KinoFlare

비개발자를 위한 무료 가이드

출시 버튼 누르기 전 20분만 주세요

AI로 만든 서비스, 사고 나기 전에 막는 23가지

안 하면 돈이 나가거나 데이터가 사라지는 것만 남겼습니다
코드는 몰라도 됩니다. 복사해서 AI에 붙여넣기만 하세요

KinoFlare · kinoflare.com

원래 41개였습니다

41개짜리 문서는 아무도 끝까지 읽지 않습니다. 그래서 **안 하면 실제로 사고가 나는 항목만** 스물세 개로 추렸습니다.

바이브코딩의 함정은 하나입니다. **화면에 잘 나오니까 다 된 줄 안다**는 것입니다. 물 밑에 남은 문제는 **사용자가 생기고 나서야** 터집니다. 하필 서비스가 잘되기 시작한 날에 말입니다.

항목마다 **이렇게 됩니다**(무슨 손해가 나는지) → **이런 일이 납니다**(왜 그렇게 되는지) → **확인**(어디를 보는지) → **프롬프트**(한 줄 복사) 순서입니다.

> 급하시면 굵은 빨간 줄만 스물세 개 읽으세요. **뜨끔한 게 있으면 그것부터** 하시면 됩니다.

이 문서를 만든 곳

KinoFlare · 졸업하는 외주

티켓팅 플랫폼, 페스티벌 가이드, 마케팅 자동화, 개발자 툴을 직접 만들어 직접 운영합니다. 여기 적힌 사고들은 대부분 저희가 겪었거나 남의 것을 고쳐드린 것입니다.

읽다가 **"이건 도저히 내가 못 하겠는데"** 싶은 항목이 나오면, 그 항목이 바로 저희가 하는 일입니다.

상담: 카카오톡 채널 · kinoflare.com/ko/launch

1부. 열쇠 — 5가지

돈이 나가는 열쇠가 코드에 그대로 적혀 있지 않은가

이렇게 됩니다

내 카드로 수백만 원이 결제됩니다

🔥 **이런 일이 납니다**

"챗봇 기능 만들어줘"라고 시키면 AI는 **키를 코드에 그대로 적어서** 줍니다. 그래야 바로 돌아가기 때문이고, 실제로 화면에서는 완벽하게 작동합니다.

그 상태로 깃허브 같은 코드 보관소에 올리면 몇 분 만에 남의 코드를 자동으로 뒤지는 프로그램이 그 키를 가져갑니다. API 키는 **내 이름으로 결제되는 신용카드**와 같아서, 자고 일어나면 결제 알림이 수십 건 쌓여 있게 됩니다.

개발 커뮤니티에서 심심찮게 보이는 글

"토이 프로젝트라 그냥 올렸는데 아침에 메일 200통 와 있었습니다. 사용량 보니까 제가 안 쓴 시간대에 계속 돌아갔더라고요."

확인 프로젝트 폴더에서 `sk-` `key` `secret` `password` 를 검색해 보세요. 코드 파일 안에 실제 값이 따옴표로 적혀 있다면 위험한 상태이며, 키가 있어야 할 곳은 `.env` 파일 하나뿐입니다.

📋 **이 한 줄만 복사해서 AI에 붙여넣으세요**

```
코드에 직접 적혀 있는 키·비밀번호를 전부 찾아 .env로 옮기고, 바꾼 파일 목록 알려줘.
```

열쇠를 모아둔 파일이 남들에게 공개되지 않는가

이렇게 됩니다

지운 줄 알았던 키가 계속 살아 있습니다

🔥 **이런 일이 납니다**

1번을 하고 나면 열쇠들이 `.env` 라는 **열쇠 보관 파일** 하나에 모입니다. 여기까지는 잘한 것입니다.

문제는 코드를 깃허브에 올릴 때 **이 파일까지 같이 올라간다**는 점입니다. 그래서 "이 파일은 올리지 마"라고 적어 두는 목록이 따로 있는데, 그 목록 파일 이름이 `.gitignore` 입니다. AI는 시키지 않으면 여기에 추가해 주지 않습니다.

더 곤란한 점은 **한 번 올라가면 나중에 지워도 소용이 없다**는 것입니다. 깃허브는 예전 기록을 계속 남기기 때문에, 열쇠를 새로 발급받기 전까지는 그대로 살아 있습니다.

확인 프로젝트 폴더에서 `.gitignore` 라는 파일을 열어 보세요. 그 안에 `.env` 라는 줄이 한 줄 있어야 합니다. 없으면 아래 프롬프트를 쓰세요. 이미 한 번 올라간 적이 있다면 **코드에서 지우는 것보다 열쇠를 새로 발급받는 일이 먼저입니다.**

📋 이 한 줄만 복사해서 AI에 붙여넣으세요

```
.env가 .gitignore에 있는지, 예전에 올라간 적 있는지 확인하고 조치 알려줘.
```

비밀 열쇠가 웹사이트에서 들여다보이지 않는가

이렇게 됩니다

아무나 내 AI 요금을 대신 씁니다

🔥 **이런 일이 납니다**

"키를 못 읽는다는 에러가 난다"고 하면 AI가 자주 제안하는 해결책이 있습니다. **변수 이름 앞에 `NEXT_PUBLIC_` 을 붙이는 것**인데, 에러가 사라지기 때문에 문제가 풀린 것처럼 보입니다.

하지만 그 접두어의 뜻은 **"이 값을 브라우저에 공개하겠다"**입니다. 누구나 F12만 누르면 OpenAI 키를 볼 수 있게 됩니다. 에러는 사라졌고 화면도 멀쩡하기 때문에, 그대로 출시되는 경우가 많습니다.

🌐 **브라우저 — 누구나 열어볼 수 있는 곳**

화면, 디자인, 공개해도 되는 키(지도 표시용 등). NEXT_PUBLIC_·VITE_ 가 붙은 값은 전부 여기 있습니다.

🔒 **서버 — 나만 볼 수 있는 곳**

OpenAI 키, 결제 시크릿, 데이터베이스 관리자 키. 여기서만 써야 하는 값들입니다.

확인 `.env` 를 열어 `NEXT_PUBLIC_` 이나 `VITE_` 로 시작하는 변수를 살펴보세요. 그 접두어는 "브라우저에 공개하겠다"는 뜻이므로, 비밀이어야 하는 키가 섞여 있으면 안 됩니다.

📋 **이 한 줄만 복사해서 AI에 붙여넣으세요**

```
브라우저에 공개되는 설정값을 판정하고, 공개되면 안 되는 건 서버로 옮겨줘.
```

관리자 비밀번호가 아직 `admin / 1234` 인가

이렇게 됩니다

관리자 화면이 통째로 넘어갑니다

🔥 **이런 일이 납니다**

"관리자 계정 하나 만들어줘"라고 하면 AI는 `admin / admin1234` 같은 계정을 만들어 줍니다. 테스트용이기 때문이고, 나중에 바꾸라는 안내도 대개 하지 않습니다.

공교롭게도 자동으로 침입을 시도하는 프로그램이 **가장 먼저 자동으로** 시도하는 조합이 정확히 그것입니다. 관리자 계정이 한 번 열리면 그 뒤의 방어는 의미가 없습니다.

확인 지금 바꾸시면 5초면 끝납니다. 그리고 회원가입 화면에서 아무나 관리자 등급을 받을 수 없는지도 함께 확인하세요.

회원 비밀번호가 읽히는 채로 저장되고 있지 않은가

이렇게 됩니다

회원 전원의 비밀번호가 함께 샙니다

🔥 **이런 일이 납니다**

"회원가입 기능 만들어줘"라고만 하면 AI는 가장 단순한 코드를 내놓습니다. 입력받은 비밀번호를 **그대로 데이터베이스에 저장**하는 방식인데, 동작 자체는 완벽합니다.

그러나 나중에 테이블을 열어 보면 `hong1234` 가 그대로 읽힙니다. 유출이 한 번 일어나면 회원 전원의 비밀번호가 함께 새고, 사람들은 여러 사이트에 같은 비밀번호를 쓰기 때문에 피해가 서비스 밖으로 번집니다.

확인 로그인을 직접 만드셨다면 아래 프롬프트로 점검하세요. Supabase나 Firebase, 소셜 로그인을 쓰신다면 알아서 처리되므로 넘어가셔도 됩니다.

📋 **이 한 줄만 복사해서 AI에 붙여넣으세요**

```
비밀번호가 평문 저장되는지 확인하고, 그렇다면 알아볼 수 없는 형태로 바꾸고 로그인 전반을 점검해줘.
```

2부. 문단속 — 4가지

관리자 화면에 주소만 알면 들어갈 수 있지 않은가

이렇게 됩니다

남의 주문이 조회되고 남의 글이 지워집니다

🔥 **이런 일이 납니다**

"관리자만 이 메뉴가 보이게 해줘"라고 시키면 AI는 시킨 대로 정확히 처리합니다. **화면에서 보이지 않게** 만드는 것입니다.

문제는 그것이 보안이 아니라는 데 있습니다. 주소창에 직접 입력하면 그대로 들어가집니다. 커튼을 친 것이지 문을 잠근 것이 아니기 때문입니다. 같은 원리로 다른 사람의 주문 내역이 조회되고, 다른 사람의 글이 삭제되기도 합니다.

확인 로그인했는지, 관리자가 맞는지, 그리고 **이 데이터의 주인이 맞는지**를 서버에서 매번 확인해야 합니다.

📋 **이 한 줄만 복사해서 AI에 붙여넣으세요**

```
로그인·관리자·데이터 소유자 확인이 서버에서 빠진 곳을 전부 찾아서 고쳐줘.
```

남의 회원 정보까지 다 보이게 열려 있지 않은가

이렇게 됩니다

회원 명부 전체가 그대로 내려받아집니다

🔥 **이런 일이 납니다**

Supabase에서 데이터가 불러와지지 않을 때 AI가 자주 내놓는 답이 있습니다. **"데이터 접근 규칙을 잠시 꺼 보세요"**라는 것인데, 끄면 곧바로 해결됩니다.

문제는 그 "잠시"가 출시까지 이어진다는 점입니다. 이 규칙은 브라우저가 데이터베이스에 직접 접근하는 구조에서 **유일한 자물쇠**입니다. 꺼 둔 채로 열면 회원 한 사람이 전체 회원 명부를 통째로 내려받을 수 있고, 여기에는 특별한 기술도 필요하지 않습니다.

이런 제보도 종종 올라옵니다

"서비스 잘 쓰다가 브라우저에서 F12를 눌러 봤는데, 다른 사람 이메일이랑 전화번호가 그냥 다 보이더라고요. 제보하려고 연락처 찾는 중입니다."

확인 모든 테이블에 데이터 접근 규칙(RLS)이 켜져 있는지, 그리고 관리자 키가 브라우저 코드에 들어가 있지 않은지 확인하세요.

📋 **이 한 줄만 복사해서 AI에 붙여넣으세요**

```
모든 테이블의 데이터 접근 규칙을 점검하고 필요한 설정을 만들어 주고, 관리자 키가 브라우저에 있는지 확인해줘.
```

입력창에 이상한 걸 넣어도 버티는가

이렇게 됩니다

데이터가 통째로 새거나 지워집니다

🔥 **이런 일이 납니다**

"검색 기능 만들어줘"라고 하면 AI가 문자열을 이어붙여 쿼리를 만드는 코드를 내놓을 때가 있습니다. 검색 자체는 잘 작동합니다.

그런데 누군가 검색어 대신 **데이터베이스 명령어**를 넣으면 그것이 그대로 실행됩니다. 주문서 비고란에 "그리고 금고 열쇠도 주세요"라고 적는 것과 같은데, 실제로 통합니다. 댓글에 악성 코드를 심는 방식은 더 곤란합니다. 공격을 받는 쪽이 내가 아니라 **그 글을 읽은 다른 회원**이기 때문입니다.

확인 사용자가 글자를 넣을 수 있는 곳은 모두 대상입니다. 검색창, 로그인창, 게시글, 댓글, 문의 폼을 차례로 살펴보세요.

📋 **이 한 줄만 복사해서 AI에 붙여넣으세요**

```
입력창을 통한 공격에 취약한 곳을 전부 찾아서 안전하게 고쳐줘.
```

하루에 나갈 수 있는 돈에 상한을 걸었는가

이렇게 됩니다

쓰지도 않은 요금이 청구됩니다

🔥 **이런 일이 납니다**

AI에게 "OpenAI 연동해줘"라고 하면 연동은 잘 해 줍니다. 다만 **"하루 지출 한도를 걸어 두세요"라고 알려 주지는 않습니다.** 시키지 않았기 때문입니다.

그 상태로 열어 두면 누군가 그 기능을 자동으로 수만 번 호출합니다. 특이한 점은 **공격자가 얻는 것이 아무것도 없다**는 것인데, 요금은 서비스를 만든 쪽이 냅니다. 한도 기능은 대부분 제공되지만 기본값이 꺼져 있습니다.

요금 사고 후기에서 반복되는 패턴

"출시하고 3일째에 카드 정지 문자가 왔습니다. 확인해 보니 누가 제 챗봇에 밤새 자동으로 질문을 넣고 있었어요. 그 사람은 아무것도 안 가져갔습니다."

확인 돈이 나가는 서비스마다 **하루 한도와 알림**을 켜 두세요. 5분이면 충분합니다. 여기에 더해 같은 사람이 무한히 반복 호출하지 못하도록 횟수 제한도 함께 걸어야 합니다. 한도만 걸어 두면 이미 나간 돈은 돌려받을 수 없기 때문입니다.

여기까지 9개. 절반 왔습니다 👏

3부. 데이터와 법 — 2가지

개인정보처리방침과 이용약관이 있는가

이렇게 됩니다

앱스토어 심사에서 막힙니다

🔥 **이런 일이 납니다**

"저희는 이메일만 받는데요"라고 하시는 경우가 많은데, 이메일 한 줄도 개인정보 수집에 해당합니다.

방침 없이 회원을 받으면 법적인 문제가 생기고, **앱스토어 심사는 거의 확실하게 막힙니다.** 출시한 뒤에 붙이는 것이 아니라 첫 사용자를 받기 전에 있어야 하는 문서입니다.

확인 수집 항목과 이용 목적, 보관 기간, 파기 절차, 문의처가 들어가야 합니다. 회원 탈퇴 기능도 함께 만드세요. **가입은 되는데 탈퇴가 안 되는 서비스**는 심사에서 바로 걸립니다.

📋 **이 한 줄만 복사해서 AI에 붙여넣으세요**

```
[서비스 설명 / 수집 정보]에 맞는 개인정보처리방침·이용약관과 그 페이지를 만들어줘.
```

데이터가 자동 저장되고, 되살려 본 적 있는가

이렇게 됩니다

데이터가 사라지고 되돌릴 수 없습니다

🔥 **이런 일이 납니다**

데이터는 해킹보다 **실수로 사라지는 경우가 훨씬 많습니다.** 잘못된 삭제 명령 한 줄이나 AI가 만들어 준 데이터 구조를 바꾸는 작업 한 번이면 충분합니다.

그리고 이런 일은 대개 서비스가 잘되기 시작한 다음에 벌어집니다. 회원이 열 명일 때는 나지 않던 사고가 천 명이 되면 납니다.

확인 무료 플랜은 백업이 꺼져 있는 경우가 많습니다. 다만 더 중요한 것은 **복구를 한 번이라도 해 보았는가**입니다. 복구해 본 적 없는 백업은 백업이 아니라 추측에 가깝습니다.

📋 **이 한 줄만 복사해서 AI에 붙여넣으세요**

```
[DB 이름] 자동 백업 설정과 복구 연습 방법을 단계별로 알려줘.
```

4부. 배포 — 3가지

열쇠를 배포한 서버에도 넣어줬는가

이렇게 됩니다

배포는 됐는데 아무 기능도 안 돕니다

🔥 **이런 일이 납니다**

"로컬에선 되는데 배포하니까 안 돼요." 바이브코딩에서 가장 자주 나오는 문장이고, 그 원인 1위가 이것입니다.

AI가 키를 `.env` 로 옮겨 주고 `.gitignore` 에도 넣어 주었습니다. 여기까지는 잘한 것입니다. 다만 **배포 서비스 대시보드에도 따로 등록해야 한다**는 안내는 하지 않습니다. 그래서 서버에는 키가 하나도 없는 상태로 배포됩니다.

확인 배포 서비스 대시보드의 설정값(환경변수) 화면에 `.env` 내용을 그대로 넣으세요. 넣은 뒤에는 **다시 배포해야** 반영되는데, 이 단계를 빠뜨리는 경우가 많습니다.

새 사이트 주소를 로그인·결제 회사에 알려줬는가

이렇게 됩니다

로그인과 결제 버튼만 먹통이 됩니다

🔥 **이런 일이 납니다**

소셜 로그인과 결제, 데이터베이스는 각자 **"이 주소에서 오는 요청만 받는다"**는 명단을 갖고 있습니다. 새로 배포한 도메인은 당연히 그 명단에 없습니다.

그래서 화면은 완벽한데 **로그인 버튼만 동작하지 않습니다.** 결제 버튼도 마찬가지입니다. 원인을 짐작하기 가장 어려운 종류의 문제입니다.

확인 구글·카카오·네이버 로그인의 로그인 후 되돌아올 주소, 데이터베이스의 허용 주소, 결제사의 성공·실패 주소를 모두 새 도메인으로 등록하세요.

사이트를 올린 직후 5분 점검을 하는가

이렇게 됩니다

회원가입이 멈춘 채 광고비만 나갑니다

🔥 이런 일이 납니다

배포에 성공하면 초록불이 켜집니다. 안심하고 하루를 마무리하게 됩니다.

그런데 그 초록불은 **코드가 올라갔다는 뜻이지 서비스가 정상이라는 뜻이 아닙니다.** 회원가입이 조용히 멈춰 있어도 초록불은 그대로 켜져 있습니다. 고객 문의로 알게 될 때는 이미 몇 시간이 지난 뒤입니다.

가장 아픈 종류의 후기

"광고를 이틀 돌렸는데 가입이 한 명도 없어서 이상하다 했더니, 회원가입 버튼이 배포 이후로 안 눌리고 있었습니다. 광고비만 나갔어요."

확인 배포할 때마다 다섯 가지만 직접 해 보세요. 첫 화면이 뜨는지, 로그인이 되는지, 핵심 기능 하나가 끝까지 가는지, 휴대폰에서도 되는지, 그리고 브라우저에서 F12를 눌렀을 때 빨간 글씨가 없는지입니다.

14개 통과. 이제 여섯 개 남았습니다 🔥

5부. 테스트 — 2가지

가입부터 결제까지 직접 해봤는가

이렇게 됩니다

사용자가 첫 화면에서 막혀 나갑니다

🔥 **이런 일이 납니다**

만든 사람은 **정상적인 순서로만** 써 보게 됩니다. 어떻게 만들었는지 알고 있으니 자연스러운 일입니다.

반면 사용자는 빈칸으로 제출하고, 버튼을 연타하고, 결제 중에 뒤로 가기를 누르고, 이름 칸에 아주 긴 글을 붙여 넣습니다. 사고는 정상적인 사용이 아니라 **정확히 이 지점에서** 납니다.

확인 회원가입부터 로그인, 핵심 기능, 결제까지 한 번에 통과해 보세요. 그다음에는 일부러 이상하게 써 보시면 됩니다. 빈칸으로 제출하고, 버튼을 연타하고, 뒤로 갔다가 다시 진행해 보세요. 지인 세 명에게 시켜 보고 옆에서 지켜보는 방법이 가장 정확합니다.

휴대폰에서도 확인했는가

이렇게 됩니다

손님 절반이 써 보지도 못하고 나갑니다

🔥 **이런 일이 납니다**

사용자 대부분은 휴대폰으로 들어오는데, 만든 사람은 데스크톱 크롬으로만 확인합니다. 사고는 이 간극에서 납니다.

아이폰 사파리에서 버튼이 눌리지 않거나, 키보드가 올라오면서 **입력창을 완전히 가려 버리는** 경우가 흔합니다. 데스크톱에서는 절대 보이지 않는 문제입니다.

확인 내 휴대폰과 기종이 다른 휴대폰에서 직접 들어가 보세요. 화면이 깨지지 않는지, 버튼이 눌리는지, 키보드가 입력창을 가리지 않는지 세 가지만 보면 됩니다.

6부. 결제 — 4가지

검증된 결제 회사를 쓰고 있는가

이렇게 됩니다

카드 사고 책임을 내가 집니다

🔥 **이런 일이 납니다**

카드 번호를 직접 받아 저장하는 순간 **지켜야 할 법적 의무와 사고 책임이 통째로 넘어옵니다.**

이것은 개인이나 소규모 팀이 감당할 수 있는 무게가 아닙니다. 하지 않는 것이 아니라 할 수 없는 영역입니다.

확인 토스페이먼츠나 아임포트, 스트라이프 같은 결제 대행사(PG사)를 쓰세요. 카드 정보는 대행사가 처리하고 내 서버에는 결제 결과만 남는 구조가 정답입니다.

결제 금액을 내 서버가 다시 확인하는가

이렇게 됩니다

10만 원짜리를 100원에 팔게 됩니다

🔥 이런 일이 납니다

"결제 붙여줘"라고 하면 AI는 대개 화면에서 금액을 받아 결제사로 넘기는 코드를 내놓습니다. 가장 단순하고 테스트도 잘 통과하기 때문입니다.

문제는 **그 금액을 사용자가 바꿀 수 있다**는 점입니다. 브라우저에서 F12를 눌러 볼 줄 아는 사람이라면 10만 원짜리를 100원에 살 수 있습니다. **이 문서에서 가장 중요한 항목입니다.**

✗ 흔한 구조

브라우저: "이거 100,000원이에요"
서버: "네 알겠습니다" → 상품 지급

사용자가 그 숫자를
100
으로 바꾸면 그대로 통과됩니다.

✓ 안전한 구조

브라우저: "결제 끝났어요"
서버 → 결제사: "얼마 결제됐죠?"
서버: 내 DB 가격과 대조 → 같을 때만 지급

브라우저에서 오는 숫자는 전부 의심하세요

확인 결제가 끝나면 서버가 결제사에 직접 물어서 실제 결제 금액을 확인해야 합니다. 그 금액이 내 데이터베이스의 상품 가격과 같을 때만 상품을 지급하세요.

📋 이 한 줄만 복사해서 AI에 붙여넣으세요

```
결제 금액을 서버에서 결제사에 재조회해 검증하도록 고쳐줘. 브라우저 금액을 믿는 코드는 전부 없애줘.
```

테스트로 결제하고 환불까지 해봤는가

이렇게 됩니다

돈은 받았는데 상품이 안 나갑니다

🔥 **이런 일이 납니다**

결제는 됐는데 상품이 나가지 않는 경우가 있습니다. 반대로 알림이 두 번 도착해서 **상품이 두 번 나가기도** 합니다.

그리고 첫 환불 요청이 들어온 날에야 환불하는 방법을 찾기 시작하게 됩니다. 그동안 고객은 기다립니다. 세 가지 모두 출시 전에 한 바퀴만 돌려 보면 걸러집니다.

첫 결제일에 자주 벌어지는 일

"첫 결제가 들어와서 신났는데, 고객이 '결제됐는데 아무것도 안 왔어요'라고 연락이 왔습니다. 자동 알림이 두 번 오는 건 알았는데 안 오는 경우는 생각을 못 했네요."

확인 테스트 모드로 결제부터 상품 지급, 환불까지 한 바퀴 돌려 보세요. 결제사가 보내는 자동 알림이 두 번 오더라도 상품이 두 번 나가지 않는지도 함께 확인하셔야 합니다.

사업자 등록과 판매 신고를 마쳤는가

이렇게 됩니다

과태료를 물고 결제가 막힙니다

🔥 **이런 일이 납니다**

돈을 받는 순간부터는 취미가 아니라 사업입니다.

신고 없이 판매하면 **과태료 대상**이고, 애초에 결제 대행사(PG사) 심사부터 통과되지 않습니다. 그리고 환불 정책을 미리 정해 두지 않으면 첫 분쟁에서 손해를 보게 됩니다. 판단할 기준이 없기 때문입니다.

확인 사업자 등록을 하고 통신판매업 신고를 마친 뒤 결제사 심사를 받는 순서입니다. 환불 정책은 어느 시점에 얼마가 돌아오는지를 문서로 정해 사이트에 올려 두세요.

7부. 속도와 비용 — 3가지

사진이 무거워서 첫 화면이 느리지 않은가

이렇게 됩니다

열어 보기도 전에 손님이 나갑니다

🔥 **이런 일이 납니다**

AI는 화면을 예쁘게 만들어 주지만 **사진 용량은 줄여 주지 않습니다.** 휴대폰으로 찍은 4메가짜리 사진을 그대로 올려도 화면에는 잘 나오기 때문에 문제를 눈치채기 어렵습니다.

그런데 손님 입장에서는 첫 화면이 뜨는 데 몇 초가 걸립니다. **3초를 넘기면 절반 가까이가 그냥 나갑니다.** 광고를 돌리는 중이라면 그 돈이 그대로 새는 셈입니다.

확인 구글에서 "PageSpeed Insights"를 검색해 내 사이트 주소를 넣어 보세요. 점수보다 **가장 무거운 사진이 무엇인지**만 보시면 됩니다.

📋 **이 한 줄만 복사해서 AI에 붙여넣으세요**

```
첫 화면을 느리게 만드는 사진을 찾아서 자동으로 가볍게 변환되도록 고쳐줘.
```

글이 많아져도 견디는가

이렇게 됩니다

손님이 늘어난 날 사이트가 멈춥니다

🔥 **이런 일이 납니다**

만드는 동안에는 글이 열 개쯤 있습니다. 그래서 무엇을 해도 빠릅니다.

AI가 짜 주는 코드는 대개 **전부 불러온 다음 화면에서 골라내는** 방식입니다. 열 개일 때는 문제가 없지만 만 개가 되면 그대로 멈춥니다. 출시 전에는 데이터가 적어서 절대 보이지 않는 문제입니다.

확인 목록 화면에 시험 삼아 데이터를 천 개쯤 넣어 보고 열리는 속도를 보세요. 눈에 띄게 느려진다면 지금 고쳐야 합니다.

📋 **이 한 줄만 복사해서 AI에 붙여넣으세요**

```
목록 화면이 데이터 1만 건에서도 견디는지 점검하고, 느린 곳을 고쳐줘.
```

무료 요금제의 한도를 알고 있는가

이렇게 됩니다

어느 날 갑자기 서비스가 멈춥니다

🔥 **이런 일이 납니다**

무료 요금제에는 반드시 한도가 있습니다. 그리고 그 한도를 넘으면 둘 중 하나가 일어납니다.

서비스가 멈추거나, 말없이 요금이 청구되기 시작하거나.

어느 쪽인지 모르고 출시하면 하필 **서비스가 잘되기 시작한 날**에 사고가 납니다. 손님이 몰린 바로 그 순간에 사이트가 열리지 않습니다.

확인 쓰고 있는 서비스마다 한도가 얼마인지, 넘으면 멈추는지 과금되는지 확인하세요. 한동안 접속이 없으면 잠자기 상태가 되는지도 함께 보시면 좋습니다.

스물세 개, 끝났습니다

출시 직전에 이 한 장만 다시 훑으세요.

- ☐ 01 · API 키가 코드에 없다
- ☐ 02 · `.env` 가 `.gitignore` 에 있다
- ☐ 03 · 브라우저에 비밀 키가 없다
- ☐ 04 · 관리자 비밀번호를 바꿨다
- ☐ 05 · 비밀번호가 그대로 저장되지 않는다
- ☐ 06 · 권한 확인을 서버에서 한다
- ☐ 07 · 데이터베이스 보안 규칙이 켜져 있다
- ☐ 08 · 입력값 공격을 막아뒀다
- ☐ 09 · 하루 지출 한도를 걸었다
- ☐ 10 · 개인정보처리방침과 약관이 있다
- ☐ 11 · 자동 백업이 켜져 있고, 복구해 봤다
- ☐ 12 · 배포 서버에 환경변수를 등록했다
- ☐ 13 · 외부 서비스에 새 주소를 등록했다
- ☐ 14 · 배포 직후 5분 점검을 했다
- ☐ 15 · 핵심 흐름을 끝까지 통과했다
- ☐ 16 · 휴대폰에서 확인했다
- ☐ 17 · 검증된 결제사를 쓴다
- ☐ 18 · 금액을 서버에서 다시 확인한다
- ☐ 19 · 테스트 결제와 환불을 통과했다
- ☐ 20 · 사업자 등록과 신고를 마쳤다

- ☐ 21 · 무거운 사진을 가볍게 바꿨다
- ☐ 22 · 데이터가 많아져도 견딘다
- ☐ 23 · 무료 요금제 한도를 안다

마지막으로 세 가지만

완벽한 보안은 없습니다. 쉬운 먹잇감만 안 되면 됩니다. 해킹 봇은 잠긴 문을 굳이 안 땁니다. 옆집이 열려 있으니까요.

사고는 출시 날이 아니라 잘되기 시작한 날 터집니다. 회원이 늘면 이 문서를 다시 꺼내 보세요.

AI에게 시키되, 확인은 사람이 하세요. 이 문서의 프롬프트들이 그 확인하는 법입니다.

출시 응원합니다. 🚀

이 문서를 만든 곳 · 졸업하는 외주

스물세 개 중에 "이건 못 하겠는데" 싶은 게 몇 개였나요

0개였다면 이 페이지는 넘기셔도 됩니다. 이미 하실 수 있는 분입니다.

3개 이상이었다면 솔직히 말씀드리겠습니다. 그 항목들은 시간을 더 쓴다고 해결되지 않습니다. 결제 검증, 권한 처리, 데이터 이관 같은 건 **몰라서 못 하는 게 아니라 한 번도 안 해봐서 못 하는 일**이고, 처음 해보는 사람이 실서비스에서 연습하면 그 비용은 고객이 냅니다.

그래서 이 구간만 대신 만들어 드립니다

랜딩 페이지나 소개 사이트는 저희 상품에서 뺐습니다. 그 정도는 이제 AI로 직접 하실 수 있으니, 저희가 돈을 받을 이유가 없습니다.

- 알림·시트 연동 같은 **서버에서 도는 자동화**
- 권한과 동시 처리가 걸린 **관리자 화면**
- **결제 · 정산 · 환불**과 스토어 배포
- 흩어진 시스템 **통합과 데이터 이관**

그리고 만들고 끝이 아니라 **코드 전량과 문서 6종을 통째로 드립니다.** 이후 수정은 대표님 쪽에서 AI에게 직접 시키시면 됩니다. **유지보수 계약을 받지 않습니다.** 다른 외주는 계속 붙어 있으려고 하죠. 저희 목표는 졸업시키는 겁니다.

위시켓 공개 평점 5.00입니다. 전문성·만족도·의사소통·일정 준수·적극성·하자보수 여섯 항목이 전부 만점이고, 저희가 정한 숫자가 아니라 클라이언트가 매긴 것입니다. **위시켓 프로필**에서 직접 확인하실 수 있습니다.

상품	정가	첫 기수	좌석
LITE 연동·자동화	250만	**49만**	8석
STANDARD 운영 백오피스	600만	**150만**	3석
PRO 서비스	1,200만	**290만**	2석
MAX 통합	3,000만	**490만**	1석

부가세 별도. 새로 만든 상품이라 공개할 사례가 없어서 **후기와 사례를 돈 주고 사는 중**입니다. 첫 기수만 이 가격이고, 사례가 쌓이는 다음 기수부터는 정가로 돌아갑니다. **좌석은 결제 순으로 차고, 다 차면 그대로 종료됩니다.**

담당자분이 직접 수정해 보는 시연을 3회까지 도와드렸는데도 못 하시면 전액 환불합니다. 못 고치실 거면 저희도 이 상품을 팔면 안 되니까요.

지금 상담 → 카카오톡 채널 바로가기

남은 좌석·전체 안내 → kinoflare.com/ko/launch

본 문서는 일반적인 기술·법률 정보를 제공하며, 개별 상황에 대한 법률 자문을 대체하지 않습니다. 사업 규모가 커지면 보안 전문가와 법률 전문가의 검토를 받는 것을 권장합니다.