[우당탕탕 생성형AI 활용기] 초보 코딩, 챗지피티 유료 결제 첫날! 행사 업무분장표 웹앱 만들기 도전기(저장소 깃허브와 Vercel 연동 무료 배포)

프로그램

2026/10/10 10:00 http://blog.naver.com/talktalk2024/224436658879



안녕하세요!!!

처음으로 Vercel을 해 본 난장이입니다.

지난번 포스팅에서는 축제 업무지원 웹앱을 만들고 깃허브(GitHub)에 비공개로 싹~ 업로드하는 것까지 해봤죠?

[우당탕탕 생성형AI 활용기] 초보 코딩, 챗지피티 유료 결제 첫날! 행사 업무분장표 웹앱 만들기 도전기(쉼표 하나에 멈춘 에러, 쉿! 깃허브 비공개 저장소 만들기!)

안녕하세요!! 업무분장표 웹앱 만들기 늪에서 빠져나오고 있는 난장이입니다. 어제에 이어서 '업무분...

blog.naver.com

오늘은 드디어!

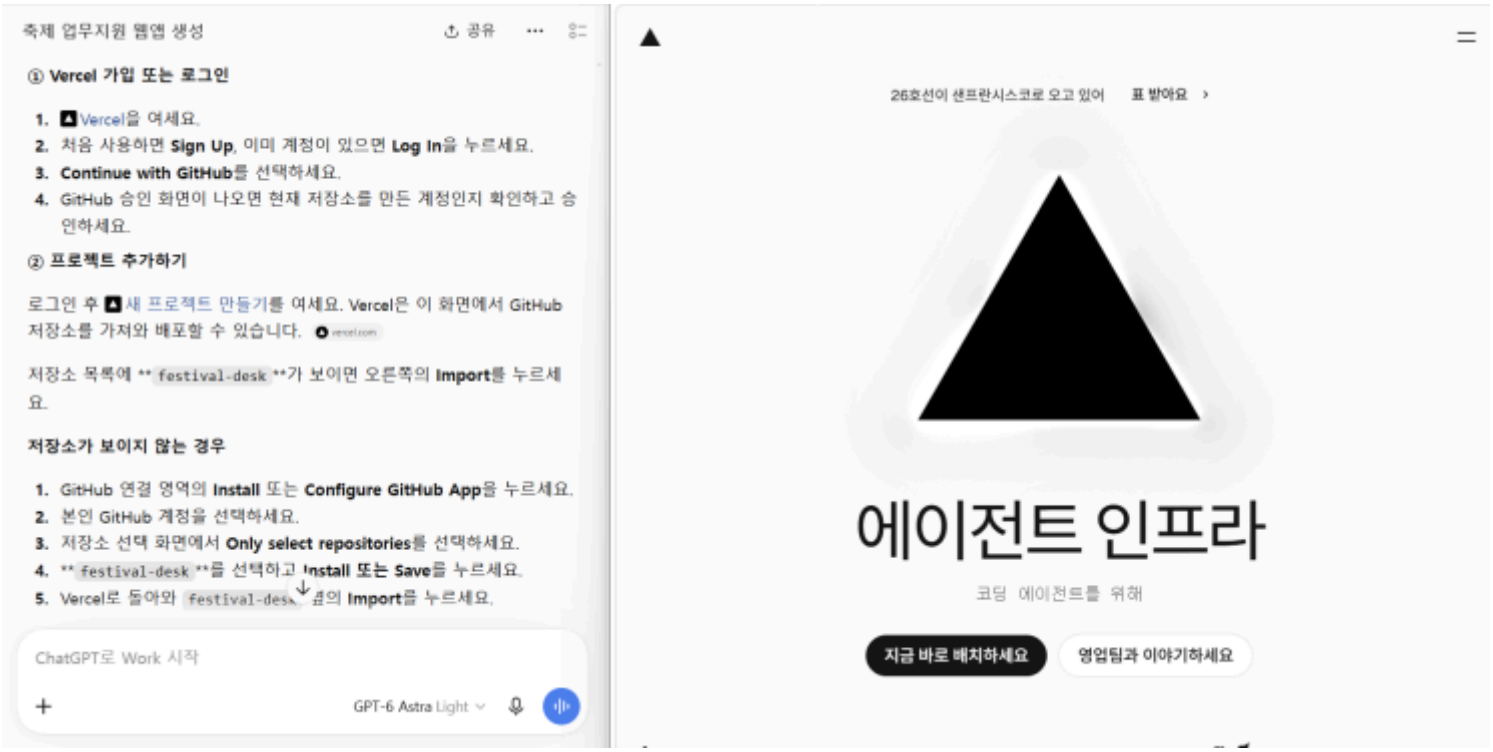
제 컴퓨터에만 있던 그 웹앱을 진짜 인터넷 세상에 띄우는 작업을 해볼 거예요.

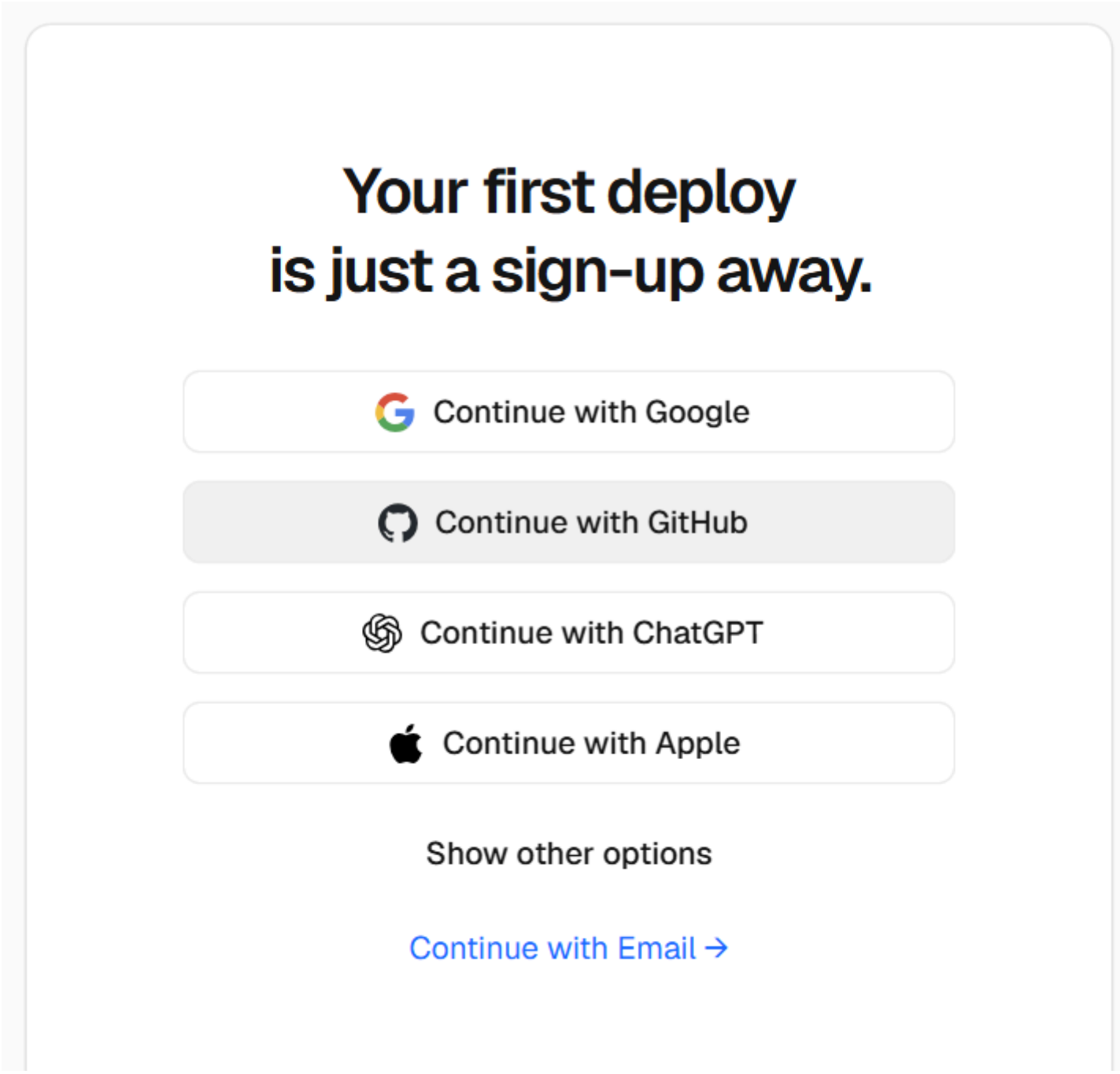
> ? 1초 용어 사전
> 배포(Deployment)란?
> 내 컴퓨터 방구석에만 꽁꽁 숨겨져 있던 가게를, 누구나 찾아올 수 있게 '진짜 인터넷 길거리'에 정식 오픈하는 것!

우선 사이트를 띄워주는 Vercel(버셀)이라는 곳에 회원가입을 했어요.

챗GPT가 친절하게 알려준 대로, 'Continue with GitHub'를 눌러서 제 깃허브 계정이랑 찰떡같이 연동해 주었답니다.

깃허브 승인 화면이 나오길래 계정을 확인하고 가볍게 승인 완료!


축제 업무지원 웹앱 생성
공유
① Vercel 가입 또는 로그인
1. Vercel을 여세요.
2. 처음 사용하면 Sign Up, 이미 계정이 있으면 Log In을 누르세요.
3. Continue with GitHub를 선택하세요.
4. GitHub 승인 화면이 나오면 현재 저장소를 만든 계정인지 확인하고 승인하세요.
② 프로젝트 추가하기
로그인 후 새 프로젝트 만들기를 여세요. Vercel은 이 화면에서 GitHub 저장소를 가져와 배포할 수 있습니다.
저장소 목록에 **festival-desk**가 보이면 오른쪽의 Import를 누르세요.
저장소가 보이지 않는 경우
1. GitHub 연결 영역의 Install 또는 Configure GitHub App을 누르세요.
2. 본인 GitHub 계정을 선택하세요.
3. 저장소 선택 화면에서 Only select repositories를 선택하세요.
4. **festival-desk**를 선택하고 Install 또는 Save를 누르세요.
5. Vercel로 돌아와 festival-desk 옆의 Import를 누르세요.
ChatGPT로 Work 시작
GPT-6 Astra Light
26호선이 샌프란시스코로 오고 있어
표 받아요
에이전트 인프라
코딩 에이전트를 위해
지금 바로 배치하세요
영업팀과 이야기하세요


Your first deploy
is just a sign-up away.
Continue with Google
Continue with GitHub
Continue with ChatGPT
Continue with Apple
Show other options
Continue with Email →



Vercel by Vercel

wants access to your GitHub account

Authorizing allows this app to

- ✓ Verify your GitHub identity
- ✓ Know which resources you can access
- ✓ Act on your behalf What does this mean?

Resources on your account

Email addresses (read)
View your email addresses

Vercel is not owned or operated by GitHub

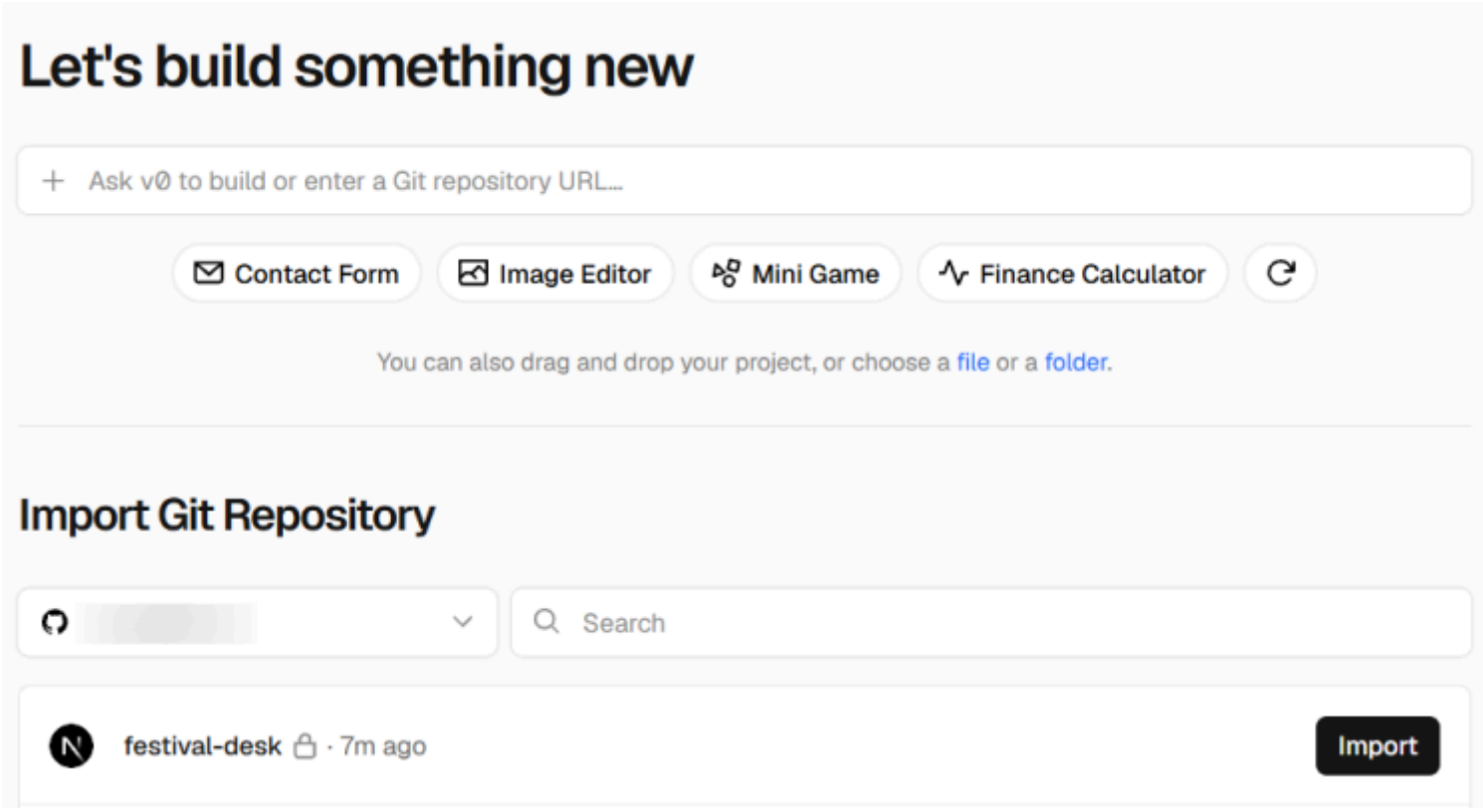
Created 9 years ago

More than 1K GitHub users

Authorize

로그인해서 '새 프로젝트 만들기'를 눌렀더니, 오! 지난번에 깃허브에 저장해둔 festival-desk 저장소가 바로 보이는 거 있죠?

반가운 마음에 오른쪽의 'Import' 버튼을 꾹 눌렀습니다.



---

* 만약 저장소가 보이지 않을 경우도 챗지피티가 알려줬어요!!

혹시나 보이지 않는 분들은 참고하시면 좋을 거 같아요.

② 프로젝트 추가하기

로그인 후 ▲ 새 프로젝트 만들기를 여세요. Vercel은 이 화면에서 GitHub 저장소를 가져와 배포할 수 있습니다. vercel.com

저장소 목록에 **`festival-desk`**가 보이면 오른쪽의 **Import**를 누르세요.

저장소가 보이지 않는 경우

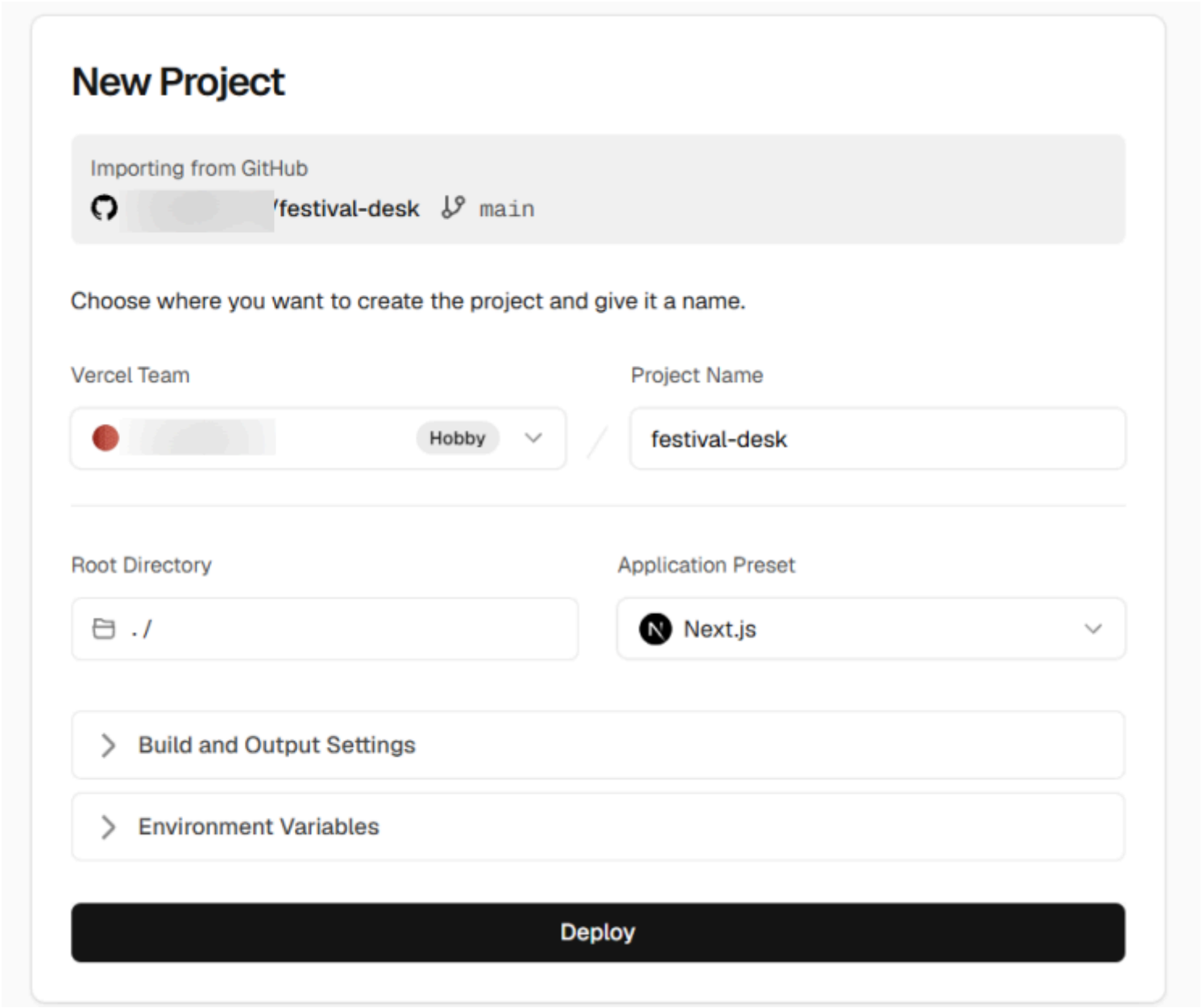
1. GitHub 연결 영역의 **Install** 또는 **Configure GitHub App**을 누르세요.
2. 본인 GitHub 계정을 선택하세요.
3. 저장소 선택 화면에서 **Only select repositories**를 선택하세요.
4. **`festival-desk`**를 선택하고 **Install 또는 Save**를 누르세요.
5. Vercel로 돌아와 `festival-desk` 옆의 **Import**를 누르세요.

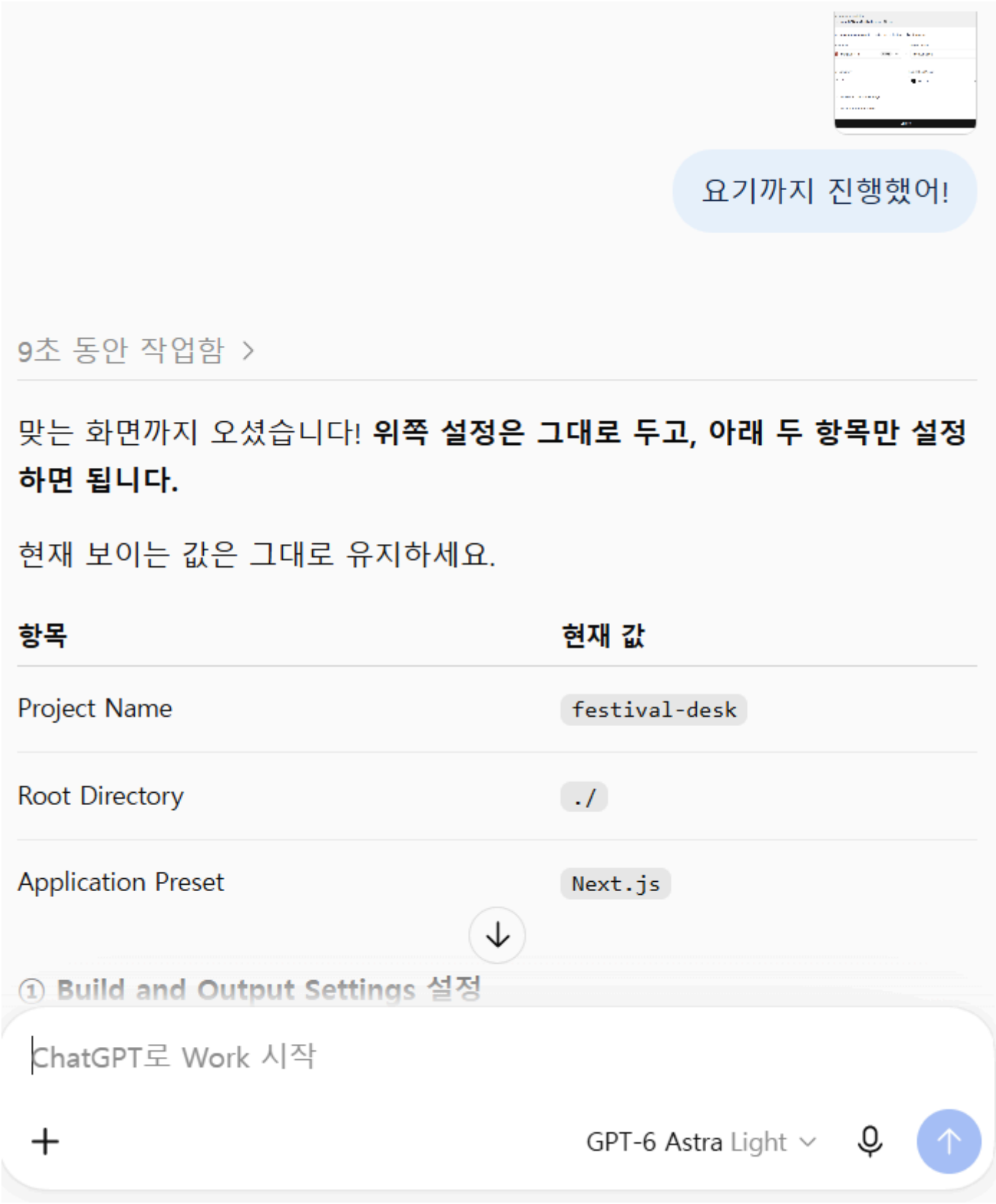
설정 화면이 떴는데 영어가 잔뜩 있어서 여기서 뭘 만져야 할지 잘 모르겠는거예요.

그래서 일단 정지!

설정 화면을 그대로 캡처해서 챗GPT에게 바로 물어봤습니다.

그랬더니 챗GPT가 화면과 현재 값을 싹 확인하더니 "아래 두 항목만 설정하면 됩니다"라고 콕 집어 알려주더라고요.


New Project
Importing from GitHub
/festival-desk
main
Choose where you want to create the project and give it a name.
Vercel Team
Hobby
Project Name
festival-desk
Root Directory
./
Application Preset
Next.js
Build and Output Settings
Environment Variables
Deploy



가장 헷갈렸던 건 Environment Variables (**환경변수**) 부분이었어요.

입력해야 할 내용이 무려 5개나 돼서 잘 맞춰서 넣어야 했거든요.

그럼 또 어떻게 한다?

캡쳐해서 챗지피티한테 또 물어봐야한다!! 아시죠??

? 1초 용어 사전

환경변수(Environment Variables)란?

우리 집 현관문 도어락 비밀번호!

남에게 절대 보여주면 안 되는 중요한 정보들만 따로 안전하게 보관하는 금고라고 생각하시면 돼요.



② **Environment Variables 설정**

아래쪽 **Environment Variables**를 펼치세요.

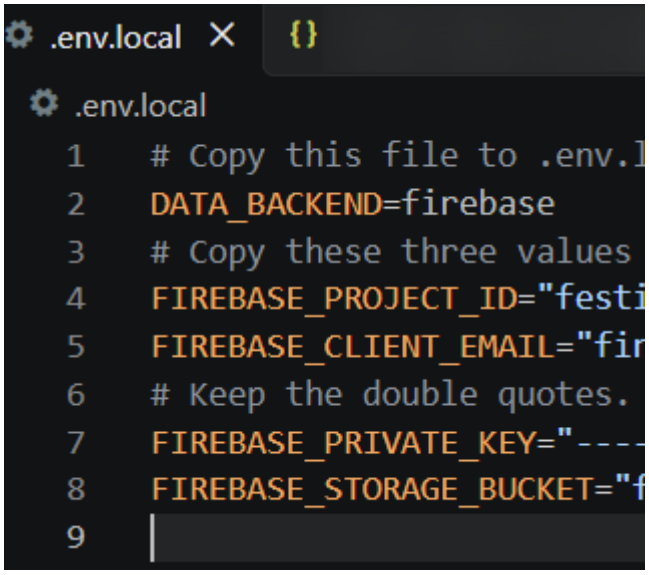
`Key` 또는 `Name`에는 설정 이름을, `Value`에는 해당 값을 입력합니다. VS Code의 `.env.local`을 보면서 **다음 5개를 한 줄씩 추가**하세요.

Key / Name	Value
`DATA_BACKEND`	`firebase`
`FIREBASE_PROJECT_ID`	현재 `.env.local`의 프로젝트 ID
`FIREBASE_CLIENT_EMAIL`	현재 `.env.local`의 서비스 계정 이메일
`FIREBASE_PRIVATE_KEY`	현재 `.env.local`의 비공개 키 전체
`FIREBASE_STORAGE_BUCKET`	현재 `.env.local`의 버킷 이름

↓

예를 들어 `.env.local`에 다음과 같이 적혀 있다면:

VS Code를 켜고 .env.local 파일을 열어보니 진짜 입력해야 할 키값들이 보였어요.



그런데 문득, "Value에 있는 큰따옴표("")도 같이 복사해서 넣어야 하나?" 고민이 되더라고요.

챗GPT에게 물어보니 "5개 항목 모두 바깥쪽 큰따옴표("")는 빼고 입력하세요!"라고 명쾌하게 대답해 줘서 그대로 복사+붙여넣기를 마쳤습니다.

그럼 Value에 넣을 때 "" 이것도 넣어? 아니면 빼고 입력해?

Vercel의 Value에는 바깥쪽 큰따옴표 `"` 를 빼고 입력하세요. 5개 항목 모두 동일합니다.

예를 들어 `.env.local` 이 이렇게 되어 있다면:

dotenv

```
FIREBASE_PRIVATE_KEY="-----BEGIN PRIVATE KEY-----\n키내용\n-----END PRIVATE KEY-----\n"
```

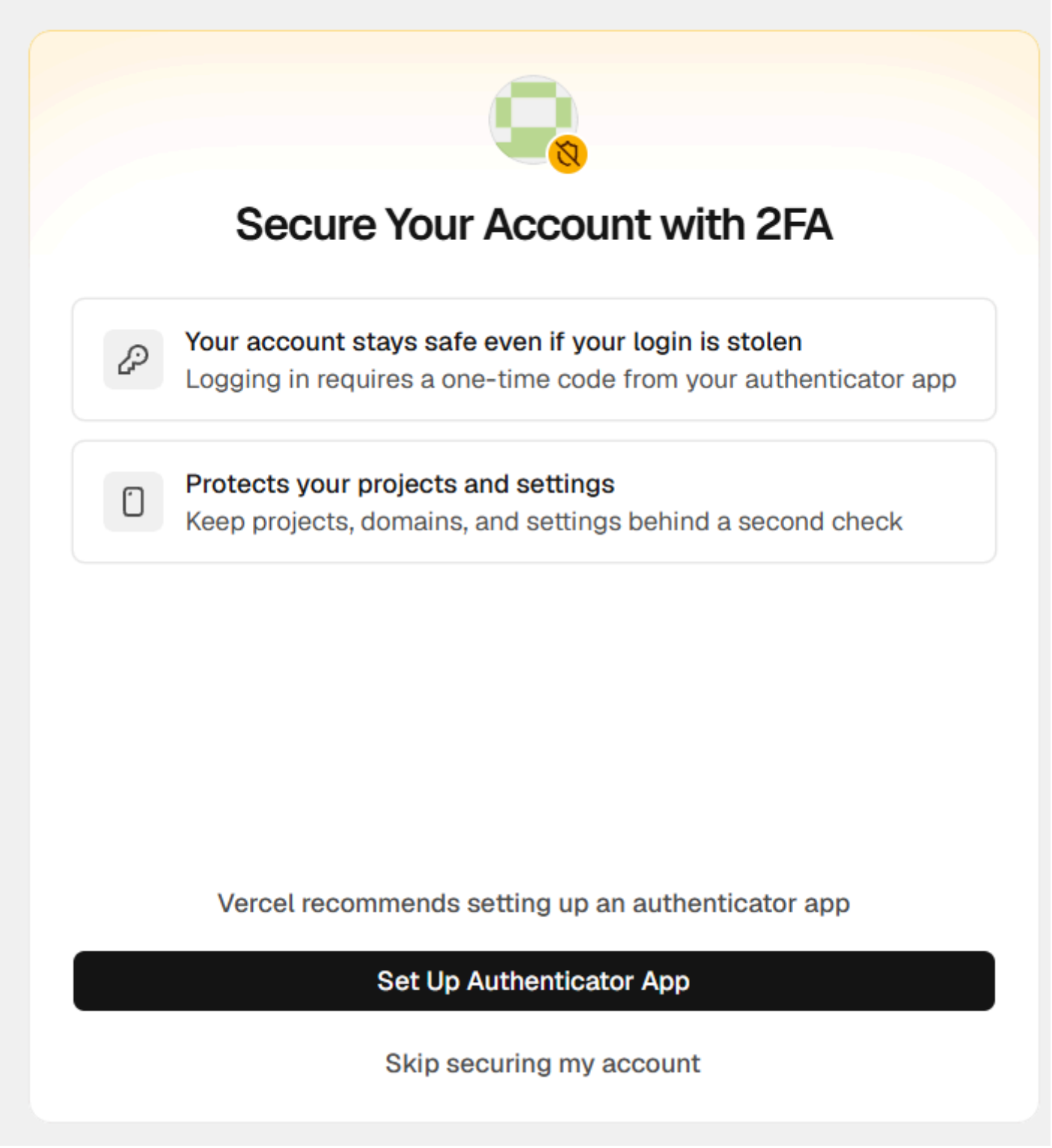
Vercel의 **Value**에는 아래 부분만 넣으세요.

일반 텍스트

```
-----BEGIN PRIVATE KEY-----\n키내용\n-----END PRIVATE KEY-----\n
```

큰따옴표와 끝의 쉼표는 제외하고, `\n` 은 그대로 유지하면 됩니다.

그다음 2단계 인증앱 설치 페이지가 나왔는데, 필수가 아니라 권장 사항이라길래 쿨하게 넘겼습니다.





이렇게 떴는데 인증앱을 설치 해야해?

8초 동안 작업함 >

지금 꼭 설치해야 하는 것은 아닙니다. 화면은 Vercel 계정의 **2단계 인증(2FA)을 권장하는 안내**입니다.

우선 배포를 계속하려면 화면 맨 아래의 **`Skip securing my account`**를 누르세요.

2단계 인증을 설정하려면 검은색 **`Set Up Authenticator App`** 버튼을 선택하면 됩니다.

이 설정은 **본인의 Vercel 관리 계정용**입니다. 나중에 축제 사이트에 접속하는 업무지원자들이 인증앱을 설치해야 한다는 뜻은 아닙니다.

이제 끝났다 싶어서 배포를 눌렀는데...

아뿔싸! ? **배포 실패 오류**가 떴습니다.

```
5  Vercel CLI 62.0.0
0  Warning: Detected "engines": { "node": ">=22.13.0" } in your `package.j
0  Detected `pnpm-lock.yaml` version 9 generated by pnpm@11.x from package
```

```
22:00:12.479  vite.config.ts(4,27): error TS2307: Cannot find module './
22:00:12.554  Failed to type check.
22:00:12.554
22:00:12.628  [ELIFECYCLE] Command failed with exit code 1.
```

당황하지 않고 오류 난 부분을 캡처해서 보여줬더니, 챗GPT가 **'.openai' 폴더가 누락되었다**고 귀신같이 찾아냈어요.



확인했습니다. **배포를 막은 것은 빨간색 오류입니다.**

- **노란색:** Node.js 버전 설정에 관한 경고입니다. 현재 보이는 배포 실패 원인은 아닙니다.
- **빨간색:** `vite.config.ts`의 4번째 줄에서 필요한 파일을 찾지 못했다는 오류입니다.

보내주신 화면은 오른쪽이 잘려 있지만, 제가 제공한 코드의 해당 줄은 다음과 같습니다.

TypeScript

```typescript
import hostingConfig from "./.openai/hosting.json";
```

따라서 **`.openai/hosting.json`이 GitHub 업로드에서 빠졌을 가능성이 높습니다.** 앞서 올려주신 저장소 화면에도 `.openai` 폴더가 보이지 않았습니다.

우선 이 파일부터 확인해 주세요.

1. GitHub의 `festival-desk` 저장소 첫 화면을 여세요.
2. `.openai` 폴더가 있는지 찾으세요.
3. 있다면 폴더 안에 `hosting.json`이 있는지 확인하세요.

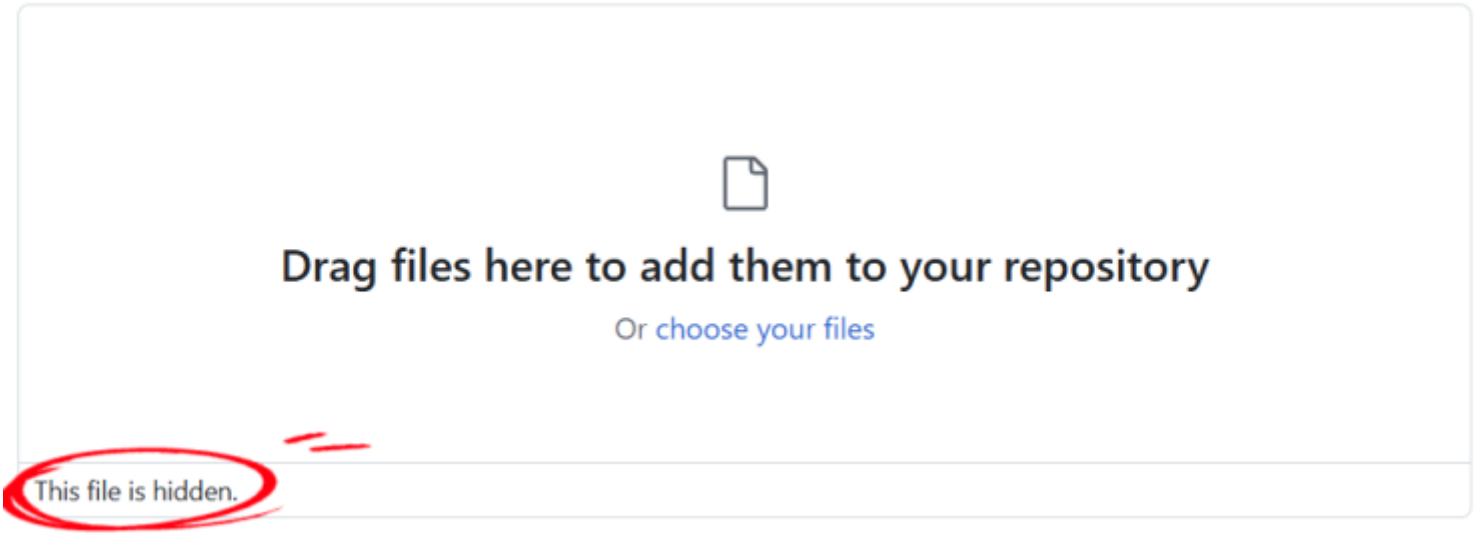
폴더가 없다면 이렇게 추가하세요.

1. 내 컴퓨터에서 `package.json`이 있는 `festival-desk` 폴더를 여세요.
2. `.openai` 폴더를 찾으세요. 안에는 `hosting.json`이 있습니다.
3. GitHub 저장소 첫 화면에서 **Add file → Upload files**를 누르세요.
4. **`.openai` 폴더 자체를** 끌어다 놓으세요.
5. 업로드 목록에 ** `.openai/hosting.json` **으로 표시되는지 확인하세요.
6. **Commit changes**를 누르세요.

이 파일은 제가 제공한 프로젝트의 호스팅 설정 파일로, **Firebase 비공개 키 파일과는 다릅니다.** `.env.local`이나 서비스 계정 JSON은 올리지 마세요.

저장하면 연결된 Vercel에서 새 배포가 시작되는지 확인해 주세요. **이미 `.openai/hosting.json` 이 있다면,** 파일을 바꾸지 말고 빨간 오류 줄의 오른쪽 끝까지 보이도록 캡처하거나 그 줄 전체를 복사해 보내주세요.

다시 깃허브에 올리려고 하니 이건 '숨김 파일'이라서 폴더째로 쓱 업로드가 안 되더라고요 ㅠㅠ



다시 해결 방법을 물어보니, 우리의 챗GPT!

아예 파일을 직접 만드는 법을 떠먹여 줬습니다.

깃허브 저장소 첫 화면에서 'Add file -> Create new file'을 누르고, 위쪽 파일 이름 란에 .openai/hosting.json을 직접

입력했어요.

그리고 VS Code에서 해당 파일 내용을 Ctrl+A로 전체 선택, Ctrl+C로 복사한 뒤! 깃허브의 본문 빈칸에 Ctrl+V로 붙여 넣었죠.

마지막으로 Commit changes를 누르니 저장소에 폴더와 파일이 뿅! 하고 제대로 생겼습니다.

네. 아래의 **`This file is hidden.`**은 숨김 파일이라는 뜻입니다. `.openai`처럼 점으로 시작하는 폴더를 업로드하는 과정에서 막힌 것으로 보입니다.

폴더를 끌어다 놓는 대신 GitHub에서 파일을 직접 만들면 됩니다. 제가 드래그로 안내해서 막히게 했네요.

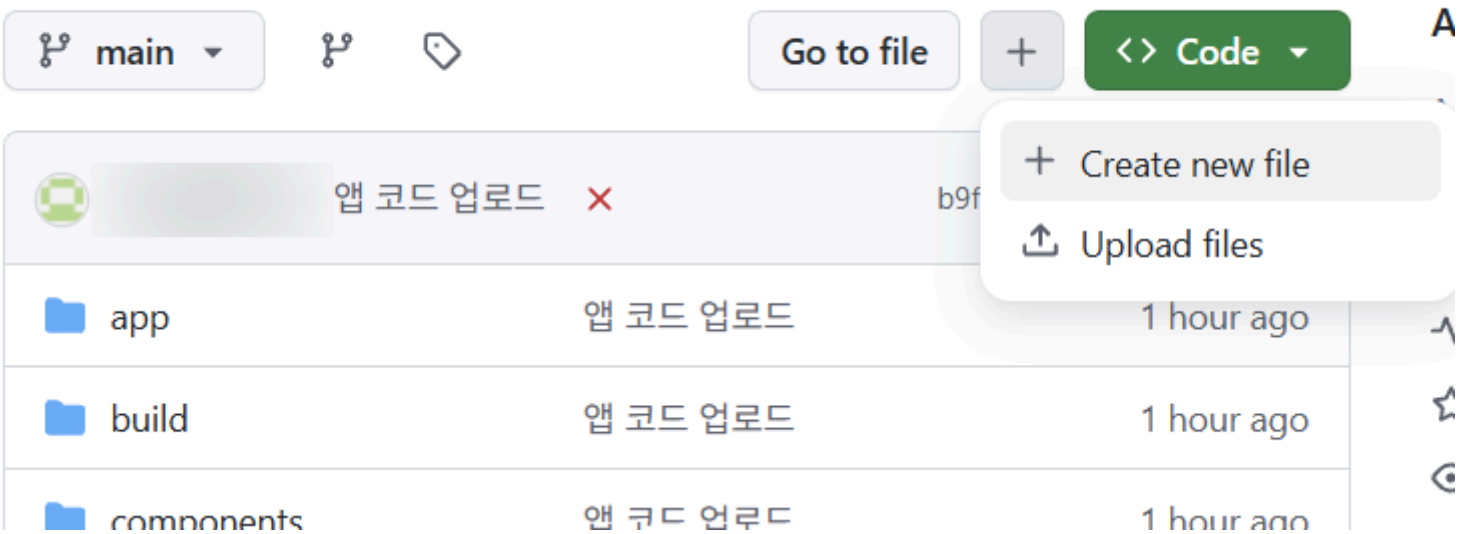
1. GitHub의 **`festival-desk` 저장소 첫 화면**으로 돌아가세요.
2. **Add file → Create new file**을 누르세요.
3. 위쪽 **Name your file...** 입력란에 다음을 정확히 입력하세요.

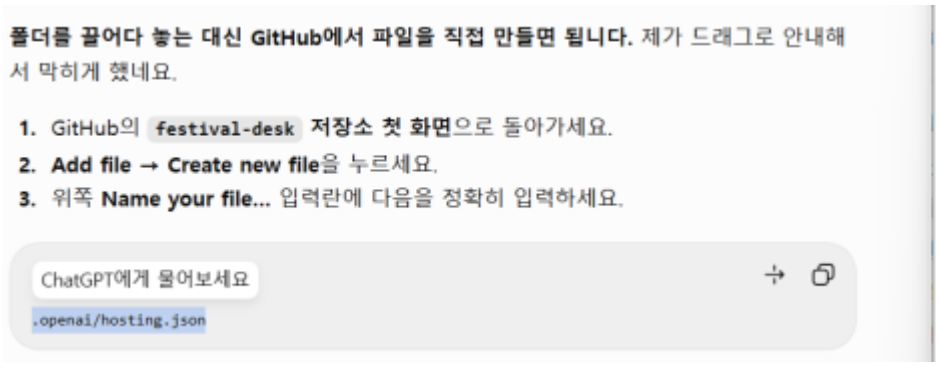
ChatGPT에게 물어보세요

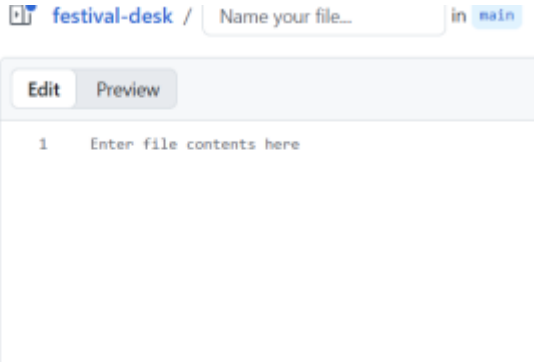
```
.openai/hosting.json
```

4. VS Code에서 **`.openai` 폴더 → `hosting.json`**을 여세요.
5. 파일 안에서 **`Ctrl + A` → `Ctrl + C`**로 내용을 복사하세요.
6. GitHub의 큰 본문 입력란에 **`Ctrl + V`**로 붙여넣으세요.
7. **Commit changes...**를 누르고, 확인 창이 뜨면 다시 **Commit changes**를 누르세요.

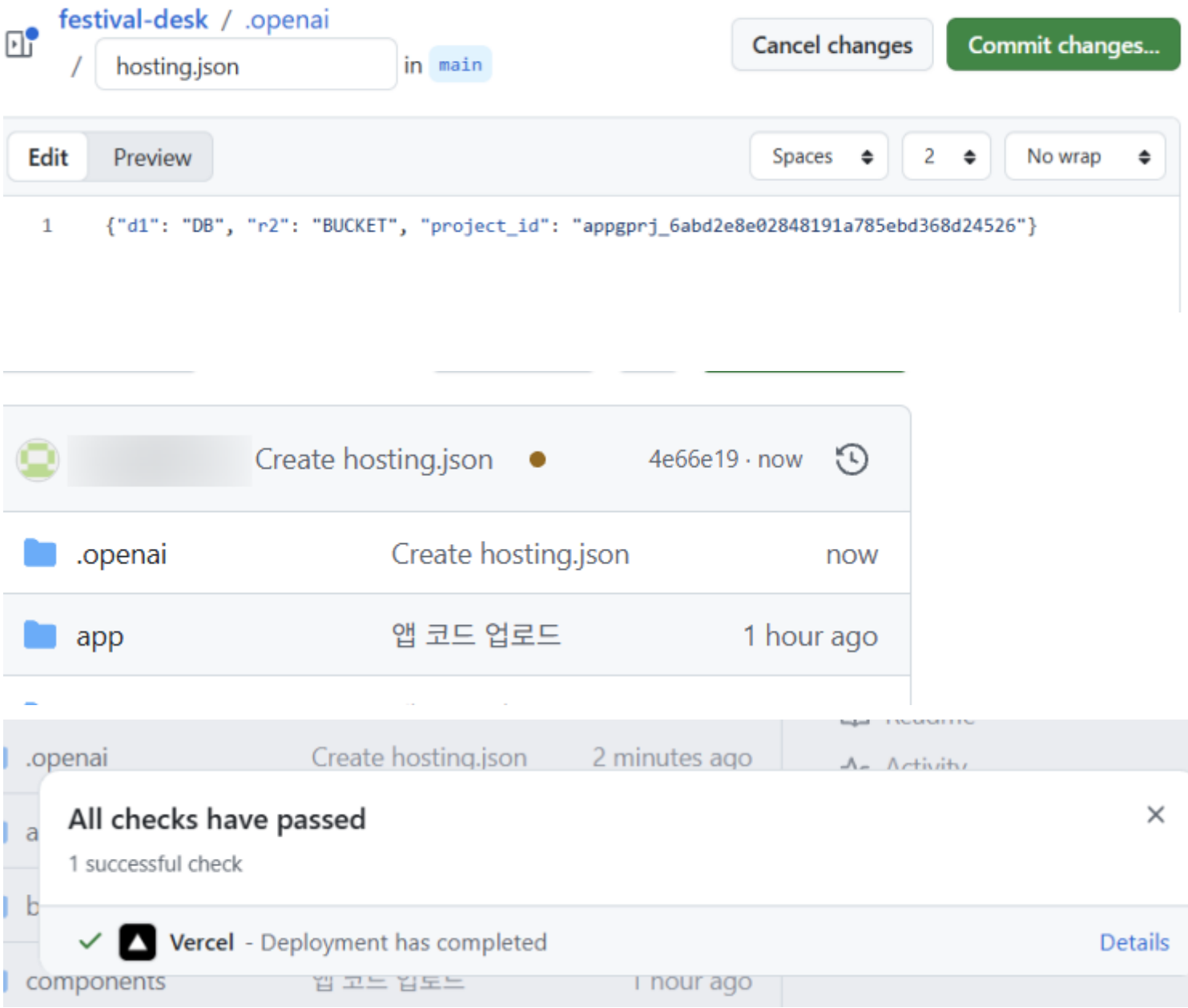
완료 후 저장소에 **`.openai` 폴더가 생기고 그 안에 `hosting.json`이 보이면 성공**입니다.
이어서 Vercel에서 새 배포가 시작되는지 확인하면 됩니다.


main
Go to file
Code
Create new file
Upload files
앱 코드 업로드
app
앱 코드 업로드
1 hour ago
build
앱 코드 업로드
1 hour ago
components
1 hour ago


폴더를 끌어다 놓는 대신 GitHub에서 파일을 직접 만들면 됩니다. 제가 드래그로 안내해서 막히게 했네요.
1. GitHub의 festival-desk 저장소 첫 화면으로 돌아가세요.
2. Add file → Create new file을 누르세요.
3. 위쪽 Name your file... 입력란에 다음을 정확히 입력하세요.
ChatGPT에게 물어보세요
.openai/hosting.json


festival-desk / Name your file... in main
Edit
Preview
1 Enter file contents here


festiv...
.next
.openai
hosting.json
.sites-runtime
.vscode
.wrangler
app
.openai > hosting.json > ...
1 {"d1": "DB", "r2": "BUCKET", "project_id": "appgr



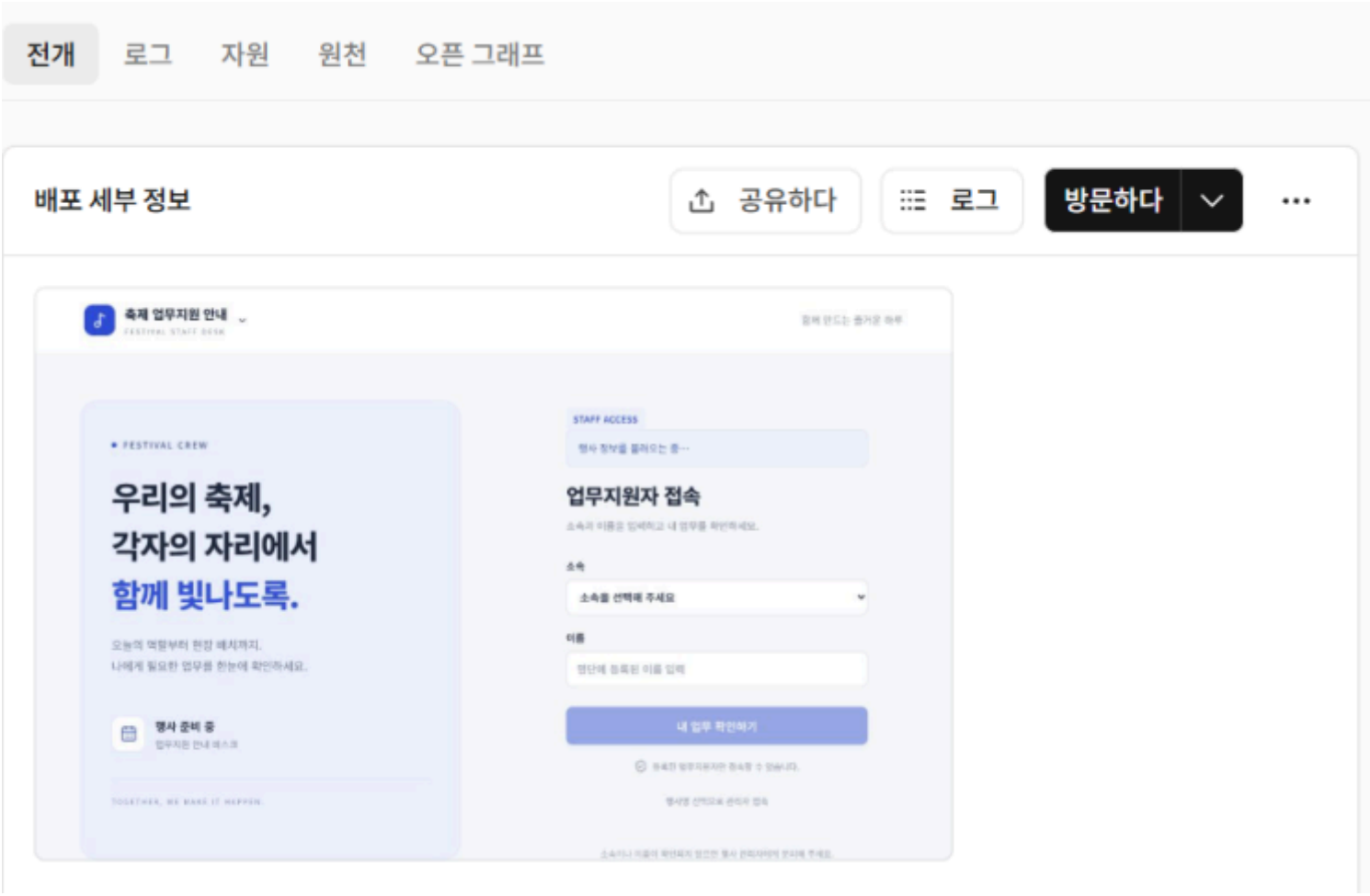
? 1초 용어 사전

커밋(Commit)이란?

게임 하다가 누르는 '세이브(저장)' 버튼!

"나 여기까지 작업 완료했으니까 확실하게 덮어쓰기 저장해 줘!" 하고 쾅 도장을 찍는 행동이에요.

오류를 싹 다 해결하고 완료된 창을 캡처해서 보냈더니...



챗GPT가 "배포가 완료됐습니다! 화면의 All checks have passed 와 Vercel - Deployment has completed 가 성공 표시입니다" 라며 축하해 주었어요!

그 문장을 보는데 너무 신났었습니다. ?

Deployment

festival-desk· [redacted] vercel.app

Domains ⊕

festival [redacted]

Status Created

● Ready Oct 6 by [redacted]

Source

main

c435252 총괄소통 외 업데이트(10.6.2.)

17초 동안 작업함 >

배포가 완료됐습니다! 화면의 `All checks have passed`와 `Vercel - Deployment has completed`가 성공 표시입니다.

이제 실제 접속 주소를 확인해 볼게요.

1. 현재 창 오른쪽의 **Details**를 누르세요.
2. Vercel 배포 화면이 열리면 **Visit** 버튼을 누르세요.
3. 축제 로그인 화면이 열리는지 확인하세요.
4. **기존 관리자 계정으로 로그인**해서 명단과 자료가 보이는지 확인하세요. 같은 Firebase를 연결했다면 기존 자료가 보여야 합니다.

그다음 **업무지원자도 접속할 수 있는지** 확인하세요.

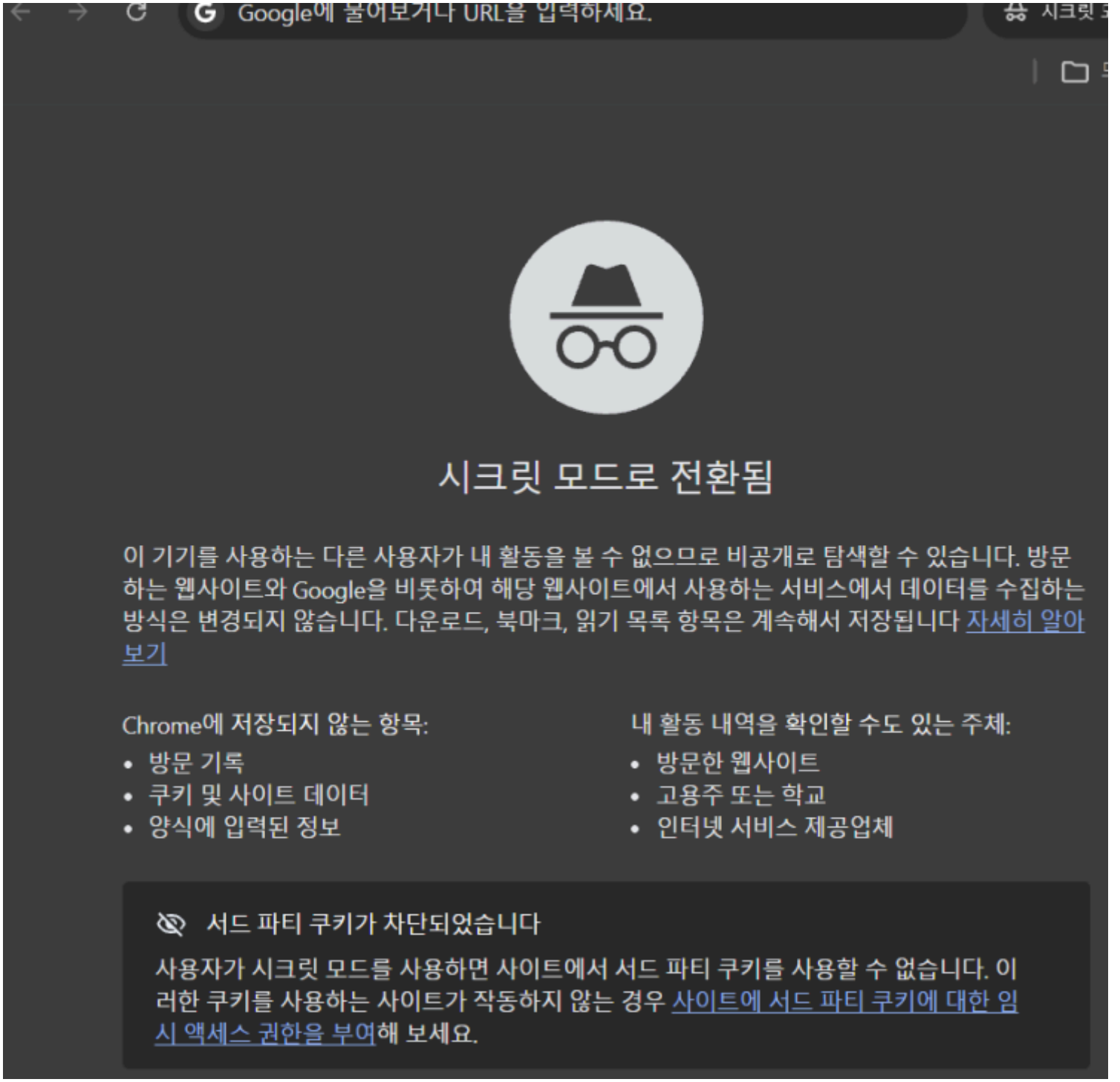
1. Vercel 프로젝트의 **Domains**에 표시된 운영 주소를 복사하세요.
2. 브라우저에서 **`Ctrl + Shift + N`**으로 시크릿 창을 여세요.
3. 복사한 주소를 붙여넣으세요.
4. Vercel 로그인 요구 없이 **축제 업무지원자 로그인 화면**이 나오면 접속 공개 여부는 확인된 것입니다.

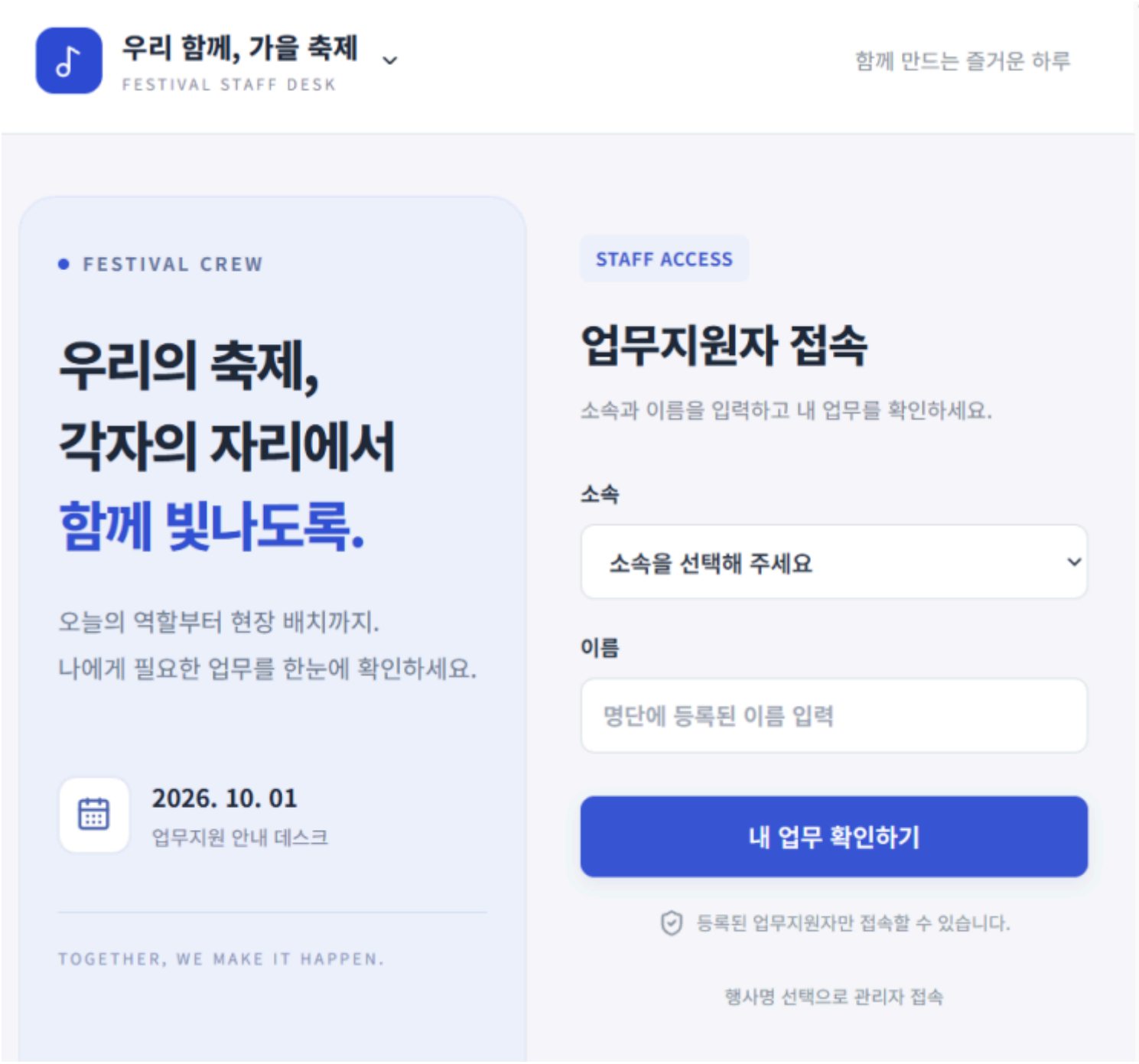
우선 `Details`를 누른 다음 화면을 보여주 ↓ . 여러 주소가 보이면 어떤 주소를 지원자에게 전달하면 되는지 짚어드리겠습니다.

마지막으로 실제 접속 주소가 잘 작동하는지 확인하는 작업을 거쳤어요.

챗GPT가 알려준 대로 브라우저에서 **'시크릿 창'을 열고, Vercel 프로젝트 Domains에 표시된 운영 주소를 복사해서 붙여 넣어 봤습니다.**

배포 상태도 Ready, 운영 환경도 Production으로 정상 표시되고, 로그인 화면이 문제없이 촥 열리는 걸 보니 가슴이 다 웅장해졌습니다.





? 1초 용어 사전

시크릿 창(Incognito Window)

내 로그인 기록이나 방문 기록을 전혀 모르는 '완전 쌩판 남의 컴퓨터' 상태로 만들어주는 마법의 창이에요!

내가 만든 사이트가 다른 사람 폰에서도 잘 열리는지 테스트할 때 최고의 꿀팁이죠.

오늘 이렇게 좌충우돌 에러도 겪어가며 무사히 웹앱 배포에 성공했습니다!

왕초보도 챗GPT가 시키는 대로만 따라 하니까 진짜 인터넷에 내 프로그램이 뜨는 게 너무 신기하지 않나요?

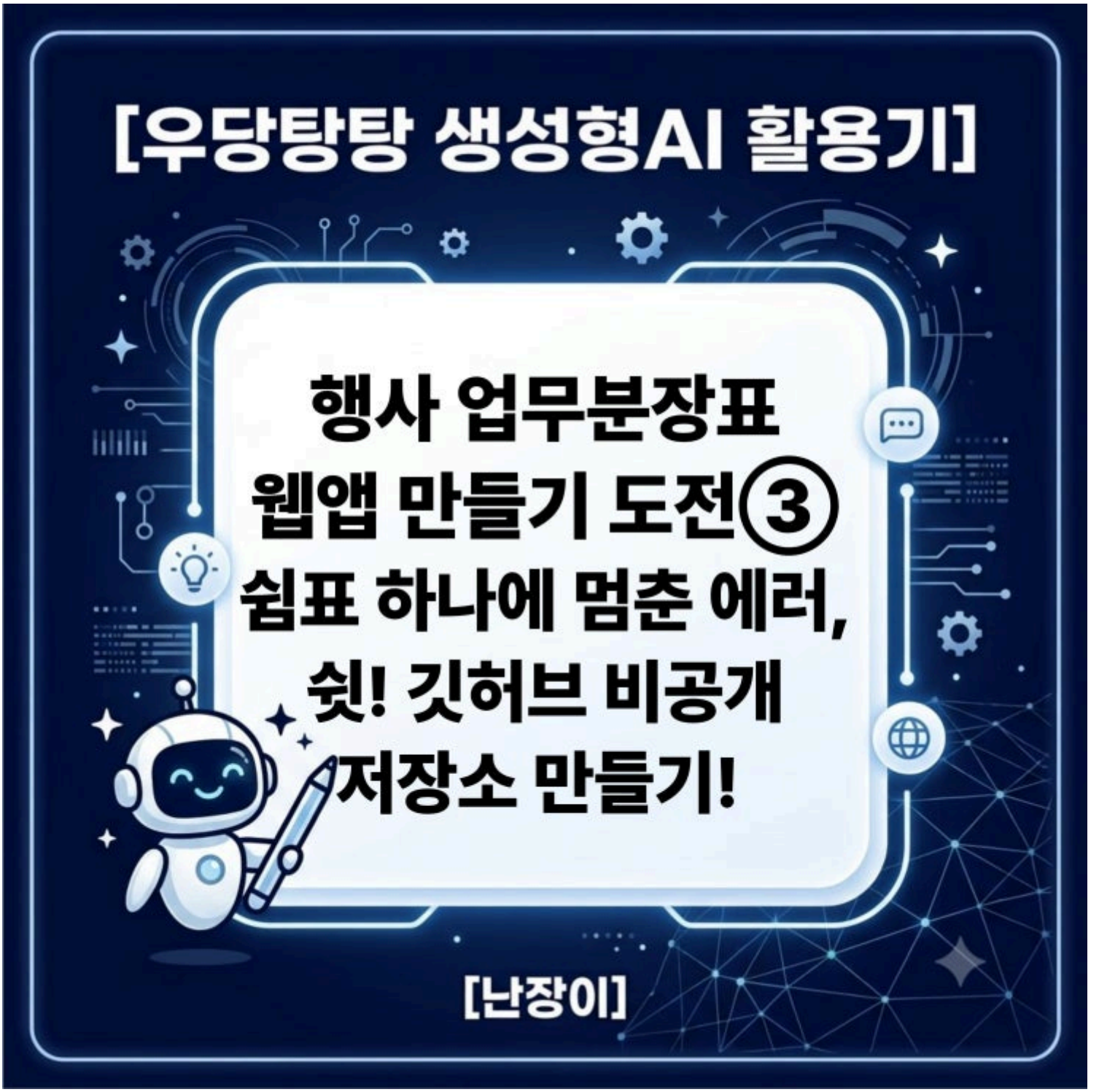
다음 포스팅에서는 **축제 업무분장표 웹앱을 업데이트하는 방법**을 들고 오겠습니다!

모두 오늘 하루도 화이팅 하시고, 공감과 응원 댓글은 큰 힘이 됩니다! ??

[우당탕탕 생성형AI 활용기] 초보 코딩, 챗지피티 유료 결제 첫날! 행사 업무분장표 웹앱 만들기 도전기(쉼표 하나에 멈춘 에러, 쉿! 깃허브 비공개 저장소 만들기!)

프로그램

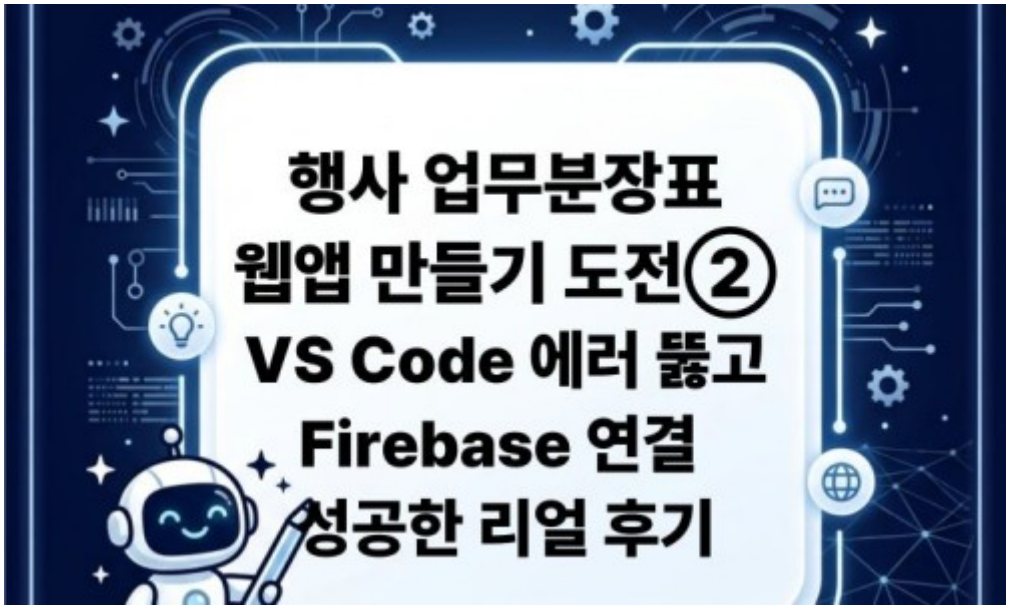
2026/10/07 08:00 http://blog.naver.com/talktalk2024/224433585896



안녕하세요!!

업무분장표 웹앱 만들기 늪에서 빠져나오고 있는 난장이입니다.

어제에 이어서 '업무분장표 웹앱 만들기' 포스팅을 시작하겠습니다.

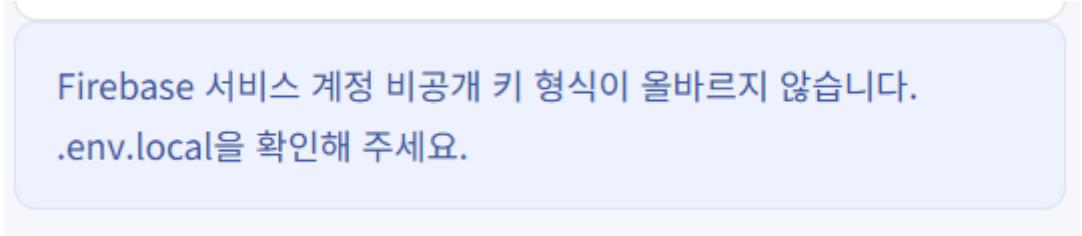


[우당탕탕 생성형AI 활용기] 초보 코딩, 챗지피티 유료 결제 첫날! 행사 업무분장표 웹앱 만들기 도전기(VS Code 에러 뚫고 Firebase 연결 성공한 리얼 후기)
안녕하세요!! 업무분장표 만드는 과정에서 고군분투하고 돌아온 난장이입니다. 오늘은 어제 만든 웹앱에 테...
blog.naver.com

이마 탁 치게 만드는 또 발생한 고비들!

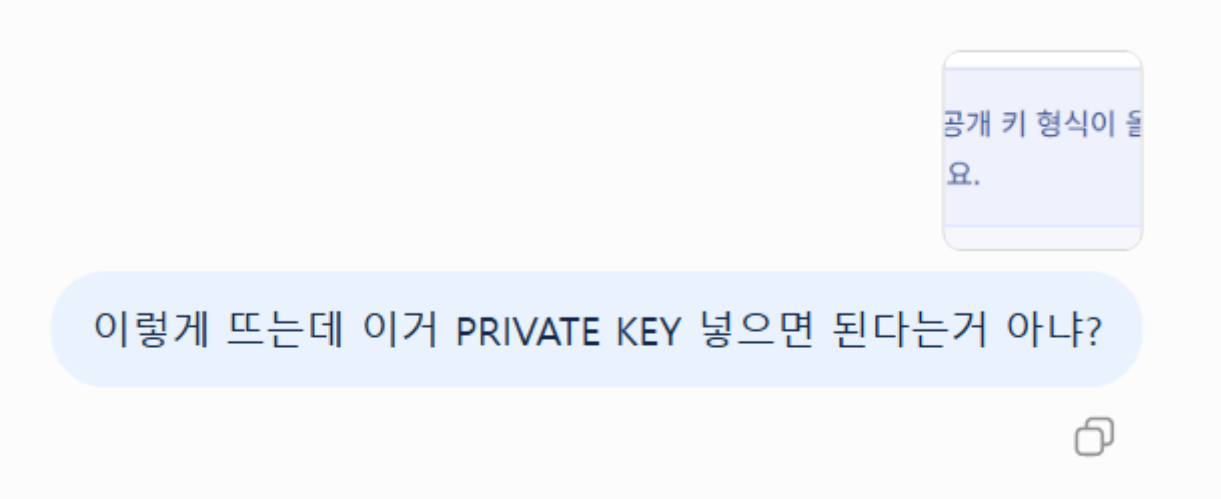
어제 또 여러 번의 고비를 넘기고 다음 단계로 넘어가기 위해 관리자로 로그인을 멋지게 하려는데 화면에 "Firebase 서비스 계정 비공개 키 형식이 올바르지 않습니다.

.env.local을 확인해 주세요."라는 에러 메시지가 떡하니 떴어요.

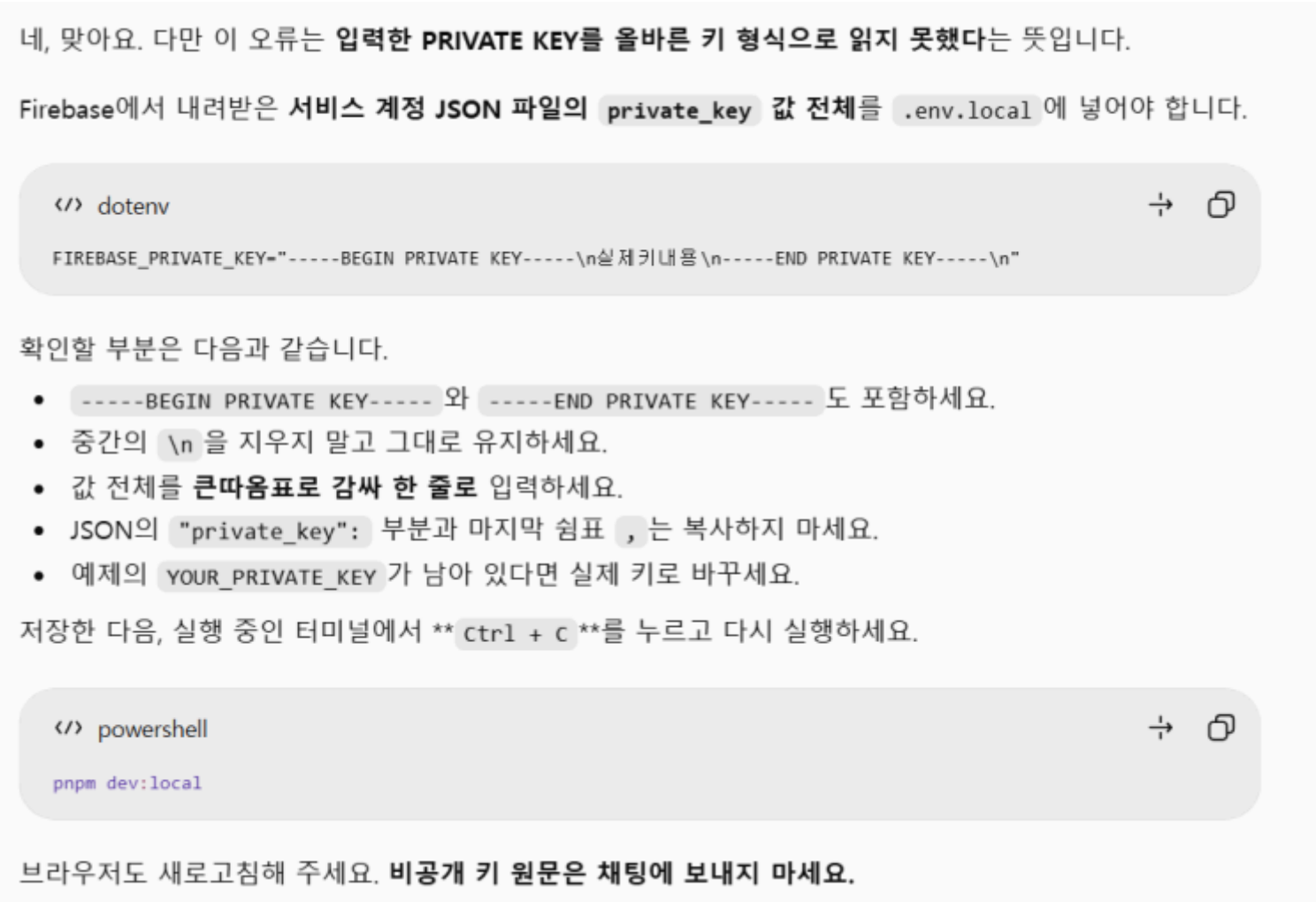


오류가 뜰 때는 어떻게 해라?

바로 또 캡쳐해서 질문 들어갔습니다.



챗GPT가 **PRIVATE KEY 값을 넣을 때**는 줄바꿈(\n)은 유지하되, 각 줄 끝의 쉼표는 빼고 큰따옴표로 감싸서 한 줄로 입력해야 한다고 정확히 짚어주더라고요.



반영한 후에 다시 로그인을 하자 이번에는 파이어베이스 서비스 계정과 키의 유효성을 확인하라는데 어디에서 어떻게 확인하면 되는지 모르니까 또 캡쳐해서 질문을 던졌어요.

비밀번호

비밀번호 입력

Firebase 인증에 실패했습니다. 서비스 계정과 키의 유효성을 확인해 주세요.

관리자 로그인



파이어베이스 서비스 계정과 키의 유효성을 확인하라는데 어디에서 어떻게 확인하면 돼?

14초 동안 작업해서 이번 오류도 잘 파악해서 해결책을 제시해줬어요.

이번 오류는 **키 형식은 읽었지만 Google에서 인증을 받아주지 않은 상태**입니다. 화면만으로는 키가 잘못됐는지, 서비스 계정과 맞지 않는지까지 구분할 수 없습니다.

먼저 **같은 JSON 파일에서 세 값을 복사했는지** 확인해 주세요.

`.env.local` 항목	서비스 계정 JSON에서 가져올 값
`FIREBASE_PROJECT_ID`	`project_id`
`FIREBASE_CLIENT_EMAIL`	`client_email`
`FIREBASE_PRIVATE_KEY`	`private_key`

특히 `FIREBASE_CLIENT_EMAIL`**에는 본인의 Gmail 주소를 넣으면 안 됩니다.** JSON에 적힌 `…@….iam.gserviceaccount.com` 주소를 그대로 넣어야 합니다. `private_key_id`도 비공개 키와 다른 값입니다.

계정과 키의 상태는 다음 위치에서 확인하세요.

1. Google Cloud 콘솔에 접속합니다.
2. 상단에서 Firebase와 **같은 프로젝트**를 선택합니다.
3. **IAM 및 관리자 → 서비스 계정**으로 들어갑니다.
4. JSON의 `client_email`과 같은 계정을 선택합니다.
5. 계정이 사용 중지되지 않았는지 확인하고, **키** 탭에서 JSON의 `private_key_id`와 같은 키 ID가 있는지 확인합니다. 삭제되거나 사용 중지된 키는 인증에 실패할 수 있습니다. developers.goo...

해당 키가 없거나 사용할 수 없다면, **Firebase 콘솔 → 프로젝트 설정 → 서비스 계정 → 새 비공개 키 생성**에서 새 JSON을 받고, 위 세 값을 모두 그 파일 기준으로 교체하세요. firebase.google....

마지막으로 저장하고 서버를 다시 실행하세요.

```powershell
# 실행 중인 터미널에서 Ctrl+C를 누른 뒤
pnpm dev:local
```

우선 `FIREBASE_CLIENT_EMAIL`에 Gmail을 넣었는지부터 확인해 주세요. 키 원문이나 JSON 파일은 보내지 않아도 됩니다.

이 내용들을 확인하고 차근차근 하나씩 따라하는데 각 줄 맨 뒤에 있는 쉼표 ','가 있는게 보였어요.

그래서 챗GPT에게 JSON 파일 캡처를 보여주며 "혹시 이거 다 쉼표(,)는 복사하지 않는거야?" 하고 물어봤죠.

알고 보니, 제가 키 값을 복사하면서 맨 끝에 붙은 **작은 쉼표 하나(,)까지 몽땅 긁어다 넣은 게 화근**이었어요!

혹시 이거 다 ,는 복사하지 않는거야?

네. **JSON에서 각 항목 끝에 붙은 쉼표 `,`는 모두 빼고 입력하세요.**

JSON 파일이 이렇게 되어 있다면:

JSON

```
"project_id": "my-festival",
"client_email": "example@my-festival.iam.gserviceaccount.com",
"private_key": "-----BEGIN PRIVATE KEY-----\n키내용\n-----END PRIVATE KEY----
```

`.env.local`에는 이렇게 넣습니다:

dotenv

```
FIREBASE_PROJECT_ID=my-festival
FIREBASE_CLIENT_EMAIL=example@my-festival.iam.gserviceaccount.com
FIREBASE_PRIVATE_KEY="-----BEGIN PRIVATE KEY-----\n키내용\n-----END PRIVATE KI
```

각 줄 끝의 쉼표는 제외하고, PRIVATE KEY의 큰따옴표와 `\n`은 유지하세

ChatGPT로 Work 시작

GPT-6 Astra Light

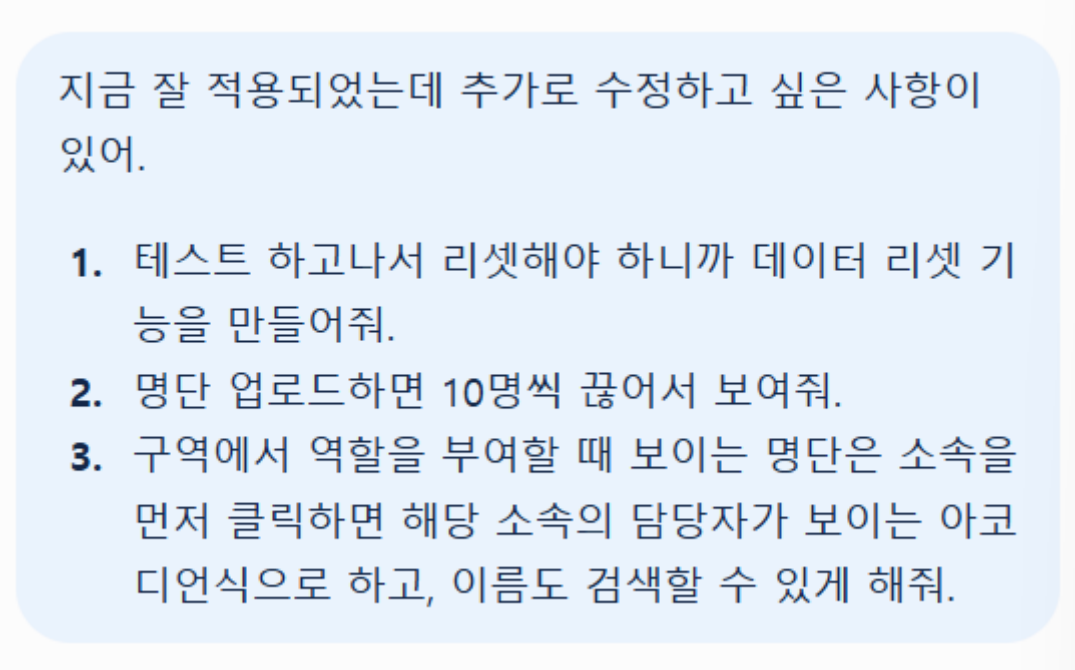
단 하나의 쉼표 오차도 허용하지 않는 코딩의 세계...

정말 세심하고 예민한 녀석이라는 걸 온몸으로 느꼈답니다.

무사히 에러를 해결하고, 내친김에 챗GPT에게 **데이터 리셋 기능, 10명씩 끊어서 보기, 소속 클릭 시 담당자가 보이는 아코디언 기능, 이름 검색 기능**까지 야무지게 추가해달라고 요청해서 업데이트를 마쳤어요.

이 기능들 중에 10명씩 끊어서 보기나 아코디언 기능은 데이터가 발생되는 양을 줄이는 데에 큰 도움이 됩니다.

(이건 나중에 파이어베이스와 같은 데이터 저장소 사용량 감소하는 방법에 대해 포스팅 해볼게요!)

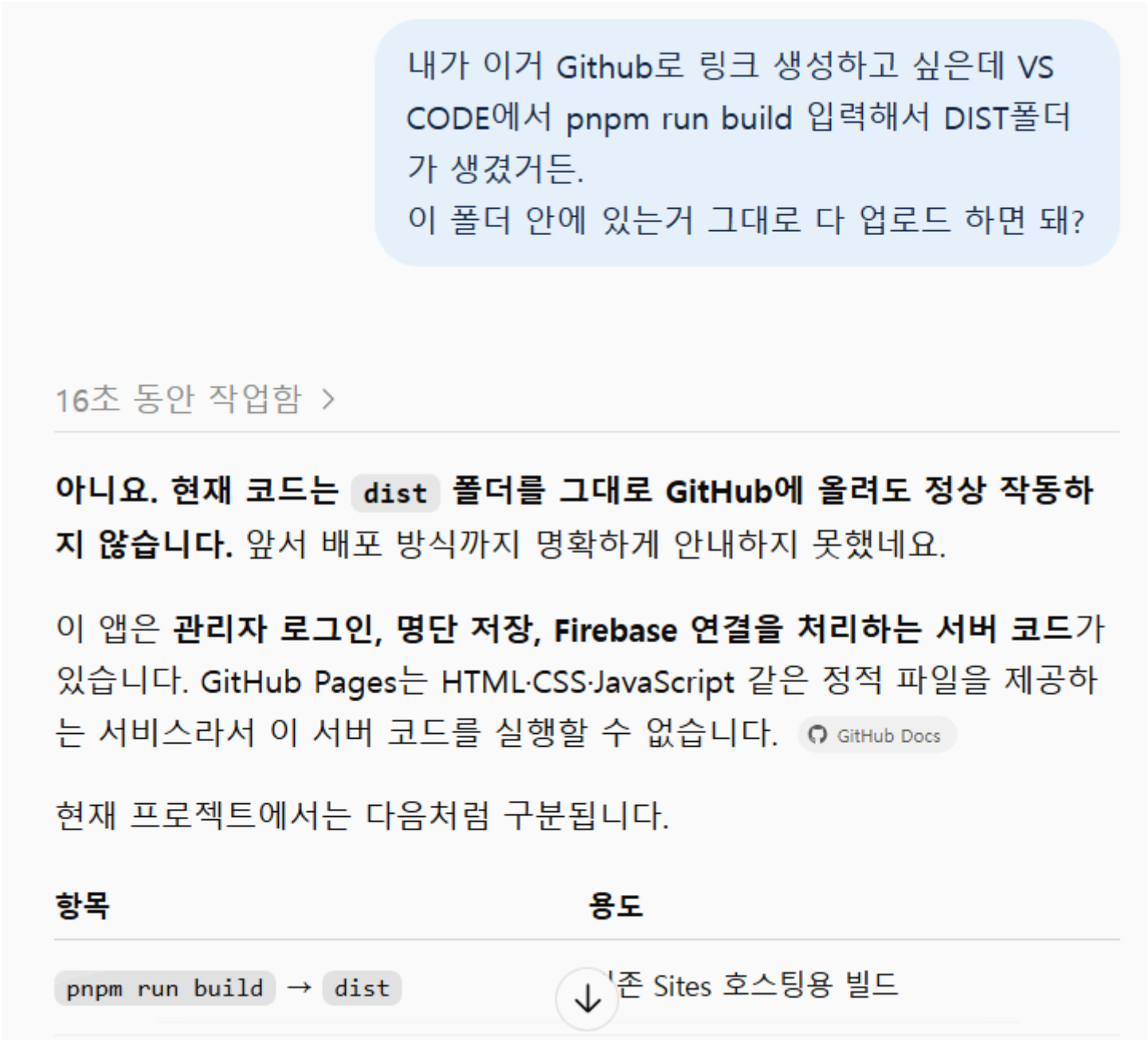


깃허브에 업로드하면 링크가 생성되는거 아니었어??

이제 드디어 완성된 프로그램을 진짜 인터넷에 올려볼 차례예요!

예전에 제미나이(Gemini)로 코딩했을 때는 VS Code 터미널에 pnpm run build를 입력해서 만들어진 dist 폴더를 깃허브에 올리기만 하면 끝이었거든요.

그래서 이번에도 당당하게 "dist 폴더 안에 있는 거 그대로 다 업로드 하면 돼?"라고 물어봤어요.



그런데 챗GPT의 단호한 대답은 **"아니요!"** 였습니다.

제가 이번에 만든 앱은 관리자 로그인이나 명단 저장 같은 '서버 코드'가 포함되어 있어서, 단순히 화면만 보여주는 깃허브 페이지(GitHub Pages)로는 정상 작동할 수 없대요.

? 1초 용어 사전

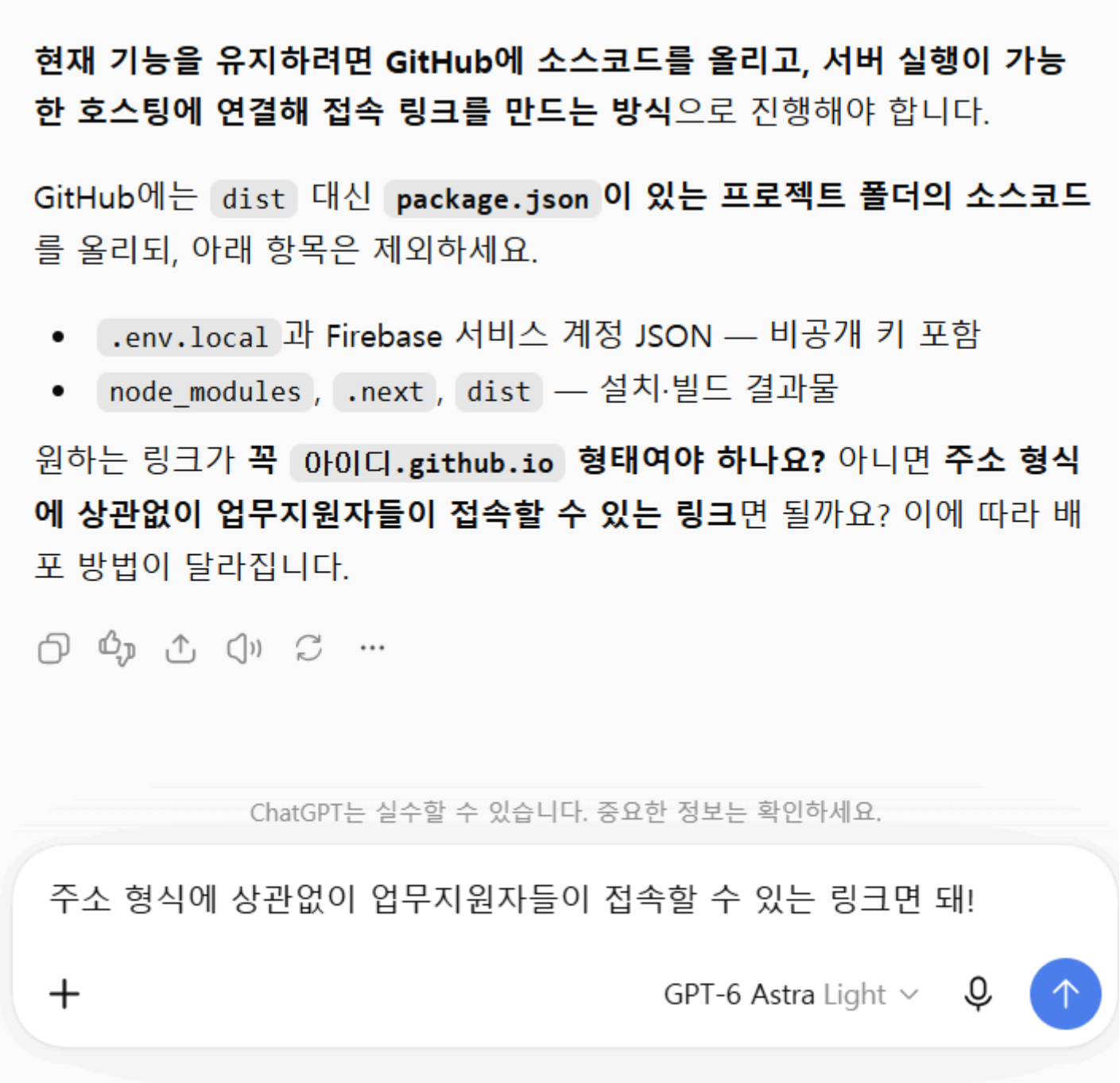
정적 파일 vs 서버 코드

정적 파일(HTML 등)은 '인쇄된 전단지'처럼 누가 보든 똑같은 그림만 보여주는 거고요.

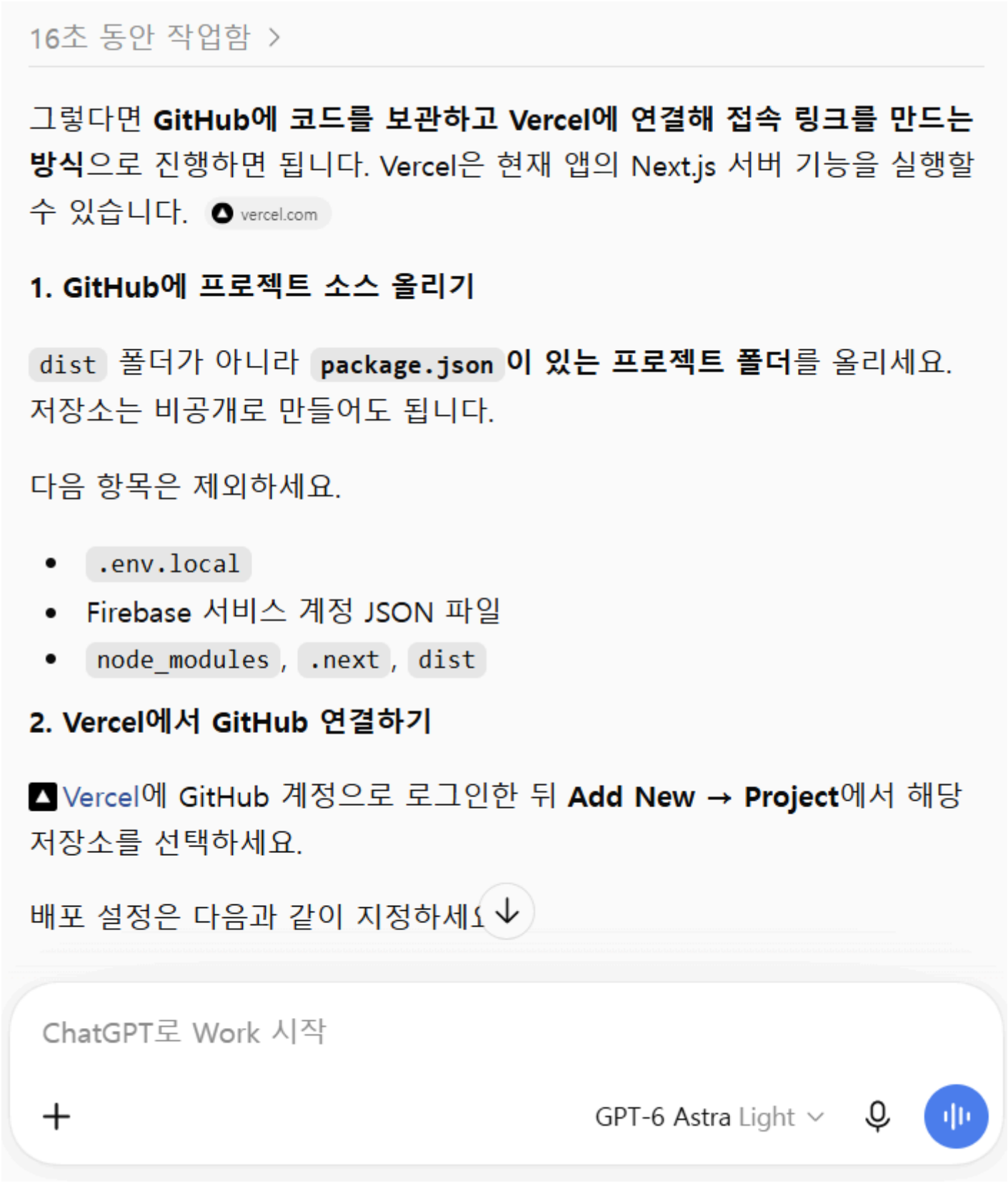
서버 코드는 '요리사가 있는 식당 주방'처럼 로그인이나 검색 같은 주문이 들어오면

그때그때 알맞게 요리해서 보여주는 똑똑한 기능이에요!

대신 현재 기능을 유지하려면, **깃허브에 소스코드를 보관하고 Vercel(버셀)이라는 호스팅 서비스에 연결해 접속 링크를 만드는 방식**으로 진행해야 한다고 하네요.



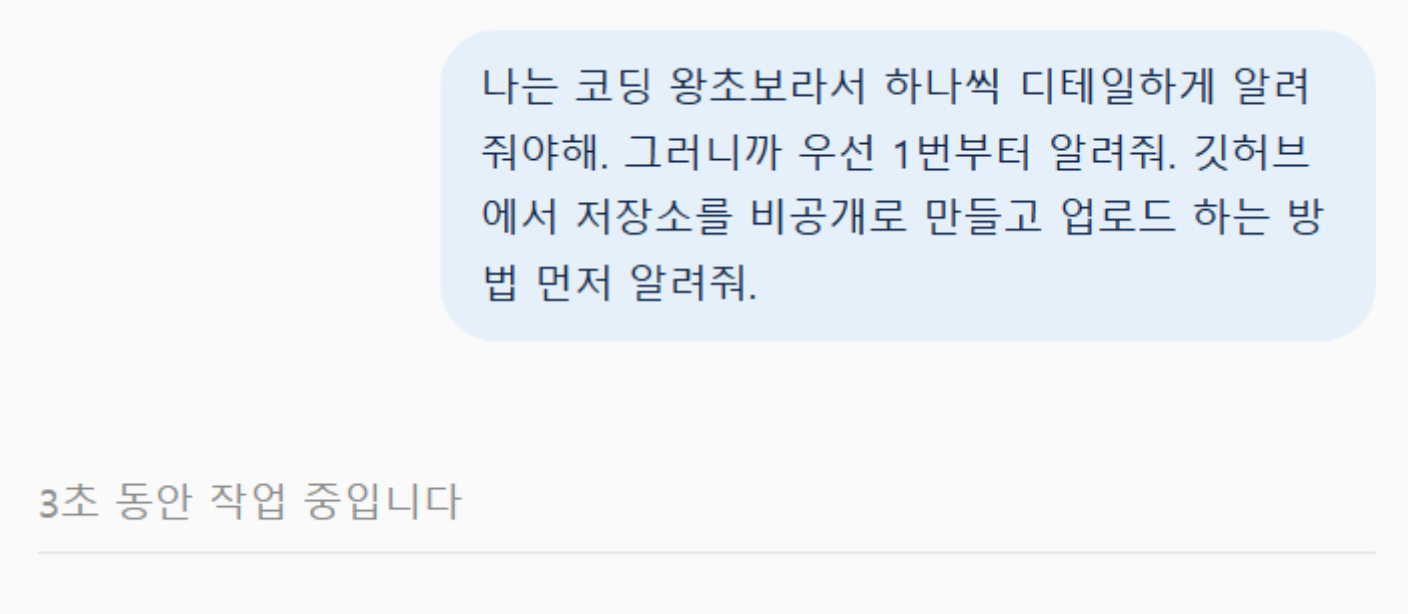
주소 형식에 상관없이 누구나 접속할 수 있는 링크를 만들 수 있다니, 이 달콤하고 새로운 방식을 지나칠 수 없죠!



하지만 저는 코딩 왕초보!

처음 듣는 Vercel 연결 방식이 나오니 덜컥 막막해졌어요.

그래서 **"나는 코딩 왕초보라서 하나씩 디테일하게 알려줘야 해. 우선 1번부터 알려줘."**라고 챗GPT에게 SOS를 쳤습니다.



나만의 깃허브 비공개 저장소 만들기 대작전!!

가장 먼저 한 일은 **깃허브(GitHub)에 '비공개 저장소' 만들기**였어요.

챗GPT가 화면 항목별로 어떤 걸 입력하고 선택해야 하는지(Visibility는 Private 선택 등) 표로 꼼꼼히 정리해 줘서, 터미널 명령어를 몰라도 웹사이트에서 마우스 클릭만으로 쉽게 만들 수 있었답니다.

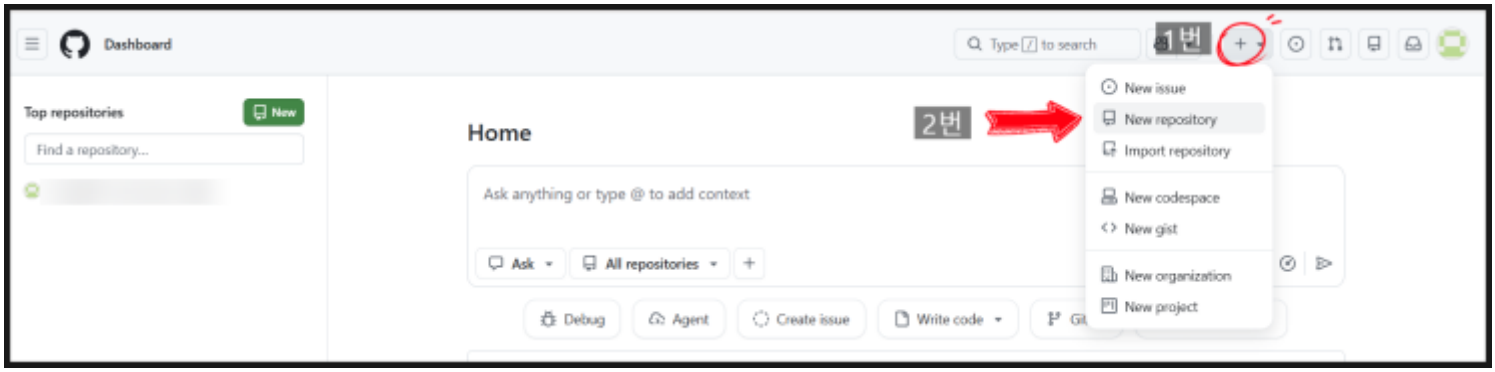
① GitHub에 비공개 저장소 만들기

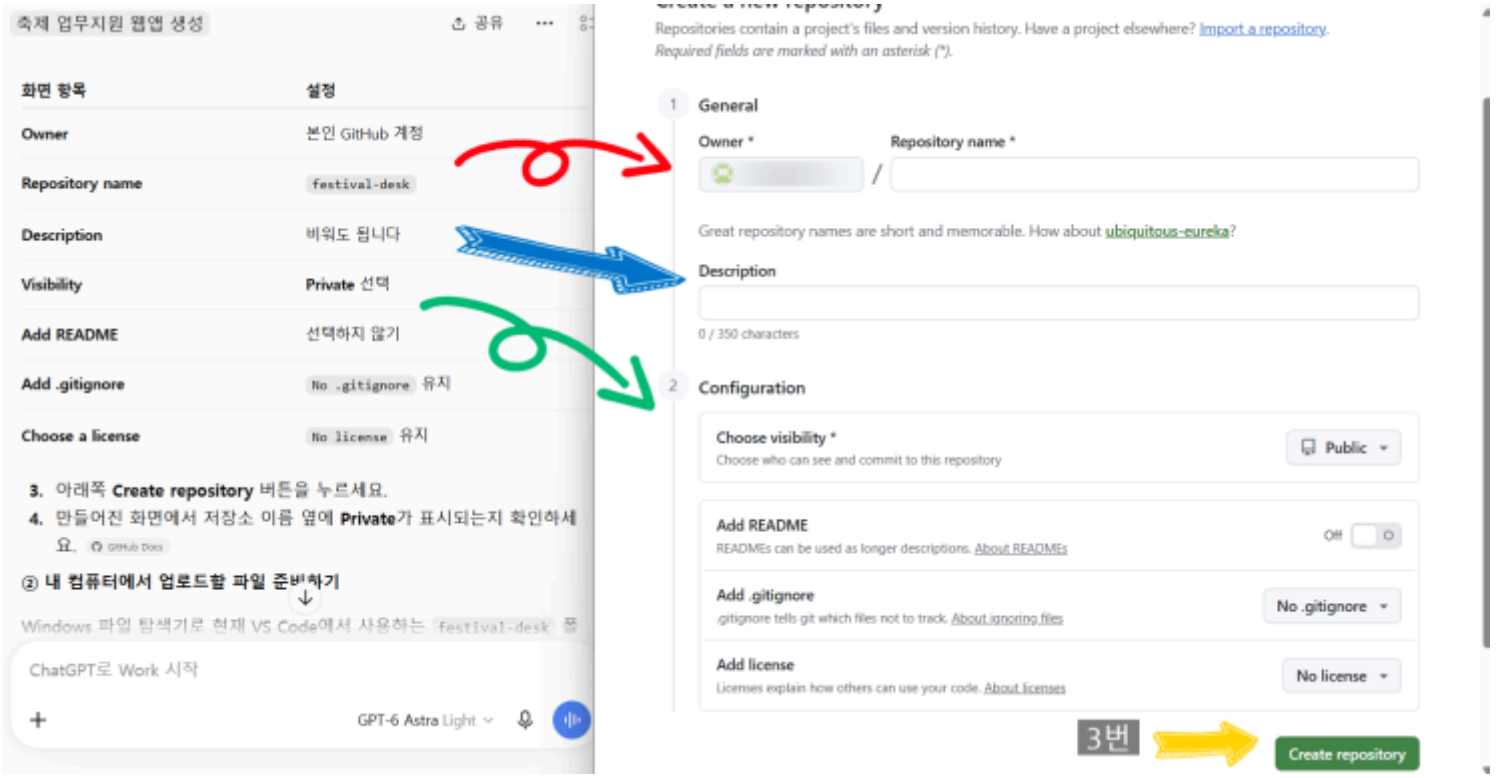
1. GitHub 새 저장소 만들기에 접속하고 로그인하세요.
2. 아래처럼 설정하세요

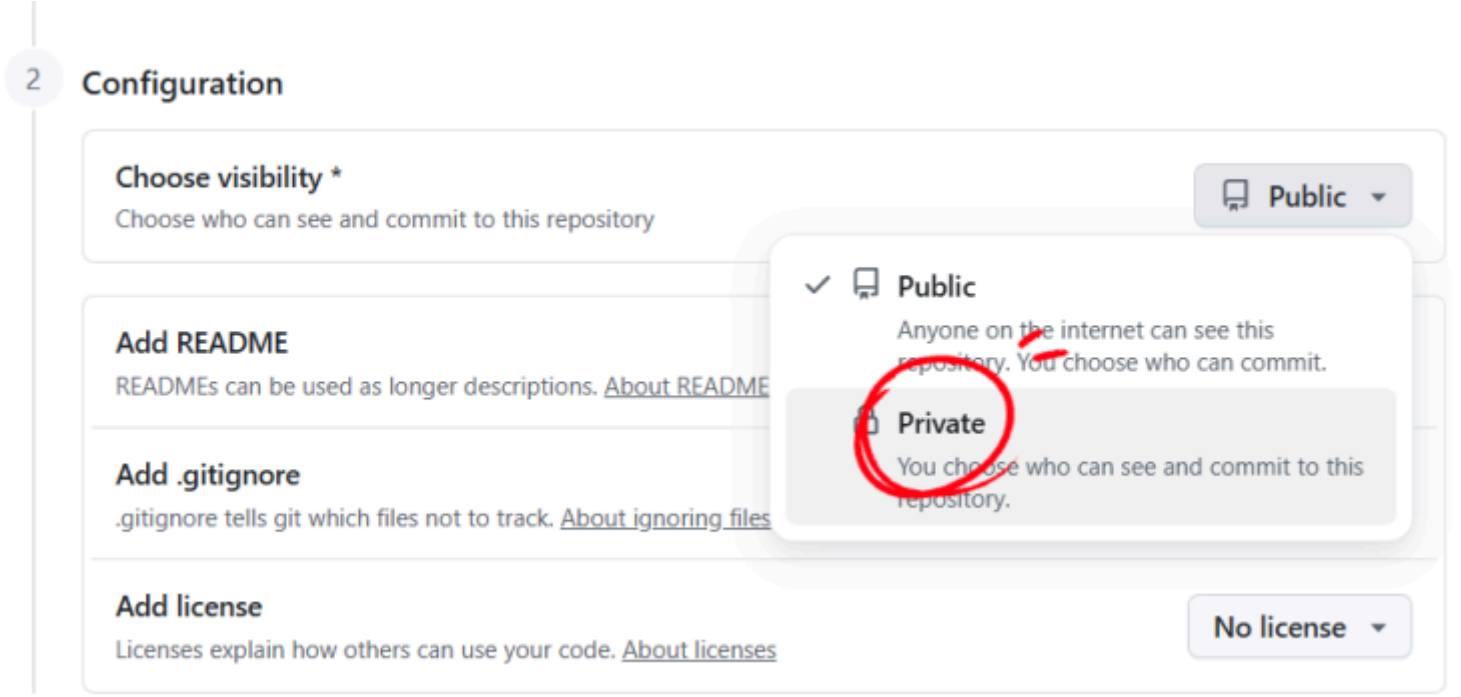
화면 항목	설정
Owner	본인 GitHub 계정
Repository name	`festival-desk`
Description	비워도 됩니다
Visibility	**Private** 선택
Add README	선택하지 않기
Add .gitignore	`No .gitignore` 유지
Choose a license	`No license` 유지

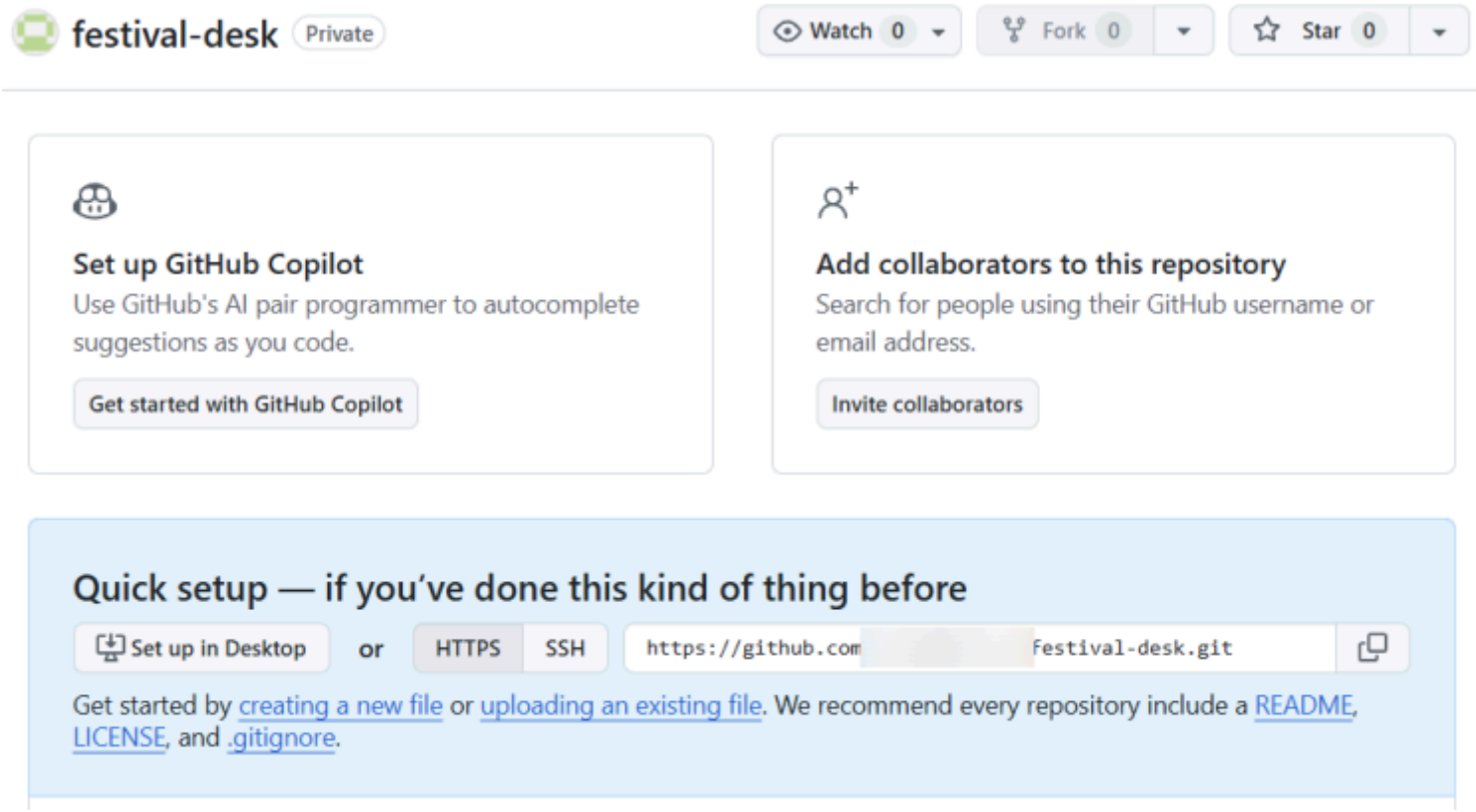
3. 아래쪽 **Create repository** 버튼을 누르세요.
4. 만들어진 화면에서 저장소 이름 옆에 **Private**가 표시되는지 확인하세요. GitHub Docs

새창을 띄워서 나란히 놓고 하나하나 차례대로 입력해봤습니다.




축제 업무지원 웹앱 생성
공유
화면 항목
설정
Owner
본인 GitHub 계정
Repository name
festival-desk
Description
비워도 됩니다
Visibility
Private 선택
Add README
선택하지 않기
Add .gitignore
No .gitignore 유지
Choose a license
No license 유지
3. 아래쪽 Create repository 버튼을 누르세요.
4. 만들어진 화면에서 저장소 이름 옆에 Private가 표시되는지 확인하세요.
② 내 컴퓨터에서 업로드할 파일 준비하기
ChatGPT로 Work 시작
GPT-6 Astra Light
Repositories contain a project's files and version history. Have a project elsewhere? Import a repository.
Required fields are marked with an asterisk (*).
1 General
Owner *
Repository name *
Great repository names are short and memorable. How about ubiquitous-eureka?
Description
0 / 350 characters
2 Configuration
Choose visibility *
Choose who can see and commit to this repository
Public
Add README
READMEs can be used as longer descriptions. About READMEs
Off
Add .gitignore
.gitignore tells git which files not to track. About ignoring files
No .gitignore
Add license
Licenses explain how others can use your code. About licenses
No license
3번
Create repository


2 Configuration
Choose visibility *
Choose who can see and commit to this repository
Public
Add README
READMEs can be used as longer descriptions. About README
Add .gitignore
.gitignore tells git which files not to track. About ignoring files
Add license
Licenses explain how others can use your code. About licenses
No license
Public
Anyone on the internet can see this repository. You choose who can commit.
Private
You choose who can see and commit to this repository.



그다음은 내 컴퓨터에서 깃허브로 업로드할 파일들을 솎아내는 준비 단계예요.

폴더 안에 package.json, app, components 같은 파일이 보이는 위치가 정확히 맞는지 꼭 확인해야 한다고 해요.

여기서 **가장 중요한 왕초보 필수 주의사항!**

업로드에서 반드시 제외해야 하는 항목들이 있어요.

특히 파이어베이스 비공개 키가 들어있는 .env.local 파일이나 JSON 파일은 절대 올리면 안 됩니다!

비공개 저장소라고 하더라도 해킹의 위험이 있기 때문에, 알아서 빠지겠거니 생각하지 말고 무조건 직접 선택에서 제외해야 한다고 신신당부를 하더라고요.

② 내 컴퓨터에서 업로드할 파일 준비하기

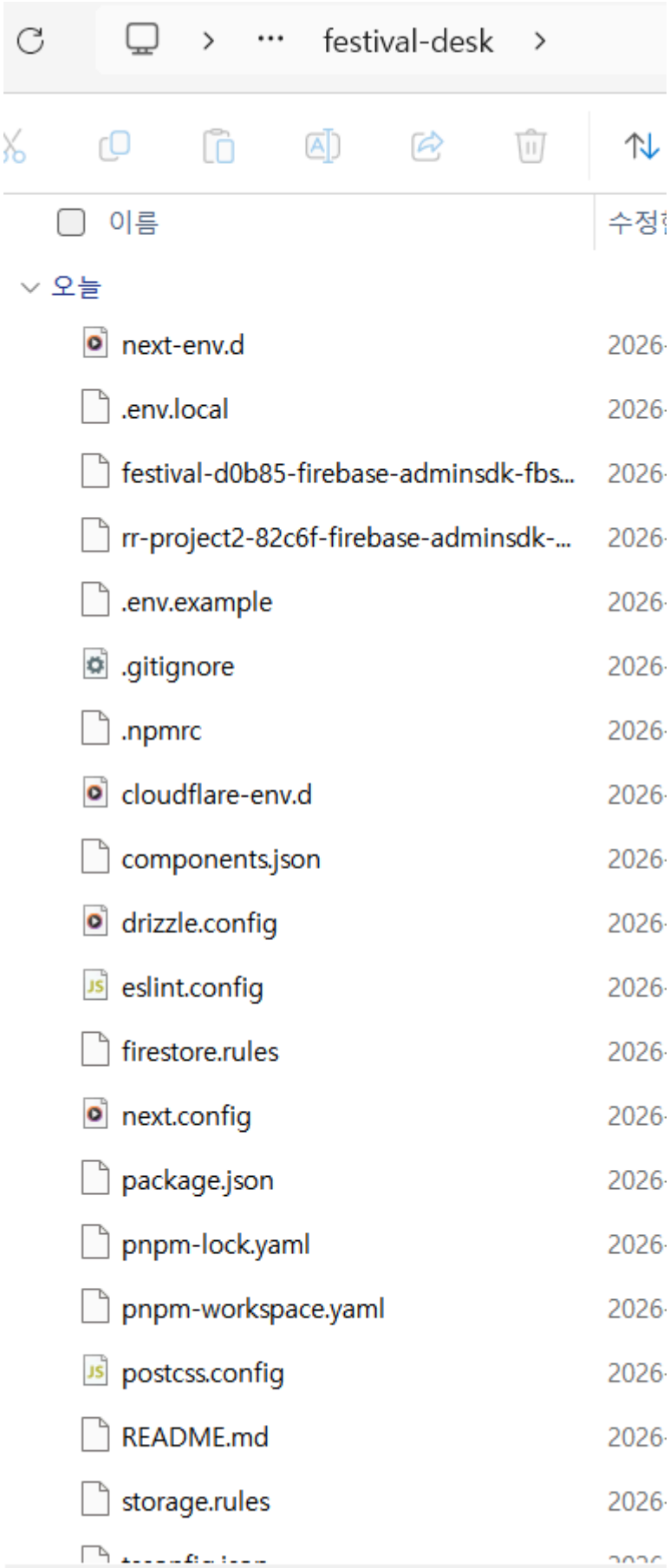
Windows 파일 탐색기로 현재 VS Code에서 사용하는 `festival-desk` 폴더를 여세요.

`package.json`, `app`, `components` 등이 보이는 위치가 맞습니다. `dist` 안으로 들어가지 마세요.

다음 항목은 업로드에서 제외해야 합니다. **컴퓨터에서 삭제하라는 뜻은 아닙니다.**

업로드하지 않을 항목	이유
`.env.local`	Firebase 비공개 키 포함
내려받은 Firebase 서비스 계정 JSON	비공개 키 포함
`node_modules`	설치된 라이브러리
`dist`, `.next`, `.wrangler`, `.sites-runtime`	빌드 결과·로컬 실행 자료
`.git`	로컬 Git 관리 자료

비공개 저장소라도 비공개 키는 올리지 마세요. 웹 업로드에서는 `.gitignore`가 알아서 제외해 줄 것이라고 생각하지 말고 직접 선택에서 빼주세요.


festival-desk
이름
수정
오늘
next-env.d 2026
.env.local 2026
festival-d0b85-firebase-adminsdk-fbs... 2026
rr-project2-82c6f-firebase-adminsdk-... 2026
.env.example 2026
.gitignore 2026
.npmrc 2026
cloudflare-env.d 2026
components.json 2026
drizzle.config 2026
eslint.config 2026
firestore.rules 2026
next.config 2026
package.json 2026
pnpm-lock.yaml 2026
pnpm-workspace.yaml 2026
postcss.config 2026
README.md 2026
storage.rules 2026

? 1초 용어 사전

.env.local 파일

남들에게 보여주면 안 되는 우리 집 비밀번호나 금고 열쇠 같은 중요한 정보들을

따로 숨겨두는 '비밀 일기장' 같은 파일이에요.

인터넷에 올리면 절대 안 돼요!

떨리는 첫 업로드, 일단 하나만 올려보자!

챗GPT가 아주 친절하게 짚어준 대로, 방금 만든 빈 저장소 화면에서 uploading an existing file이라는 작은 문구를 찾아서 클릭했어요.

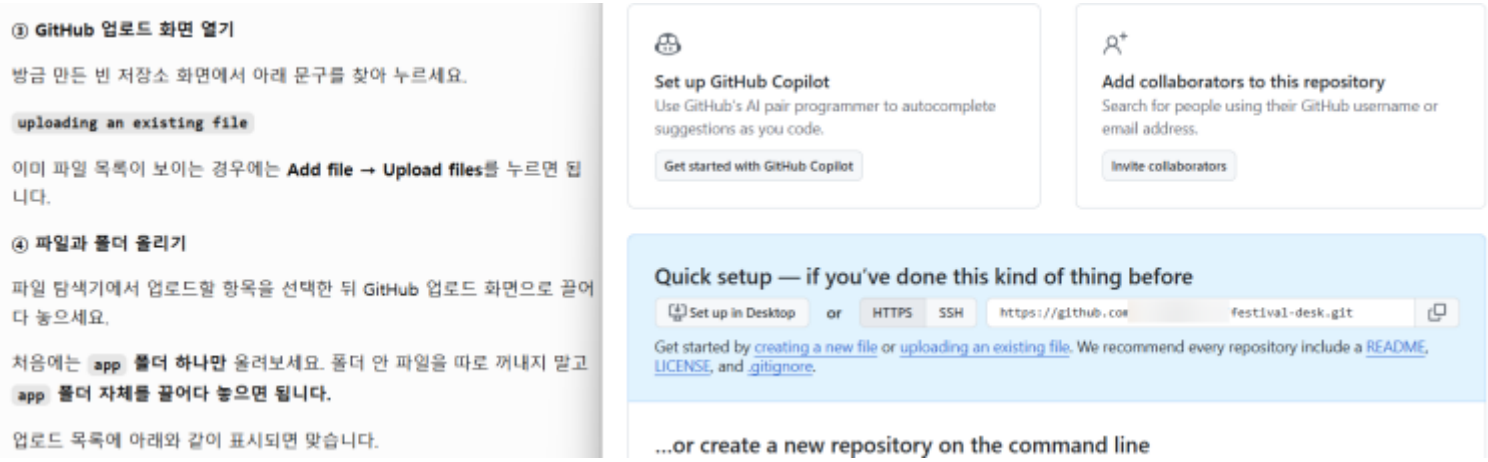
그랬더니 마치 탐색기처럼 파일을 끌어다 놓을 수 있는(드롭) 네모난 화면이 짠 하고 나타나더라고요.

? 1초 용어 사전

깃허브(GitHub) 저장소

내가 짠 코드들을 안전하게 보관해 주는 '인터넷상의 전용 클라우드 방'이라고 생각하면 돼요!

네이버 마이박스나 구글 드라이브 같은 건데, 코드 보관에 최적화된 곳이랍니다.



혹시라도 잘못 올라갈까 봐 무서워서, 처음에는 챗GPT의 조언대로 app 폴더 딱 하나만 살포시 끌어다 놓아봤어요.

다행히 업로드 목록에 폴더 안의 내용들이 예쁘게 표시되는 걸 확인하고 안심했죠!

④ 파일과 폴더 올리기

파일 탐색기에서 업로드할 항목을 선택한 뒤 GitHub 업로드 화면으로 끌어다 놓으세요.

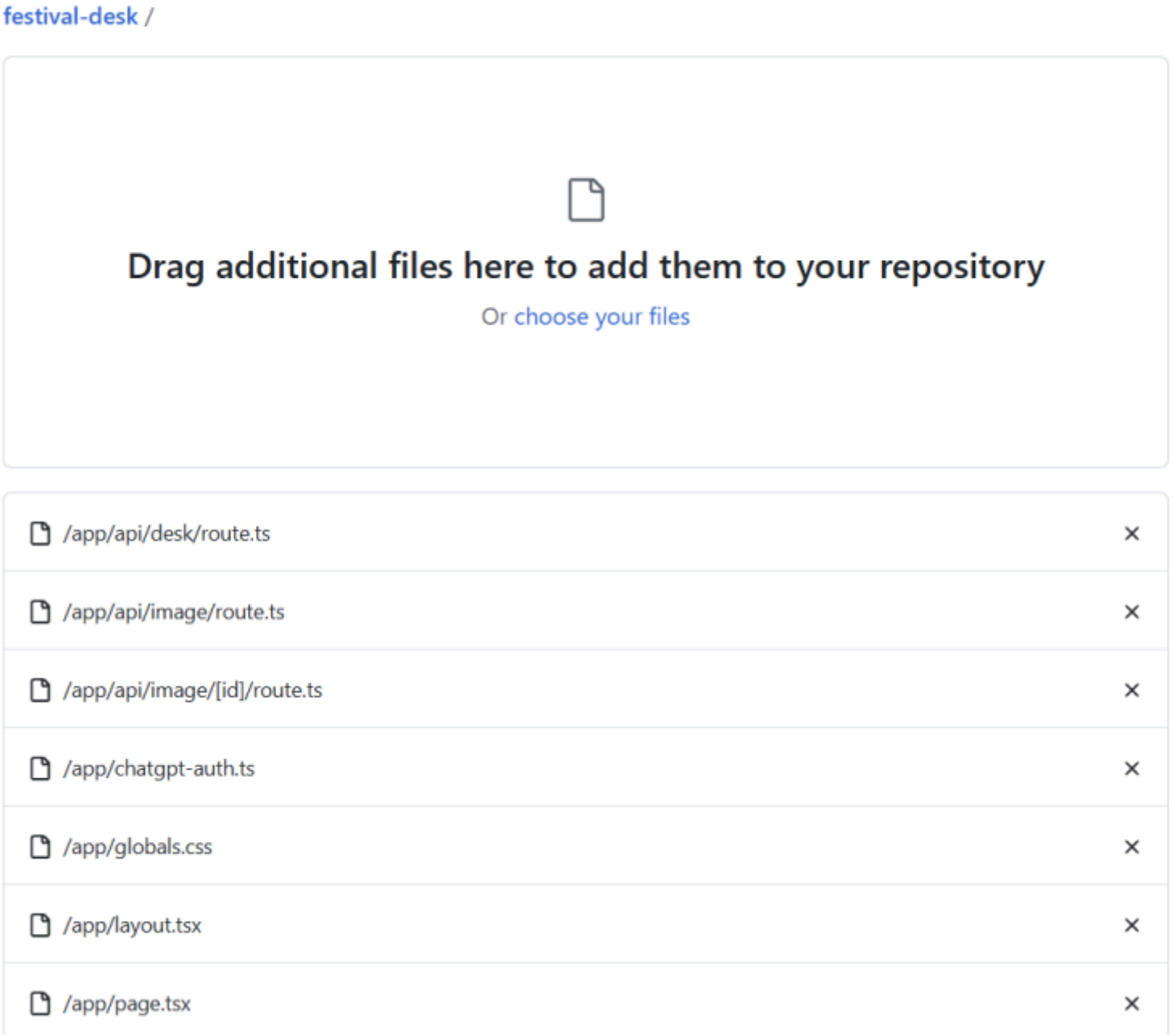
처음에는 `app` **폴더 하나만** 올려보세요. 폴더 안 파일을 따로 꺼내지 말고 `app` **폴더 자체를 끌어다 놓으면 됩니다.**

업로드 목록에 아래와 같이 표시되면 맞습니다.

```
일반 텍스트

app/page.tsx
app/layout.tsx
app/globals.css
```

화면 아래의 **Commit changes** 버튼을 누르면 저장됩니다. 설명 입력란에는 `앱 코드 업로드` 라고 적어도 됩니다.

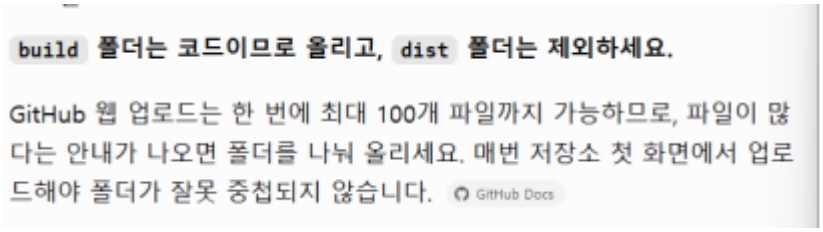


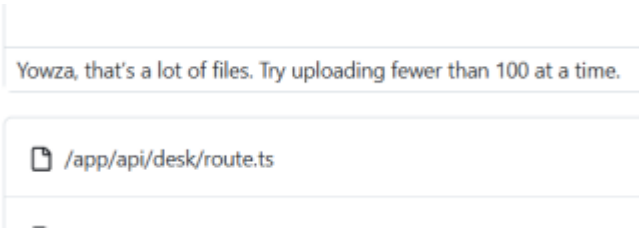
아뿔싸! 100개의 벽에 부딪히다

하나가 잘 올라가는 걸 확인했으니, 이제 자신감이 붙었잖아요?

나머지 수많은 파일과 폴더들을 마우스로 쫙 긁어서 한 번에 호기롭게 드롭해 버렸습니다.

그런데... 빙글빙글 돌더니 "Yowza, that's a lot of files. Try uploading fewer than 100 at a time." 이라는 영어 문구가 뜨면서 업로드가 멈춰버렸어요.





알고 보니 여기서 왕초보들이 자주 하는 실수가 있더라고요.

중요한 사실!!

깃허브 웹사이트에서 직접 업로드할 때는 **한 번에 최대 100개 파일까지만** 올릴 수 있대요.

그래서 파일이 너무 많으면 저처럼 에러가 나니까, 폴더를 여러 번에 나눠서 올려야 한답니다.

나눠서 올리고 도장 쾅쾅 찍기

침착하게 폴더를 몇 개씩 쪼개서 다시 드롭했어요.

화면 아래에 있는 Commit changes 초록색 버튼을 눌러야 최종적으로 저장이 되는데요.

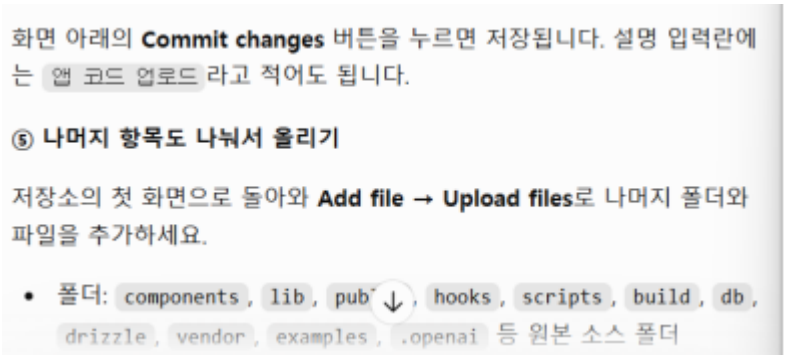
? 1초 용어 사전

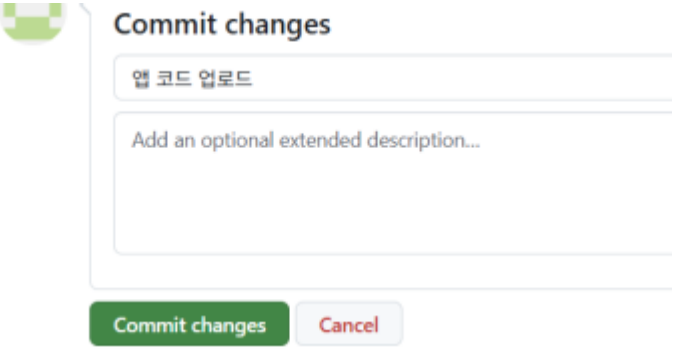
커밋(Commit)

오락실 게임 하다가 중간에 '세이브(저장)' 하는 거랑 똑같아요!

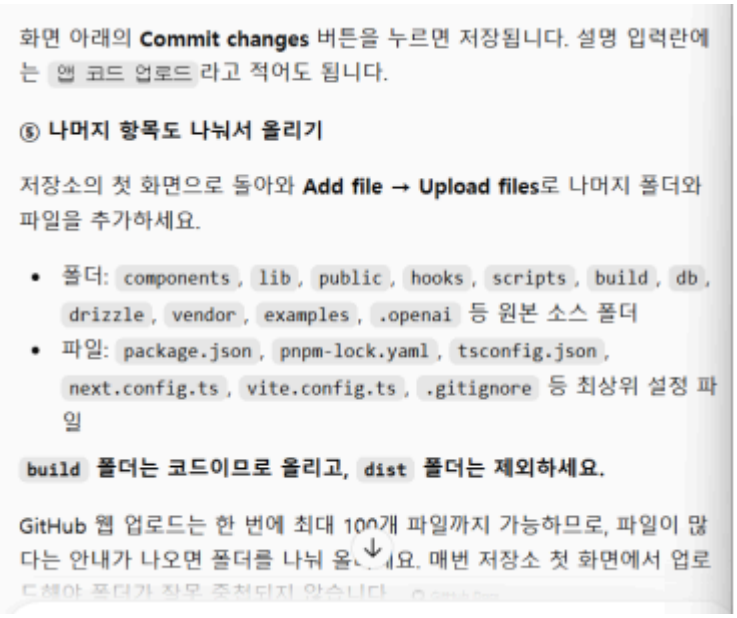
"나 여기까지 작업 완료했어!" 하고 깃허브에 도장을 쾅 찍어두는 거랍니다.

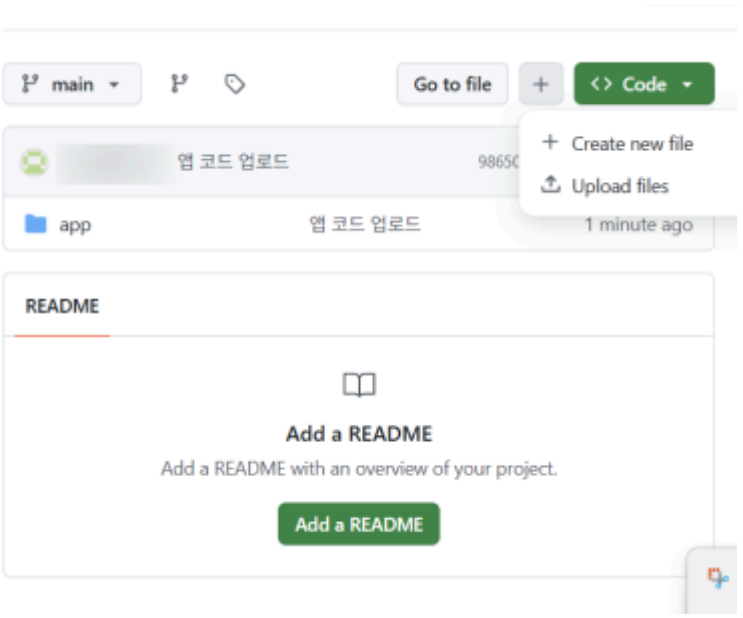
저장하기 전에 설명 입력란에 앱 코드 업로드처럼 지금 올린 파일이 뭔지 적어두면 나중에 알아보기 훨씬 쉽다고 해서 꼼꼼히 메모도 남겼어요.





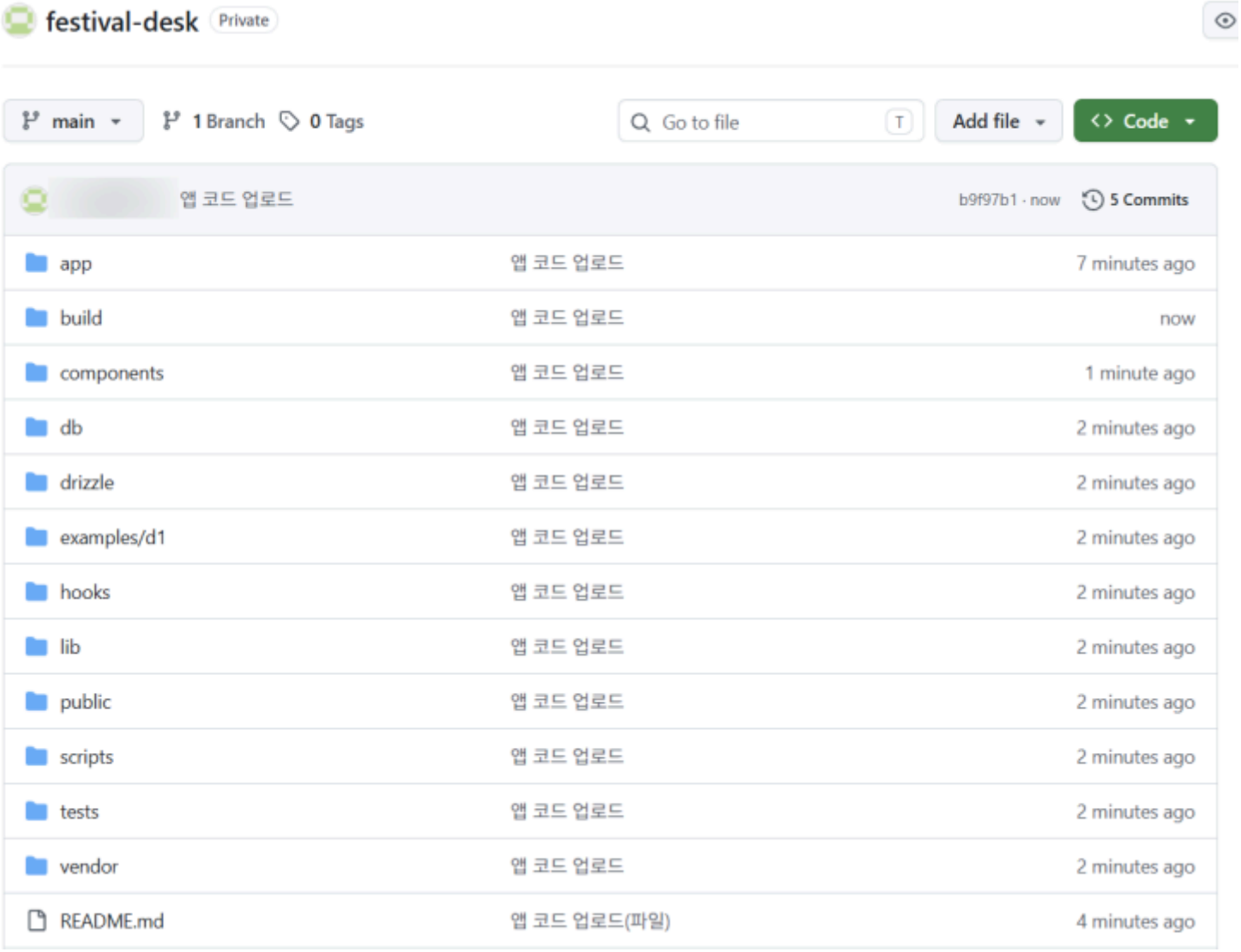
나머지 폴더들도 계속 올려야 하는데, 이때는 저장소 첫 화면으로 돌아가서 Add file -> Upload files를 누르거나, 오른쪽 상단의 Go to file 옆에 있는 **+ 버튼**을 눌러서 추가해야 폴더가 잘못 꼬이지 않는대요.





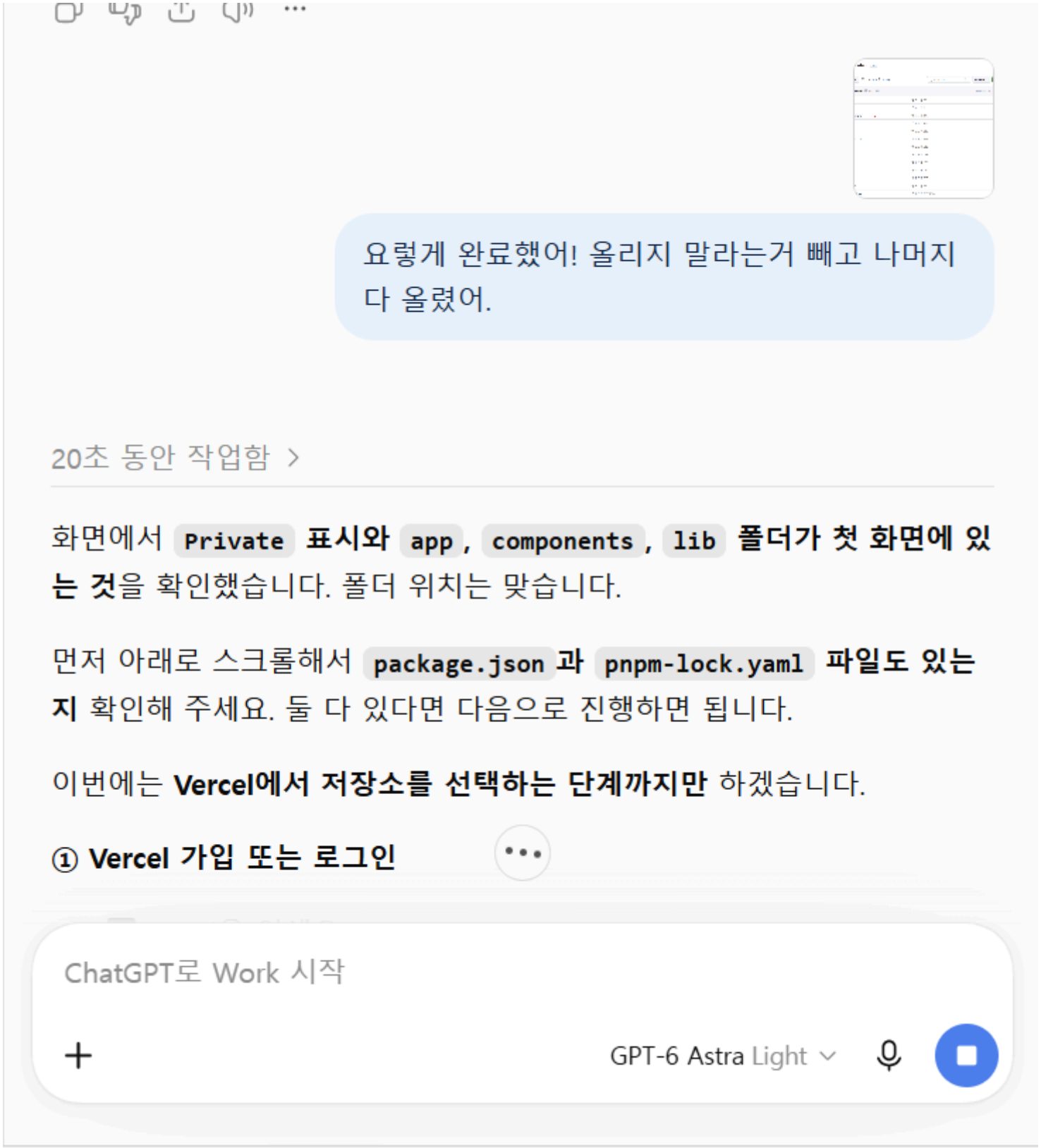
챗GPT의 칭찬으로 마무리!

여러 번의 반복 작업 끝에, 드디어 빼야 할 파일(비공개 키 등)은 쏙 빼고 나머지 소스코드들을 모두 무사히 깃허브에 올렸습니다!



마지막으로 제가 잘했는지 검사받으려고 캡처 화면을 챗GPT에게 보여줬어요.

"요렇게 완료했어!" 했더니, "Private 표시와 app, components, lib 폴더가 첫 화면에 있는 것을 확인했습니다. 폴더 위치는 맞습니다."라며 아주 든든하게 칭찬과 합격을 주더라고요!



코딩 1도 모르는 제가 제 손으로 깃허브에 소스코드를 저장하다니, 오늘 밤은 왠지 뿌듯해서 잠이 잘 올 것 같아요.

내일은 드디어 이 코드들을 인터넷 세상에 공개하는 마지막 관문!

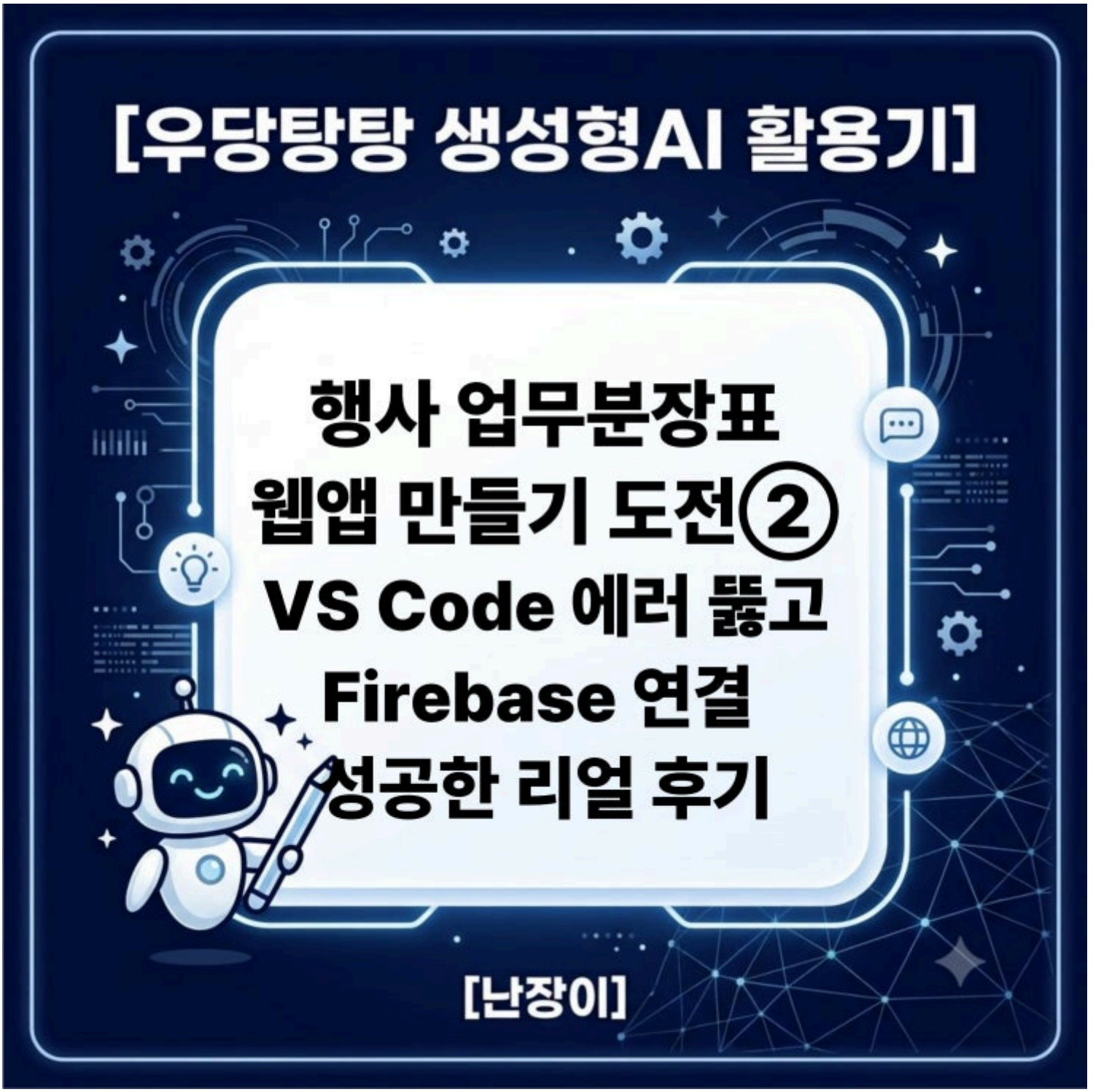
Vercel(버셀)에 가입하고 저장소를 연결하는 단계를 포스팅해 보겠습니다.

저처럼 하나하나 더듬어가며 배우고 계신 초보분들, 내일도 포기하지 말고 같이 달려봐요! ?

[우당탕탕 생성형AI 활용기] 초보 코딩, 챗지피티 유료 결제 첫날! 행사 업무분장표 웹앱 만들기 도전기(VS Code 에러 뚫고 Firebase 연결 성공한 리얼 후기)

프로그램

2026/10/06 08:00 http://blog.naver.com/talktalk2024/224432438338



안녕하세요!!

업무분장표 만드는 과정에서 고군분투하고 돌아온 난장이입니다.

오늘은 어제 만든 웹앱에 테스트용 명단을 엑셀로 올리는 작업부터, 내 컴퓨터에 진짜 화면을 띄워보는 것까지 생생한 에러 해결 과정을 공유해드릴게요!

우선 본격적인 포스팅을 하기 전, 관리자로 로그인 하면 나오는 관리자 페이지를 영상으로 한번 보여드릴게요.

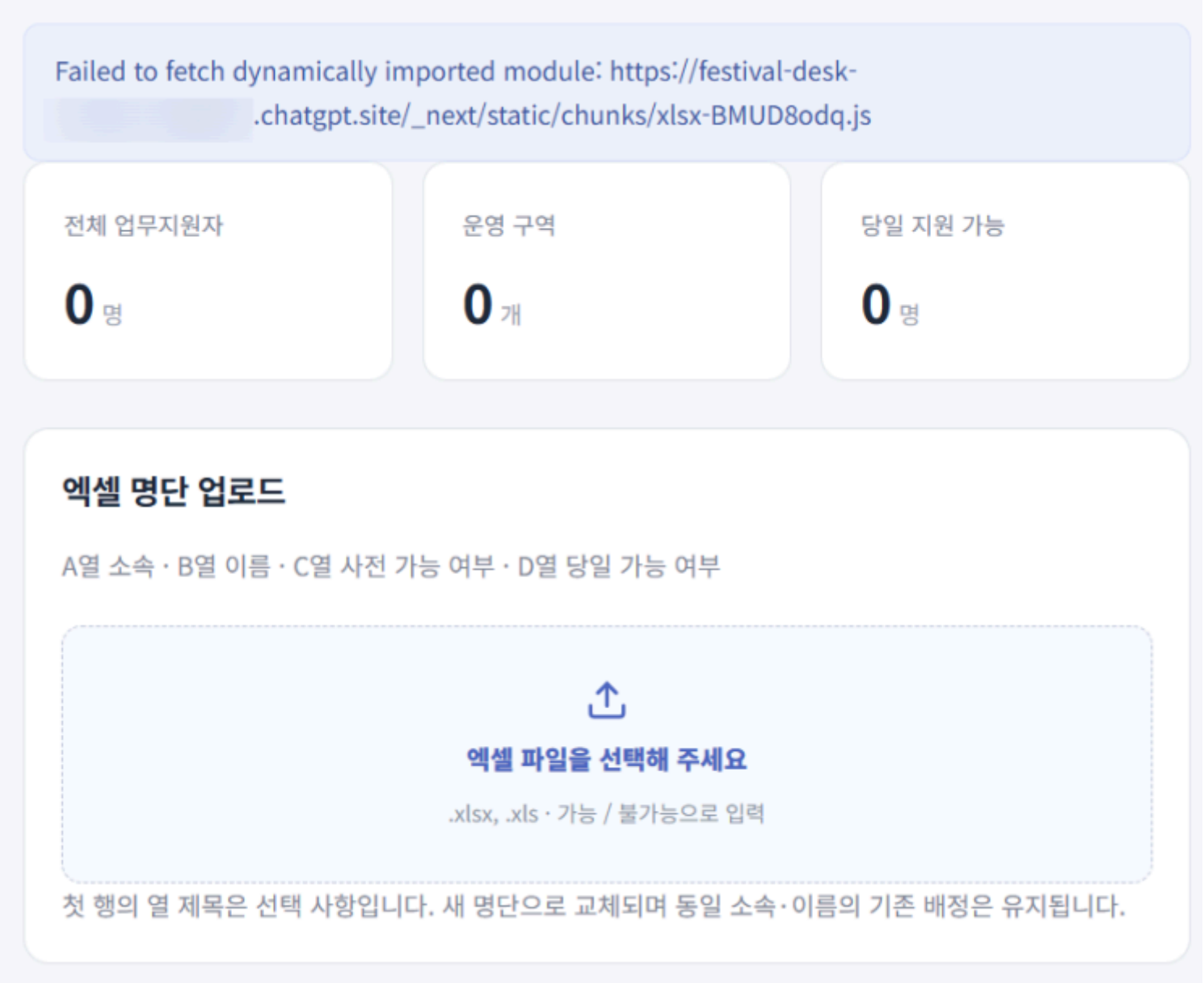
? 두 번째 난관: 엑셀 업로드가 안 돼요!

먼저 챗GPT에게 웹앱 테스트를 위해 50명 명단을 엑셀로 다운받게 해달라고 부탁했어요.

엑셀을 다운로드하고 기쁜 마음으로 냅다 업로드를 해봤죠!

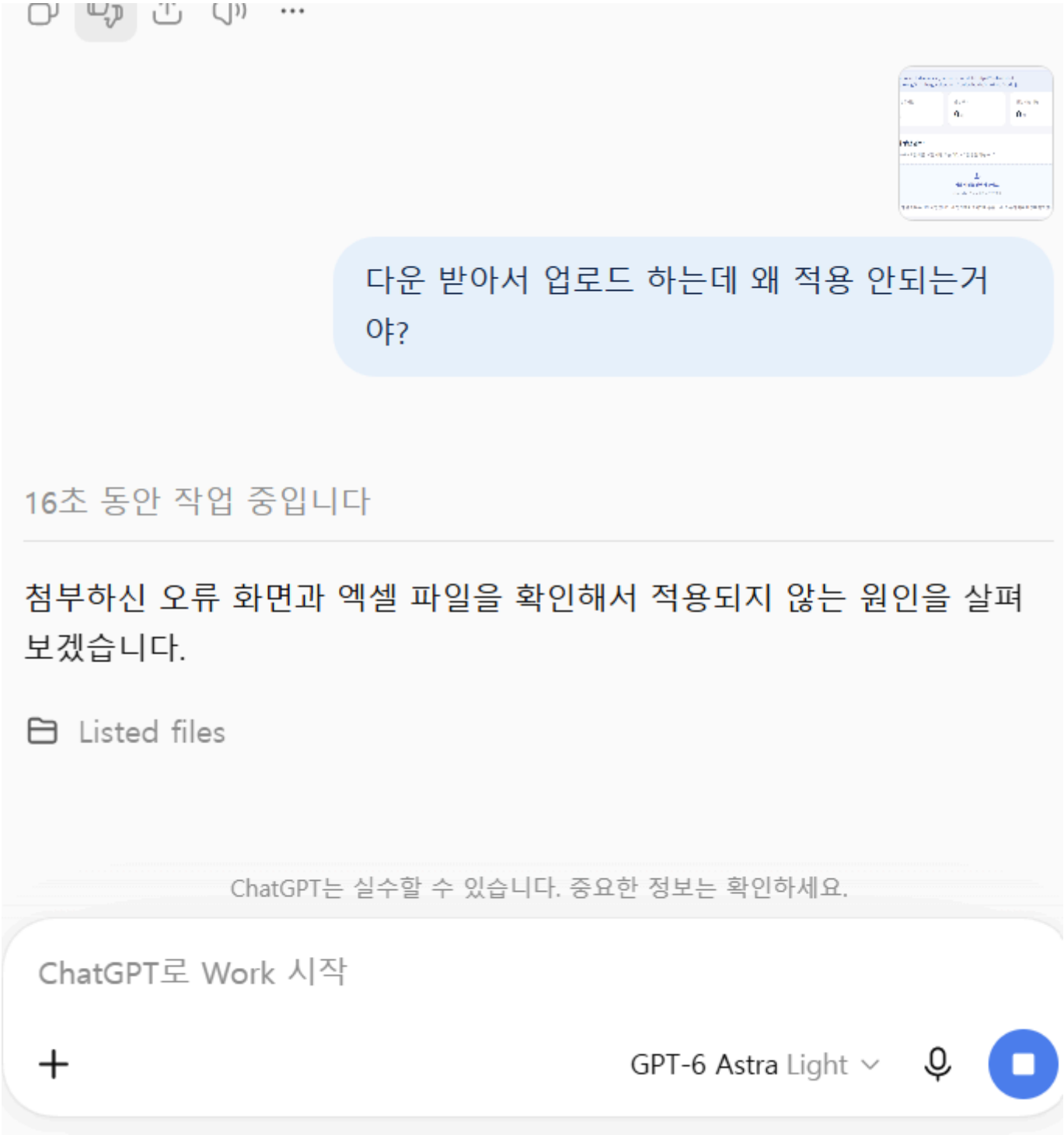
그런데... 또 오류가 발생했습니다.

명단을 읽지 못하고 인원이 0명으로 나오면서 멈춰버렸어요.



당황하지 않고, 오류 화면을 그대로 캡처해서 챗GPT에게 물어봤어요.

"다운 받아서 업로드 하는데 왜 적용 안되는거야?"라고요.



챗GPT가 정확한 원인 파악을 위해 사용 중인 프로젝트 전체를 압축해서 달라고 하더라고요.

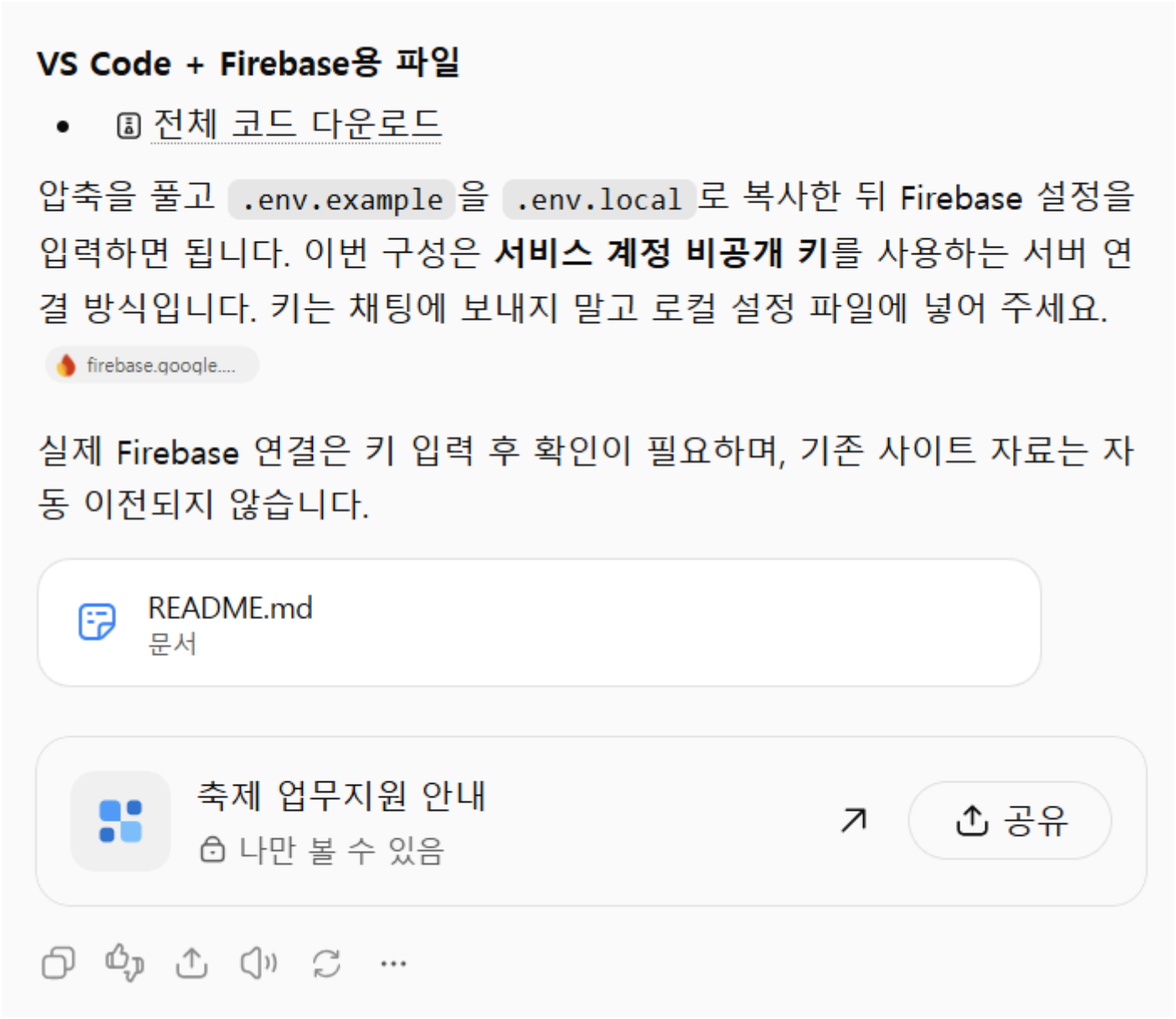
그래서 파일을 통째로 첨부해 줬더니, 원인을 고쳐서 다시 다운로드할 수 있게 **수정된 전체 코드**를 새로 짜줬어요!

화면의 오류는 **웹앱이 엑셀을 읽는 프로그램 파일(`xlsx-….js`)을 불러오지 못했다**는 뜻입니다. 명단을 읽기 전 단계에서 멈춰서 인원이 0명으로 표시되고 있어요.

제가 만든 엑셀에는 요청하신 4개 열과 50명 데이터가 들어 있는 것을 확인했습니다.

1. 사이트에서 **Ctrl + Shift + R**로 강력 새로고침한 뒤 다시 업로드해 주세요.
2. 같은 오류가 계속되면 **사이트의 엑셀 업로드 코드 또는 배포 파일을 수정해야 합니다.**

정확한 원인은 코드 확인이 필요합니다. **VS Code에서 사용 중인 프로젝트를 ZIP으로 첨부해 주시면**, 엑셀 업로드 부분을 확인해 수정하겠습니다. 첨부할 때 `node_modules`와 `.env`, Firebase 서비스 계정 키 파일은 제외해 주세요.



? 1초 용어 사전

바이브코딩(Vibe Coding)

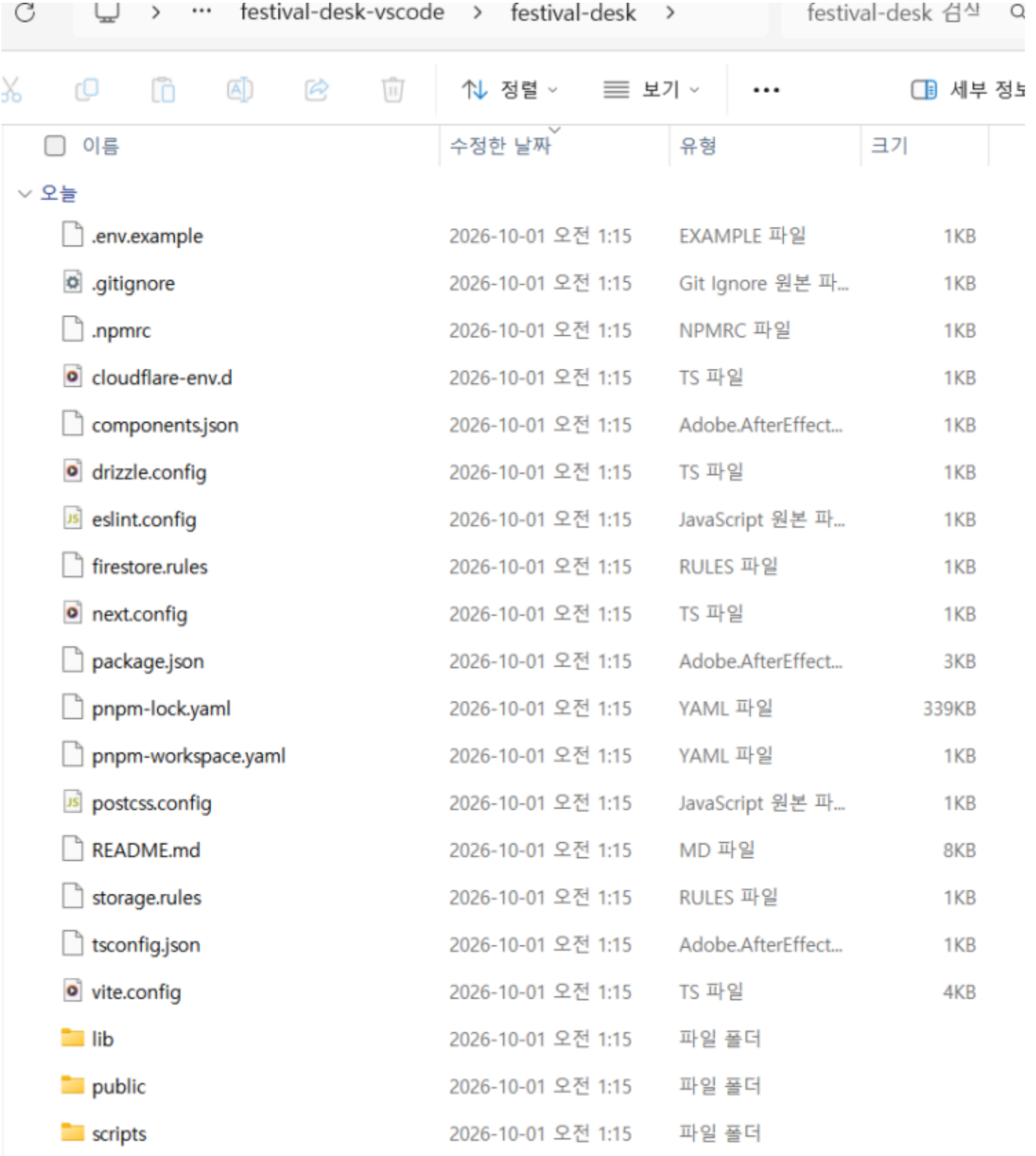
직접 복잡한 코드를 짜는 대신, 챗GPT 같은 AI와 대화하며(바이브를 맞추며)

프로그램을 뚝딱뚝딱 만들어내는 방식을 말해요.

"AI야, 이거 만들어줘!" 하고 말로 하는 요술봉 같죠!

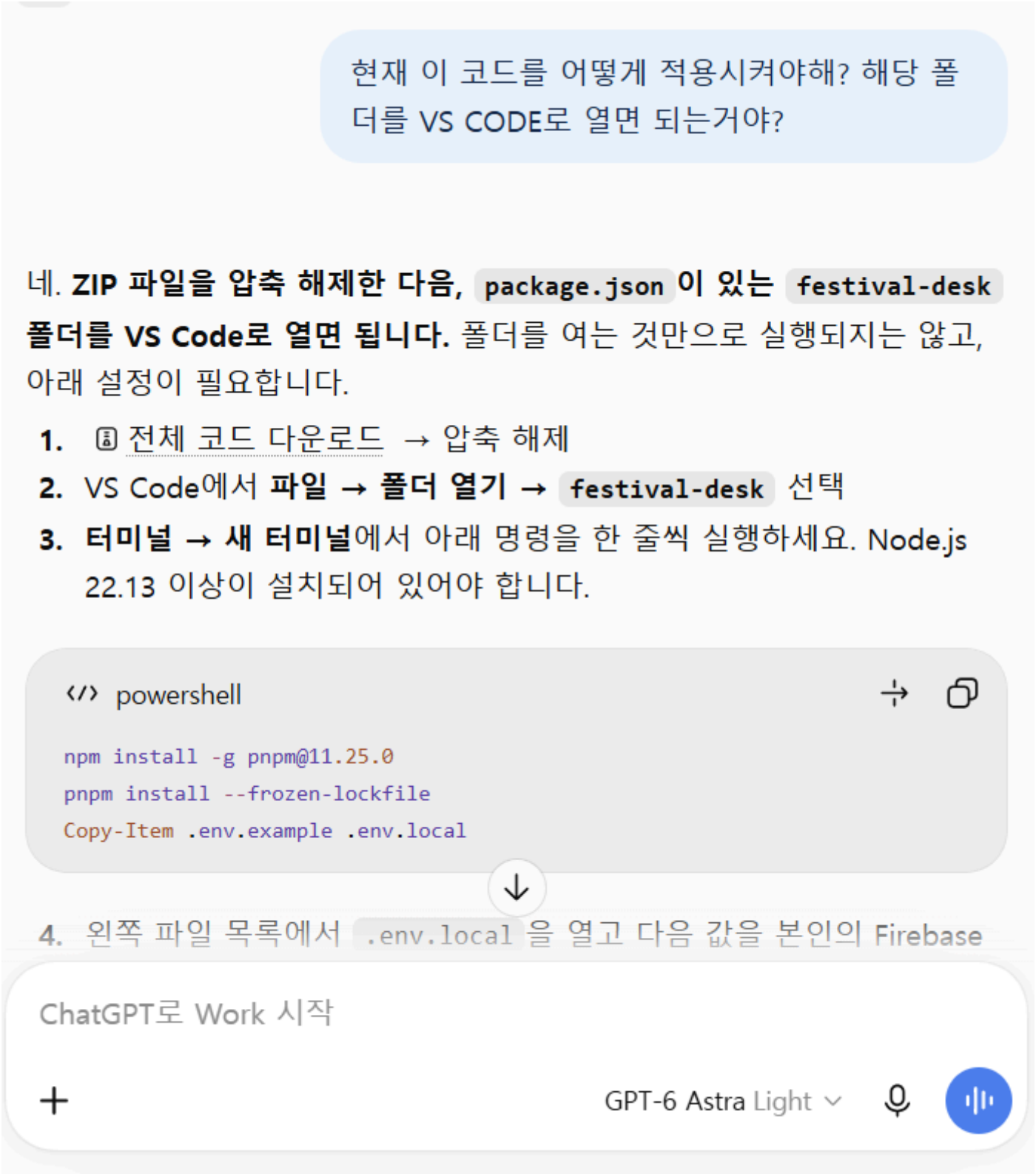
? 세 번째 난관: 빨간색 에러 글씨의 습격

새로운 코드를 다운로드해서 압축을 풀었는데, 낯선 파일이 너무 많아서 이걸 어떻게 써야 할지 또 막막해졌어요.



그래서 바로 또 물어봤죠.

"현재 이 코드를 어떻게 적용시켜야해? 해당 폴더를 VS CODE로 열면 되는거야?"



제가 이번에 바이브코딩을 하면서 확실히 느낀 점!

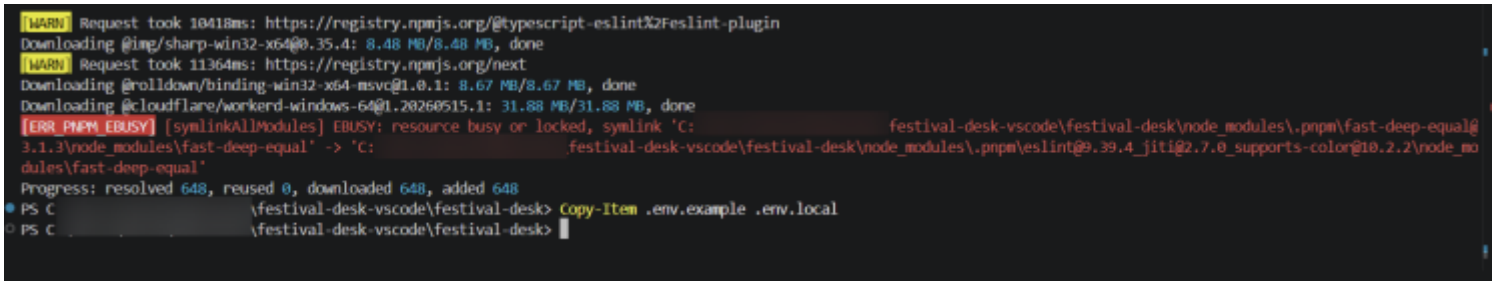
모르는 초보라면 무조건, 끊임없이 질문을 던져야 한다는 거예요.

챗GPT가 친절하게 순서대로 터미널에 입력할 명령어들을 알려줬어요.

그런데 알려준 대로 차근차근 실행하는데...

앗! 무서운 빨간색 에러 글씨(ERR_PNPM_EBUSY)가 떴어요.

> ? 1초 용어 사전
>
> 터미널 (Terminal)
>
> 마우스 클릭 대신, 글자를 쳐서 컴퓨터에게 "이것 좀 실행해 줘!"라고
>
> 직접 말로 명령을 내릴 수 있는 '검은색 명령어 입력창'이에요!



초보인 저는 코딩을 잘 모르니, 이때도 바로 화면을 캡처해서 첨부하고 "일단 3번을 하는 도중에 이런 에러가 떴어. 해결하고 넘어가고 싶어."라고 요청했습니다.

해결책은 의외로 간단했어요!

켜둔 다른 프로그램과 VS Code를 모두 닫고 다시 열어서 진행하라는 거였죠.

그대로 따라 했더니, 드디어 에러 없이 깔끔하게 설치가 완료되었습니다!



일단 3번을 하는 도중에 이런 에러가 떴어. 이것부터 해결하고 4번으로 넘어가고 싶어.

17초 동안 작업함 >

화면의 핵심 오류는 **`ERR_PNPM_EBUSY: resource busy or locked`**입니다. 설치 중 파일이 잠겨서 멈춘 상태입니다. 위쪽 노란색 `WARN`은 다운로드 지연 안내입니다.

우선 아래 순서로 진행해 주세요.

1. VS Code와 해당 프로젝트를 열어 둔 다른 프로그램을 모두 닫으세요.
2. VS Code를 다시 열고 `festival-desk` 폴더를 여세요.

ChatGPT로 Work 시작

+ GPT-6 Astra Light

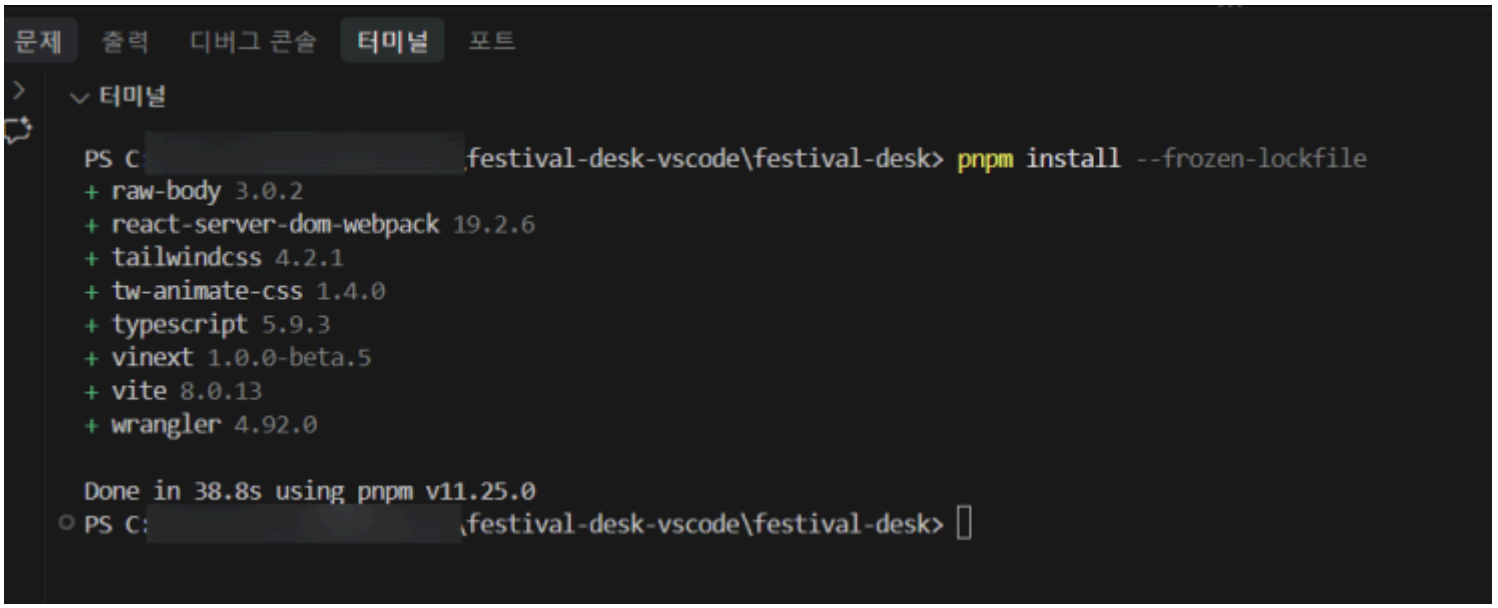
3. **터미널 → 새 터미널**에서 다음 명령만 다시 실행하세요.

```powershell
pnpm install --frozen-lockfile
```

같은 `EBUSY` 오류가 반복되면 **컴퓨터를 재시작한 뒤 위 명령을 다시 실행**해 주세요. 보안 프로그램을 끄거나 관리자 권한으로 실행할 필요는 없습니다.

마지막에 빨간 오류 없이 `Done in ...`이 나오면 설치가 완료된 것입니다.

참고로 화면에 있는 **`Copy-Item .env.example .env.local`은 오류 없이 실행됐으므로 다시 실행하지 마세요.** 설치만 완료되면 4번의 `.env.local` 설정으로 넘어가면 됩니다.



? 네 번째 난관: 스토리지 버킷은 또 뭐야?

이제 3번을 실행했으니 4번을 할 차례겠죠?

4번은 파이어베이스 설정을 교체하는 과정이었어요.

4. 왼쪽 파일 목록에서 `.env.local`을 열고 다음 값을 본인의 Firebase 설정으로 교체하세요.

항목	입력할 값
`DATA_BACKEND`	`firebase` 그대로 유지
`FIREBASE_PROJECT_ID`	프로젝트 ID
`FIREBASE_CLIENT_EMAIL`	서비스 계정 JSON의 `client_email`
`FIREBASE_PRIVATE_KEY`	서비스 계정 JSON의 `private_key`
`FIREBASE_STORAGE_BUCKET`	Storage 버킷 이름

비공개 키의 **큰따옴표와 `\n`을 유지**하세요. 상세 과정은 폴더 안의 `README.md`에 있습니다.

찾아서 하나씩 입력하는데 스토리지 버킷 이름..??

어디 있는지 알 턱이 없죠?

망설임 없이 또 챗GPT에게 물어봤답니다.



? 1초 용어 사전

스토리지 버킷(Storage Bucket)

웹사이트에서 사용하는 사진이나 엑셀 파일 같은 자료들을

잃어버리지 않게 담아두는 '인터넷상의 커다란 바구니'라고 생각하면 돼요!

챗GPT가 Firebase(파이어베이스) 콘솔 화면에서 확인하는 방법을 친절하게 알려줬어요.



알려준 주소에서 gs:// 부분만 빼고, .env.local이라는 파일 안에 설정값으로 야무지게 입력해 줬답니다.

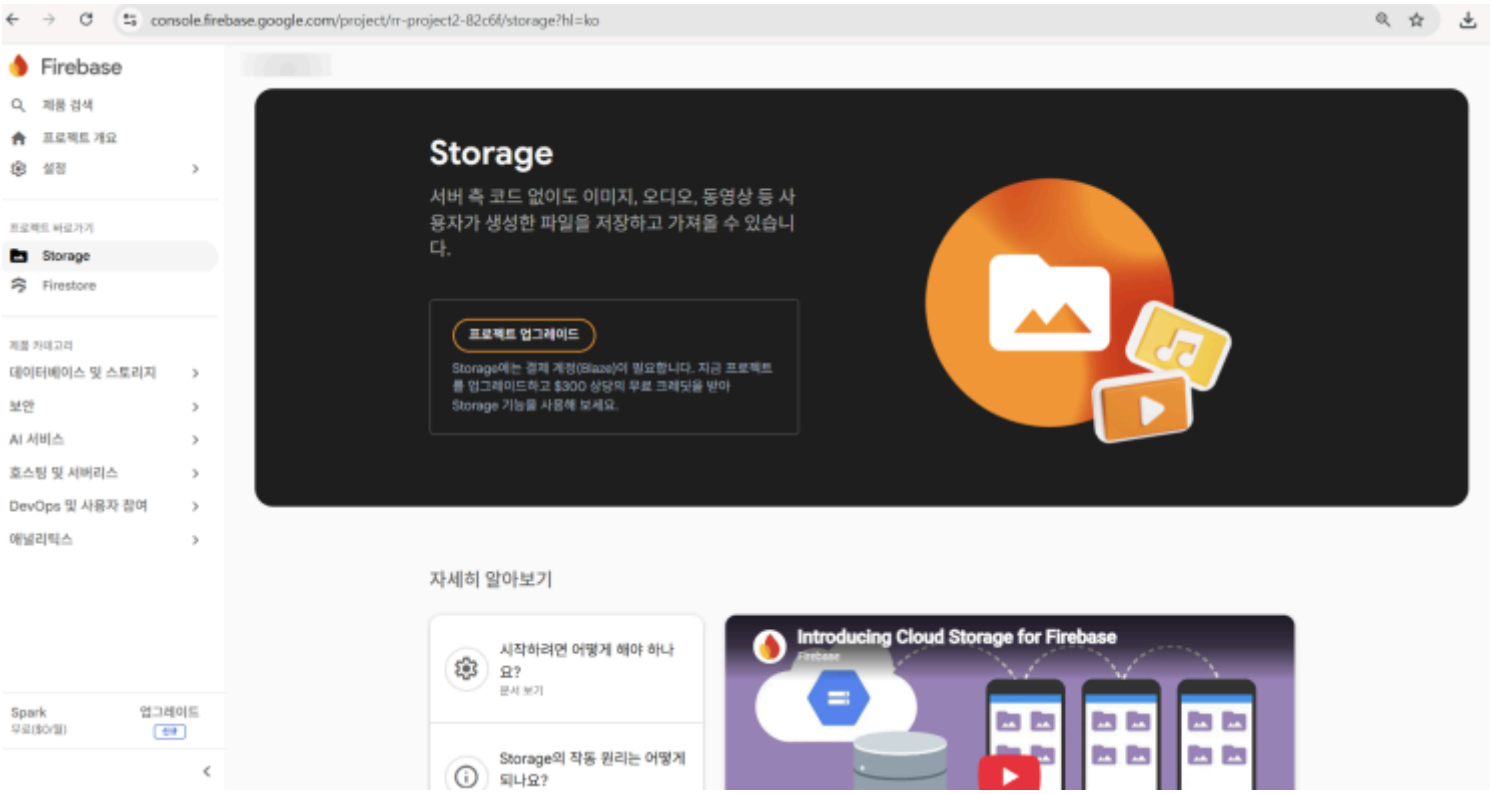
? 1초 용어 사전

.env (환경변수 파일)

남들에게 보여주면 안 되는 우리 집 비밀번호나 금고 열쇠 같은 중요한 정보들을

따로 숨겨두는 '비밀 일기장' 같은 파일이에요!

다 된 줄 알았는데 파일 화면 대신 '시작하기' 버튼이 보여서 또 질문했더니, **저장 공간을 쓰려면 요금제 연결(무료 구간 있음)이 필수**라고 해서 순서대로 업그레이드까지 마쳤습니다.



네. **현재 화면에서는 Blaze 요금제에 결제 계정을 연결해야 Storage를 사용할 수 있습니다.** 화면 중앙에도 "Storage에는 결제 계정(Blaze)이 필요합니다"라고 표시되어 있어요. 버킷 이름은 이 화면에 표시되지 않습니다.

앞서 "요금제 연결을 요구할 수 있다"고 안내했는데, 현재는 **필수**입니다. 정확하지 않게 안내해 죄송합니다. 무료 사용 구간은 있지만, 사용량에 따라 요금이 발생할 수 있습니다. firebase.google....

진행한다면 순서는 다음과 같습니다.

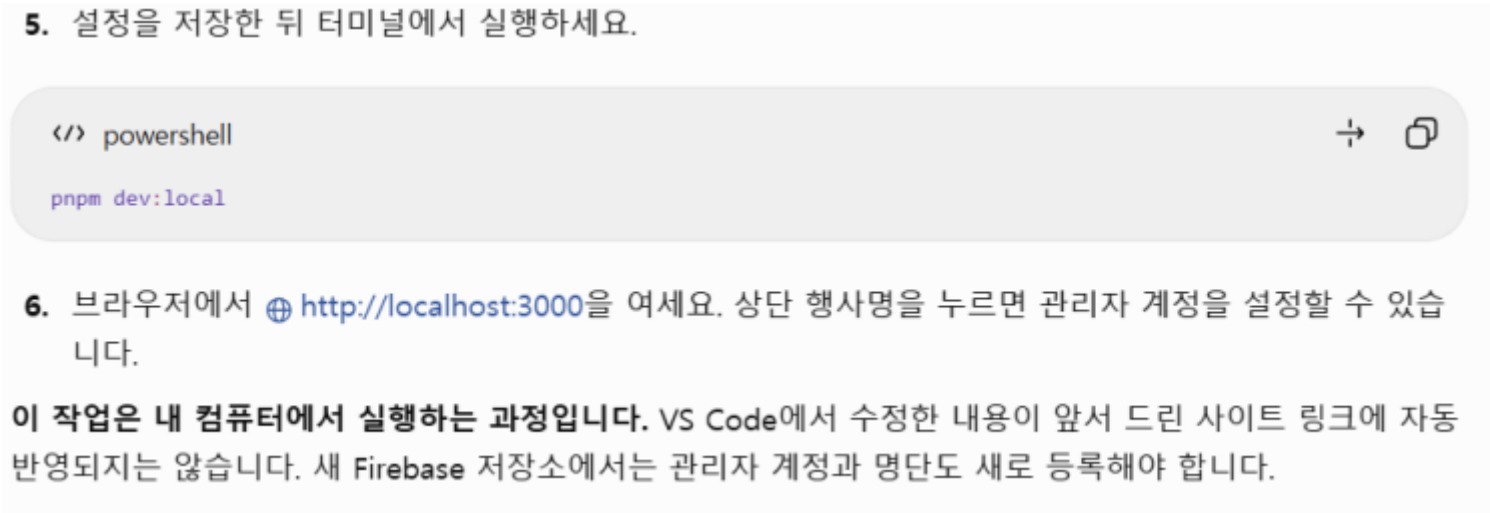
1. 화면 중앙 **프로젝트 업그레이드** 선택
2. Blaze 요금제 및 결제 계정 연결
3. Storage로 돌아와 **시작하기**로 버킷 생성
4. 생성 후 **파일 탭 상단의** `gs://...` 주소 확인
5. `.env.local`에 `gs://`를 제외한 버킷 이름 입력

결제 계정을 연결하고 싶지 않다면 지금 업그레이드하지 않아도 됩니다. 다만 현재 코드는 Firebase Storage를 사용하도록 되어 있어, 결제 연결 없이 진행하려면 이미지 저장 방식을 수정해야 합니다.

참고로 파이어베이스는 유료로 전환해도 무료구간이 먼저 소진되기 때문에 바로 요금이 발생하지는 않아요!

그리고 유료 한도액을 설정하면 해당액이 달성되기 전에 미리 메일로 몇프로 사용되고 있는지도 알려줍니다.

? 드디어 내 컴퓨터에 웹앱이 떴다!



설정을 모두 저장한 뒤 VS CODE 터미널에 pnpm dev:local이라는 마법의 주문을 입력했어요.

그리고 브라우저 주소창에 http://localhost:3000을 열어보라고 하더라고요.

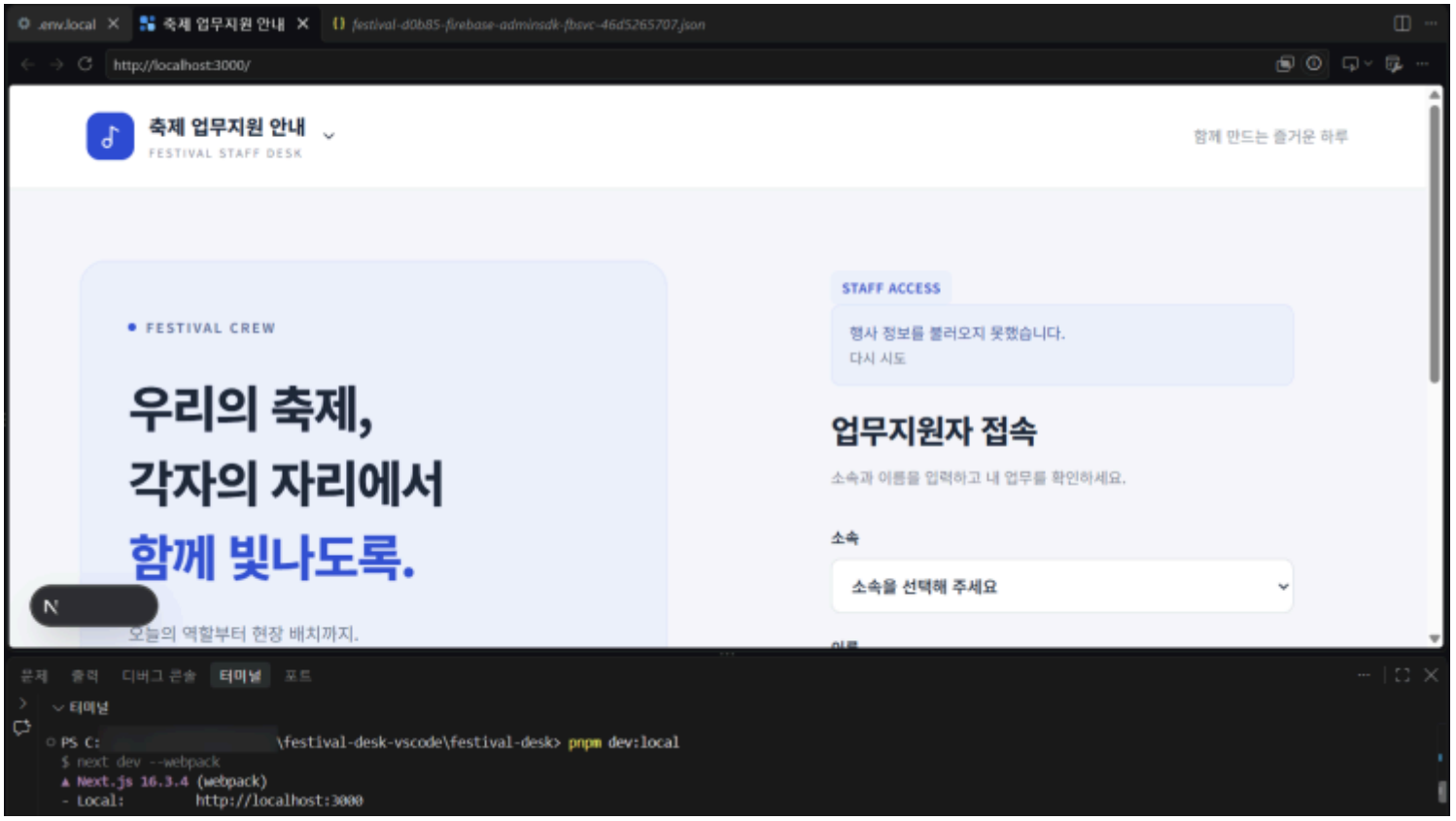
? 1초 용어 사전

localhost (로컬호스트)

진짜 인터넷에 정식으로 식당을 오픈하기 전에,

내 컴퓨터 안에서만 나홀로 조용히 요리해보고 테스트해보는 '가게 가오픈 상태'를 말해요!

화면이 예쁘게 뜨는 걸 보고 너무 신났어요!



그런데 문득, **코드를 다시 수정하고 싶으면 켜져 있는 서버를 어떻게 끄고 켜야 하는지** 모르겠더라고요.

그래서 챗GPT에게 "코드 수정 후 다시 열고 싶어. 명령어 알려줘."라고 물어봤습니다.



터미널 창을 클릭하고 Ctrl + C를 누르면 서버가 멈춘대요!

그다음 코드를 고치고 Ctrl + S로 저장한 뒤 다시 아까 명령어를 치면 끝이랍니다.

정말 쉽죠?

> 막히면 무조건 화면을 캡처해서
> AI에게 물어보자!
> 캡처는 초보의 가장 강력한 무기다.

진짜 코딩의 원리는 아직 100% 모르지만, AI가 하라는 대로 복사하고 붙여넣고 끊임없이 질문하다 보니 제 컴퓨터에 프로그램이 굴러가는 게 너무 신기하네요.

저처럼 덜컥 겁부터 나는 코딩 왕초보분들도 일단 화면 캡처부터 시작해 보세요!

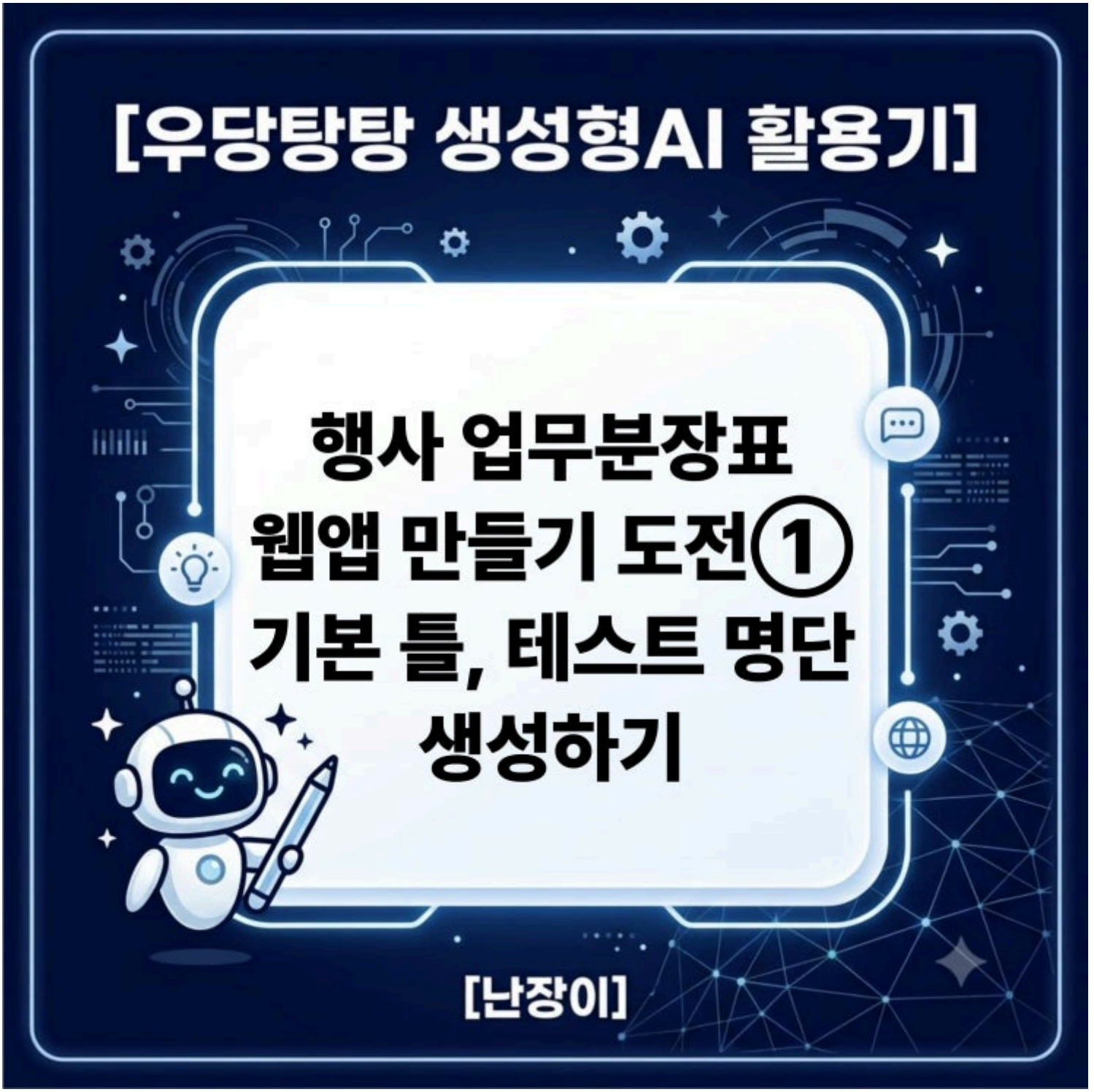
다음은 파이어베이스 서비스 계정 키의 유효성 확인과 깃허브에 비공개 저장소 만드는 방법과 또 어떤 오류가 발생했는지에 대한 내용을 작성해볼게요!

포스팅도 기대해 주세요!

[우당탕탕 생성형AI 활용기] 초보 코딩, 챗지피티 유료 결제 첫날! 행사 업무분장표 웹앱 만들기 도전기(기본 틀, 테스트 명단 생성하기)

프로그램

2026/10/05 11:30 http://blog.naver.com/talktalk2024/224430764093



안녕하세요!

드디어... **챗지피티(ChatGPT) 유료 결제**를 저지르고 온 난장이 입니다.

짝짝짝 ?

비싸서 고민만 하다가 플러스 모드를 쓸 수 있게 되어서 너무 신나요.

사실 전에 챗지피티 유료결제를 했었으나 제미나이로 코딩 하는 것을 알게 된 후에 갈아탔었는데, 최근 챗지피티로 하는 코딩이 코덱스와 결합해서 좋다는 말을 듣고 몇번 테스트를 했었거든요.

그러다가 최근 제미나이로 코딩하는데 업데이트 되서 그런건지 평소에 알던 제미나이가 아니었어요!!!

그래서 챗지피티가 좋다는 얘기를 듣고 며칠을 고민하다 드디어 유료결제를 다시 했습니다!!

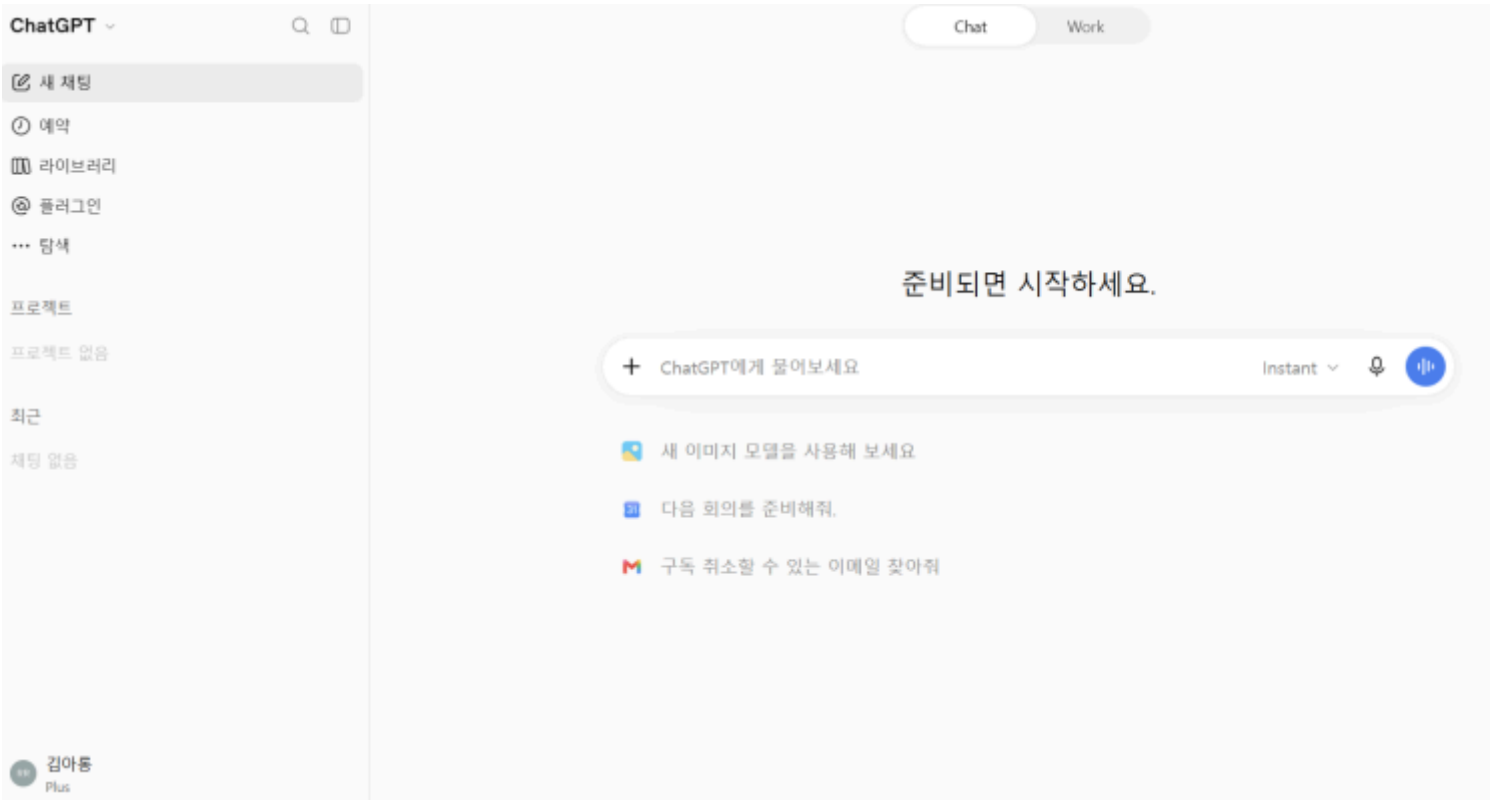
결제 기념으로 뭘 해볼까 하다가, 일상의 불편함을 하나 해결해 보기로 했어요.

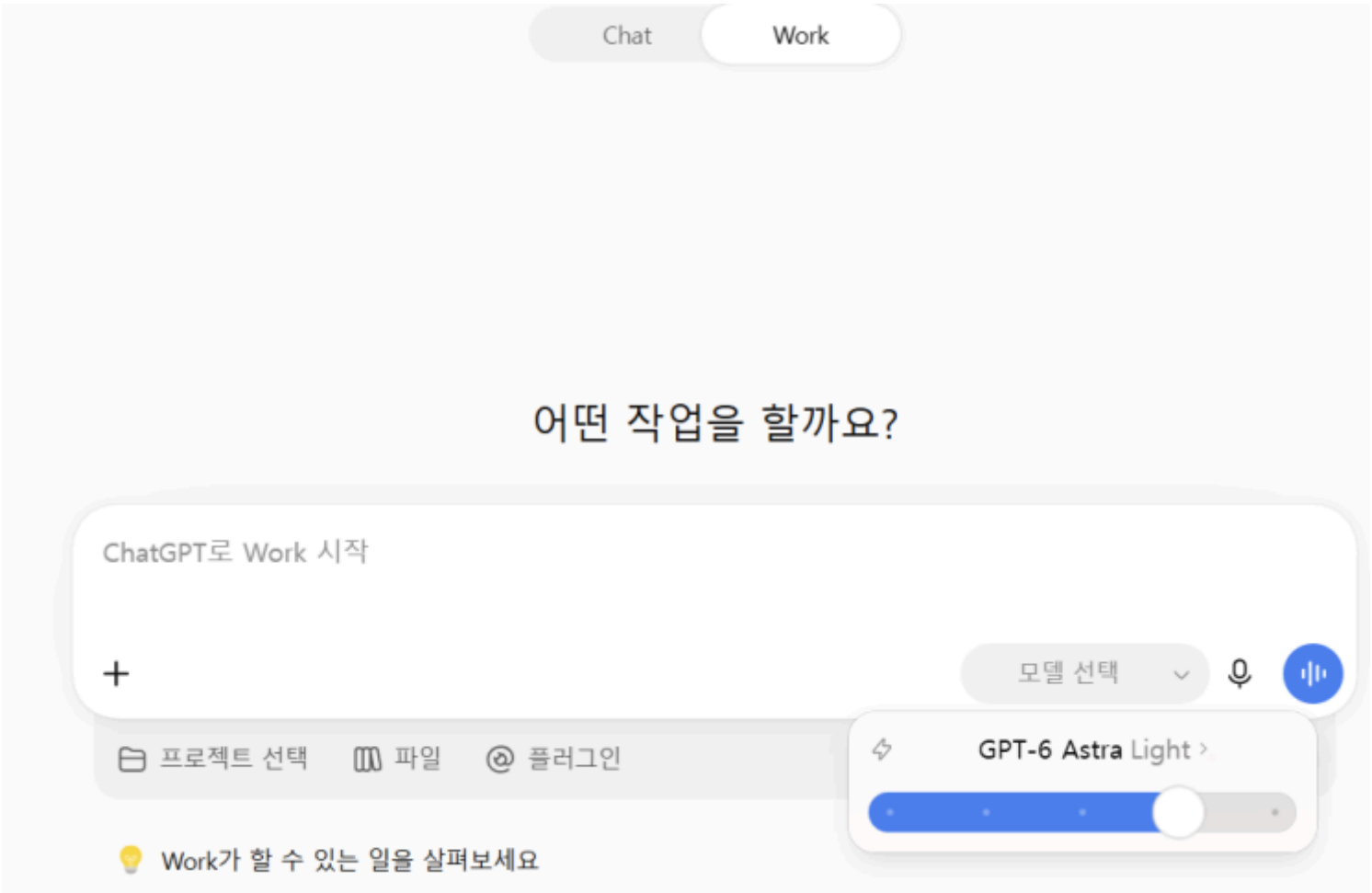
곧 큰 행사가 있는데, 업무 지원 나온 직원들이 자기 업무를 스마트폰으로 편하게 확인할 수 있는 '업무분장표 프로그램'을 만들어보기로 했답니다.

유료 결제한 챗지피티는 어떻게 생성해줄 지 너무나 기대되었어요!!

챗지피티 플러스, 첫 만남과 프롬프트 작성!

우선 새롭게 만난 **GPT-6 Astra 라이트 모드**로 설정을 맞췄어요.





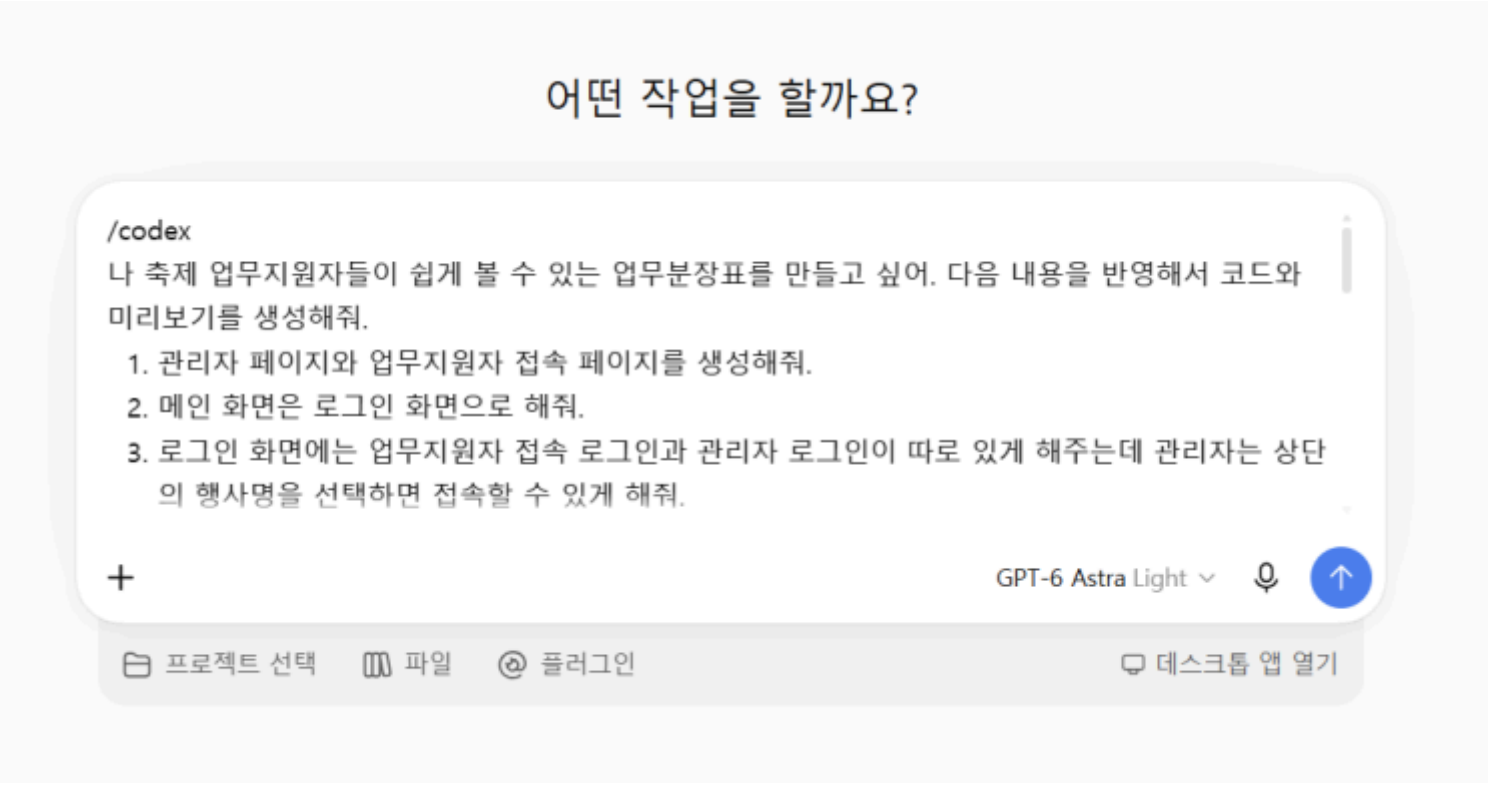
참고해 보시면 화면이 정말 깔끔하죠?

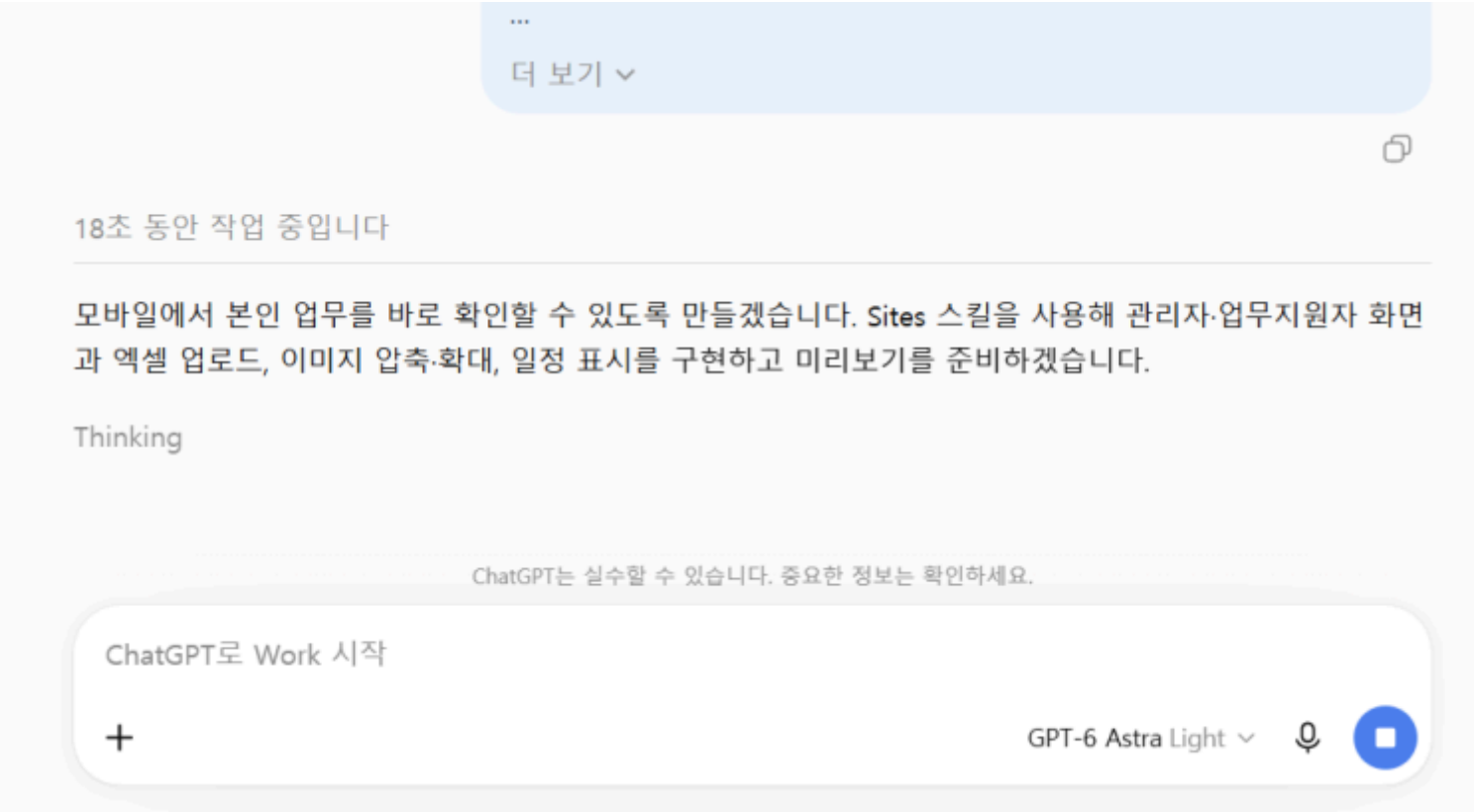
어디서 본 건 있어서 대화창 첫 줄에 멋지게 /codex라고 적어봤어요.

(사실 이게 진짜 반영된 건지는 초보라 저도 잘 모르겠네요 ?)

제가 구상한 프로그램을 위해 무려 15개의 조건을 조목조목 적어서 심부름 쪽지를 날렸어요.

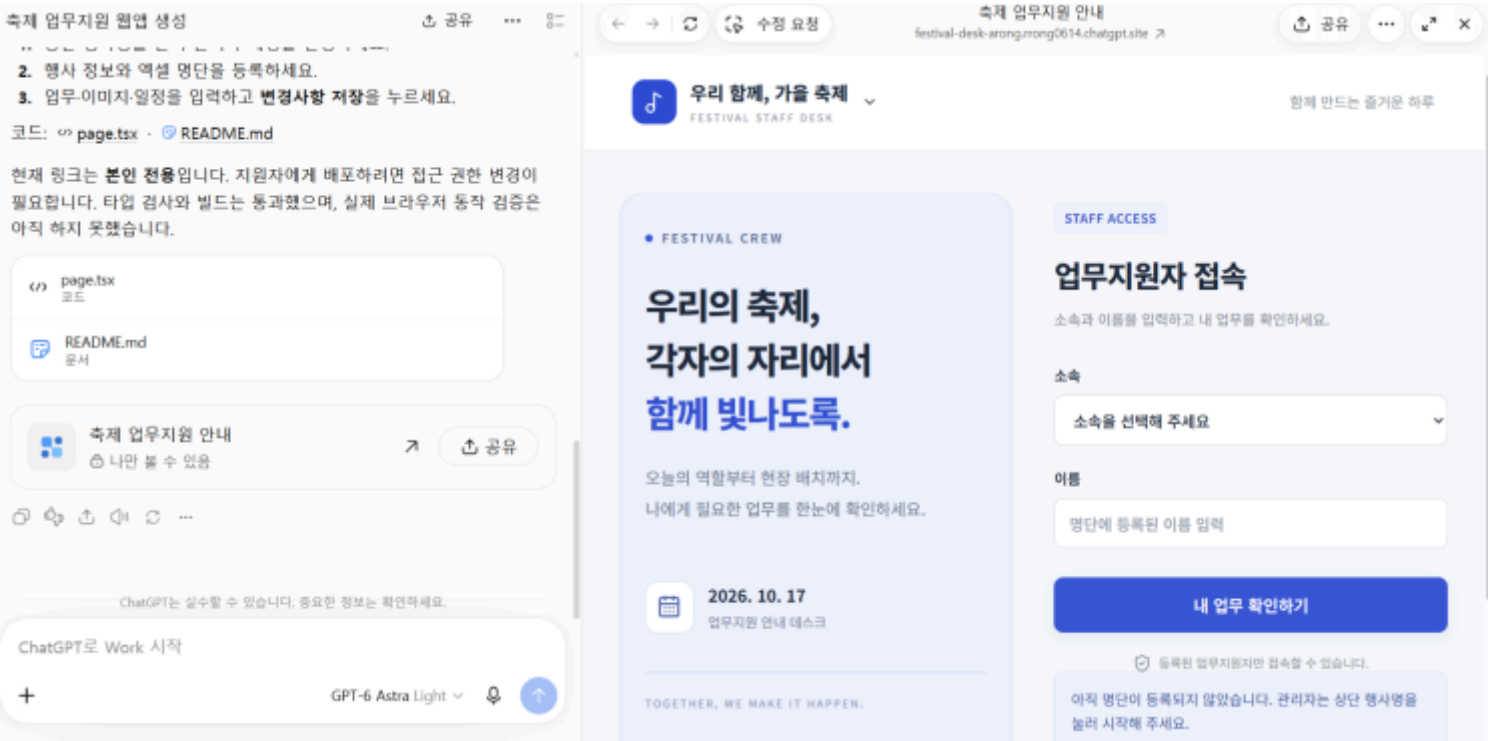
"/codex
나 축제 업무지원자들이 쉽게 볼 수 있는 업무분장표를 만들고 싶어.
다음 내용을 반영해서 코드와 미리보기를 생성해줘.
관리자 페이지와 업무지원자 접속 페이지를 생성해줘.
메인 화면은 로그인 화면으로 해줘..."





그랬더니 AI가 뚝딱뚝딱 생각하더니, 약 10분 38초 만에 초안을 딱! 만들어주지 뭐예요?

제미나이 쓸 때랑은 다르게 화면 왼쪽에 코드, 오른쪽에 **웹앱 미리보기가 바로 짠~** 하고 나타나서 정말 신기했어요!



하지만 코드가 촤르륵 뜨니까 살짝 '어려운 거 아닌가?' 덜컥 겁도 났지만, 뭐든 처음 파악해 보는 게 중요하니까요!

순탄치 않은 에러 해결기

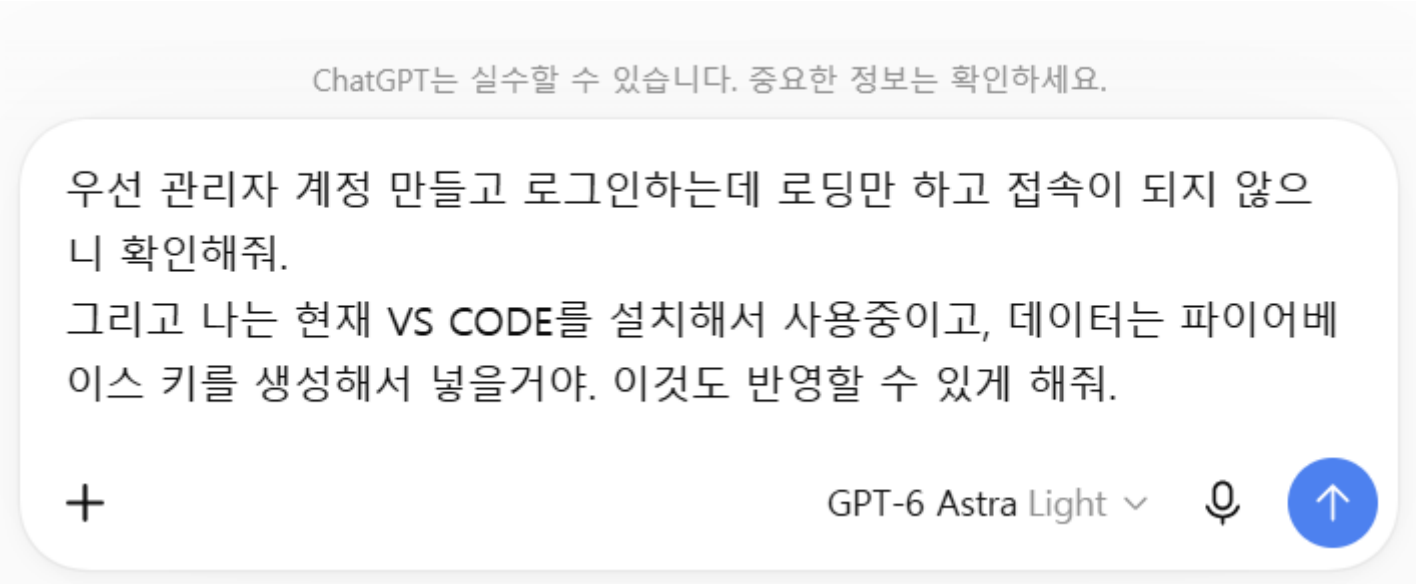
하지만 초보의 길에 한 번에 성공이란 없죠 ㅠㅠ

여기서 **첫 번째 오류가 발생**했어요.

야심 차게 관리자 계정을 생성하고 로그인을 누르는데... 화면이 뱅글뱅글 로딩만 되고 접속이 되질 않는 거예요.

나머지 테스트를 할 수가 없어서 순간 땀이 삐질 났어요.
그래서 다시 챗지피티에게 오류를 확인해달라고 부탁했답니다!

이왕 고치는 김에, 제가 쓰려고 준비해 둔 VS CODE와 **파이어베이스**도 함께 반영해 달라고 덧붙였어요.

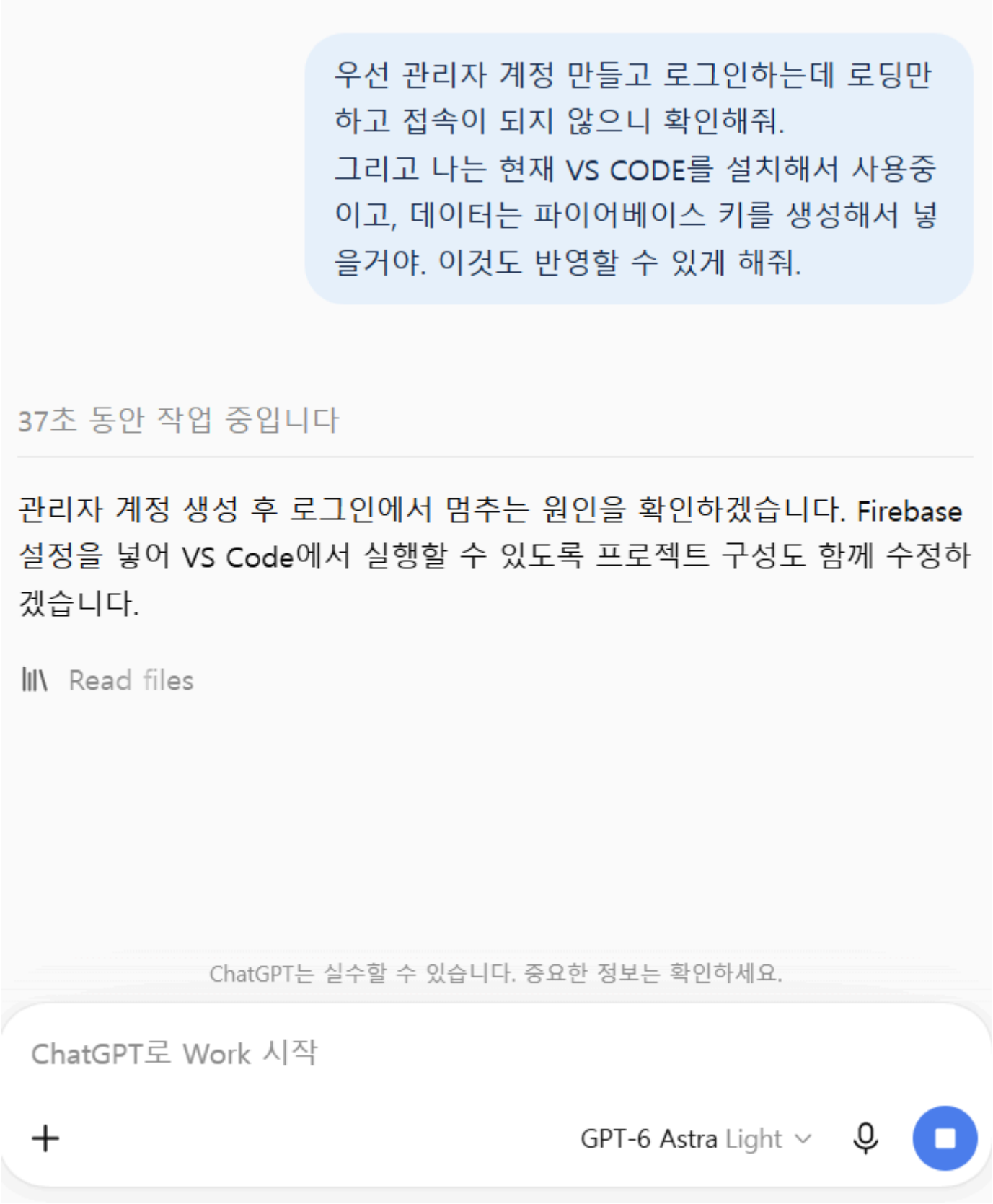


? 1초 용어 사전
VS CODE
코딩을 할 때 쓰는 '만능 스케치북 프로그램'이에요!
글을 쓸 때 한글이나 워드를 켜듯,
코드를 작성할 땐 이 스케치북을 열고 글씨(코드)를 쓴답니다.

? 1초 용어 사전
파이어베이스(Firebase)
구글에서 빌려주는 '안전한 온라인 창고'예요!
우리가 앱에 직원 명단을 올리거나 로그인 정보를 저장하면,
내 컴퓨터가 아니라 이 창고에 차곡차곡 보관되는 거랍니다.

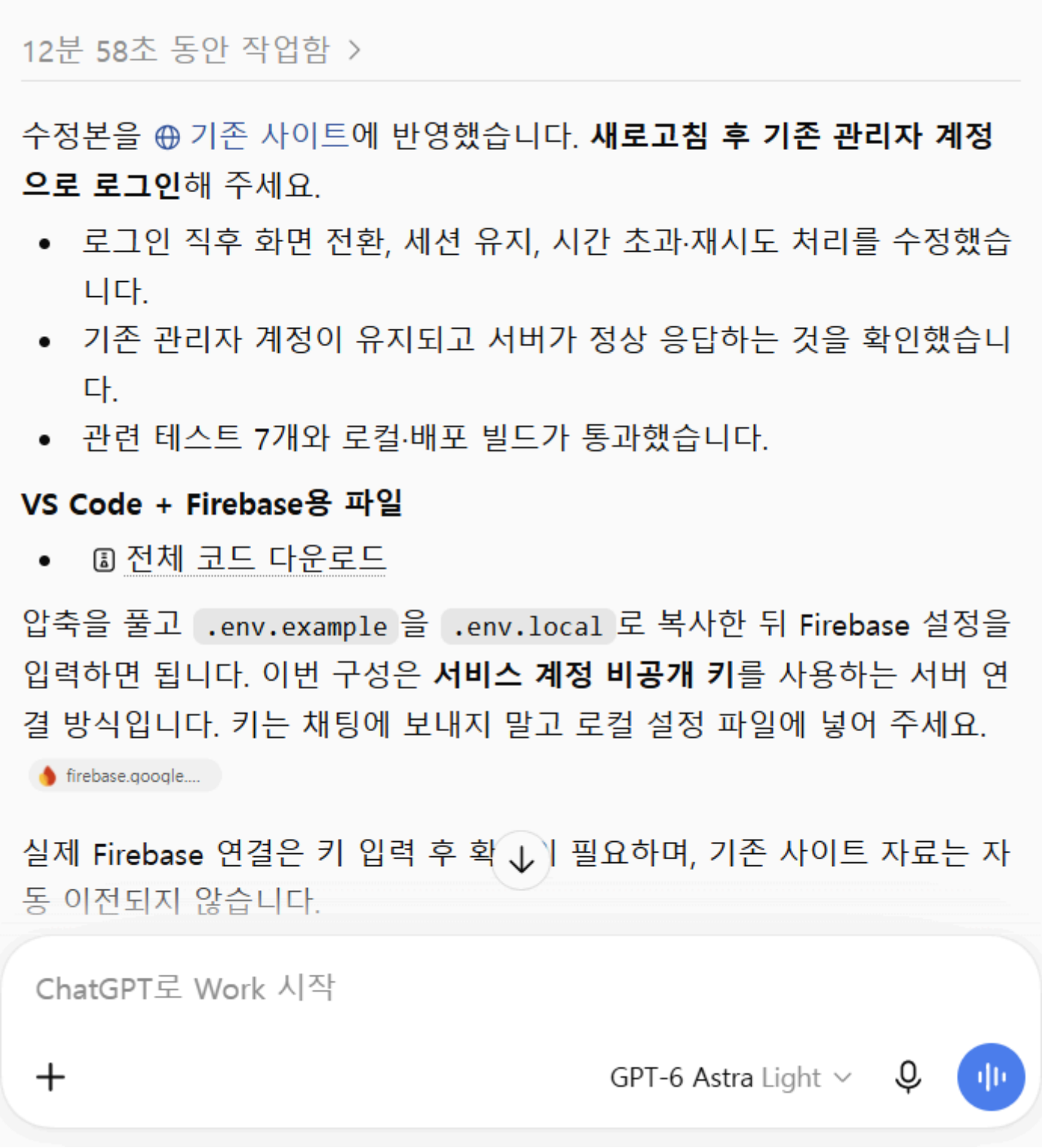
이번 작업은 AI에게도 복잡했는지 10분이 넘게 걸리더라고요.

기다리는 동안 혹시라도 이 화면을 벗어나면 작업이 멈출까 봐 쫄아서 ㅋㅋㅋ 조심스럽게 새 창을 띄우고 다른 사이트 구경을 하면서 기다렸답니다.



오류 해결! 그리고 다음 단계를 위한 꼼수

다행히 얼마 뒤 수정본이 완벽하게 반영되었어요!



VS Code에 적용하는 방법도 자세히 알려줬는데...

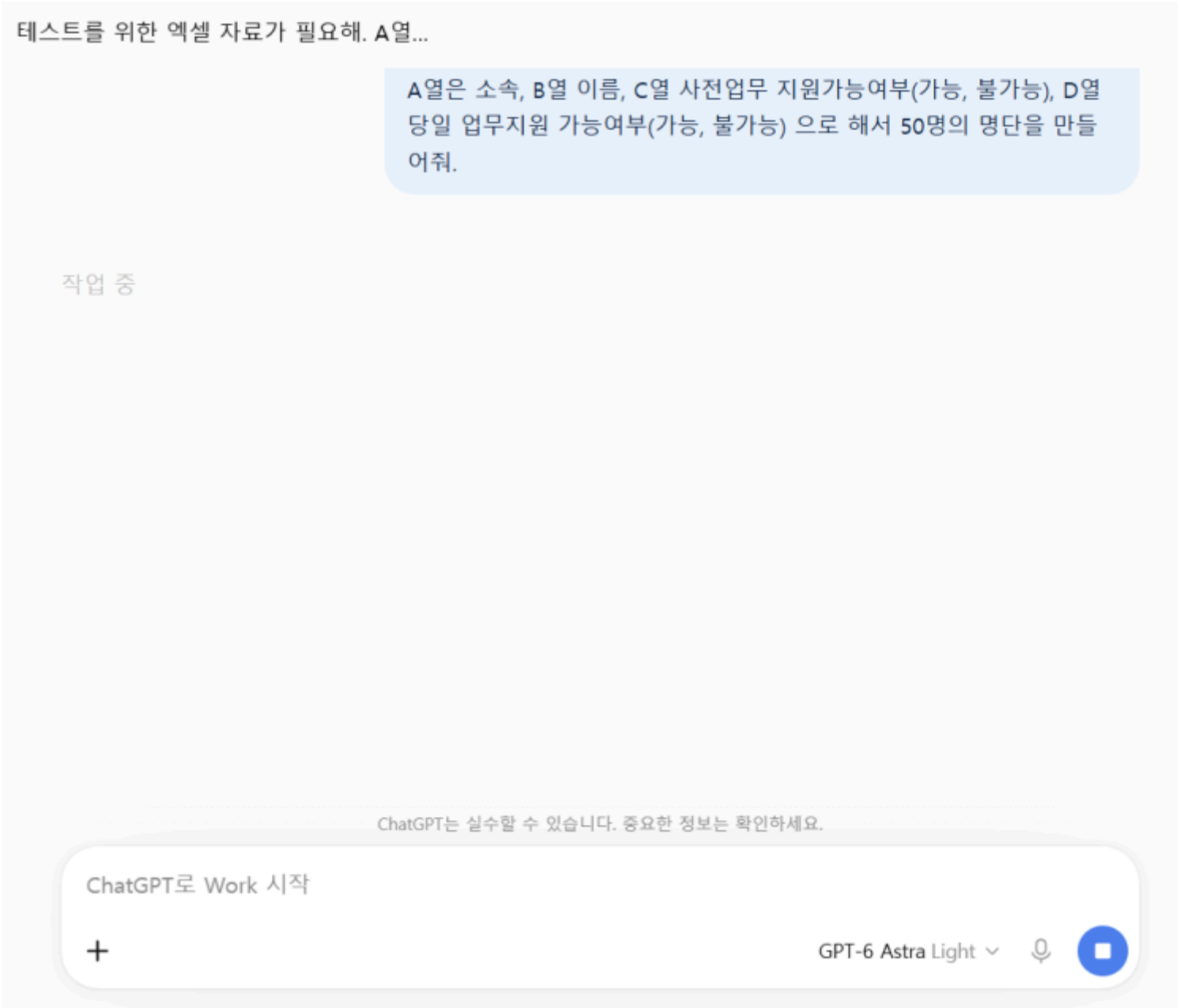
솔직히 제미나이로 할 때랑 또 느낌이 달라서 "앗, 이건 너무 어렵다! 이따가 하나씩 아주 자세히 다시 물어봐야지~" 하고 살짝 미뤄뒀어요. ?

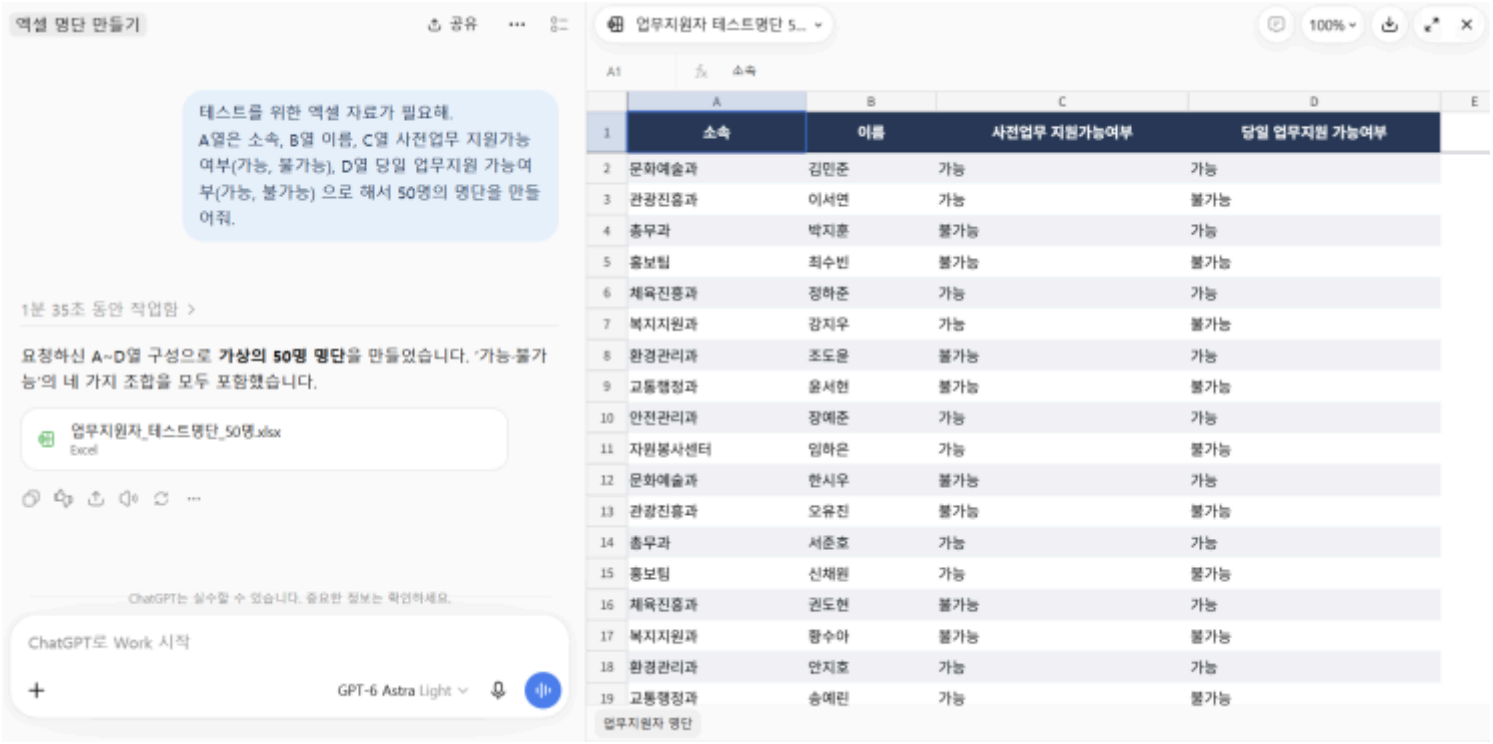
이제 관리자 계정이 잘 작동하니까, 실제 직원 명단을 올려서 테스트해 볼 차례예요!

그런데 진짜 직원 명단은 회사 컴퓨터에 있고, 지금 제 노트북엔 없지 뭐예요?

그래서 똑똑한 AI에게 **"테스트용 명단 엑셀로 50명만 만들어줘!"** 하고 편하게 부탁했어요.

(이때, 새 창을 열어서 새 채팅으로 요청했어요!)





	A	B	C	D
1	소속	이름	사전업무 지원가능여부	당일 업무지원 가능여부
2	문화예술과	김민준	가능	가능
3	관광진흥과	이서연	가능	불가능
4	총무과	박지훈	불가능	가능
5	홍보팀	최수빈	불가능	불가능
6	체육진흥과	정하준	가능	가능
7	복지지원과	강지우	가능	불가능
8	환경관리과	조도윤	불가능	가능
9	교통행정과	윤서현	불가능	불가능
10	안전관리과	장예준	가능	가능
11	자원봉사센터	임하은	가능	불가능
12	문화예술과	한시우	불가능	가능
13	관광진흥과	오유진	불가능	불가능
14	총무과	서준호	가능	가능
15	홍보팀	신채원	가능	불가능
16	체육진흥과	권도현	불가능	가능
17	복지지원과	황수아	불가능	불가능
18	환경관리과	안지호	가능	가능
19	교통행정과	송예린	가능	불가능

돈 쓴 보람이 팍팍 느껴지는 챗지피티 플러스 첫 경험이었어요!

오늘은 여기까지 뼈대를 만들고 에러를 잡는 것만으로도 완전 뻗어버렸지만, 너무 뿌듯한 하루였답니다.

엑셀 명단을 넣어서 테스트한 내용과, 그 어려운 VS Code에 한 땀 한 땀 적용해 보는 과정은 다음에 이어서 포스팅해 볼게요!

저 같은 코딩 왕초보도 클릭 몇 번에 그럴싸한 프로그램 초안을 만들어낼 수 있으니, 여러분도 겁먹지 말고 꼭 한 번 챗지피티와 대화해 보세요! ?

오늘 제 우당탕탕 도전기가 누군가에게 작은 용기가 되길 바라며, 공감과 댓글 많이 부탁드려요!

내일 또 만나요