

# MCP 설치 및 설정 지침서 (88OEMBoard 프로젝트 전용)

## 1. 프로젝트 개요

**프로젝트명:** 88OEMBoard 대시보드 시스템 구축

**목표:** - Notion 기반 제품/바이어 관리 데이터를 자동 연동하여 시각화된 대시보드 제공 - 내부 NAS 또는 클라우드 DB에 실시간 상태 저장 및 조회 - 회사 도메인 `db.88toy.co.kr` 을 통해 내부와 외부 바이어에 분리 제공

**특징:** - 내부 관리 용도 중심 → 로그인 보안 포함 - 추후 바이어용 OEM 포털 확장 가능성 고려 - 다양한 시각화 방식 (카드형, 표, 그래프 등)

**사용 기술 및 인프라:** - Next.js 기반 웹사이트 구축 - Notion API + NAS DB 연동 - MCP (Model Context Protocol)를 통한 Claude 자동화 - Git 버전 관리 및 팀협업 기반 구조

**추가 문서 연동:** - [Project Plan 88oemboard](#) 문서에 세부 기획 및 폴더 구조, MCP 종류, 설정 정보 포함됨

**프로젝트 정의 및 진행 순서:** 1. 프로젝트 기획 및 요구사항 정리 (완료) 2. MCP 연결 및 개발 환경 구축 (진행 중) 3. Notion API 연동 테스트 및 데이터 구조 설계 4. NAS 또는 클라우드 DB 연결 및 테스트 5. Next.js 기반 대시보드 UI 구성 6. 사용자 인증 및 외부 바이어 뷰 설계 7. 배포 및 `db.88toy.co.kr` 도메인 연결 8. 내부 테스트 및 피드백 반영

## 2. 전제 사항

- 플랫폼: 클로드 데스크탑 (Windows 환경)
- 루트 폴더 위치: 예시 `C:\Projects\88oemboard`
- NAS DB 및 Notion API 연동 예정
- MCP 도구: `filesystem`, `github`, `memory`, `puppeteer`

## 🔗 3. MCP 설치 상태 및 기능

**현재 설치된 MCP 서버 목록:**

### 1. Filesystem 📁

- 상태: 정상 작동
- 기능: 파일/폴더 읽기, 쓰기, 생성, 삭제, 검색
- 경로: `C:\Projects\88oemboard`

### 2. GitHub

- 상태: 설치됨 (토큰 설정 필요)
- 기능: 저장소 관리, 파일 업로드, Issue/PR 관리

- 설정 필요: `GITHUB_PERSONAL_ACCESS_TOKEN`

### 3. Memory

- 상태: 정상 작동
- 기능: 지식 그래프 생성, 엔티티 관리, 대화 기록
- 테스트: `88oemboard` 프로젝트 정보 저장됨

### 4. Puppeteer 🐼

- 상태: 정상 작동
- 기능: 웹 자동화, 스크린샷, 폼 입력, 스크래핑
- 테스트: 구글 홈페이지 스크린샷 성공

### 5. Everything ✨

- 상태: 정상 작동
- 기능: MCP 프로토콜 테스트, Echo, 계산 등
- 테스트: Echo 및 덧셈 계산 성공

총 5개 서버 중 4개 완전 작동, 1개 토큰 설정 대기 모든 주요 기능(파일 관리, 웹 자동화, 메모리 관리)이 준비되었습니다!

## 4. 설정 정보 파일 위치

| 항목           | 파일명               | 설명   |
|--------------|-------------------|--|
| GitHub 토큰    | <code>.env</code> | <code>GITHUB_TOKEN=ghp_...</code>  |
| Notion API 키 | <code>.env</code> | <code>NOTION_API_KEY=sk-...</code>   |
| NAS DB 연결    | <code>.env</code> | <code>DB_HOST=...</code> , <code>DB_USER=...</code> , <code>DB_PASS=...</code> |

`.env` 파일은 `lib/config.ts` 또는 `next.config.js` 에서 참조됨

## 클로드 데스크탑 작업 유의사항

- `filesystem` MCP는 필수 MCP이며, 다른 모든 작업의 기반
- Claude Desktop에서 `write` → `append` 방식으로 파일 조작 권장
- `project_plan.md` 에 항상 변경 사항을 `append`하여 반영해야 함

## 참고 유튜브 영상 (강력 추천)

1. [플레이션트에 GPT 이미지 MCP 탑재](#)
2. [Context7 활용법 \(차단 우회, 코딩 보조\)](#)
3. [커서 AI로 구글링, 코딩 자동화](#)
4. [클로드 MCP로 최신 자료 크롤링, DB설계](#)

## 5. [Git MCP 실습 영상](#)

---

이 지침은 프로젝트가 진화할수록 계속 업데이트되며, `project_plan.md` 와 함께 참고해야 합니다.