

고유웹 무료 가이드

바이브코딩 A to Z

상업적으로 써도 되는 AI 홈페이지 만들기

무료 전자책

1판 · 고유웹



들어가며

AI로 홈페이지 만드는 법을 알려주는 영상은 많습니다. 그런데 거의 다 같은 데서 끝납니다. "무료로 전 세계에 공개됩니다."

저는 홈페이지 만드는 일을 합니다. 그러다 보니 이런 건 약관을 한 번 읽어보게 됩니다. 읽어보니 대부분의 영상이 추천하는 그 무료 플랜은, **사업자가 쓰면 안 되는 플랜**이었습니다.

이 책은 그 얘기부터 시작해서, 코딩을 안 배운 사장님이 자기 홈페이지를 만들어 **상업적으로 써도 되는 곳에** 공개하기까지의 전 과정을 담았습니다.

이 책을 읽으면

- 내 사이트를 무료 플랜에 올려도 되는지 스스로 판단할 수 있습니다
- ChatGPT 데스크톱 앱의 Codex로 사이트를 만들 수 있습니다
- 문의 폼에 들어온 내용이 실제로 저장되게 만들 수 있습니다
- 키를 잘못 다뤄서 데이터가 새는 일을 피할 수 있습니다
- 내 컴퓨터가 꺼져 있어도 열리는 주소를 갖게 됩니다

이 책을 읽어도 안 되는 것

- 내 도메인 연결, 문의 알림 받기, 실제 월 비용 계산 — 2편에서 다룹니다
- 쇼핑몰, 결제, 회원가입 — 범위 밖입니다

준비물

준비물	비용	없으면
맥 또는 윈도우 PC	—	필수입니다
ChatGPT 계정	무료 시작 가능	구글 계정으로 가입
GitHub 계정	무료	3장에서 만듭니다
Supabase 계정	무료	7장에서 만듭니다
Cloudflare 계정	무료	9장에서 만듭니다

유료로 쓸 가능성이 있는 것은 ChatGPT 하나뿐입니다. 나머지는 이 책 범위에서 전부 무료입니다.

표기 약속

그대로 붙여넣으세요 — 이 상자 안의 글은 Codex 입력창에 그대로 복사해 넣으면 됩니다.

⚠ 이것만은 — 건너뛰면 나중에 문제가 되는 것입니다.

0장 · 5분 요약

전체 지도

① 도구 세팅	ChatGPT 데스크톱 앱 설치 → Codex 모드
② 저장할 곳	Git · GitHub 연결 (Codex가 거의 다 해줌)
③ 화면 만들기	"이런 사이트 만들어줘" (프론트엔드)
④ 뒤편 붙이기	Supabase 연결 (백엔드 · 데이터베이스)
⑤ 공개하기	Cloudflare 배포 (주소가 생김)

걸리는 시간

처음 하시면 두 시간에서 반나절입니다. 대부분 기다리는 시간이고, 손으로 하는 건 계정 만들기과 버튼 몇 번입니다.

한 달에 드는 돈

항목	이 책 범위에서
ChatGPT	무료도 가능, 권장은 Plus 월 20달러
GitHub	0원
Supabase	0원
Cloudflare	0원

홈페이지 외주 견적이 보통 200만 원부터입니다. 한 달 3만 원 정도로 사이트를 여러 개 만들 수 있습니다.

가장 중요한 세 가지

1. 무료 배포처를 아무 데나 고르지 마세요. 사업용으로 쓰면 안 되는 곳이 있습니다. (1장)

- 2. **비밀 키를 AI 대화창에 붙여넣지 마세요.** 공식 문서가 채팅으로 주고받지 말라고 못 박아 뒀습니다. (8장)
 - 3. **RLS를 켜세요.** 안 켜면 문의 내용이 아무한테나 열립니다. (8장)
-

1장 · 시작 전에: 무료 배포의 함정

영상들이 알려주지 않는 문장

AI로 만든 사이트를 공개하는 방법으로 가장 많이 추천되는 곳이 Vercel입니다. 무료 플랜 이름은 Hobby입니다. 그 Hobby 플랜 약관에 이런 문장이 있습니다.

Commercial usage

Hobby teams are restricted to non-commercial personal use only. All commercial usage of the platform requires either a Pro or Enterprise plan.

Commercial usage is defined as any Deployment that is used for the purpose of financial gain of **anyone** involved in **any part of the production** of the project, including a paid employee or consultant *writing the code*. Examples of this include, but are not limited to

Hobby teams are restricted to non-commercial personal use only. All commercial usage of the platform requires either a Pro or Enterprise plan.

무료(Hobby) 플랜은 비상업적 개인 용도로만 사용할 수 있습니다. 플랫폼의 모든 상업적 사용은 Pro 또는 Enterprise 플랜이 필요합니다.

그리고 바로 아래에 '상업적 사용'의 예시가 나옵니다.

Commercial usage is defined as any Deployment that is used for the purpose of financial gain of **anyone** involved in **any part of the production** of the project, including a paid employee or consultant writing the code. Examples of this include, but are not limited to, the following:

- Any method of requesting or processing payment from visitors of the site
- Advertising the sale of a product or service
- Receiving payment to create, update, or host the site
- Affiliate linking is the primary purpose of the site

- Advertising the sale of a product or service **상품이나 서비스 판매를 광고하는 것**
- Receiving payment to create, update, or host the site **사이트를 만들거나 수정하거나 호스팅하는 대가로 돈을 받는 것**

내 사이트는 해당될까

판단표입니다. 현재 공식 약관 기준입니다.

내 사이트	무료 플랜
내 가게·회사 소개 사이트	❌ 상업적 사용입니다
서비스 안내와 문의 폼이 있는 사이트	❌ 상업적 사용입니다
클라이언트에게 돈 받고 만들어 준 사이트	❌ 상업적 사용입니다
취미로 만든 개인 블로그 (광고·수익 없음)	○ 괜찮습니다
공부용 연습 사이트	○ 괜찮습니다

정리하면, 홈페이지로 뭔가를 팔거나 알리려는 순간 무료 플랜 밖입니다.

그래서 큰일이 나느냐

당장 무슨 일이 생기는 건 아닐 수 있습니다. Pro도 월 20달러부터라 아주 비싸지도 않습니다. Vercel이 나쁜 서비스라는 얘기도 아닙니다.

다만 사업자에게 홈페이지는 브랜드 그 자체입니다. "여기 걸면 안 된다"고 써 있는 자리에 우리 간판을 걸어두고 몇 년을 모른 채 가는 것, 이건 만들기 전에 알아야 하는 일이라고 생각합니다.

이 책이 Cloudflare를 쓰는 이유

제 사이트도, 고객 사이트들도 전부 Cloudflare에 올라가 있습니다. 여기는 무료 플랜을 상업적으로 써도 됩니다.

요금 페이지, 이용약관, 한도 문서까지 확인했는데 그런 제한 문구가 없습니다. 무료 플랜에서 막아둔 것은 **사이트에서 카드 결제를 직접 받는 경우** 정도입니다. 소개 사이트나 문의 폼은 해당이 없습니다.

원문 링크는 부록 D에 있습니다. 직접 확인해 보세요. 약관은 바뀔니다. 이 책의 판단은 집필 시점 기준입니다.

2장 · 도구 고르기

질문하는 AI와 대신 일하는 AI

ChatGPT, 제미나이, 클로드. 이런 걸 LLM이라고 합니다. 엄청난 데이터로 학습한 두뇌입니다. 그런데 두뇌만 있으면 질문에 답만 합니다.

실제로 내 컴퓨터에 파일을 만들고 코드를 쓰려면 손발이 필요합니다. 그게 **코딩 에이전트**입니다.

	하는 일	예
LLM	물어보면 답한다	ChatGPT 대화, 제미나이, 클로드
코딩 에이전트	파일을 만들고 고친다	Codex, Claude Code, Cursor

이 차이만 기억하시면 됩니다.

예전에는 그랬습니다

에이전트를 쓰려면 검은 터미널 창을 열고, 영어 명령어로 설치하고, 편집기를 따로 깔고, Git도 세팅해야 했습니다. 뭔지도 모르면서 따라 해야 했습니다.

```
Last login: Sat Sep 20 05:31:12 on ttys003
user@mac ~/my-site % npm install
npm warn deprecated inflight@1.0.6: This module is not supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
* reify:react: timing reifyNode:node_modules/react Completed in 412ms
* reify:next: http fetch GET 200 https://registry.npmjs.org/next/-/next-15.3.1.tgz
* reify:@supabase/supabase-js: timing reifyNode Completed in 903ms
* reify:typescript: http fetch GET 200 https://registry.npmjs.org/typescript

added 412 packages, and audited 413 packages in 18s
142 packages are looking for funding
  run npm fund for details

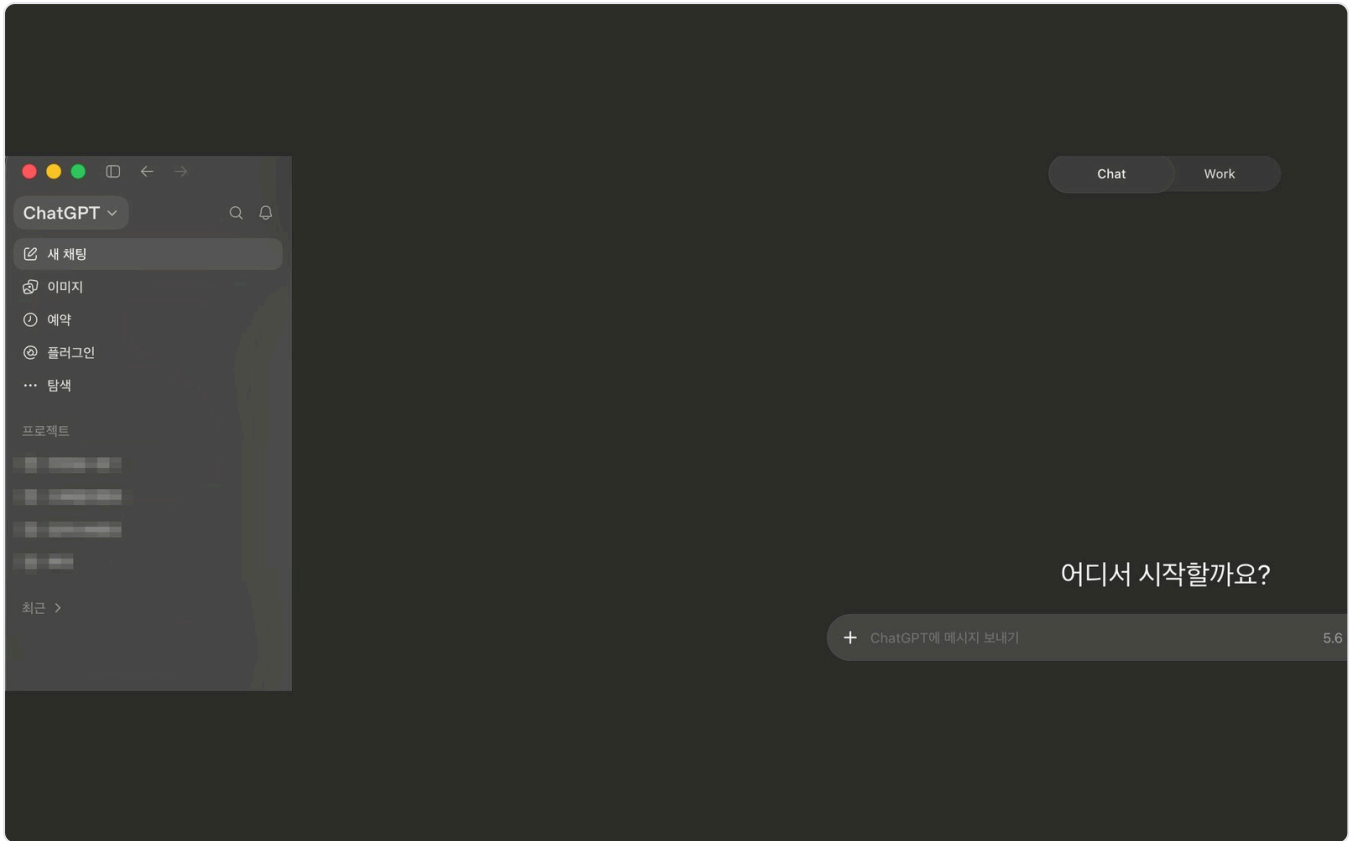
found 0 vulnerabilities

user@mac ~/my-site % git init
Initialized empty Git repository in /Users/user/my-site/.git/
user@mac ~/my-site % code .
user@mac ~/my-site % █
```

그래서 VS Code 같은 편집기 통합 도구가 인기였습니다.

지금은 앱 하나면 됩니다

이 책은 **Codex**를 씁니다. OpenAI가 만든 코딩 에이전트인데, 지금은 ChatGPT 데스크톱 앱 안에 그냥 들어 있습니다. 앱 하나 설치하면 끝이고, 결과물이 좋습니다.



요금제 고르기

집필 시점 공식 요금입니다.

플랜	월 요금	Codex
Free	0달러	됩니다. 한도가 적습니다
Go	8달러	가벼운 작업용
Plus	20달러	이 책의 권장
Pro	100달러부터	Plus의 5배 또는 20배 한도

무료 플랜에서도 Codex가 열립니다. 다만 한도가 적어서 사이트 하나 만들다 보면 금방 막힙니다. 가장 싼 유료는 Go 월 8달러입니다.

사이트 하나 만드는 동안 한도 걱정 안 하시려면 **Plus 월 20달러**를 권합니다. 더 비싼 요금제는 나중에 필요할 때 생각하시면 됩니다.

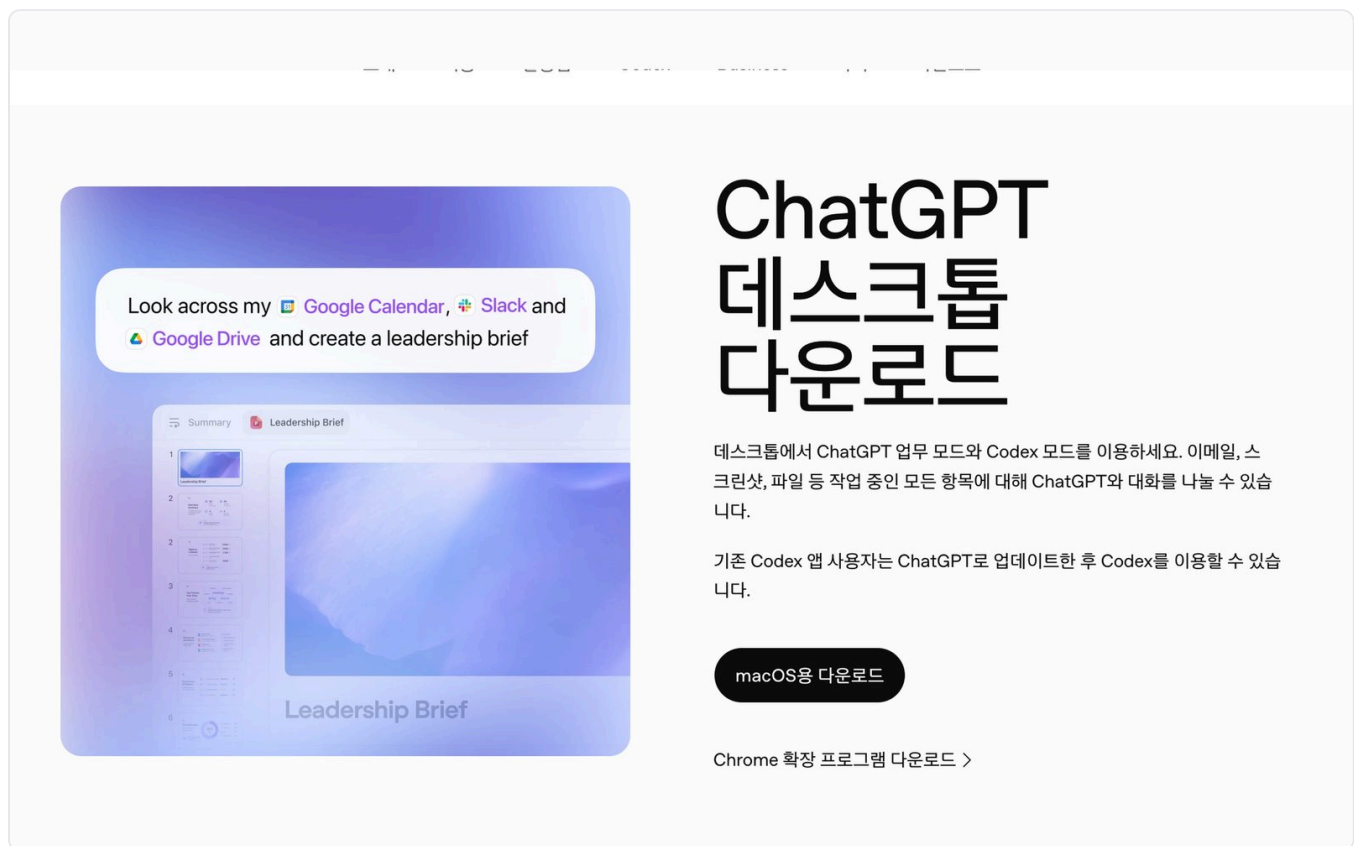
 **이것만은** — 한도는 5시간 단위로 다시 채워집니다. 막혔다고 결제부터 올리지 마시고, 몇 시간 뒤에 이어서 해보세요.

3장 · 설치

전부 다섯 단계입니다. 누르는 자리를 사진으로 표시했습니다.

① 내려받기

구글에 **ChatGPT 데스크톱**을 검색하거나 `chatgpt.com/download` 로 갑니다. 내 컴퓨터에 맞는 파일을 받으시면 됩니다.



맥은 `macOS용 다운로드` , 윈도우는 윈도우용 버튼을 누릅니다.

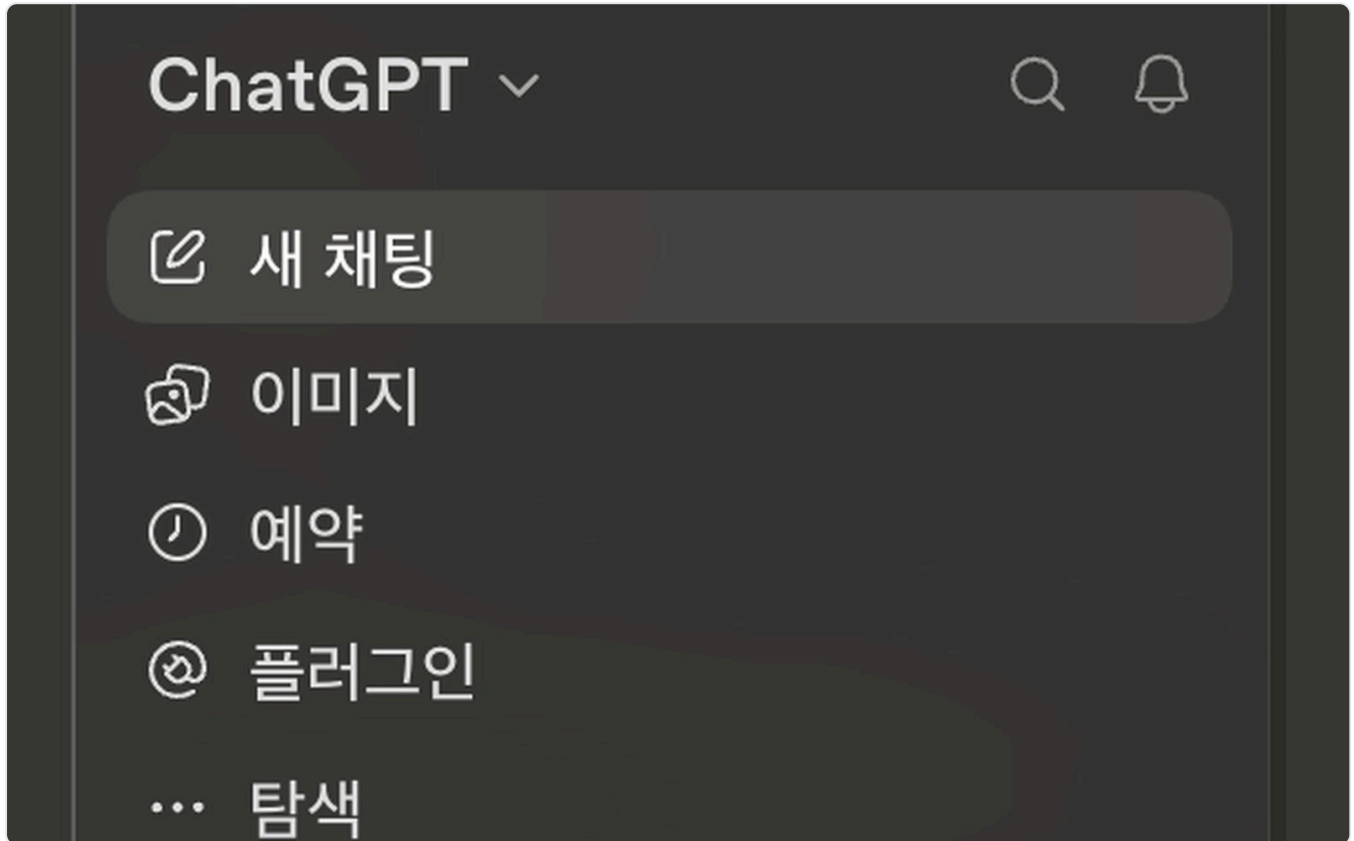
② 로그인

처음 실행하면 로그인 창이 뜹니다. **구글 계정으로 로그인**하시거나, 없으면 계정을 하나 만들어서 들어가시면 됩니다.

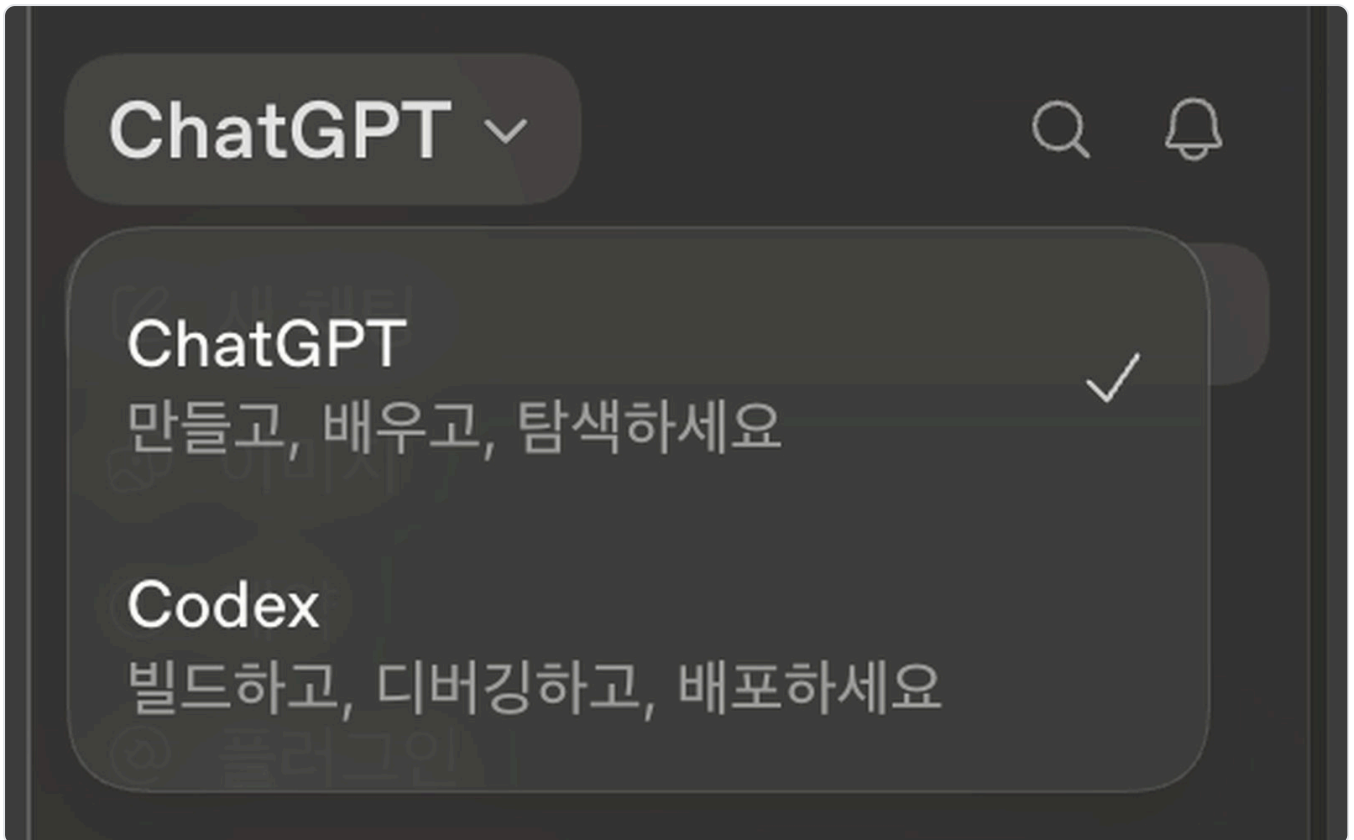
③ Codex 모드로 바꾸기 — 여기가 핵심입니다

앱을 열면 그냥 ChatGPT입니다. 이 상태로 코딩 도구가 아닙니다.

왼쪽 위에 ChatGPT ▾ 라고 써 있는 걸 누르세요.



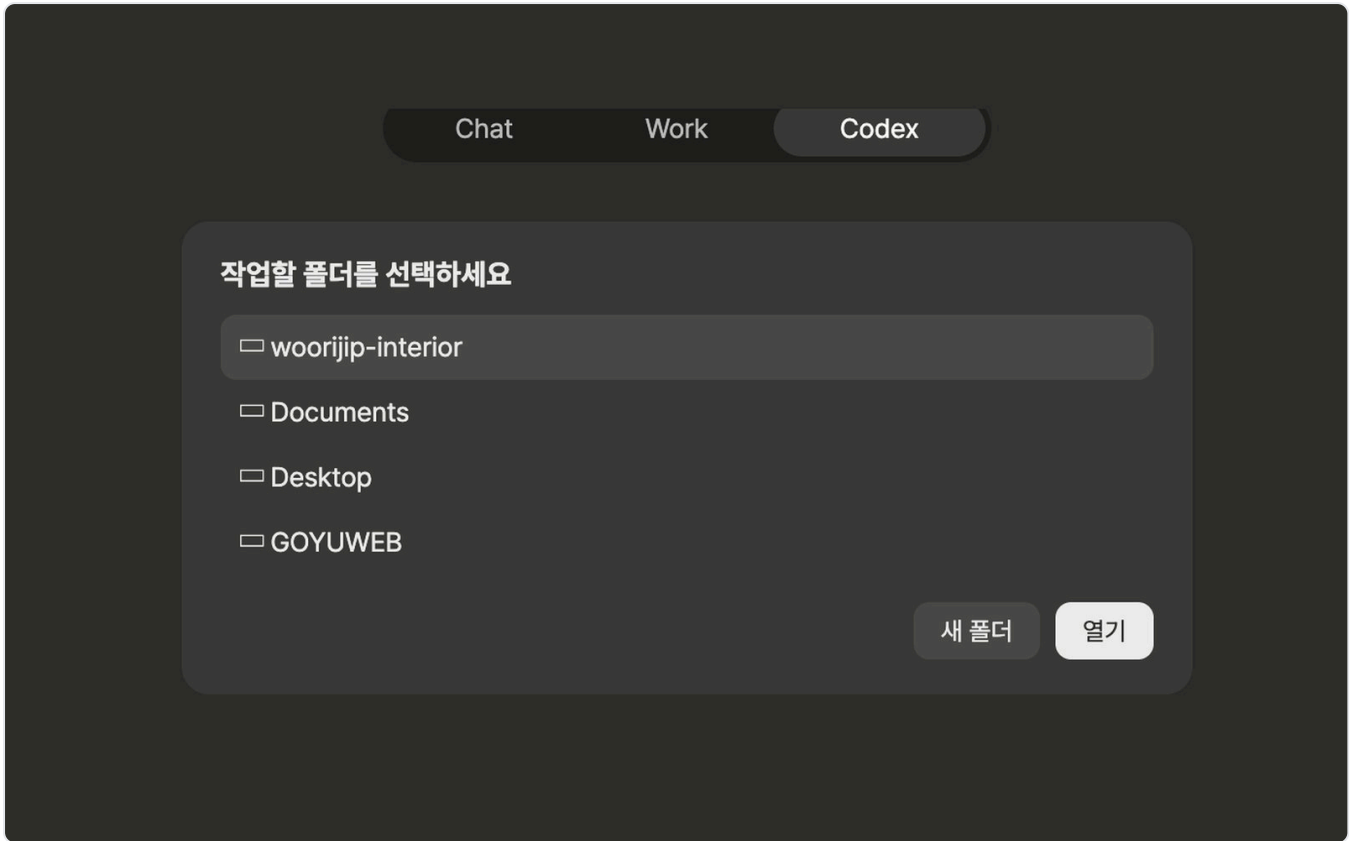
누르면 모드를 고르는 메뉴가 열립니다. 여기서 딱 하나만 기억하시면 됩니다. **Codex**.



⚠ 이것만은 — 일반 채팅(ChatGPT)이 아니라 Codex를 눌러야 파일을 만들 수 있습니다. 사이드바를 아무리 찾아도 없습니다. 왼쪽 위 이름을 누르는 게 맞습니다.

④ 작업 폴더 고르기

Codex를 누르면 어느 폴더에서 작업할지 물어봅니다. **새 폴더를 하나 만들어서** 고르시면 됩니다. 사이트 하나에 폴더 하나라고 생각하시면 편합니다.



⑤ 끝

대화창이 뜹니다. 여기에 한국어로 말하면 코드를 짜주고, 과정도 보여주고, 미리보기까지 됩니다.

무엇을 만들어 볼까요?

+ 코덱스에 작업 요청하기

GPT-5.6 ▾



설치는 이걸로 끝입니다.

4장 · Git과 GitHub

왜 필요한가

카카오톡 대화 백업을 아실 겁니다. 폰을 바꿔도 대화가 넘어오는 게 백업 덕분입니다.

코드도 똑같습니다. 수정할 때마다 저장해 두고, 망치면 되돌려야 합니다. 이걸 해주는 게 **Git**입니다. 그리고 그 백업을 인터넷 서버에 올려두는 게 **GitHub**입니다.

나중에 사이트를 공개할 때 GitHub가 필요하니까 지금 연결해 둡니다.

Codex한테 시키면 됩니다

그대로 붙여넣으세요

GitHub 저장소 만들고 이 프로젝트 연결해줘. 필요한 거 있으면 설치도 해줘.

무엇을 만들어 볼까요?

+ GitHub 저장소 만들고 이 프로젝트 연결해줘. 필요한 거 있으면 설치도 해줘.

GPT-5.6 ▾ ⬆

그러면 Git이 없으면 깔아주고, GitHub 연결 도구도 깔고, 저장소도 만듭니다.

GitHub 저장소 만들고 이 프로젝트 연결해줘. 필요한 거 있으면 설치도 해줘.

```
✓ git 확인 중... git version 2.39.5 (Apple Git-154)
✓ GitHub CLI 설치 brew install gh
✓ gh 설치 완료 gh version 2.63.2
● GitHub 인증이 필요합니다. 브라우저에서 승인해 주세요...
gh auth login --web
```

🔄 브라우저 승인 기다리는 중

사람이 눌러야 하는 단 한 곳

딱 한 군데, 사람이 직접 눌러야 하는 데가 있습니다. **여기입니다.**



GitHub

Authorize Codex

ChatGPT Codex Connector by OpenAI wants to access your **goyuweb** account.

Repositories · Read and write access

Cancel

Authorize

GitHub가 "이 컴퓨터, 진짜 너 맞아?" 하고 물어보는 겁니다. **이건 AI가 대신 못 누릅니다.**

- GitHub 계정이 없으면 여기서 가입하시면 됩니다
- 있으면 로그인하고 **Authorize**를 누르시면 됩니다

그다음은 알아서 끝납니다

GitHub 저장소 만들고 이 프로젝트 연결해줘. 필요한 거 있으면 설치도 해줘.

```
✓ git 확인 완료
✓ GitHub CLI 설치 완료
✓ GitHub 인증 완료 goyuweb
✓ 저장소 생성 github.com/goyuweb/woorijip-interior
✓ 첫 커밋 및 푸시 완료 main · 14 files
```

GitHub 연결이 끝났습니다.

한 일

- GitHub CLI(gh)를 설치하고 goyuweb 계정으로 로그인했습니다.
- goyuweb/woorijip-interior 저장소를 만들고 현재 프로젝트를 연결했습니다.
- 지금까지의 파일을 첫 커밋으로 올렸습니다.

앞으로 "GitHub에 올려줘"라고 하면 그 시점 코드를 저장합니다.

이제 "지금까지 작업한 거 GitHub에 올려줘"라고 하면 그 시점 코드가 저장됩니다.

⚠ **이것만은** — 컴퓨터 환경에 따라 설치나 로그인을 직접 하셔야 할 수도 있습니다. Codex가 "이 명령을 실행할까요?"라고 물으면 승인해 주시면 됩니다. 이 승인 하나만 넘기면 나머지는 AI가 마무리합니다.

5장 · 사이트 만들기

먼저 업체 정보를 정리하세요

이걸 미리 넣어주면 문구까지 내 업체 걸로 채워줍니다. 빈칸을 채워 두세요.

항목	내 정보
업체 이름	
업종	
지역	
연락처	
경력·규모	
강점 세 가지	
필요한 페이지	
올릴 곳	Cloudflare

요청 문장

이 책의 예시는 가상 업체 **우리집인테리어**입니다.

그대로 붙여넣으세요 (내 정보로 바꿔서)

우리집인테리어라는 업체 홈페이지를 만들 거야. 업체 정보는 이래. 부산 해운대, 주거 인테리어·상가 인테리어, 010-0000-0000, 10년 경력 1인 시공. 이 정보에 맞게 메인, 시공 사례, 서비스 안내, 문의 페이지로 홈페이지 만들어줘. 모바일에서도 잘 보이게. 나중에 Cloudflare에 올릴 거야.

무엇을 만들어 볼까요?

우리집인테리어라는 업체 홈페이지를 만들 거야. 업체 정보는 이래. 부산 해운대, 주거 인테리어·상가 인테리어, 010-0000-0000, 10년 경력 1인 시공. 이 정보에 맞게 메인, 시공 사례, 서비스 안내, 문의 페이지로 홈페이지 만들어줘. 모바일에서도 잘 보이게. 나중에 Cloudflare Pages에 올릴 거야.]

GPT-5.6



"나중에 어디에 올릴 거야"를 미리 말해두세요. 그래야 거기 맞게 설정까지 해줍니다. 9장에서 시간을 아깁니다.

Codex가 먼저 계획을 잡습니다

우리집인테리어라는 업체 홈페이지를 만들 거야. 업체 정보는 이래. 부산 해운대, 주거 인테리어·상가 인
테리어, 010-0000-0000, 10년 경력 1인 시공. 이 정보에 맞게 메인, 시공 사례, 서비스 안내, 문의 페이
지로 홈페이지 만들어줘. 모바일에서도 잘 보이게. 나중에 Cloudflare Pages에 올릴 거야.

좋습니다. 업체 정보를 반영해서 이렇게 만들겠습니다.

구성

- 메인 · 히어로(부산 해운대 10년 · 1인 시공), 강점 3가지, 시공 사례 미리보기, 문의 유도
- 시공 사례 · 주거/상가 필터, 사진 그리드
- 서비스 안내 · 주거 인테리어 / 상가 인테리어 / 진행 절차
- 문의 · 이름·연락처·공간 유형·지역·예산·내용 폼

기술 선택

여기서 내용이 마음에 안 들면 이때 고쳐 달라고 하시면 됩니다. 파일이 만들어진 뒤보다 훨씬 빠릅니다.

파일을 하나씩 만듭니다

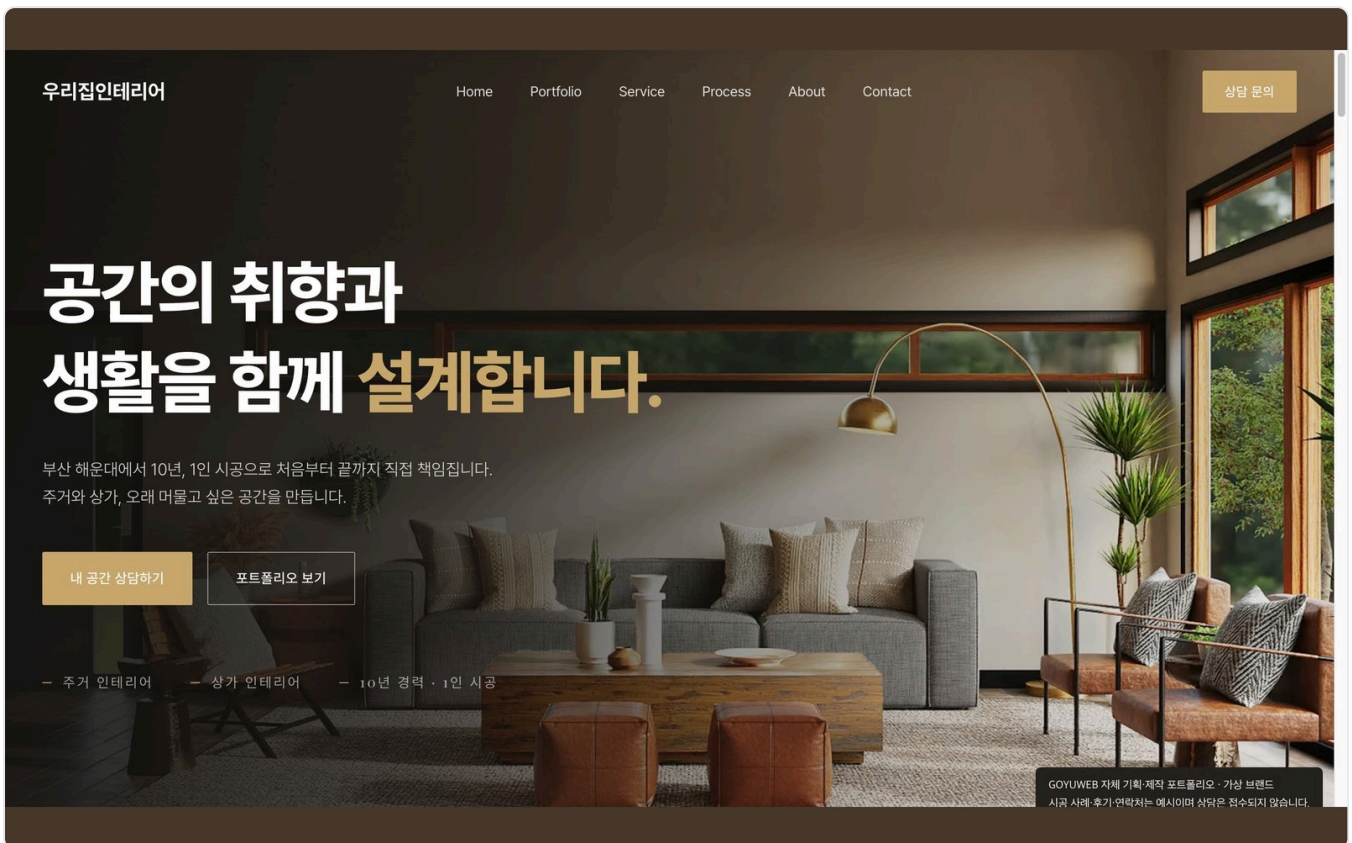
우리집인테리어라는 업체 홈페이지를 만들 거야. 업체 정보는 이래. 부산 해운대, 주거 인테리어·상가 인테리어, 010-0000-0000, 10년 경력 1인 시공. 이 정보에 맞게 메인, 시공 사례, 서비스 안내, 문의 페이지로 홈페이지 만들어줘. 모바일에서도 잘 보이게. 나중에 Cloudflare Pages에 올릴 거야.

업체 정보를 반영해서 순수 HTML + CSS + JavaScript로 4페이지 구성을 만듭니다.

- + index.html
- + css/style.css
- + js/script.js 생성

🔄 파일 만드는 중

5분 정도면 화면이 뜹니다



Codex가 알려주는 `http://localhost:3000` 같은 주소를 브라우저에 넣으면 보입니다.

⚠ 이것만은 — 이걸 아직 **내 컴퓨터 안에서만** 보이는 상태입니다. 인터넷에 올라간 게 아닙니다. 이 주소를 남에게 보내도 안 열립니다. 공개는 9장에서 합니다.

6장 · 왜 문의 폼이 먹통인가

직접 해보세요

만든 사이트의 문의 폼에 이름을 넣고 보내기를 눌러 봅니다.

공간 정보가 구체적일수록 더 정확한 상담이 가능합니다.

이름 * 연락처 *

김민수 010-1234-5678

이메일 공간 유형 *

이메일을 입력해주세요. 아파트 ▼

이름 * 연락처 *

"문의가 접수되었습니다"가 뜹니다.

예) 500

예) 5,000원

희망 상담 방식

선택해주세요.

문의 내용 *

공간에 대한 고민이나 원하는 분위기 등을 자유롭게 입력해주세요.

☐ 개인정보 수집 및 이용에 동의합니다. (필수)

상담 문의 남기기 →

문의가 접수되었습니다. 확인 후 순차적으로 연락드리겠습니다.

그런데 이게 어디로 갔을까요? Codex한테 물어봅시다.

그대로 붙여넣으세요

방금 문의 폼에 입력한 내용 어디 저장됐어?

우리집인테리어라는 업체 홈페이지를 만들 거야. 업체 정보는 이래. 부산 해운대, 주거 인테리어-상가 인테리어, 010-0000-0000, 10년 경력 1인 시공. 이 정보에 맞게 메인, 시공 사례, 서비스 안내, 문의 페이지로 홈페이지 만들어줘. 모바일에서도 잘 보이게. 나중에 Cloudflare Pages에 올릴 거야.

완성했습니다. <http://localhost:3000> 에서 확인해 보세요.

방금 문의 폼에 입력한 내용 어디 저장됐어?

지금은 아무 데도 저장되지 않습니다.

현재 문의 폼은 화면에서 입력값을 검사한 뒤 "접수되었습니다" 메시지만 보여주고 끝납니다. 데이터를 받아서 보관할 서버나 데이터베이스가 연결되어 있지 않기 때문입니다.

실제로 저장하려면 백엔드가 필요합니다. 가장 간단한 방법은 Supabase 같은 서비스에 테이블을 만들고, 폼 제출 시 그 테이블에 기록하도록 연결하는 것입니다. 원하시면 바로 연결해 드릴게요.

"지금은 아무 데도 저장되지 않습니다."

화면에서 입력값을 검사한 뒤 "접수되었습니다" 메시지만 보여주고 끝납니다. 데이터를 받아서 보관할 서버나 데이터베이스가 연결되어 있지 않기 때문입니다.

식당으로 비유하면

식당	홈페이지	하는 일
손님이 보는 홀	프론트엔드	화면, 버튼, 글자
안 보이는 주방	백엔드	받아서 처리하는 곳
주방 안 냉장고	데이터베이스	쌓아두는 곳

지금은 홀만 있고 주방이 없는 겁니다. 손님이 주문서를 써서 냈는데 받아가는 사람이 없는 상태입니다.

여기가 제일 많이 포기하는 지점입니다

서버, 데이터베이스, API. 원래 배울 게 많습니다. 그래도 이 부분은 개념을 알고 가셔야 합니다. 나중에 문제가 생겼을 때 AI한테 뭘 요청할지 감이 오기 때문입니다.

문의 폼 하나가 되려면 두 가지가 필요합니다.

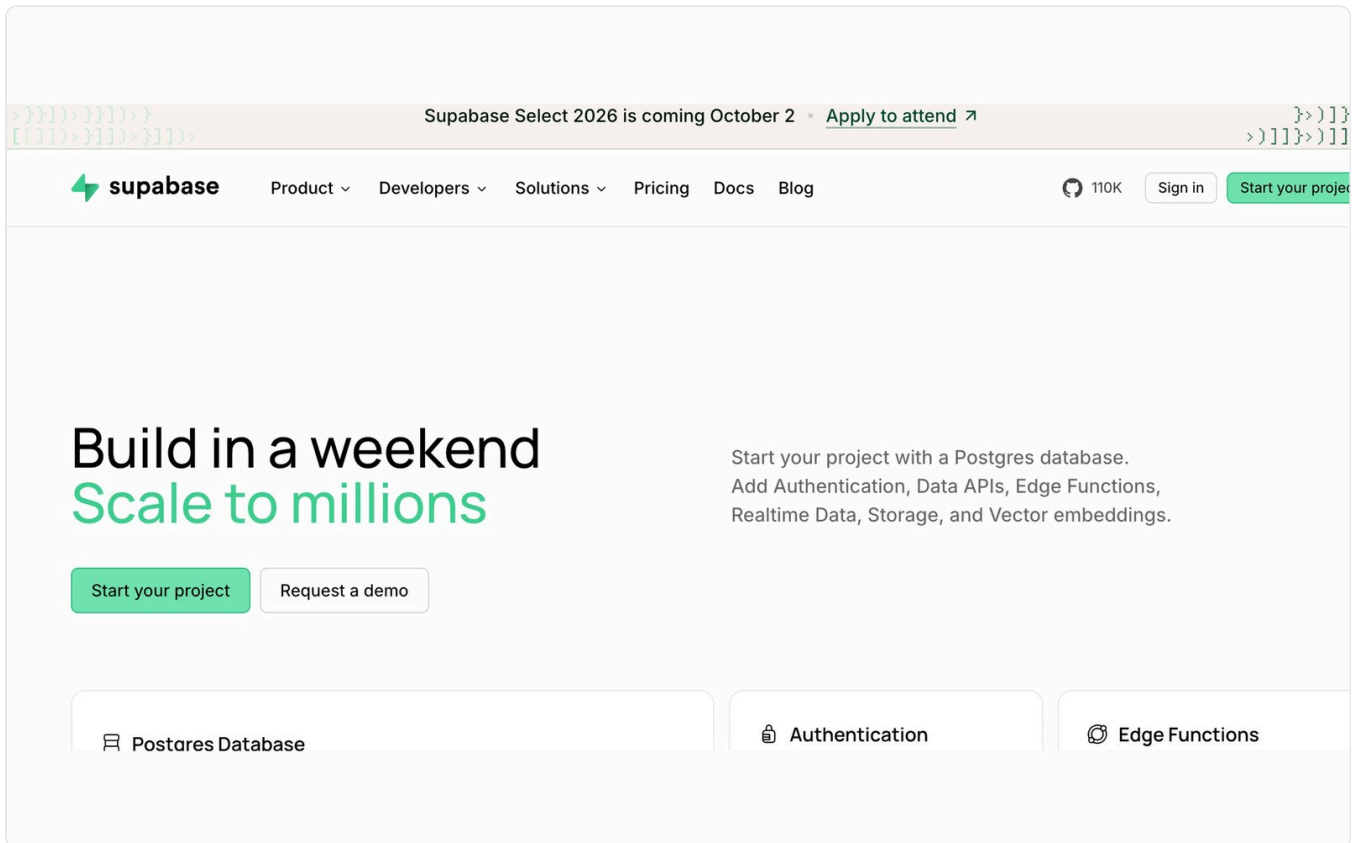
1. 문의 내용을 **저장할 곳** (냉장고)
2. 화면과 저장소가 **소통하는 코드** (주방)

예전에는 이걸 위해 서버 컴퓨터를 빌리고, 코드를 올리고, 프로그램을 깔아야 했습니다. 지금은 주방과 냉장고를 통째로 빌려주는 서비스가 있습니다. **Supabase**입니다.

7장 · Supabase 연결

① 가입

supabase.com 에 접속합니다. 4장에서 만든 GitHub 계정으로 가입하면 제일 편합니다.



Start your project 를 누릅니다.

② 조직과 프로젝트 만들기

Your organizations

Create an organization
Manage your team and projects in one place.

+ New organization

조직을 하나 만들고, 이어서 프로젝트를 만듭니다.

Create a new project

Your project will have its own dedicated instance and full Postgres database.
An API will be set up so you can easily interact with your new database.

Organization

goyulab

FREE

GitHub (optional)

Connect GitHub

Ideal for agent-first workflows. Update your schema in code and push it to GitHub. Supabase deploys the changes. [Learn more](#)

Project name

goyuweb's Project

Database password

Type in a strong password

This is the password to your Postgres database, so it must be strong and hard to guess. [Generate a password](#).

항목	넣을 것
Project name	사이트 이름 (예: woorijip-interior)
Database password	Generate a password 버튼을 누르고 따로 적어두세요
Region	Asia-Pacific (Northeast Asia) — 한국에서 가까운 곳

⚠ 이것만은 — Database password는 다시 볼 수 없습니다. 생성 버튼을 누른 즉시 메모장이든 어디든 적어두세요. 이 책 범위에서는 쓸 일이 없지만, 나중에 필요할 때 없으면 곤란합니다.

③ 서버가 생겼습니다

goyulab FREE
/
woorijip-interior
/
main PRODUCTION
Connect

woorijip-interior

https://geiakhpnhikiwqsjgavp.supabase.co
Copy

STATUS
Healthy

COMPUTE
NANO

GITHUB
No repository connected

RECENT BRANCH
No branches

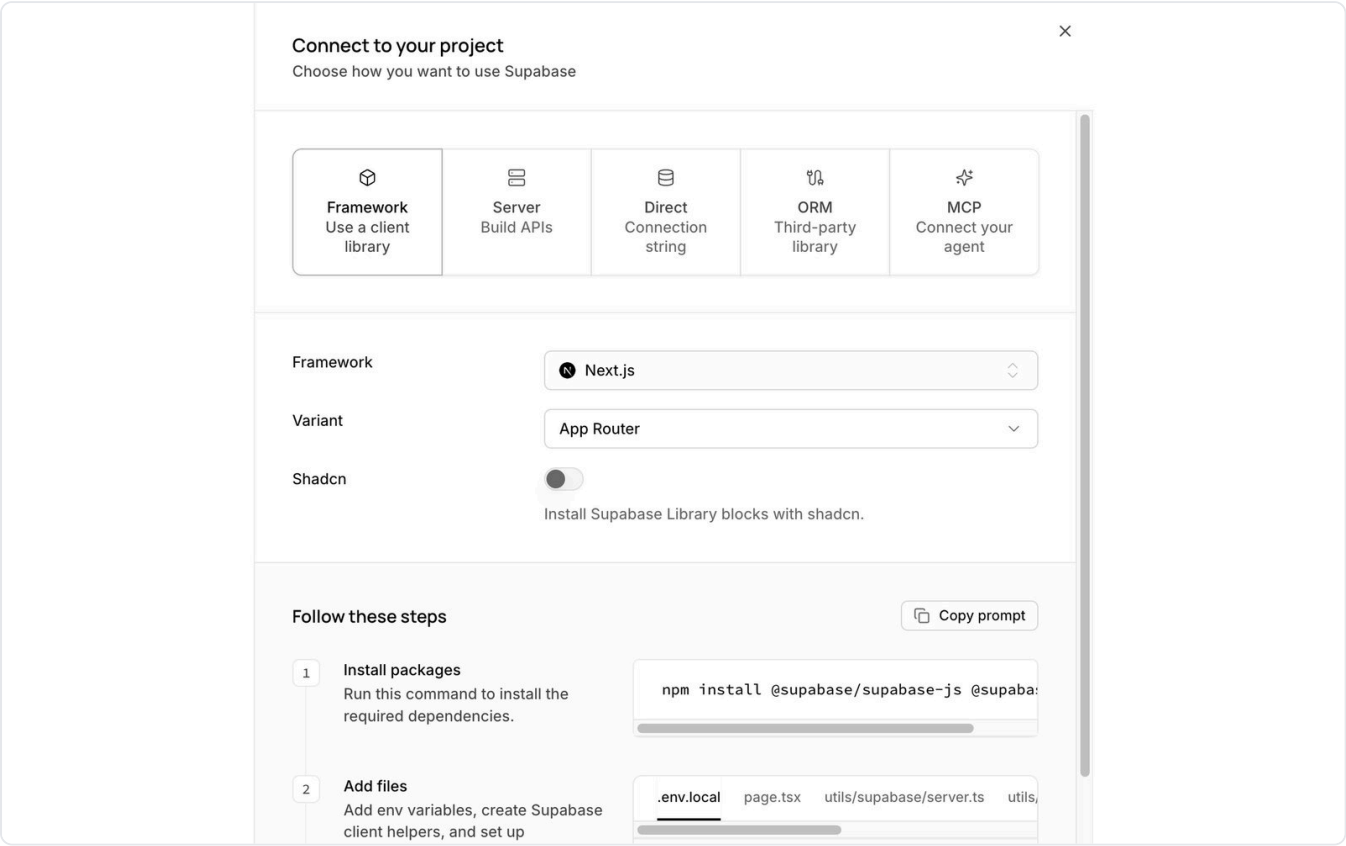
LAST MIGRATION
No migrations

LAST BACKUP
No backups

이 순간 내 서버가 실제로 생긴 겁니다. 아직 텅 빈 냉장고고요. 칸을 나눠야 합니다.

④ 주소와 열쇠 찾기

프로젝트 화면 위쪽의 **Connect** 버튼을 누릅니다.



여기서 필요한 게 두 개입니다.

이름	생긴 모양	비유
Project URL	https://xxxxx.supabase.co	주소
Publishable key (공개 키)	sb_publishable_...	열쇠

API Keys

Configure API keys to securely control access to your project

Docs

Publishable and secret API keys Legacy anon, service_role API keys

Your new API keys are here

We've updated our API keys to better support your application needs.

[Join the discussion on GitHub](#)

Having trouble with the new API keys? [Contact support](#)

Publishable key

This key is safe to use in a browser if you have enabled Row Level Security (RLS) for your tables and configured policies.

+ New publishable key

NAME	API KEY
default No description	sb_publishable_FGfLX6TtkmLbKl37kzBADg_j-ELX...
Publishable keys can be safely shared publicly	



⚠ 이것만은 — 같은 화면 아래에 **Secret keys**도 있습니다. 그건 지금 안 씁니다. 복사하지 마세요. 이유는 8장에 있습니다.

⑤ Codex에 붙여넣기

그대로 붙여넣으세요 (내 값으로 바뀌서)

지금 만든 사이트를 Supabase랑 연결해줘. 문의 폼 내용이 저장되게. 키는 환경 파일에 따로 저장해줘.

URL: `https://내프로젝트.supabase.co` 키: `sb_publishable_...`

지금 만든 사이트를 Supabase랑 연결해줘. 문의 폼 내용이 저장되게. 키는 환경 파일에 따로 저장해줘.
URL: https://geiakphnhikiwqsgavp.supabase.co
키: sb_publishable_FGfL.....|

GPT-5.6

그러면 연결 코드를 짜주고, 냉장고 칸을 나누는 명령문(SQL)도 만들어 줍니다.

지금 만든 사이트를 Supabase랑 연결해줘. 문의 폼 내용이 저장되게. 키는 환경 파일에 따로 저장해줘.
URL: https://geiakphnhikiwqsgavp.supabase.co
키: sb_publishable_FGfL.....

- ✓ js/config.js 연결 정보 분리 (키는 여기만)
- ✓ js/script.js 폼 제출 → Supabase 저장으로 변경
- ✓ index.html config.js 불러오기 추가

연결 코드를 넣었습니다. 이제 Supabase 쪽에 문의를 담을 표(테이블)가 있어야 합니다. 아래 SQL을 Supabase의 **SQL Editor**에 붙여넣고 **Run**을 누르세요.

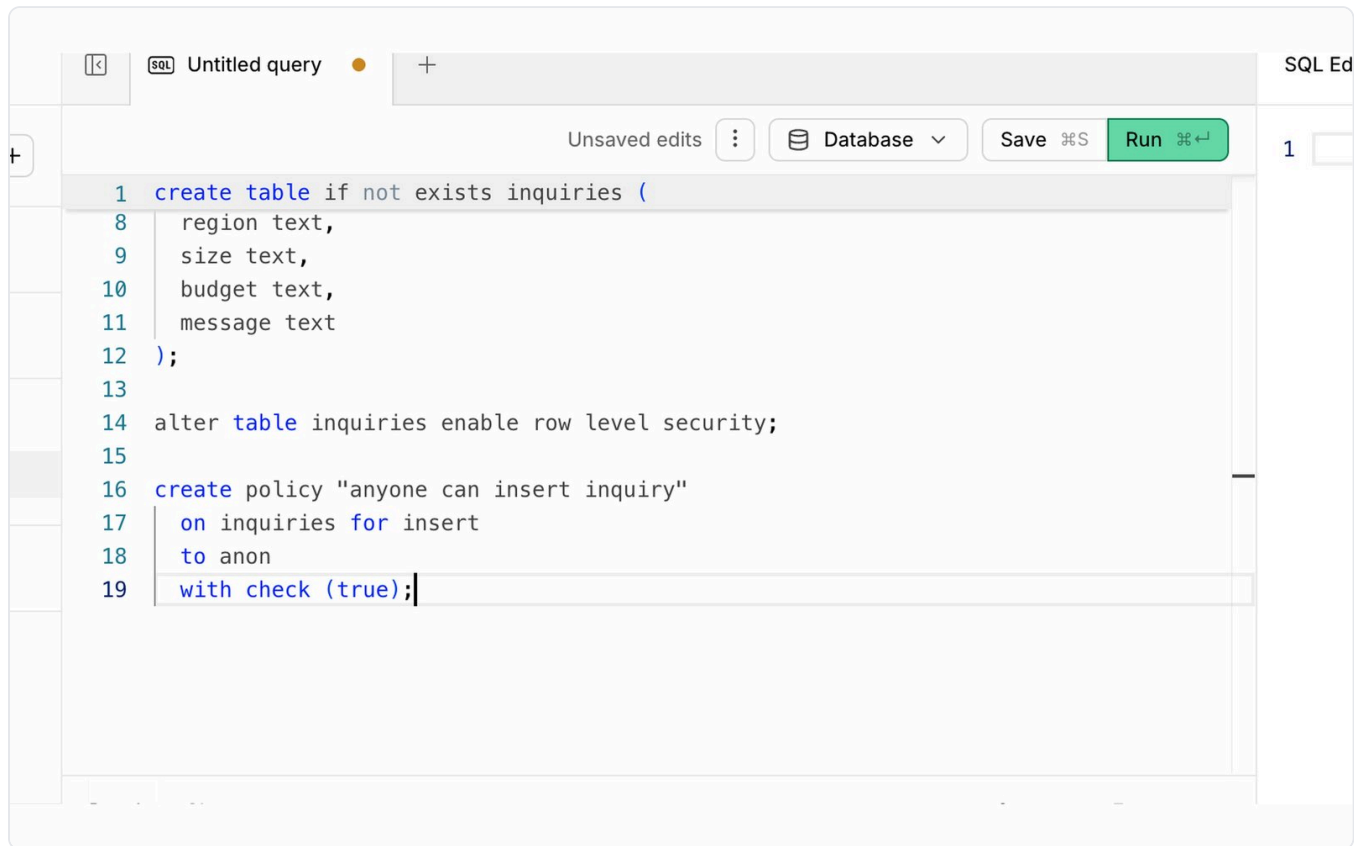
```
-- 문의 테이블 만들기
create table inquiries (
  id bigint generated always as identity primary key,
  created_at timestampz default now(),
  name text not null,
  phone text not null,
  email text, space_type text, region text,
  size text, budget text, message text
);

-- 누구나 문의는 넣을 수 있게 (읽기는 막음)
alter table inquiries enable row level security;
create policy "anyone can insert" on inquiries
  for insert to anon with check (true);
```

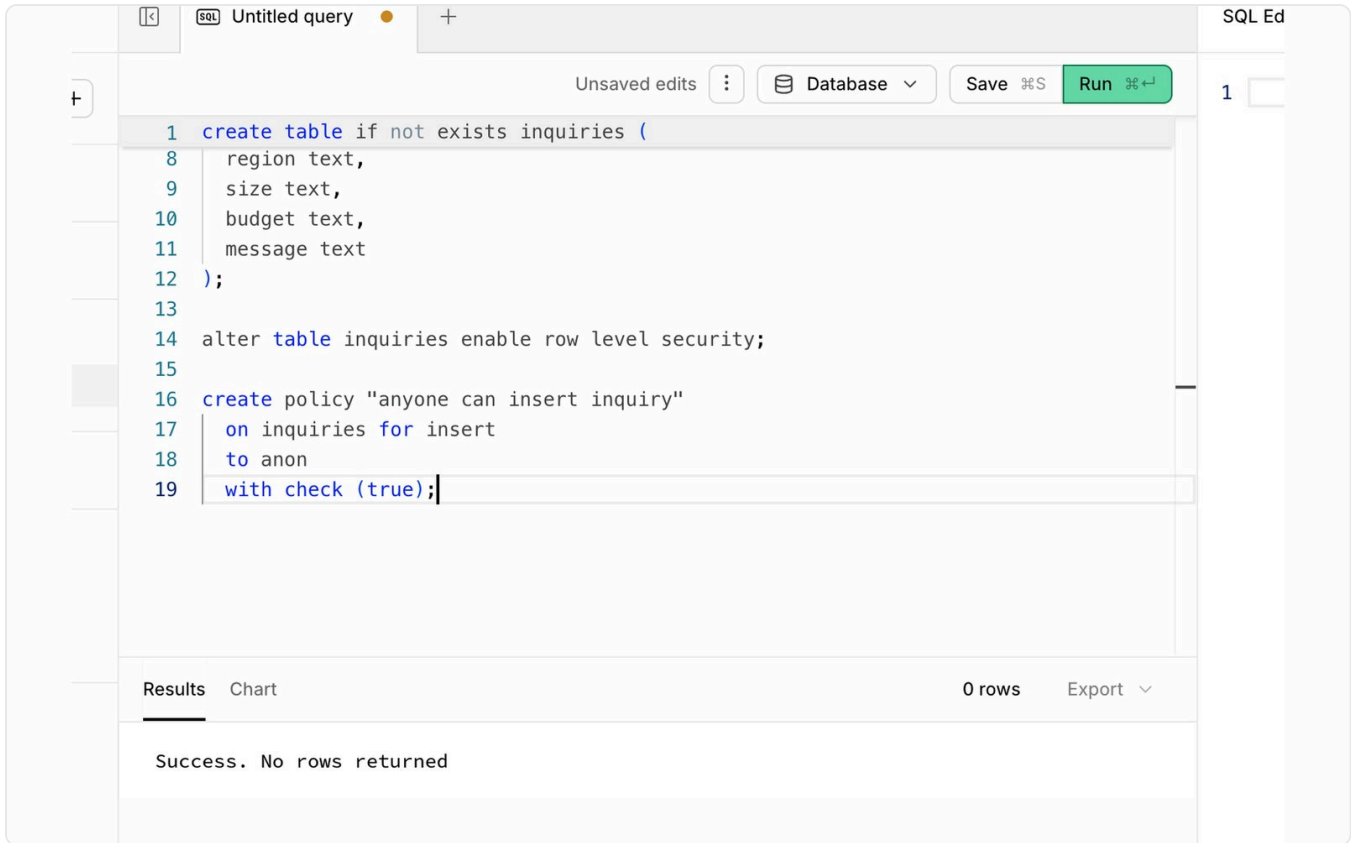
실행 후 사이트에서 문의를 보내면 Supabase의 Table Editor에 바로 나타납니다.

⑥ SQL 에디터에서 실행

Codex가 준 SQL을 복사해서 Supabase의 **SQL Editor**에 붙여넣고 **Run**을 누릅니다.



Success. No rows returned 가 나오면 성공입니다.



⑦ 진짜 저장되는지 확인

사이트에서 문의를 한 번 넣어 보고, Supabase의 **Table Editor**에서 그 줄이 쌓였는지 보세요. 여기까지 되면 주방과 냉장고가 붙은 겁니다.

8장 · 키와 보안 — 이 장이 제일 중요합니다

이 장만 제대로 읽으셔도 이 책 값은 합니다. 여기가 틀리면 문의 내용이 통째로 새거나, 남이 내 데이터베이스를 마음대로 쓸 수 있게 됩니다.

키는 두 종류입니다

Supabase는 키를 두 가지 줍니다. 이름이 비슷해서 헷갈리는데, 성격은 정반대입니다.

	공개 키 (Publishable)	비밀 키 (Secret)
생긴 모양	<code>sb_publishable_...</code>	<code>sb_secret_...</code>
어디서 쓰나	브라우저, 앱, 소스 코드	서버에서만
공개돼도 되나	됩니다	절대 안 됩니다
무엇을 볼 수 있나	RLS가 허용한 것만	전부

공개 키는 왜 공개돼도 되는가

공식 문서가 직접 이렇게 적어놨습니다.

Safe to expose online: web page, mobile or desktop app, GitHub actions, CLIs, source code.

온라인에 노출해도 안전합니다: 웹페이지, 모바일·데스크톱 앱, GitHub Actions, CLI, 소스 코드.

Anyone can read it, so it only reaches what Row Level Security allows. 누구나 읽을 수 있기 때문에, Row Level Security가 허용하는 데까지만 닿습니다.

즉 공개 키는 "누가 봐도 상관없게 설계된 키"입니다. 브라우저에 그대로 들어갑니다. 숨기려고 애쓸 필요가 없습니다.

비밀 키는 이 책에서 아예 안 씁니다

비밀 키는 모든 데이터에 그냥 접근되는 마스터 키입니다. 서버에서만 씁니다.

오늘 만드는 문의 폼에는 비밀 키가 필요 없습니다. 꺼내실 일이 없습니다.

절대 하지 말 것

⚠ 비밀 키를 AI 대화창에 붙여넣지 마세요.

공식 문서가 비밀 키에 대해 이렇게 못 박아 뒀습니다.

How they must never travel: over chat, email, or SMS. 절대 이렇게 주고받으면 안 됩니다: 채팅, 이메일, 문자.

Where they must never appear: a web page, a public document, source code, or a bundled mobile, desktop, or CLI package. A browser, even on localhost. 절대 나타나면 안 되는 곳: 웹페이지, 공개 문서, 소스 코드, 배포된 앱 패키지. 브라우저, localhost라도.

Codex 대화창은 **채팅**입니다. 여기에 비밀 키를 넣는 건 문서가 금지한 그 행동입니다.

나중에 비밀 키를 쓸 일이 생기면 이렇게 하세요.

1. AI에게 "환경 파일(`.env`) 만들어 줘, 키 자리는 비워둬"라고 시킵니다
2. 그 파일을 내가 직접 열어서 키를 붙여넣습니다
3. 그 파일이 GitHub에 안 올라가게 `.gitignore` 에 넣습니다 — 이걸 AI에게 시켜도 됩니다

키를 넣는 행위만 사람이 하고, 나머지는 전부 AI에게 시키면 됩니다.

그럼 공개 키가 공개돼 있는데 데이터는 뭐가 지키나

가게로 생각하면 쉽습니다. 손님이 들어오게 하려면 **가게 문은 열어둬야** 합니다. 대신 **돈은 금고에 넣고 잠급니다.**

공개 키가 열어둔 가게 문이고, 금고 자물쇠에 해당하는 게 **RLS(Row Level Security)**입니다. 누구에게 어느 줄까지 보여줄지를 데이터베이스가 직접 판단합니다.

RLS 꺼짐	RLS 켜짐
공개 키만 있으면 표 전체를 읽을 수 있음	규칙이 허용한 줄만
문의 내용이 아무한테나 열림	넣기만 되고 읽기는 막힘

중요한 함정

⚠ SQL 에디터로 표를 만들면 이 자물쇠가 기본으로 안 걸립니다.

문은 열어두고 금고는 안 잠근 셈입니다. 공식 문서도 표마다 직접 켜라고 되어 있습니다. Codex가 만들어 준 SQL에 RLS 문장이 들어 있는지 꼭 확인하세요.

```
alter table inquiries enable row level security;

create policy "anyone can insert inquiry"
  on inquiries for insert
  to anon
  with check (true);
```

위 두 문장이 하는 일은 이렇습니다.

- 첫 문장: 이 표에 잠금장치를 켜다
- 둘째 문장: 누구나 넣을 수는 있게 한다 (읽기는 허용하지 않았으므로 막힙니다)

문의 폼은 이 조합이면 충분합니다. 누구나 문의를 남길 수 있고, 남긴 내용은 남이 못 봅니다.

그대로 붙여넣으세요

방금 만든 테이블에 RLS 켜고, 익명 사용자는 insert만 되고 select는 안 되게 정책 만들어줘. SQL로 줘.

배포 전에 확인할 것

1. 소스 코드에 `sb_secret_` 으로 시작하는 문자열이 없는가
2. `.env` 파일이 `.gitignore` 에 들어 있는가
3. 내 테이블에 RLS가 켜져 있는가
4. 문의를 넣어보고, 브라우저에서 그 표를 읽으려 하면 막히는가

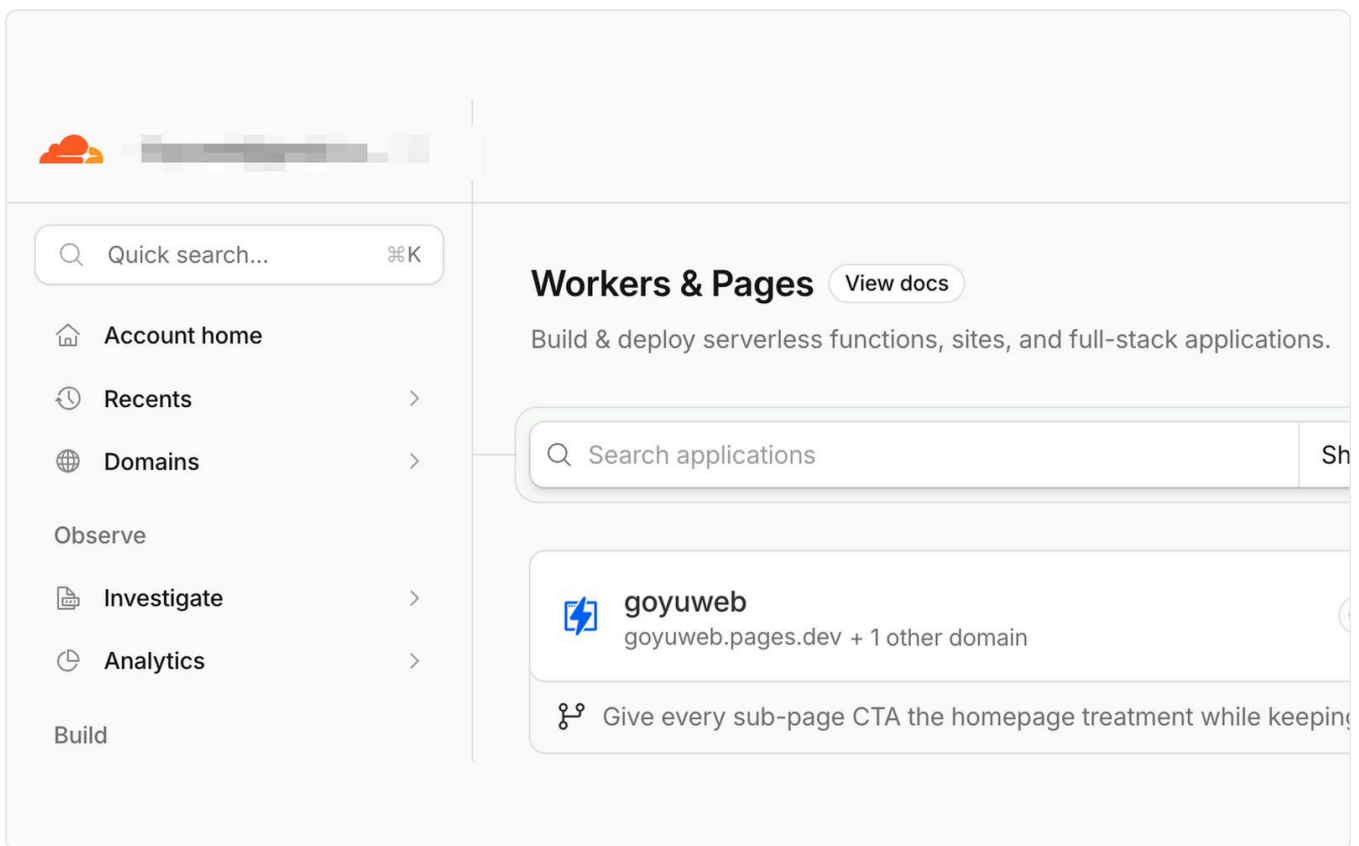
전체 체크리스트는 부록 B에 있습니다.

9장 · 공개하기

지금까지 만든 건 내 컴퓨터 안에서만 돌아갑니다. 유튜브로 치면 편집 다 하고 업로드를 안 한 상태입니다. 이제 업로드 버튼을 누릅니다.

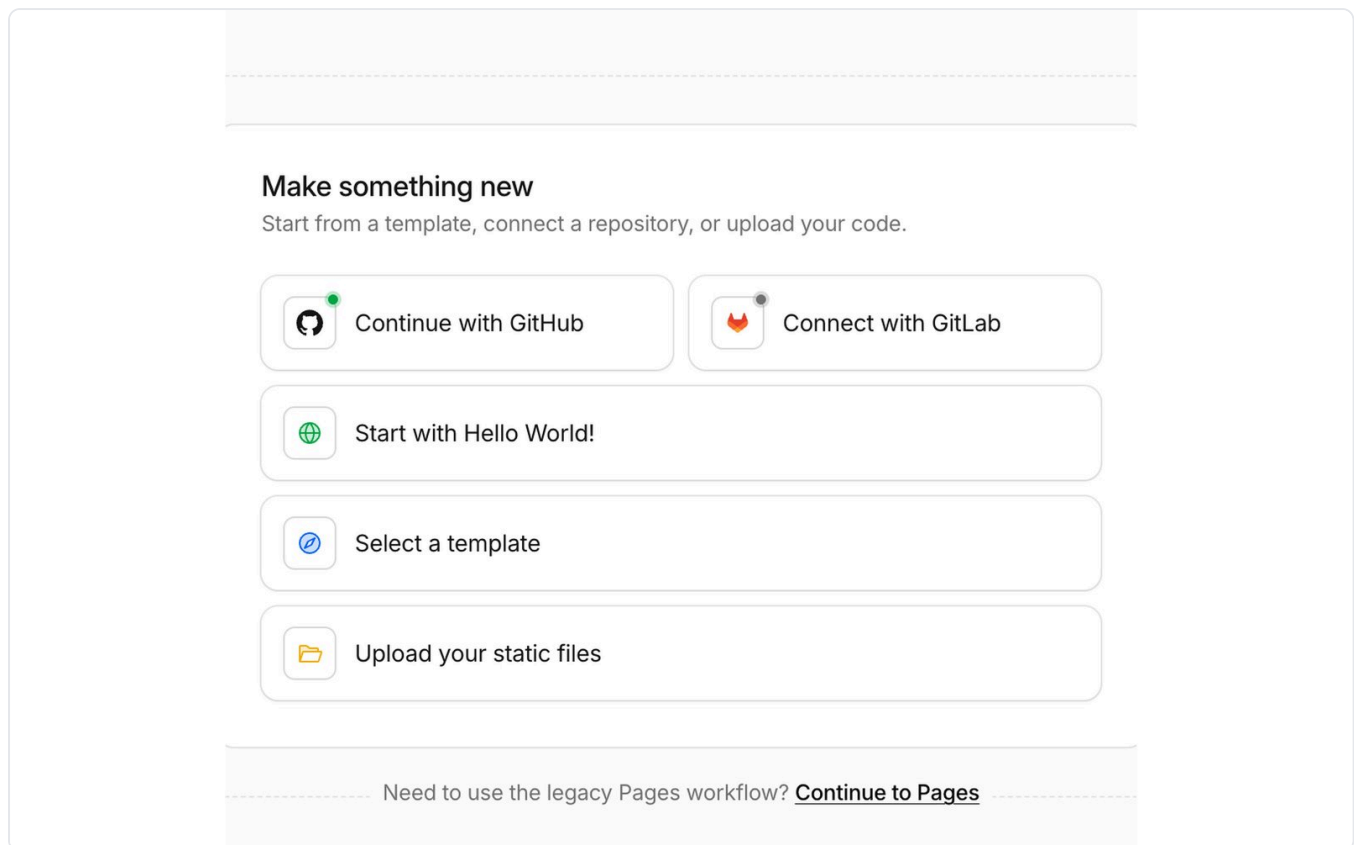
① Cloudflare 가입

cloudflare.com 에 가입하고 대시보드로 들어갑니다. 왼쪽에서 **Workers & Pages**를 찾습니다.



Create application을 누릅니다.

② GitHub 연결



Continue with GitHub를 누릅니다. 4장에서 만든 계정으로 연결하면 됩니다.

③ 저장소 고르기

내 저장소 목록이 나옵니다. 아까 만든 사이트 저장소를 고르고 **Next**를 누릅니다.

Need to use the legacy Pages workflow? [Continue to Pages](#)

Select a repository

Select a repository from your GitHub account to deploy.

goyuweb

Search repositories...

woorijip-interior

goyuweb

goyuweb-main

Clone a public repository via Git URL >

Back

Next →

④ 빌드 설정

빌드 설정 화면이 나옵니다. 뭘 넣을지 모르겠으면 Codex한테 물어보시면 됩니다.

그대로 붙여넣으세요

Cloudflare 빌드 설정 알려줘.

5장에서 "나중에 Cloudflare에 올릴 거야"라고 미리 말해줬으니 바로 알려줍니다.

이 사이트는 빌드 과정이 없는 정적 HTML이라 설정이 아주 간단합니다.

Cloudflare 배포 설정

- **Build command** · 비워두기
- **Deploy command** · `npx wrangler deploy` (기본값 그대로)

wrangler.toml

프로젝트에 이 파일을 추가했고 GitHub에도 올렸습니다.

```
name = "woorijip-interior"
compatibility_date = "2026-09-01"

[assets]
directory = "./"
```

Cloudflare에서 저장소만 고르고 **Deploy**를 누르면 됩니다.

순수 HTML로 만든 사이트라면 대개 이렇습니다.

항목	값
Build command	비워두기
Deploy command	<code>npx wrangler deploy</code> (기본값 그대로)

그대로 넣고 **Deploy**를 누릅니다.

goyuweb/woorijip-interior

Project name

woorijip-interior

Build command

Optional ⓘ

Deploy command ⓘ

npx wrangler deploy

☒ Builds

 for non-production branches

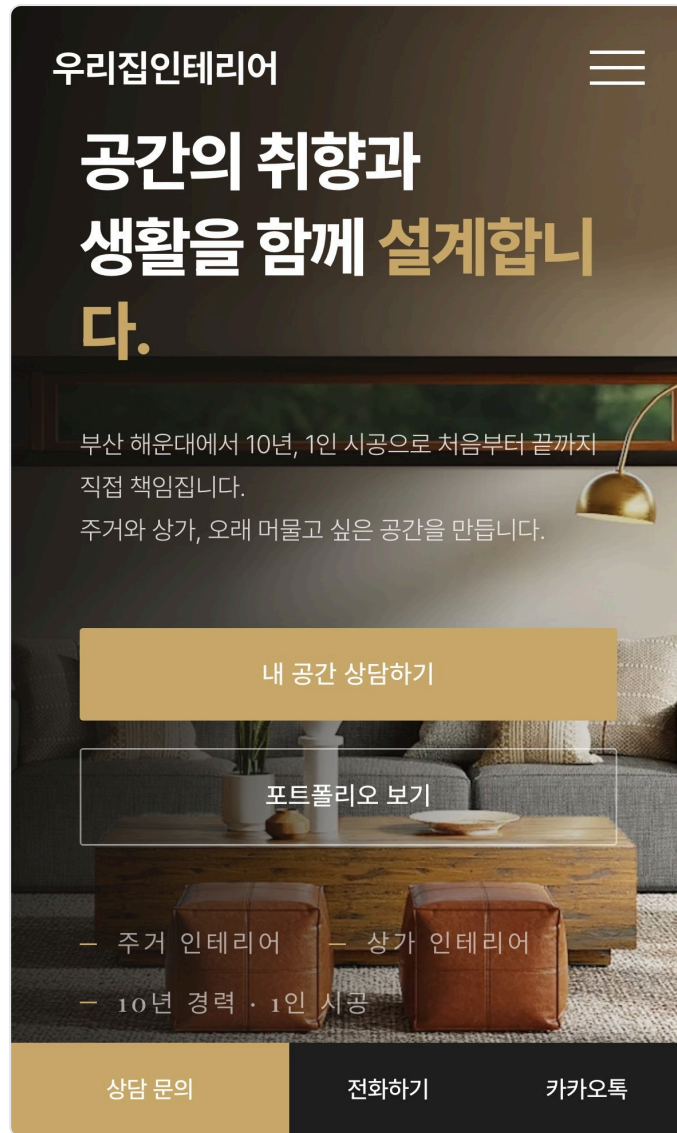
Back

Deploy

⑤ 주소가 생깁니다

잠깐 기다리면 프로젝트이름.내계정.workers.dev 주소가 생깁니다.

내 컴퓨터가 꺼져 있어도 전 세계 누구나 들어올 수 있습니다.



휴대폰으로도 한 번 열어보세요.

⑥ 앞으로는 자동입니다

앞으로 Codex로 수정해서 GitHub에 올릴 때마다 Cloudflare가 알아서 최신으로 바꿔줍니다. 배달 기사가 알아서 새 물건을 갖다 놓는 셈입니다.

그대로 붙여넣으세요

수정한 거 GitHub에 올려줘.

이 한 줄이면 됩니다.

10장 · 막혔을 때

원칙 하나

오류 메시지는 버전마다 바뀝니다. 그런데 **물어보는 법은 안 바뀝니다**. 이 장은 개별 해결책보다 **Codex에게 제대로 물어보는 법**을 다룹니다.

대부분의 사람이 이렇게 묻습니다.

✗ 안 돼요 ✗ 오류 났어요

이러면 Codex도 추측만 합니다. 이렇게 물으세요.

○ 무엇을 하려다가 / 무엇을 했더니 / 무엇이 나왔는지 세 가지를 같이

만능 문장

그대로 붙여넣으세요

[하려던 일]을 하려고 [한 행동]을 했는데 [나온 결과]가 나왔어. 화면에 뜬 메시지는 이거야: [메시지 그대로 복사] 원인이 뭔지 찾아서 고쳐줘. 고치기 전에 뭘 바꿀 건지 먼저 설명해줘.

마지막 줄이 중요합니다. **고치기 전에 설명하게 하면** 엉뚱한 파일을 건드리는 걸 미리 막을 수 있습니다.

상황별 문장

상황	이렇게 물으세요
명령어가 없다고 나옴	이 명령이 없다고 나오는데, 내 컴퓨터에 설치부터 해줘. 관리자 권한이 필요하면 내가 뭘 해야 하는지 알려줘.
승인 창이 안 뜸	브라우저 승인 창이 안 열려. 대신 터미널에서 인증하는 방법으로 바꿔줘.
폼을 보내도 표에 안 쌓임	폼을 제출했는데 Supabase 테이블에 줄이 안 생겨. 브라우저 개발자 도구 네트워크 탭에서 뭘 확인해야 하는지 알려주고, 코드도 점검해줘.
권한 오류가 뜸	Supabase에서 권한 관련 오류가 나. RLS 정책이 지금 어떻게 되어 있는지 확인하는 SQL이랑, 문의 폼에 맞는 정책 SQL을 줘.
배포는 됐는데 빈 화면	배포된 주소가 빈 화면이야. 브라우저 콘솔에 나오는 오류는 이거야: [복사]. 빌드 설정이나 경로 문제인지 봐줘.
수정했는데 사이트가 그대로	GitHub에 올렸는데 사이트가 안 바뀌어. 지금 커밋이 올라갔는지 확인하고, 배포가 걸렸는지 보는 방법 알려줘.
모바일에서 깨짐	휴대폰에서 [어느 부분]이 [어떻게] 깨져. 반응형으로 고쳐줘.
뭘 고쳤는지 모르겠음	방금 뭘 바꿨는지 파일별로 정리해서 알려줘.
되돌리고 싶음	방금 변경사항 되돌려줘. 마지막 커밋 상태로.
한도가 찼음	몇 시간 뒤에 다시 하세요. 5시간 단위로 다시 채워집니다.

되돌릴 수 있다는 걸 기억하세요

4장에서 GitHub를 연결해 둔 이유가 이겁니다. 망쳐도 마지막 저장 시점으로 돌아갈 수 있습니다. **과감하게 시도하세요** 됩니다.

그대로 붙여넣으세요

지금 상태 GitHub에 저장해줘. 메시지는 "문의 폼 연결 전"으로.

큰 작업 전에 이 한 줄을 습관으로 두세요.

부록 A · 복사해서 쓰는 요청 문장

내 정보로 바뀌서 그대로 붙여넣으시면 됩니다.

1. 사이트 만들기

[업체명]이라는 업체 홈페이지를 만들 거야. 업체 정보는 이래.
[지역], [업종], [연락처], [경력·규모].
이 정보에 맞게 [필요한 페이지들]로 홈페이지 만들어줘.
모바일에서도 잘 보이게. 나중에 Cloudflare에 올릴 거야.

2. GitHub 연결

GitHub 저장소 만들고 이 프로젝트 연결해줘. 필요한 거 있으면 설치도 해줘.

3. 지금 상태 저장

지금 상태 GitHub에 저장해줘. 메시지는 "[무엇을 하기 전]"으로.

4. Supabase 연결

지금 만든 사이트를 Supabase랑 연결해줘. 문의 폼 내용이 저장되게.
키는 환경 파일에 따로 저장해줘.
URL: [내 프로젝트 URL]
키: [공개 키]

5. RLS 켜기

방금 만든 테이블에 RLS 켜고, 익명 사용자는 insert만 되고 select는 안 되게
정책 만들어줘. SQL로 줘.

6. 빌드 설정 물어보기

Cloudflare 빌드 설정 알려줘.

7. 문구만 바꾸기

[어느 페이지]의 [어느 문구]를 "[새 문구]"로 바꿔줘. 다른 건 건드리지 마.

8. 색·분위기 바꾸기

전체 분위기를 [따뜻한 / 차분한 / 깔끔한]으로 바꿔줘. 색은 [색]을 기본으로.
글자 크기랑 여백도 같이 정리해줘.

9. 오류 물어보기

[하려던 일]을 하려고 [한 행동]을 했는데 [나온 결과]가 나왔어.
화면에 뜬 메시지는 이거야: [메시지 그대로]
원인이 뭔지 찾아서 고쳐줘. 고치기 전에 뭘 바꿀 건지 먼저 설명해줘.

10. 되돌리기

방금 변경사항 되돌려줘. 마지막 커밋 상태로.

부록 B · 공개 전 체크리스트

☐ 소스 코드에 `sb_secret_` 으로 시작하는 문자열이 없다 ☐ `.env` 파일이 `.gitignore` 에 들어 있다 ☐ 문의 테이블에 RLS가 켜져 있다 ☐ 문의를 넣어보면 Supabase 표에 줄이 쌓인다 ☐ 남이 그 표를 읽으려 하면 막힌다 ☐ 전화번호·주소·영업시간이 실제 값이다 ☐ 휴대폰에서 열어봤고 안 깨진다 ☐ 모든 버튼과 링크를 눌러봤다 ☐ 문의 폼 필수 항목이 실제로 필수로 걸려 있다 ☐ 개인정보 수집 동의 문구가 있다 ☐ 오타를 소리 내어 읽으며 한 번 확인했다 ☐ 마지막 상태를 GitHub에 저장해뒀다

부록 C · 용어 사전

용어	한 줄 설명
LLM	물어보면 답하는 AI 두뇌. ChatGPT·제미나이·클로드
코딩 에이전트	실제로 파일을 만들고 고치는 AI. Codex 등
Codex	OpenAI의 코딩 에이전트. ChatGPT 데스크톱 앱 안에 있음
프론트엔드	손님이 보는 화면. 식당의 홀
백엔드	화면 뒤에서 처리하는 곳. 식당의 주방
데이터베이스	데이터를 쌓아두는 곳. 냉장고
HTML	화면의 뼈대를 만드는 언어
CSS	색·크기·배치를 정하는 언어
자바스크립트	화면을 움직이게 하는 언어
프레임워크	자주 쓰는 걸 미리 묶어놓은 밀키트. 리액트·넥스트 등
Git	수정 기록을 저장하고 되돌리는 도구
GitHub	그 기록을 인터넷에 올려두는 곳
저장소(repository)	GitHub에서 프로젝트 하나가 들어가는 방
커밋	"여기까지 저장" 한 번
배포	만든 사이트를 인터넷에 올려 남이 볼 수 있게 하는 것
Supabase	백엔드와 데이터베이스를 통째로 빌려주는 서비스
Cloudflare	사이트를 인터넷에 올려주는 서비스. 무료 상업 이용 가능
API	프로그램끼리 주고받는 창구
공개 키	브라우저에 들어가도 되는 키

용어	한 줄 설명
비밀 키	서버에서만 쓰는 마스터 키. 채팅에 붙여넣지 말 것
RLS	누구에게 어느 줄까지 보여줄지 데이터베이스가 판단하는 잠금장치
SQL	데이터베이스에 시키는 명령어
localhost	내 컴퓨터 안에서만 열리는 주소

부록 D · 원문 링크

약관과 요금은 바뀝니다. 이 책의 판단은 집필 시점 기준입니다. **직접 확인해 보세요.**

무엇	어디
Vercel 공정 사용 지침 (Commercial usage)	vercel.com/docs/limits/fair-use-guidelines
Cloudflare 요금	cloudflare.com/plans
Cloudflare 이용약관	cloudflare.com/terms
Supabase API 키 (공개·비밀)	supabase.com/docs/guides/api/api-keys
Supabase RLS	supabase.com/docs/guides/database/postgres/row-level-security
ChatGPT 요금	openai.com/chatgpt/pricing

마치며

여기까지 오셨으면 **말만 하면 사이트가 만들어지는 상태**가 된 겁니다.

세 줄로 정리하면 이렇습니다.

- 도구는 **Codex**로 프론트엔드
- 뒤편은 **Supabase**
- 공개는 **Cloudflare**

무료고, 상업 이용 됩니다.

남은 것

이 책은 '만들어서 올리기'까지입니다. 사업에 실제로 쓰려면 남은 게 있습니다.

- 내 도메인 연결하기
- 문의가 오면 알림 받기
- 한 달에 실제로 얼마 드는지

무료 조합부터 안정 조합까지 표로 정리해서 **2편**에서 다룹니다.

직접 하기 버거우시면

혼자 만들고 혼자 운영하는 분들을 위해 이런 걸 계속 실험하고 공개합니다. 시간을 쓰는 대신 맡기고 싶으시면 고유웹으로 연락 주세요.

goyuweb.com