 위 5개의 PDF 파일을 동시에 업로드했습니다.

다중 문서 기반 질의응답: 업로드된 모든 문서의 내용을 포괄하는 질문을 한국어로 입력하여 답변을 확인했습니다.

입력 질문: "5개 파일의 내용들을 융합하여 간략히 설명하고 중요한 핵심만 추려서 설명해줘"

모델 답변: 모델은 한국어 질문에도 불구하고 영어로 답변을 생성했습니다. 하지만 답변 내용은 5개 PDF 파일의 핵심 내용을 모두 정확하게 요약하고 추출했습니다. 이는 llama2 모델이 언어적 한계에도 불구하고 RAG 기능 자체는 완벽하게 수행했음을 증명합니다.



다수 PDF 파일을 업로드하고 모델에 질문하고,
다수 문서의 내용을 융합하여 질문에 모델이 영어로 답하는 화면

3.4. RAG 설정 심층 분석

로컬 LLM이 문서 기반 질문에 정확하게 답하려면, 단순히 파일을 업로드하는 것 외에 몇 가지 설정을 최적화해야 합니다. Open WebUI의 화면 왼쪽 하단에 있는 프로필 사진 > '관리자 패널' > '설정' > '문서' 탭에서 아래 항목들을 조정하여 RAG 성능을 향상시킬 수 있습니다.

1. 청크 크기 (Chunk Size)

개념: PDF나 텍스트 문서를 LLM이 처리하기 좋은 작은 덩어리로 나누는 기준입니다.

적정값: 문서 내용과 질문의 복잡성에 따라 다르지만, 512~1024 사이의 값을 추천합니다.

예시: 매뉴얼처럼 논리적으로 정리된 문서는 1024로 설정하여 큰 문맥을 유지하는 것이 좋습니다. 반면, 짧은 정보가 반복되는 문서라면 512처럼 더 작은 값으로 나누는 것이 효율적일 수 있습니다.

2. 청크 중첩 (Chunk Overlap)

개념: 문서 조각이 나뉠 때, 앞뒤 내용이 서로 겹치게 하는 크기입니다. 이 중첩을 통해 문맥이 끊어지는 것을 방지

할 수 있습니다.

적정값: 일반적으로 50~100 사이의 값을 설정하여 적절한 문맥을 이어주는 것이 좋습니다.

3. 임베딩 모델 (Embedding Model)

개념: 텍스트를 AI가 이해할 수 있는 숫자 데이터(벡터)로 변환하는 모델입니다. 이 모델의 성능이 문서 검색의 정확도를 결정하는 가장 중요한 요소입니다.

설정: Open WebUI에서는 기본적으로 'nomic-embed-text'와 같은 모델이 설정되어 있습니다. 현재 단계에서는 이 기본 모델을 사용해도 충분하지만, 나중에 더 높은 성능을 원한다면 다른 임베딩 모델을 Ollama에서 다운로드하여 적용할 수 있습니다.

작성자: 이한솔