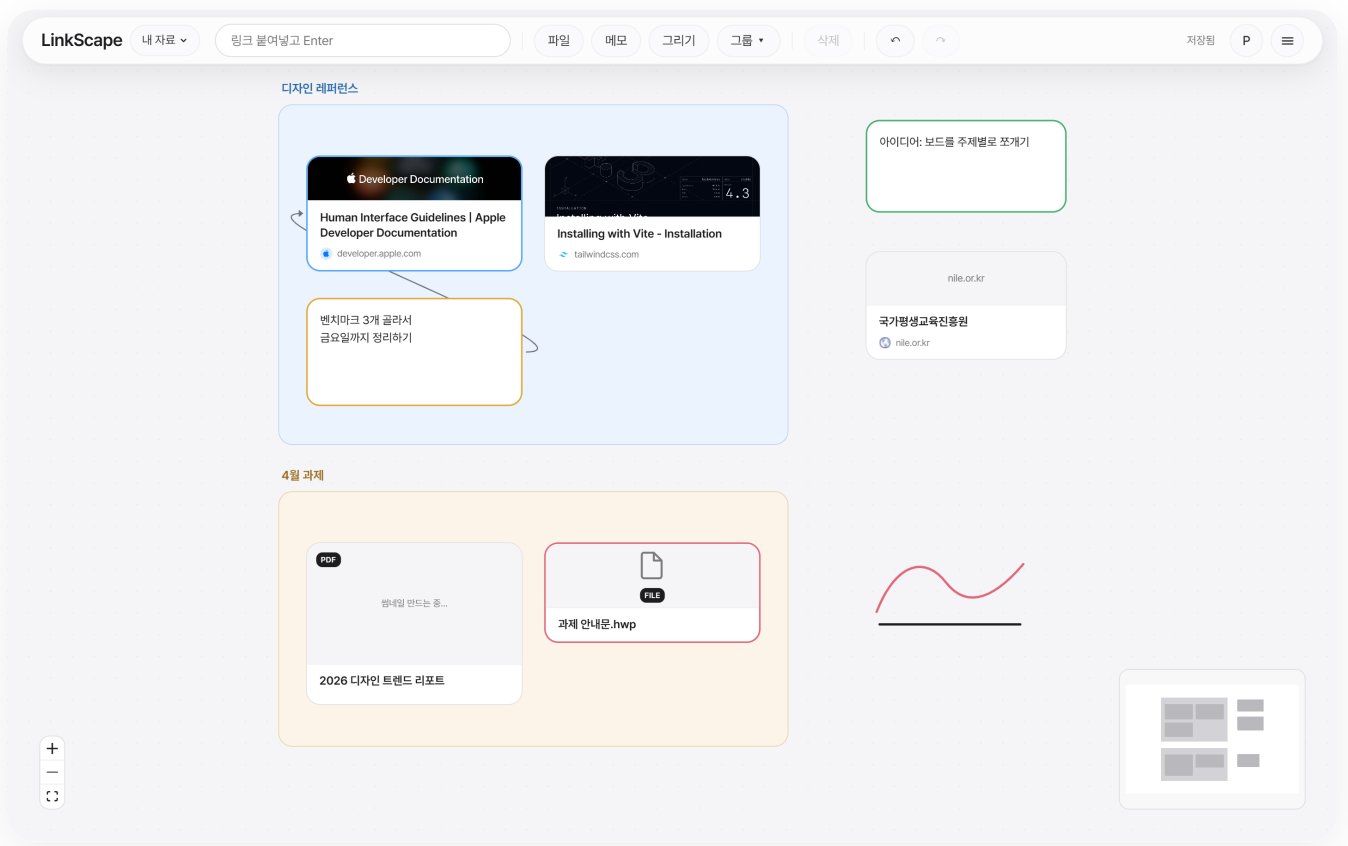




링크와 PDF, 놓는 자리가 분류가 됩니다

무한 캔버스 위에 자료를 펼쳐두는 개인 아카이브. 폴더도, 태그도, 정리 규칙도 없이 —
카드를 옮겨 놓는 것이 곧 정리가 되는 서비스입니다.



실제 서비스 화면 — 그룹·카드·펜 잉크·연결선이 한 캔버스 위에 놓여 있다

웹 (pdflinkin.vercel.app)

웨일·크롬 브라우저 확장

Windows 데스크톱 (3.43MB)

Android 앱 · PWA

LinkScape는 이런 서비스입니다

LinkScape는 링크·PDF·이미지·한글(HWP)·오피스 문서를 **무한 캔버스 위에 카드로 펼쳐두는 개인 아카이브**입니다.

자료를 담으면 카드가 되고, 카드를 어디에 놓느냐가 곧 분류가 됩니다. 웹·브라우저 확장·Windows 데스크톱·Android 네 플랫폼이 하나의 캔버스를 실시간으로 공유하므로, 어디서 담아도 어디서든 같은 보드가 열립니다.

왜 만들었나

많은 사람이 자료를 카카오톡 '나에게 보내기'에 던져 둡니다. 담기는 쉽지만 꺼내기는 어렵습니다 — 한 달만 지나면 무한 스크롤 속에서 그 링크를 다시 찾을 수 없습니다. 브라우저 북마크는 반대로, 폴더를 정하고 이름을 붙여야 해서 담는 것부터가 일입니다. 화이트보드 도구(FigJam·Miro)는 문서 내용을 보여주지 못하고, 노선은 목록 구조라 자료가 쌓일수록 한눈에 들어오지 않습니다. LinkScape는 이 사이의 빈 자리 — **"담는 건 메신저만큼 쉽고, 찾는 건 눈으로 보듯 빠른 보관함"** — 을 채웁니다.

1

위치가 곧 분류

폴더명·태그 입력 없음. 카드를 놓는 자리, 묶는 그룹, 테두리 색이 분류의 전부입니다. "왼쪽 위 그 근처에 뒀지" — 공간 기억이 검색보다 빠릅니다.

2

담기에 마찰 제로

브라우저에서 우클릭 한 번, Ctrl+V 한 번, 드래그 한 번. 입력할 것이 하나라도 있으면 사람은 담지 않는다는 전제로 설계했습니다.

3

모든 자료가 한 화면에

PDF는 본문까지 검색되고, 한글 파일은 브라우저 안에서 바로 열립니다. 전체 자료가 캔버스 하나에 보이므로 "어디 있더라"가 사라집니다.

사용 흐름은 세 단계입니다

- | | |
|-------|---|
| ① 담는다 | 웹서핑 중 우클릭·붙여넣기·드롭으로 자료를 던져 넣는다. 담는 순간 열려 있는 캔버스에 즉시 나타난다. |
| ② 놓는다 | 시간이 날 때 카드를 끌어 자리를 잡는다. 그룹에 넣으면 프레임이 알아서 늘어나고, 빼면 줄어든다. |
| ③ 찾는다 | 눈으로 훑거나, Ctrl+K 로 PDF 본문까지 검색하거나, 목록 보기로 그룹별 정렬을 본다. 클릭하면 바로 열린다. |

기능 ① 담기 — 어디서든, 마찰 없이

자료를 만나는 모든 장소에 담는 경로를 두었습니다. 담는 순간 Supabase Realtime 으로 열려 있는 모든 기기의 캔버스에 새로고침 없이 즉시 카드가 나타납니다.

담는 경로 10가지

위치	경로
웹 캔버스	 커서 자리에 링크·파일·텍스트 붙여넣기 · 파일·링크 드래그앤드롭 · 툴바 "파일" 버튼(폴더에서 선택) · 툴바 링크 입력창 · 빈 곳 더블클릭 = 메모 · 우클릭 → 붙여넣기
브라우저 확장 (웨일·크롬)	우클릭 메뉴 4종(링크·이미지·선택 텍스트·현재 페이지) · 팝업(현재 탭 담기 · Ctrl+V · 파일 드롭/선택) · 아무 페이지에서 드래그를 시작하면 우하단에 나타나는 드롭존
Android	다른 앱의 공유 시트에서 "LinkScape" 선택 → 링크가 보드에 카드로 담긴다 (PWA share target)

카드가 되는 자료 6종

종류	담기면 이렇게 됩니다
링크	제목·미리보기 이미지·파비콘을 자동 수집(OG 메타). 확장으로 담은 링크도 웹에서 열람하는 순간 자동 보정
PDF	브라우저에서 첫 페이지 썸네일 생성 + 본문 텍스트 추출 → 전문 검색 대상
이미지	썸네일 카드. 확장에서 담으면 원본을 실제 다운로드해 업로드(하링크 폴백)
메모	더블클릭으로 만들고 더블클릭으로 편집하는 텍스트 카드
일반 파일	한글(.hwp)·워드·엑셀·PPT·압축 등 모든 형식. 확장자 배지가 붙은 파일 카드
캔버스 잉크	펜으로 그린 획이 카드가 아니라 캔버스에 그대로 남는 투명 드로잉

정리를 강요하지 않습니다

개발 중 태그 입력 기능을 만들었다가 **일부러 제거**했습니다. 담을 때 입력할 것이 있으면 담는 속도가 느려지기 때문입니다. 분류는 나중에, 위치와 색으로 충분합니다.

기능 ② 정리 — 옮기고 묶는 것이 전부입니다

그룹 (프레임)

- **만드는 방법 4가지** — 사각형 올가미로 영역 감싸기 · 자유형 올가미 · 선택한 카드 묶기 · 빈 그룹 추가 후 채우기 (모바일은 카드를 탭해 고르는 "묶기 모드")
- **드래그로 소속 변경** — 카드를 프레임 안으로 끌면 들어가고, 밖으로 끌면 빠진다. 브라우저 확장의 목록에서도 행을 드래그해 그룹에 넣고 뺄 수 있다
- **유동 프레임** — 카드가 들어오면 프레임이 늘어나고 빠지면 줄어든다. 포함 여부가 항상 눈에 보인다. 들어온 카드는 형제와 겹치지 않게 계단식으로 비켜 놓인다

색 · 그리기 · 연결

- **색 분류** — 카드 테두리 색 5종(sky emerald amber rose 중립) + 컬러 피커로 아무 색이나. 색은 테두리로만 — 면을 칠해 내용을 가리지 않는다
- **펜 그리기** — 색 6종으로 캔버스에 직접 긁는다. 완료하면 잉크가 캔버스에 그대로 남는다(밀줄·동그라미·화살표 메모). 그리는 중 Ctrl+Z = 마지막 획 취소, Ctrl+휠 = 줌
- **연결선** — 카드와 카드를 화살표로 잇는다 (관계 표시)

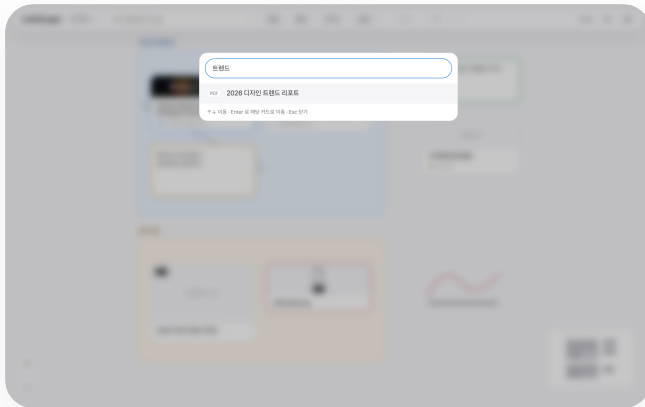
선택과 되돌리기

- **올가미 선택** — 빈 곳을 드래그하면 걸친 카드가 모두 선택. Ctrl+드래그 는 **그룹(프레임)** 위에서도 올가미가 되어 그룹 안의 카드만 골라낼 수 있다
- **다중 선택** — Ctrl+클릭 으로 하나씩 추가/제외
- **언두/리두** — 이동·삭제·묶기 등 모든 조작을 Ctrl+Z / Ctrl+Shift+Z 로 되돌린다
- **휴지통** — 삭제는 소프트 삭제. 휴지통에서 복원 가능, 15일 지나면 저장 파일까지 자동 영구 삭제

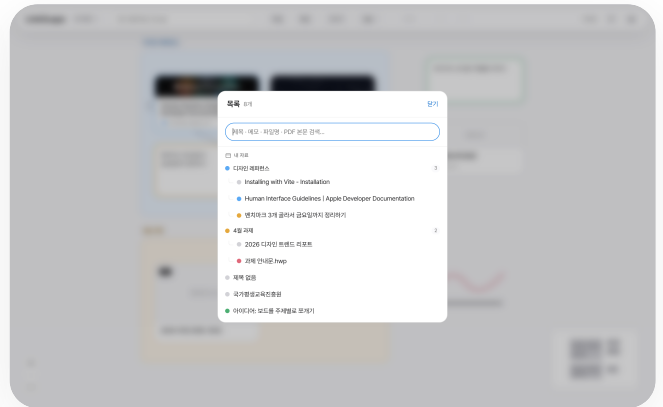
단축키

Ctrl+V 담기	Ctrl+K 검색	Ctrl+Z 되돌리기	Ctrl+D 복제
Delete 휴지통	F 화면 맞추기	Esc 닫기·취소	Space·휠클릭 팬

기능 ③ 찾기·보기 — 제목이 기억나지 않아도



Ctrl+K 검색 — PDF 본문 단어로도 카드를 찾는다



목록 보기 — 보드별 → 그룹별(↳자 종속) · 색 순서 정렬

세 가지 찾기

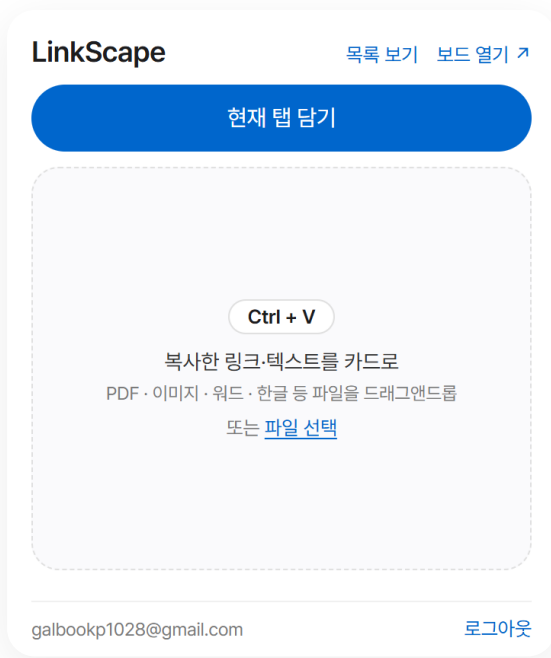
- **검색(Ctrl+K)** — 제목·메모·파일명은 물론 **PDF 본문 전문**까지. 결과를 고르면 그 카드가 있는 자리로 화면이 이동한다
- **목록 보기** — 전 보드의 자료를 그룹별로 나열. 행 클릭 = 문서·링크 바로 열기, 보조 아이콘 = "보드에서 보기"(카드 위치로 딥링크) · 이름 수정 · 삭제. 웹과 브라우저 확장 어디서나 같은 목록
- **공간 기억** — 애초에 자리를 기억하니, 캔버스를 훑는 것 자체가 검색이 된다

인앱 뷰어 — 프로그램 없이 바로 열립니다

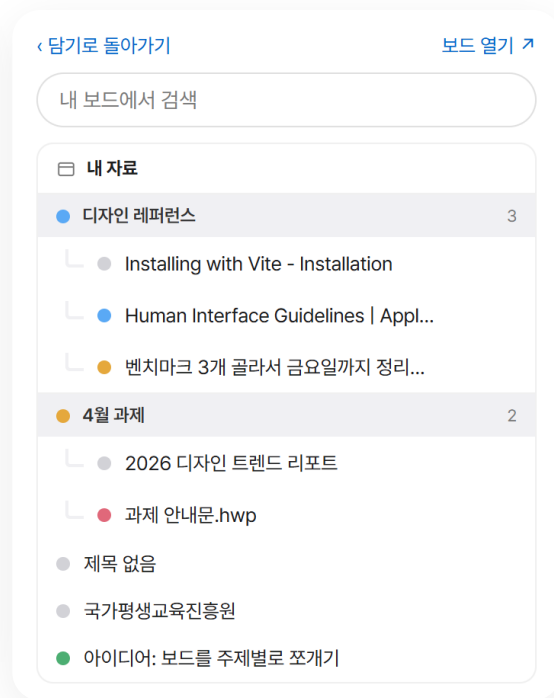
형식	처리
PDF · 이미지	내장 뷰어. PDF는 이어읽기(마지막 페이지 기억)
한글 (.hwp)	브라우저에서 직접 렌더링(hwp.js) — 한컴오피스가 없어도, 크롬·엣지·웨일 어디서든 열린다
워드·엑셀·PPT	Office Online 임베드 뷰어로 표시
그 밖의 파일	브라우저 미리보기 시도 + 다운로드 안내
다운로드	어떤 경로로든 원래 파일명 그대로 저장(한글 파일명 안 깨짐)

실제 화면 — 브라우저 확장과 데스크톱

아래는 모두 **지금 동작 중인 실제 화면**입니다. 확장 팝업은 담기와 목록을 겸하는 미니 클라이언트이고, 데스크톱 앱은 트레이에 조용히 상주하다가 클릭 한 번에 보드를 엽니다.



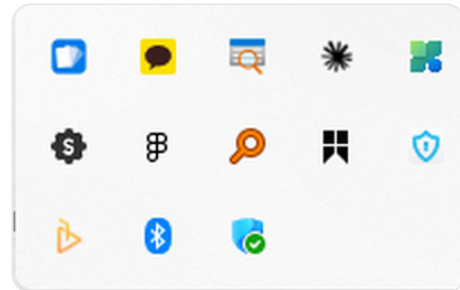
확장 팝업 — 현재 탭 담기 · Ctrl+V · 드래그 · 파일 선택



확장 목록 — 웹과 같은 그룹 구조, 행 드래그로 그룹 이동까지

데스크톱은 트레이에 삽니다

- 창을 닫아도 종료되지 않고 **트레이에 상주** — 아이콘을 클릭하면 보드가 바로 열린다
- 창이 닫히면 브라우저 엔진(WebView2)을 통째로 해제해 상주 메모리는 **42MB**
- 실행 파일 3.43MB · 설치 파일 1.26MB 의 초경량 네이티브 앱



Windows 트레이에 상주 중인 LinkScape (왼쪽 위 첫 아이콘)

하나의 보드, 4개의 플랫폼

모든 플랫폼이 같은 계정·같은 보드를 실시간으로 공유합니다. 확장으로 담으면 데스크톱 창에 바로 나타납니다.

플랫폼	역할
웹	메인 작업 공간. 반응형(모바일 하단 액션 바) + PWA 설치형. Vercel 배포 — pdflinkin.vercel.app
웨일·크롬 확장	담기의 전초기지. 우클릭·팝업·드롭존 + 목록 보기(그룹 이동·이름 수정·검색·삭제)까지 갖춘 미니 클라이언트
Windows 데스크톱	Tauri 기반 초경량 네이티브 앱. 트레이 상주 + 보드 열람. 창을 닫으면 브라우저를 엔진을 통째로 해제하는 메모리 설계
Android	Capacitor 앱 + PWA. 공유 시트에서 "LinkScape"로 바로 담기

데스크톱 앱의 숫자 (실측)

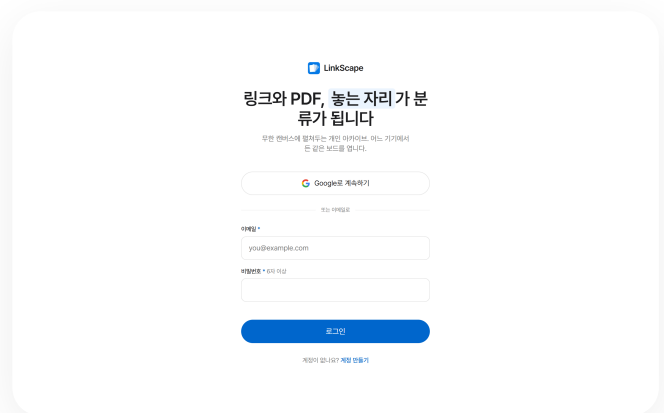
3.43MB 실행 파일 (Electron 대비 1/40)	1.26MB 설치 파일 (NSIS)	42MB 트레이 상주 메모리	92% 창 닫을 때 메모리 절감 (513→42MB)
---	-------------------------------	---------------------------	--

계정

- 구글 로그인 또는 이메일 가입 — 같은 이메일이면 자동으로 하나의 계정으로 연결
- 상단 프로필에서 구글 프로필 사진·이름 표시, 로그아웃
- 계정 삭제 시 모든 데이터 연쇄 삭제

보드

- 보드를 여러 개 만들어 주제별로 분리 (생성·이름변경·삭제)
- 보드 전환은 상단 로고 옆 스위처에서 한 번에



로그인 — Google 또는 이메일

기술 구성 — 가볍게, 브라우저가 일하게

영역	선택
웹	Next.js 16 (App Router) · React 19 · TypeScript · Tailwind CSS v4 · React Flow(캔버스) · zustand
백엔드	Supabase — Postgres · Auth(이메일+Google OAuth) · Storage · Realtime
문서 처리	pdf.js (PDF 렌더·본문 추출) · hwp.js (한글 렌더) — 전부 브라우저에서 실행
확장	순수 HTML/CSS/JS, Manifest V3 (빌드 도구 없음 — 웨일·크롬 겸용)
데스크톱 / 모바일	Tauri v2 (Rust) + WebView2 / Capacitor 8
배포	GitHub → Vercel 자동 배포 (push 하면 몇 분 안에 서비스 반영)

아키텍처 원칙

- **서버는 최소한만** — 서버 함수는 링크 미리보기 수집(unfurl) 딱 하나. PDF 파싱·썸네일 생성·링크 처리는 전부 사용자 브라우저에서 처리해 서버 비용이 들지 않는다
- **보안은 데이터베이스에서** — 모든 테이블에 행 단위 보안(RLS), 파일은 비공개 버킷 + 서명 URL. 계정 간 데이터는 구조적으로 격리
- **저장은 한 경로로** — 모든 변경이 스냅샷 diff 큐 하나를 지나므로 언두/리두·실시간 반영·저장이 어긋나지 않는다
- **실시간 동기화** — 다른 기기·확장에서 담은 카드가 열려 있는 캔버스에 즉시 나타난다 (예코 방지 설계 포함)

규모

약 1만 줄 웹 7,751 + 확장 2,320 + Rust·SQL	서버 함수 1개 나머지는 전부 클라이언트	환경변수 2개 Supabase URL · anon key 뿐	0원 현재까지 서버 비용 (무료 티어)
---	----------------------------------	--	------------------------------------

만든 방법 — 비전공자가 AI와 함께 13일

LinkScape 는 개발 전공자가 아닌 1인이 AI 코딩 에이전트(Anthropic Claude Code)와 함께 **2026년 7월 13일부터 13일간(실제 작업 8일, 자투리 시간)** 만들었습니다. 4개 플랫폼 전부가 이 기간의 결과물입니다.

13일

기획부터 4개 플랫폼
배포까지

1명

비전공 개인 + AI 에이전트

57회

GitHub 커밋 (작업 단위
기록)

4종 검증

타입·문법·빌드·브라우저
E2E

AI를 다룬 방식 — 시키는 것이 아니라 운영하는 것

방법	내용
기획 먼저 (/plan)	구현 전에 반드시 기획 문서를 쓰게 하고, AI가 임의로 가정한 부분은 질문으로 남기게 했다. 코드가 없을 때 방향을 바꾸는 것은 공짜다
인수인계 (/handover)	작업이 끝날 때마다 진행 상황·해결한 문제·지켜야 할 원칙을 문서로 축적 — 새 대화에서도 맥락이 이어진다
커밋 단위 기록	작업 단위마다 GitHub에 커밋 — 잘못되면 되돌릴 지점이 항상 있다. push 하면 Vercel이 자동 배포
실행으로 검증	"완료했습니다"라는 말 대신, AI가 직접 브라우저를 띄워 로그인·클릭·DB 조회까지 하는 자동 테스트를 만들어 통과해야 커밋

반복되는 지시는 슬래시 명령어(스킬)로 직접 만들어 재사용했습니다 — 좋은 프롬프트를 한 번 찾는 것보다, 매번 똑같이 재사용되는 형태로 만드는 것이 결과를 바꿨습니다.



지금 바로 사용해 볼 수 있습니다

서비스 주소 <https://pdflinkin.vercel.app>
이메일 또는 구글 계정으로 즉시 가입 · 무료