AI 경진대회 출품작

1. **생활 속 AI 활용** — AI 에이전트로 1인 홈서버를 운영하다
2. **업무 생산성 증가를 위한 AI 활용** — 흩어진 스크린샷 수십 장을 하나의 보고서로

생활 속 AI 활용

AI 에이전트로 1인 홈서버를 운영하다

— 미니PC 두 대를 '집 안의 비서'로 만든 기록

1. 배경 — 늘어나는 개인 서비스, 줄지 않는 손

집에서 쓰는 디지털 도구가 늘면서 자연스럽게 직접 운영하는 셀프호스팅 서비스가 쌓였다. 메모 동기화, 북마크 관리, 자동화 작업 등 하나하나는 편리하지만, 문제는 **관리 부담**이었다.

- 컨테이너를 띄우려면 매번 설정 파일과 명령어를 찾아 헤맸고,
- 서비스가 늘수록 "이건 어느 머신에 올렸더라" 하는 혼선이 생겼으며,
- 자동화를 걸어두고 싶어도 워크플로우를 짜는 일 자체가 또 다른 일거리였다.

결국 **서비스를 쓰는 시간보다 서비스를 관리하는 시간**이 더 들 위험이 있었다. 이 문제를 AI 에이전트를 '운영 보조'로 끼워 넣어 풀어보기로 했다.

2. 구성 — 역할이 나뉜 미니PC 두 대

홈서버는 작은 머신 두 대로 역할을 분담했다.

머신	역할	비고
맥미니	AI 에이전트 호스트(OpenClaw)	다른 컴퓨터에서 원격으로 열어 사용
CASAOS 미니PC	자동화 엔진(N8N) + 셀프호스팅 서비스	새 서비스는 이쪽에 집중 배치

두 머신은 **MCP(Model Context Protocol)**로 연결했다. CASAOS의 N8N이 MCP를 통해 맥미니의 AI 에이전트를 호출하는 구조로, "자동화 흐름 중간에 판단이 필요한 단계를 AI에게 위임"하는 형태가 가능해졌다.

설계 원칙은 단순했다.

- **맥미니는 AI 에이전트 전용**으로 깔끔하게 유지하고,

- **새로 올리는 서비스는 기본적으로 CASAOS 쪽**에 모은다.
- AI 추론에 GPU 가속이 꼭 필요한 경우에만 맥미니로 분리한다.

역할 경계를 미리 정해두니, 이후 "무엇을 어디에 올릴지"를 매번 고민하지 않아도 됐다.

3. AI 활용 포인트 — 무엇을 AI에게 맡겼나

(1) 자연어로 컨테이너 배포하기

새 서비스를 올릴 때, 설정 파일을 처음부터 작성하는 대신 **"이 서비스를 CASAOS에 올리고 싶다"**고 말하면 AI 에이전트가 구성 초안을 만들고 배치 위치(역할 분담 원칙에 맞게 CASAOS)를 제안하도록 했다.

실제로 이 방식으로 추가한 서비스:

- **Obsidian Self-hosted LiveSync** (CouchDB 기반) — 여러 기기 간 메모 실시간 동기화
- **Karakeep** — 북마크를 모으고, 연결된 LLM이 **자동으로 태그를 달아주는** 북마크 매니저

(2) 북마크에 사람 손이 닿지 않게 하기

가장 체감이 컸던 부분은 Karakeep의 **AI 자동 태깅**이었다. 링크를 저장만 하면 내용에 맞는 태그가 알아서 붙는다. "나중에 정리해야지" 하고 쌓아두던 북마크가, 저장하는 순간 분류까지 끝나는 경험은 일상에서 AI가 만드는 차이를 가장 직접적으로 느낀 지점이었다.

(3) 자동화 흐름 속에 AI를 끼워 넣기

N8N은 정해진 규칙대로 흐르는 자동화 엔진이다. 여기에 MCP로 AI 에이전트를 연결하니, 규칙만으로는 처리하기 애매한 "내용을 보고 판단해야 하는 단계"를 자연어 지시로 위임할 수 있게 됐다. 딱딱한 자동화와 유연한 판단을 한 흐름 안에서 섞는 셈이다.

4. 성과 — 일상에서 달라진 것

- **관리에서 운영으로**: 설정 파일과 명령어를 검색하던 시간이, 무엇을 만들지 말로 설명하는 시간으로 바뀌었다.
- **방치되던 데이터가 살아남**: 쌓아두기만 하던 북마크·메모가 저장 즉시 분류·동기화되며 실제로 다시 꺼내 쓰게 됐다.

- **확장의 심리적 비용 감소**: "새 서비스 하나 올리는 게 부담"이라는 진입장벽이 낮아져, 필요할 때 바로 시도하게 됐다.

특별한 장비도, 클라우드 구독도 아닌 **작은 미니PC 두 대와 AI 에이전트**만으로 1인 홈서버를 '관리 대상'이 아니라 '함께 일하는 비서'로 바꾼 경험이었다.

5. 돌아보며

AI 활용이라고 하면 흔히 거창한 모델이나 서비스를 떠올리지만, 정작 일상을 바꾼 건 **"반복적이고 귀찮은 운영 작업을 자연어로 넘길 수 있게 된 것"**이었다. 중요한 건 AI의 성능 그 자체보다, **AI가 내 일상 시스템 안에 자연스럽게 들어올 자리를 만들어 둔 설계**였다. 역할을 나눈 두 대의 미니PC와 그 사이를 잇는 AI 에이전트가, 그 자리를 만들어 주었다.

업무 생산성 증가를 위한 AI 활용

흩어진 스크린샷 수십 장을 하나의 보고서로

— AI와 함께 만든 시장조사 정리 워크플로우

1. 문제 — 자료는 많은데, 보고서가 안 나온다

콘텐츠 시장조사 업무는 자료 수집보다 **정리**에서 막힌다. 경쟁 서비스를 둘러보며 화면을 캡처하고, 떠오르는 메모를 적어두지만, 정작 이 자료들이 하나의 보고서로 모이기까지가 가장 오래 걸린다.

전형적인 병목은 이랬다.

- 스크린샷이 수십 장, 메모는 여러 차례에 걸쳐 흩어져 쌓인다.
- 매번 보고서마다 표·색·레이아웃을 처음부터 다시 잡는다.
- 정리하는 사람의 해석이 끼어들면서, **"사실"과 "추측"의 경계가 흐려진다.**

마지막 항목이 특히 컸다. 시장조사 보고서의 가치는 정확한 사실 정리에 있는데, 정리 과정에서 작성자(혹은 AI)의 판단이 섞이면 실무에 오히려 방해가 된다는 피드백을 받은 적이 있다.

2. 해결 — AI는 '정돈'하되 '판단'하지 않는다

이 문제를 AI와 함께 풀되, **명확한 역할 경계**를 두고 워크플로우를 설계했다.

워크플로우 구조

```
[1차 자료]                    [AI의 역할]                  [산출물]
스크린샷 + 직접 작성한 원문 → 형식만 정돈 (표현·줄바꿈 보존) → 구조화된 HTML 보고서
(raw 그대로 보존)              사실 재배치 / 추측 추가 금지      (양식 컴포넌트 재사용)
```

핵심 설계 원칙은 세 가지였다.

① 원문(raw)은 절대 손상하지 않는다.
직접 작성한 사이트 설명·메모는 별도 raw 파일에 원본 그대로 보존하고, 보고서에 옮길 때도 표현과 줄바꿈을 보존한다. AI는 형식만 정돈한다.

② AI는 사실만 옮기고, 추측·조언·결론을 덧붙이지 않는다.
"시사점", "권장 사항", "관찰 포인트" 같은 섹션을 자의로 만들지 않는다. 출처 없는 메커니즘 설명도 넣지 않는다. **역설적으로, AI에게 "판단하지 마라"고 강하게 제약한 것이 보고서의 신뢰성을 끌어올렸다.**

③ 양식을 자산으로 만든다.
한 번 잡은 보고서 양식(공통 색 토큰, figure·표·태그 같은 컴포넌트, 인쇄 대응)을 재사용 가능한 틀로 정리해, 새 보고서마다 디자인을 다시 잡지 않는다.

3. 실제 적용 — 수십 장의 자료를 하나로

이 워크플로우로 다수의 경쟁 서비스를 한 보고서에 담았다.

- 차수를 나눠 들어온 메모를 raw 파일에 **이어 붙이는** 방식으로 누적했고,
- 사이트별로 구조·기능·세일즈포인트 같은 **동일한 항목 틀**에 맞춰 정렬했으며,
- 결제 방식처럼 비교가 필요한 정보는 표와 태그로 한눈에 보이게 정돈했다.

작성자는 "무엇을 조사했는가(사실)"에만 집중하고, "어떻게 보기 좋게 배치할 것인가(형식)"는 AI에게 맡기는 분업이 자리 잡았다.

4. 성과 — 무엇이 빨라지고, 무엇이 좋아졌나

- **정리 시간의 이동**: 레이아웃을 잡고 자료를 옮겨 적던 수작업이 줄고, 자료를 **읽고 판단하는** 본질 작업에 시간을 쓰게 됐다.
- **일관성 확보**: 양식을 자산화하니, 여러 보고서가 같은 품질·같은 구조로 나온다.
- **신뢰성 향상**: AI의 자의적 해석을 제도적으로 차단하니, 보고서가 "사실 기록"으로서의 신뢰를 유지했다.

5. 돌아보며 — 'AI에게 무엇을 시키지 않을지'가 생산성을 만든다

AI 활용이라고 하면 보통 "AI가 얼마나 많은 일을 대신해 주는가"를 따진다. 하지만 이 작업에서 생산성을 만든 결정적 요인은 정반대였다. **AI가 손대도 되는 영역(형식 정돈)과 손대면 안 되는 영역(사실 판단)을 명확히 그은 것**이다.

사람은 사실을 수집·판단하고, AI는 그것을 흠 없이 정돈한다. 이 경계가 분명할 때, AI는 일을 **대신**하는 도구를 넘어 업무의 품질과 속도를 동시에 끌어올리는 **신뢰할 수 있는 동료**가 된다.