한칸스튜디오 · 현직 개발자의 바이브코딩 노트

바이브코딩 실전 가이드

코딩 몰라도 내 아이디어를 실제 서비스로 만드는 법

2026년 10월 · 5부 27장 + 특별 부록 · hankanstudio.com

차례

PART 01 · 일단 하나 만들어보자

01 바이브코딩 시작하기

02 AI에게 말 한마디로 첫 앱 만들기

03 원하는 디자인과 기능으로 바꾸기

04 잘 만든 결과물 저장하고 되돌리기

05 내 앱을 인터넷에 공개하고 공유하기

PART 02 · 이제 내가 원하는 걸 만들자

06 막연한 아이디어를 실제 앱으로 구체화하기

07 AI와 함께 기능·화면·사용 흐름 정하기

08 처음부터 크게 만들지 않는 법

09 AI에게 개발 계획을 맡기고 단계별로 실행하기

PART 03 · 바이브코딩을 잘하는 사람들의 노하우

10 AI가 내 말을 제대로 이해하게 만드는 요청법

11 기존 기능을 망가뜨리지 않고 수정하는 법

12 오류가 반복될 때 해결하는 방법

13 AI가 만든 기능이 정말 작동하는지 검증하는 법

14 AI가 이전 작업을 잊어버리지 않게 관리하는 법

15 프로젝트가 복잡해져도 흔들리지 않는 작업 방식

16 AI에게 반복 작업을 맡기고 사용 비용 줄이기

PART 04 · 진짜 서비스처럼 만들어보자

17 데이터를 저장하고 다시 불러오기

18 여러 사용자가 같은 데이터를 공유하기

19 회원가입과 로그인 기능 붙이기

20 외부 API와 AI 기능 연결하기

21 결제·알림·파일 업로드 기능 추가하기

22 모바일에서도 앱처럼 사용할 수 있게 만들기

PART 05 · 출시하고 나서도 문제없이 운영하기

23 배포 전에 반드시 확인해야 할 것들

24 개인정보·API 키·보안 사고 예방하기

25 예상치 못한 요금 폭탄 막기

26 업데이트와 백업, 오류 복구하기

27 사용자 피드백으로 서비스 개선하기

부록 특별 부록 · 복사해서 사용하는 실전 자료

PART 01

일단 하나 만들어보자

설명보다 먼저 직접 만들어보는 구간

1장

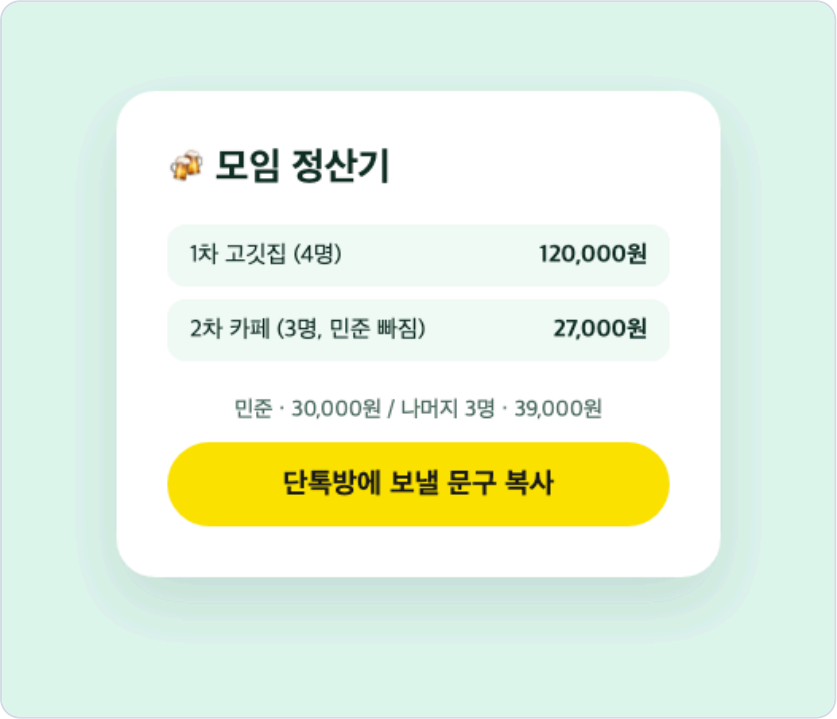
바이브코딩 시작하기

이 장을 마치면 준비물과 AI 도구를 고르고, 앱을 만들 준비가 끝나요. 코딩 공부는 필요 없어요.

바이브코딩이란

코드를 직접 쓰지 않고, AI에게 한국어로 말해서 앱을 만드는 방식이에요. "총 금액이랑 인원 넣으면 1인당 얼마인지 알려주는 계산기 만들어줘"라고 쓰면 AI가 만들어 줘요.

PART 01에서는 모임에서 쓰는 **정산기**를 하나 만들어요. 5장을 마치면 친구에게 링크로 보낼 수 있어요.



PART 01을 다 따라 하면 이런 앱이 생겨요. 예시 화면이에요.

준비물

- Mac 또는 Windows 컴퓨터 (휴대폰·태블릿은 안 돼요)
- AI 도구 구독 하나

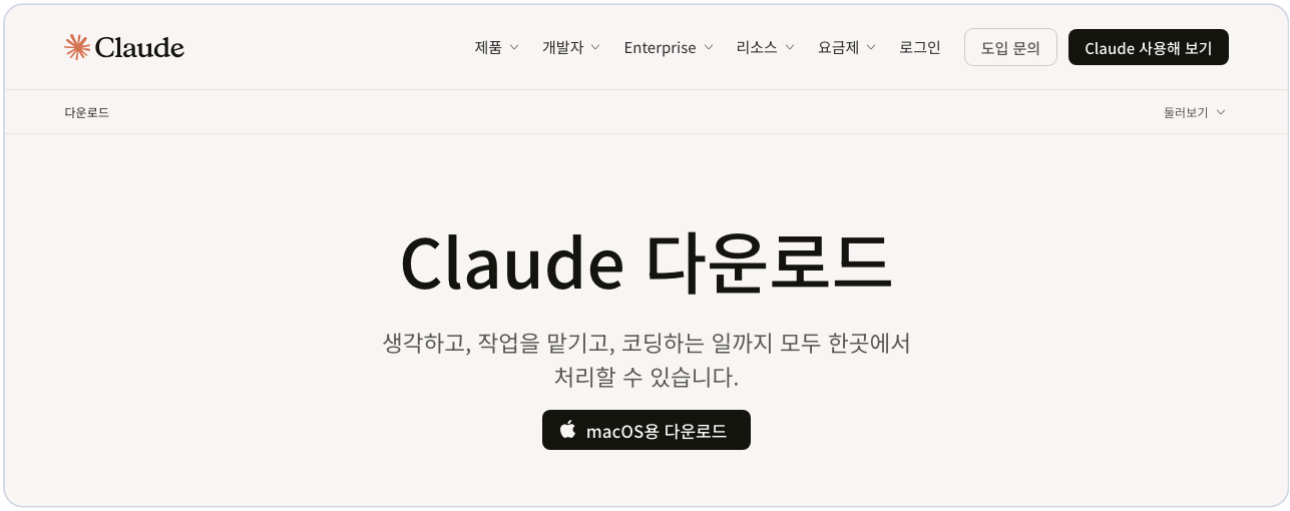
AI 도구 고르기

도구	어디서	이런 분께
Claude 데스크톱 앱 (추천)	claude.com/download	이 책을 그대로 따라 하고 싶은 분. Pro 구독이 필요해요
ChatGPT 데스크톱 앱의 Codex	chatgpt.com	이미 ChatGPT를 쓰는 분

이 책은 Claude 데스크톱 앱 기준이에요. 책의 문장은 Codex에도 그대로 붙여 넣으면 돼요. 요금은 특별 부록 E에 있어요.

1단계. Claude 앱 내려받기

claude.com/download에 들어가서 내 컴퓨터용 앱을 내려받아요.



2단계. 설치하고 로그인

- **Mac:** 내려받은 파일을 열고, Claude 아이콘을 응용 프로그램 폴더로 끌어다 놓아요.
- **Windows:** 내려받은 설치 파일을 실행해요.

앱을 열고 구독한 계정으로 로그인해요.

3단계. Code 모드 켜기

앱 왼쪽 위의 `</>` 버튼을 눌러요. 이 모드에서는 AI가 내 컴퓨터에 파일을 만들 수 있어요.

4단계. 작업 폴더 만들기

바탕화면에 `정산기` 라는 폴더를 하나 만들어 두세요. PART 01에서 만드는 파일은 전부 여기에 모여요.

막히면

이럴 때	이렇게 하세요
`</>` 버튼이 없음	앱을 최신 버전으로 업데이트해요. 무료 플랜이면 Code 모드가 없어요
브라우저에서 로그인했는데 앱으로 안 돌아옴	브라우저 위쪽에 뜨는 "앱 열기"를 허용해요

2장

AI에게 말 한마디로 첫 앱 만들기

이 장을 마치면 총 금액과 인원을 넣으면 1인당 금액이 나오는 계산기가 생겨요.

1단계. 정산기 폴더 고르기

Claude 앱의 Code 모드에서 폴더를 고르라고 나오면 1장에서 만든 `정산기` 폴더를 골라요.

2단계. 이 문장 붙여 넣고 보내기

복사해서 쓰기

```
총 금액이랑 인원을 넣으면 1인당 얼마인지 알려주는 계산기 앱 만들어줘. 나 비개발자야.
브라우저에서 바로 열리는 파일 하나로 만들고, 다 만들면 화면에 띄워줘.
```

3단계. 허용 누르기

AI가 일하다가 "이 작업을 허용할까요?" 하고 물어요. **허용**을 눌러요.

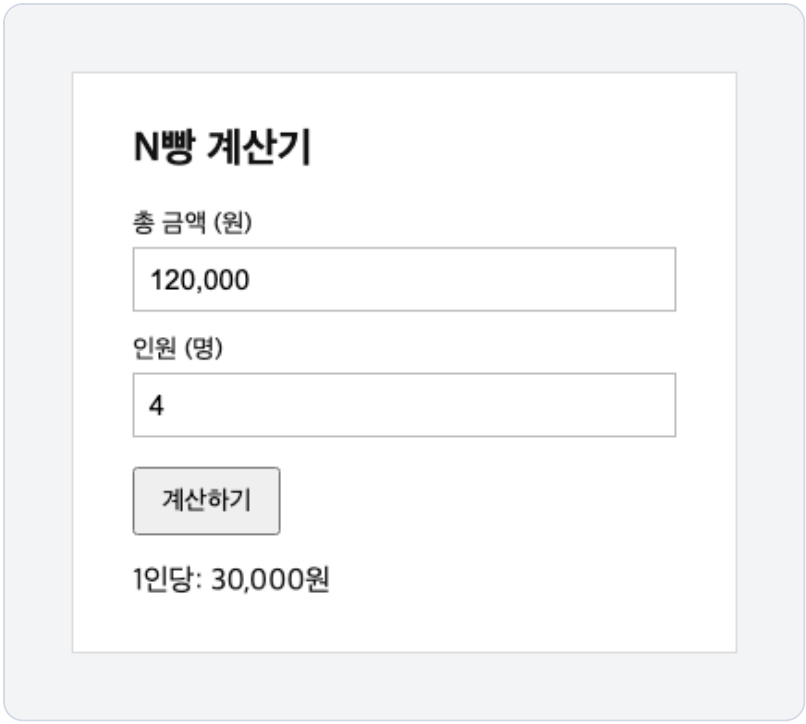
뭘 하려는 건지 궁금하면 이렇게 물어봐요.

복사해서 쓰기

```
방금 허용해 달라고 한 거, 무슨 일 하는 건지 쉬운 말로 설명해줘.
```

4단계. 결과 확인

잠시 기다리면 계산기가 화면에 떠요.



예시 화면이에요. 실제 모양은 매번 조금씩 달라요. 꾸미는 건 3장에서 해요.

화면이 안 뜨면 이렇게 보내요.

```
만든 거 어떻게 열어? 열어줘
```

5단계. 직접 써보기

총 금액에 `120000`, 인원에 `4`를 넣고 계산 버튼을 눌러요. **30,000원**이 나오면 성공이에요.

숫자가 이상하게 나오면 보이는 그대로 말해요.

```
120000이랑 4를 넣었더니 OO이 나와. 원래는 30000이 나와야 해.
```

막히면

이럴 때	이렇게 하세요
빨간 에러 메시지	메시지를 통째로 복사해서 "이 에러 고쳐줘"와 함께 보내요

이럴 때	이렇게 하세요
화면이 하얗거나 깨짐	화면을 캡처해서 붙여 넣고 "이렇게 보여"라고 보내요
어려운 말이 나옴	"이해하기 쉽게 말해줘"라고 보내요

이 장의 문장

상황	붙여 넣을 문장
계산기 만들기	총 금액이랑 인원을 넣으면 1인당 얼마인지 알려주는 계산기 앱 만들어줘. 나 비개발자야. 브라우저에서 바로 열리는 파일 하나로 만들고, 다 만들면 화면에 띄워줘.
허용 창이 떴을 때	방금 허용해 달라고 한 거, 무슨 일 하는 건지 쉬운 말로 설명해줘.
화면이 안 보일 때	만든 거 어떻게 열어? 열어줘
결과가 이상할 때	OO을 넣었더니 OO이 나와. 원래는 OO이 나와야 해.

3장

원하는 디자인과 기능으로 바꾸기

이 장을 마치면 계산기가 보기 좋게 바뀌고, 1차·2차 정산과 단톡방 문구 복사까지 돼요.

2장 계산기가 열린 상태에서, 같은 대화창에 아래 문장을 **하나씩** 보내요. 한 번 보낼 때마다 화면을 확인하고 다음으로 넘어가요.

1단계. 색과 글자 바꾸기

복사해서 쓰기

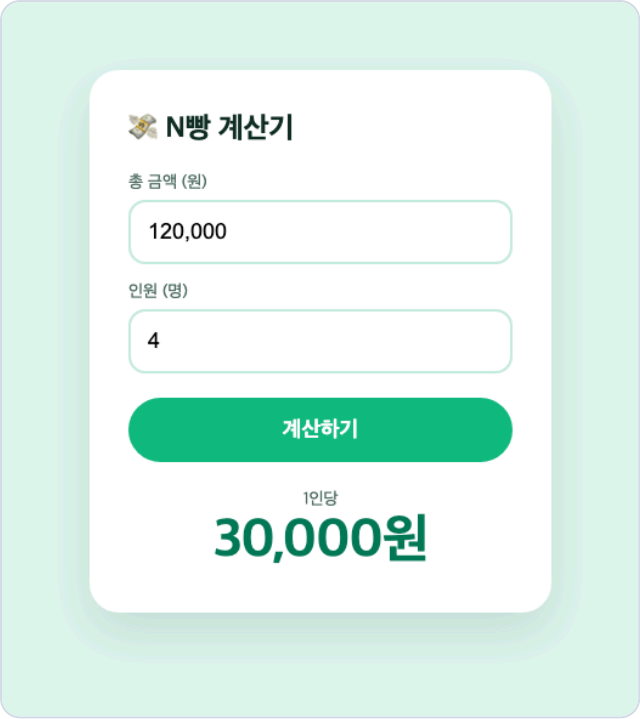
```
배경을 연한 민트색으로 바꿔줘
```

복사해서 쓰기

```
1인당 금액을 아주 크고 굵게 보여줘
```

복사해서 쓰기

```
계산 버튼을 크고 둥글게, 초록색으로 바꿔줘
```



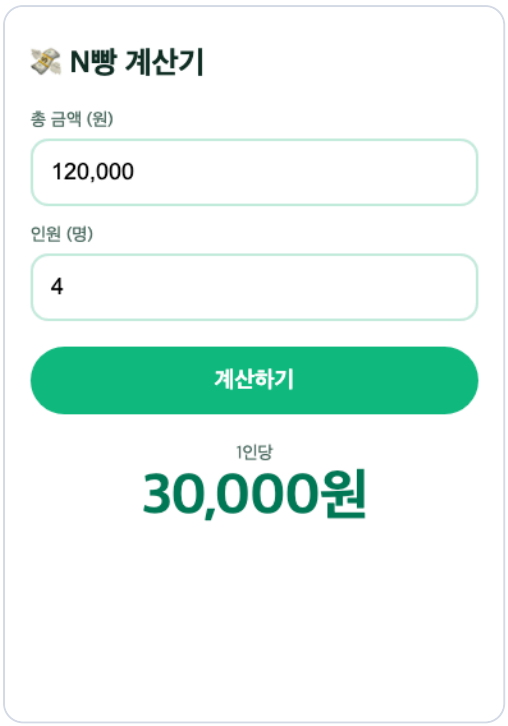
예시 화면이에요.

2단계. 휴대폰 화면에 맞추기

복사해서 쓰기

```
휴대폰 화면에서도 잘 보이게 바꿔줘. 글자랑 버튼이 손가락으로 누르기 편하게.
```

브라우저 창을 좁게 줄여 보면 휴대폰에서 어떻게 보일지 확인할 수 있어요.



3단계. 차수 나누고 사람마다 계산하기

복사해서 쓰기

1차, 2차처럼 여러 번 정산할 수 있게 해줘. 차수마다 이름이랑 금액을 따로 넣고, 차수를 추가하는 버튼도 만들어줘.

복사해서 쓰기

같이 간 사람 이름을 넣고, 차수마다 누가 참석했는지 체크할 수 있게 해줘. 맨 아래에 사람마다 내야 할 금액을 보여줘.



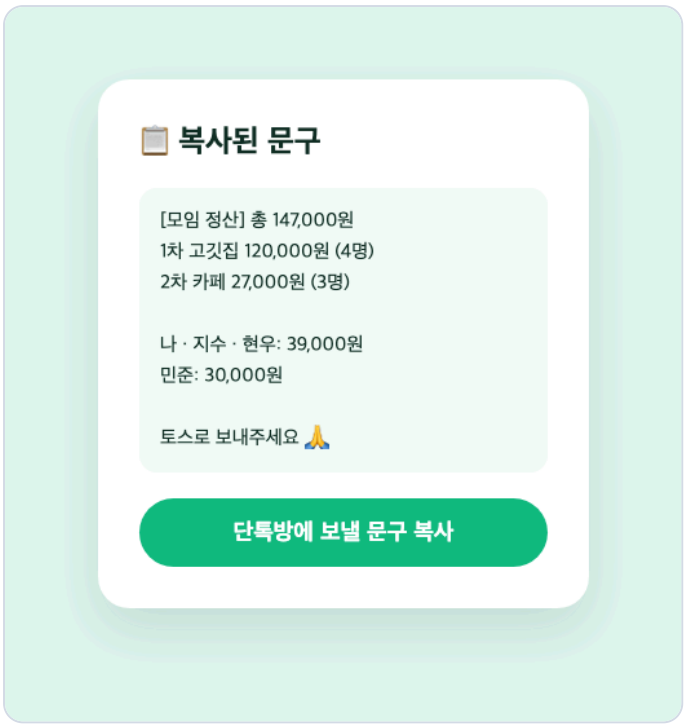
아래처럼 넣었을 때 **나·지수·현우 39,000원, 민준 30,000원**이 나오면 맞아요.

차수	금액	참석
1차 고깃집	120,000원	나, 지수, 현우, 민준
2차 카페	27,000원	나, 지수, 현우 (민준 빠짐)

4단계. 단톡방 문구 복사 버튼

복사해서 쓰기

정산 결과를 단톡방에 보낼 문구로 만들어서, 버튼을 누르면 복사되게 해줘. 차수별 금액이랑 사람마다 낼 금액이 들어가게.



마음에 안 들면

```
복사해서 쓰기
```

```
방금 바꾼 거 취소하고 그 전으로 돌려줘
```

원하는 모습은 이 틀로 말하면 돼요.

```
복사해서 쓰기
```

```
00을 00처럼 바꿔줘
```

막히면

이럴 때	이렇게 하세요
말한 대로 안 바뀜	보이는 그대로 말해요. "버튼 색이 그대로야. 초록색이었으면 좋겠어"
기능을 넣었더니 계산이 틀림	"방금 바꾼 거 취소하고 그 전으로 돌려줘"
원하는 느낌을 말로 못 하겠음	마음에 드는 앱 화면을 캡처해서 붙여 넣고 "이런 느낌으로 바꿔줘"

이 장의 문장

상황	붙여 넣을 문장
색 바꾸기	배경을 연한 민트색으로 바꿔줘
결과 크게	1인당 금액을 아주 크고 굵게 보여줘
휴대폰에 맞추기	휴대폰 화면에서도 잘 보이게 바꿔줘. 글자랑 버튼이 손가락으로 누르기 편하게.
차수 나누기	1차, 2차처럼 여러 번 정산할 수 있게 해줘. 차수마다 이름이랑 금액을 따로 넣고, 차수를 추가하는 버튼도 만들어줘.
사람마다 계산	같이 간 사람 이름을 넣고, 차수마다 누가 참석했는지 체크할 수 있게 해줘. 맨 아래에 사람마다 내야 할 금액을 보여줘.
문구 복사 버튼	정산 결과를 단톡방에 보낼 문구로 만들어서, 버튼을 누르면 복사되게 해줘.
방금 것 취소	방금 바꾼 거 취소하고 그 전으로 돌려줘
내 마음대로	OO을 OO처럼 바꿔줘

4장

잘 만든 결과물 저장하고 되돌리기

이 장을 마치면 잘 될 때 저장해 두고, 망가지면 그때로 돌아갈 수 있어요.

게임의 세이브와 같아요. 앞으로 기능을 더하다 보면 앱이 망가질 때가 있는데, 저장해 두면 걱정 없어요.

1단계. 지금 상태 저장하기

복사해서 쓰기

```
지금 상태를 저장해줘. 이름은 "정산기 1차 완성"으로. 저장 기능이 아직 없으면 켜고 해줘.
```

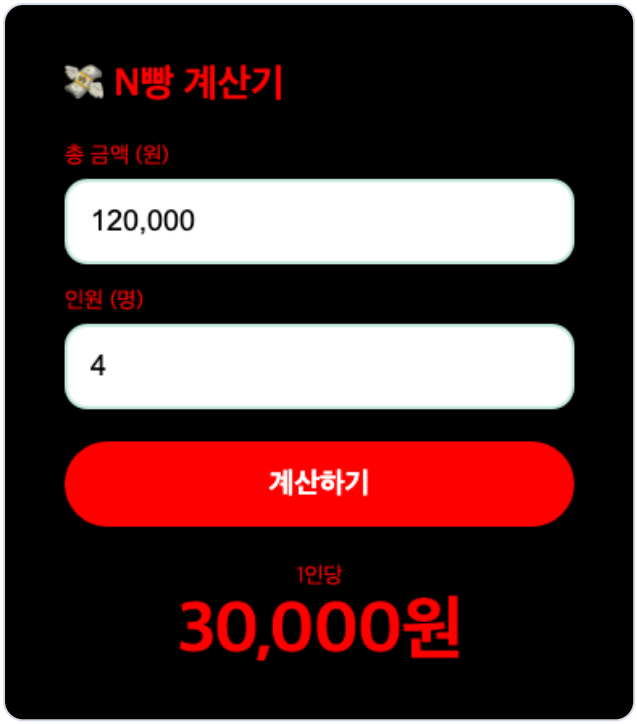
허용 창이 뜨면 허용을 눌러요. "저장했어요" 같은 답이 오면 끝이에요.

2단계. 일부러 망가뜨려 보기

되돌리기를 연습하려고 일부러 이상하게 바꿔 봐요.

복사해서 쓰기

```
배경을 검은색으로, 글자를 전부 빨간색으로 바꿔줘
```



3단계. 저장한 때로 되돌리기

복사해서 쓰기

```
"정산기 1차 완성"으로 저장한 때로 되돌려줘. 저장 기록은 지우지 말고.
```

화면이 원래 모습으로 돌아오면 성공이에요. 바로 안 바뀌면 브라우저에서 새로고침을 눌러요.

4단계. 저장 목록 보기

복사해서 쓰기

```
지금까지 저장한 거 목록 보여줘. 날짜랑 이름만 최신순으로.
```

언제 저장할까요

- 원하는 대로 잘 됐을 때
- 크게 바꾸기 직전 ("디자인 전부 바꿔줘" 같은 부탁 전)

복사해서 쓰기

```
여기까지 저장해줘. 이름은 "00"로.
```

막히면

이럴 때	이렇게 하세요
"Git이 없다"는 말이 나옴	"Git 설치해줘"라고 보내요. 저장 기능에 필요한 프로그램이에요
되돌렸는데 화면이 그대로	브라우저에서 새로고침을 눌러요
어떤 이름으로 저장했는지 모르겠음	"지금까지 저장한 거 목록 보여줘"

이 장의 문장

상황	붙여 넣을 문장
처음 저장	지금 상태를 저장해줘. 이름은 "OO"으로. 저장 기능이 아직 없으면 켜고 해줘.
이름 붙여 저장	여기까지 저장해줘. 이름은 "OO"로.
되돌리기	"OO"로 저장한 때로 되돌려줘. 저장 기록은 지우지 말고.
저장 목록	지금까지 저장한 거 목록 보여줘. 날짜랑 이름만 최신순으로.

5장

내 앱을 인터넷에 공개하고 공유하기

이 장을 마치면 정산기에 인터넷 주소가 생기고, 단톡방에 링크를 보내면 미리보기 카드와 함께 친구가 폰으로 바로 열 수 있어요.

지금까지 만든 정산기는 내 컴퓨터에서만 열려요. **Vercel**이라는 무료 서비스에 올리면 인터넷 주소가 생겨요.

1단계. Vercel 가입하기

vercel.com에 들어가서 **Sign Up**을 눌러 가입해요. 무료 요금제(Hobby)를 고르면 돼요. 가입은 내가 직접 해요.

2단계. 올려 달라고 하기

복사해서 쓰기

```
이 정산기를 인터넷 주소로 열 수 있게 Vercel에 올려줘. 나 비개발자야. 내가 직접 해야 하는 단계가 있으면 하나씩 알려줘.
```

중간에 브라우저 창이 열리고 Vercel 로그인이나 승인을 물어봐요. 가입한 계정으로 로그인하고 승인을 눌러요.

3단계. 주소 받기

다 되면 AI가 `https://이름.vercel.app` 같은 주소를 알려줘요. 눌러서 정산기가 열리는지 확인해요.

복사해서 쓰기

```
주소를 moim-jungsan.vercel.app처럼 알아보기 쉬운 이름으로 바꿔줘
```

4단계. 공유 미리보기 카드 만들기

주소를 카톡으로 보내면 주소만 덩그러니 보여요. 미리보기 카드를 만들어요.

복사해서 쓰기

```
카톡이나 문자로 링크를 보내면 미리보기 카드가 예쁘게 뜨게 해줘. 제목은 "우리 모임 정산기", 설명은 "1차·2차 나눠서 계산하고 바로 송금", 그림은 정산기 분위기에 맞게 만들어줘. 다 되면 Vercel에도 다시 올려줘.
```



카톡은 한 번 본 링크의 미리보기를 한동안 기억해요. 바로 안 바뀌면 주소 끝에 `?v=2` 를 붙여 보내 봐요.

5단계. 휴대폰으로 열어 보기

주소를 나에게 카톡으로 보내고 휴대폰에서 열어 봐요. 잘 열리면 단톡방에 보내요.

6단계. 고친 뒤 다시 올리기

복사해서 쓰기

```
고친 거 저장하고 Vercel에도 다시 올려줘
```

주소는 그대로예요.

알아둘 것

주소를 아는 사람은 누구나 열 수 있어요.

막히면

이럴 때	이렇게 하세요
로그인 창이 안 뜸	"Vercel 로그인 창이 안 떠. 내가 직접 열 수 있는 주소를 알려줘"

이럴 때	이렇게 하세요
주소를 열었더니 에러 화면	화면을 캡처해서 붙여 넣고 "올린 주소를 열었더니 이렇게 보여"
미리보기에 그림이 안 나옴	"카톡 미리보기에 그림이 안 나와. 그림 주소가 인터넷에서 열리는 전체 주소로 들어갔는지 확인해줘"

이 장의 문장

상황	붙여 넣을 문장
처음 올리기	이 정산기를 인터넷 주소로 열 수 있게 Vercel에 올려줘. 나 비개발자야. 내가 직접 해야 하는 단계가 있으면 하나씩 알려줘.
주소 이름 바꾸기	주소를 OO.vercel.app처럼 알아보기 쉬운 이름으로 바꿔줘
미리보기 카드	카톡이나 문자로 링크를 보내면 미리보기 카드가 예쁘게 뜨게 해줘. 제목은 "OO", 설명은 "OO", 그림은 분위기에 맞게 만들어줘.
다시 올리기	고친 거 저장하고 Vercel에도 다시 올려줘

PART 02

이제 내가 원하는 걸 만들자

예제를 넘어 나만의 프로젝트로 확장하기

6장

막연한 아이디어를 실제 앱으로 구체화하기

이 장을 마치면 "이런 거 있으면 좋겠다"는 생각이 한 문단짜리 앱 설명으로 바뀌어요.

PART 02에서는 정산기 대신 내 아이디어로 앱을 만들어요. 예시로 **가족 냉장고 유통기한 앱**을 끝까지 같이 만들어요.

1단계. 불편한 순간 적기

앱 아이디어는 "불편했던 순간"에서 나와요. 메모장에 세 줄만 적어 봐요.

질문	예시
언제 불편했나요?	냉장고 뒤쪽에서 유통기한 지난 우유를 또 발견했을 때
누가 겪나요?	우리 가족 네 명
지금은 어떻게 하나요?	그냥 냉장고 문을 열고 하나씩 봐요

2단계. AI와 아이디어 다듬기

아직 아무것도 만들지 말고 대화만 해요.

복사해서 쓰기

```
앱 아이디어가 있어. 아직 만들지 말고 같이 다듬어줘. 나 비개발자야.
불편한 점: 냉장고에서 유통기한 지난 음식을 자꾸 늦게 발견해.
쓰는 사람: 우리 가족 네 명.
나한테 질문 5개만 해서 이 앱이 꼭 해야 할 일을 정리해줘.
```

AI가 질문하면 아는 만큼 대답해요. 모르면 "모르겠어. 추천해줘"라고 하면 돼요.

3단계. 한 문단으로 정리하기

복사해서 쓰기

```
지금까지 이야기한 걸 앱 설명 한 문단으로 정리해줘. 누가, 언제, 무엇을 하려고 쓰는지가 들어가게.
```

이런 문단이 나와요.

> 우리 가족 네 명이 장을 보고 온 날 음식 이름과 유통기한을 적어 두면, 기한이 가까운 순서로 보여주고 3일 남은 음식은 빨간색으로 표시해 주는 앱.

예시예요.

4단계. 비슷한 앱 찾아보기

복사해서 쓰기

```
이 아이디어랑 비슷한 앱이나 서비스가 이미 있는지 알려줘. 있다면 내 앱이 다르게 할 수 있는 점 2개만.
```

이미 있어도 괜찮아요. 내가 쓰기 편한 버전을 만드는 게 목적이에요.

5단계. 파일로 남기기

새 폴더(`냉장고`)를 만들어 고르고, 앞의 대화를 정리해 저장해요.

복사해서 쓰기

```
지금까지 정리한 앱 설명을 기획.md 파일로 저장해줘.
```

막히면

이럴 때	이렇게 하세요
아이디어가 아예 없음	"나는 OO 일을 하고, 평소에 OO가 불편해. 화면 하나로 만들 수 있는 앱 아이디어 5개 추천해줘"
설명이 너무 거창해짐	"더 작게 줄여줘. 처음엔 가족 네 명만 쓰는 걸로"

이 장의 문장

상황	붙여 넣을 문장
아이디어 다듬기	앱 아이디어가 있어. 아직 만들지 말고 같이 다듬어줘. 나 비개발자야. 불편한 점: OO. 쓰는 사람: OO. 나한테 질문 5개만 해서 이 앱이 꼭 해야 할 일을 정리해줘.

상황	붙여 넣을 문장
한 문단 정리	지금까지 이야기한 걸 앱 설명 한 문단으로 정리해줘. 누가, 언제, 무엇을 하려고 쓰는지가 들어가게.
비슷한 앱 찾기	이 아이디어랑 비슷한 앱이나 서비스가 이미 있는지 알려줘. 있다면 내 앱이 다르게 할 수 있는 점 2개만.
파일로 남기기	지금까지 정리한 앱 설명을 기획.md 파일로 저장해줘.
아이디어 추천	나는 OO 일을 하고, 평소에 OO가 불편해. 화면 하나로 만들 수 있는 앱 아이디어 5개 추천해줘

7장

AI와 함께 기능·화면·사용 흐름 정하기

이 장을 마치면 앱에 들어갈 기능, 화면, 사용 순서가 정리된 기획 문서가 생겨요.

6장의 `기획.md`에 이어서 써요. 여전히 아무것도 만들지 않아요.

1단계. 기능 목록 뽑기

복사해서 쓰기

```
기획.md를 읽고, 이 앱에 들어갈 기능을 목록으로 뽑아줘. 꼭 필요한 것과 있으면 좋은 것으로 나눠서.
```

꼭 필요한 것	있으면 좋은 것
음식 이름과 유통기한 넣기	사진으로 등록
기한 가까운 순으로 보기	가족에게 알림
다 먹은 음식 지우기	장보기 목록

예시예요.

2단계. 화면 정하기

복사해서 쓰기

```
꼭 필요한 기능만으로 화면을 몇 개 만들면 될지 정해줘. 화면마다 무엇이 보이는지 그림 그리듯 글로 설명해줘.
```

처음에는 화면이 **하나나 둘**이면 좋아요. 셋 이상이면 줄여 달라고 해요.

복사해서 쓰기

```
화면을 하나로 합칠 수 있으면 합쳐줘.
```

3단계. 사용 흐름 정하기

사용하는 사람이 처음부터 끝까지 무엇을 누르는지 정해요.

복사해서 쓰기

```
엄마가 장을 보고 와서 우유를 등록하고, 3일 뒤에 확인하는 순서를 한 단계씩 적어줘. 누르는 버튼까지.
```

이렇게 정리돼요.

1. 앱을 연다
2. "추가" 버튼을 누른다
3. "우유", 날짜를 고르고 저장한다
4. 3일 뒤 앱을 열면 우유가 맨 위에 빨간색으로 보인다

예시예요.

흐름이 어색한 곳이 보이면 고쳐 달라고 해요.

복사해서 쓰기

```
날짜 고르는 게 번거로울 것 같아. "3일 뒤", "1주 뒤" 같은 버튼으로 고를 수 있게 흐름을 바꿔줘.
```

4단계. 기획 문서 완성하기

복사해서 쓰기

```
지금까지 정한 기능 목록, 화면 설명, 사용 흐름을 기획.md에 이어서 정리해줘.
```

막히면

이럴 때	이렇게 하세요
기능이 너무 많이 나옴	"꼭 필요한 건 3개만 남겨줘. 나머지는 나중 목록으로"
화면 설명이 상상이 안 됨	"이 화면을 아주 간단한 그림으로 먼저 만들어서 보여줘. 기능은 안 돼도 괜찮아"

이 장의 문장

상황	붙여 넣을 문장
기능 목록	기획.md를 읽고, 이 앱에 들어갈 기능을 목록으로 뽑아줘. 꼭 필요한 것과 있으면 좋은 것으로 나눠서.
화면 정하기	꼭 필요한 기능만으로 화면을 몇 개 만들면 될지 정해줘. 화면마다 무엇이 보이는지 그림 그리듯 글로 설명해줘.
사용 흐름	OO가 OO하는 순서를 한 단계씩 적어줘. 누르는 버튼까지.
화면 그림 먼저	이 화면을 아주 간단한 그림으로 먼저 만들어서 보여줘. 기능은 안 돼도 괜찮아
기획 정리	지금까지 정한 기능 목록, 화면 설명, 사용 흐름을 기획.md에 이어서 정리해줘.

8장

처음부터 크게 만들지 않는 법

이 장을 마치면 꼭 필요한 기능 하나로 된 첫 버전을 만들고, 나머지는 "나중 목록"에 남겨둘 수 있어요.

기획을 하다 보면 하고 싶은 게 계속 늘어나요. 한 번에 다 만들려고 하면 거의 반드시 꼬여요. 처음 버전은 **써볼 수 있는 가장 작은 것**으로 만들어요.

1단계. 첫 버전 기능 하나만 고르기

복사해서 쓰기

```
기획.md를 읽고, 이 앱이 쓸모 있으려면 꼭 있어야 하는 기능 딱 하나만 골라줘. 나머지는 "나중에"로 옮겨줘.
```

냉장고 앱이라면 이렇게 돼요.

- **첫 버전:** 음식 이름과 유통기한을 넣으면 기한 가까운 순으로 보여주기
- **나중에:** 사진 등록, 가족 알림, 장보기 목록, 여러 사람이 같이 보기

예시예요.

2단계. 나중 목록 따로 두기

복사해서 쓰기

```
"나중에" 기능들을 나중에할것.md 파일로 저장해줘. 첫 버전을 다 만들기 전에는 손대지 않을 거야.
```

만들다가 좋은 생각이 나면 바로 만들지 말고 여기에 적어요.

복사해서 쓰기

```
"냉장고 칸별로 나눠 보기"를 나중에할것.md에 추가해줘. 지금은 만들지 마.
```

3단계. 첫 버전 만들기

복사해서 쓰기

```
기획.md의 첫 버전 기능만으로 앱을 만들어줘. 나 비개발자야. 브라우저에서 바로 열리는 파일 하나로 만들고, 다 만들면 화면에 띄워줘.
```

4단계. 며칠 써보기

바로 다음 기능을 붙이지 말고 실제로 며칠 써봐요. 쓰다 보면 기획 때 생각한 것과 진짜 필요한 게 달라요.

써본 뒤 이렇게 정리해요.

복사해서 쓰기

```
며칠 써보니까 이런 점이 불편했어: OO. 나중에할것.md를 보고 다음에 만들 기능 하나를 추천해줘.
```

첫 버전이 너무 큰지 확인하는 법

이러면 너무 커요	이렇게 줄여요
화면이 셋 이상	하나로 합치거나 하나만 남기기
로그인이 필요함	처음엔 나 혼자 쓰는 버전으로
여러 사람이 같이 써야 함	처음엔 내 폰에서만 되게
결제·알림이 들어감	나중 목록으로

이 장의 문장

상황	붙여 넣을 문장
기능 하나 고르기	기획.md를 읽고, 이 앱이 쓸모 있으려면 꼭 있어야 하는 기능 딱 하나만 골라줘. 나머지는 "나중에"로 옮겨줘.
나중 목록 만들기	"나중에" 기능들을 나중에할것.md 파일로 저장해줘. 첫 버전을 다 만들기 전에는 손대지 않을 거야.

상황	붙여 넣을 문장
나중 목록에 추가	"OO"를 나중에할것.md에 추가해줘. 지금은 만들지 마.
첫 버전 만들기	기획.md의 첫 버전 기능만으로 앱을 만들어줘. 나 비개발자야. 브라우저에서 바로 열리는 파일 하나로 만들고, 다 만들면 화면에 띄워줘.
다음 기능 고르기	며칠 써보니까 이런 점이 불편했어: OO. 나중에할것.md를 보고 다음에 만들 기능 하나를 추천해줘.

9장

AI에게 개발 계획을 맡기고 단계별로 실행하기

이 장을 마치면 큰 기능도 AI가 세운 계획대로 한 단계씩 확인하면서 완성할 수 있어요.

8장에서 고른 다음 기능(예: 가족이 같이 보기)처럼 큰 기능은 한 문장으로 바로 시키면 반쯤 되다 말기 쉬워요. 계획부터 받아요.

1단계. 만들기 전에 계획부터 받기

복사해서 쓰기

```
OO 기능을 넣고 싶어. 아직 만들지 말고, 화면이 어떻게 바뀌는지랑 해야 할 일을 단계별로 정리해줘. 내가 OK하면 그때 만들어.
```

2단계. 계획 읽고 고치기

복사해서 쓰기

```
3번은 빼고, 버튼은 목록 맨 위에 있었으면 좋겠어. 이렇게 고쳐서 계획 다시 보여줘.
```

계획이 마음에 들 때까지 반복해요. 이 단계에서는 아무것도 바뀌지 않아서 마음껏 고쳐도 돼요.

계획이 어려우면 이렇게 보내요.

복사해서 쓰기

```
계획을 쉬운 말로, 5단계 이내로 줄여줘
```

3단계. 계획을 파일로 남기기

복사해서 쓰기

```
지금 정한 계획을 계획.md 파일로 저장해줘. 끝난 단계는 체크 표시해줘.
```

4단계. 한 단계씩 만들기

복사해서 쓰기

```
계획.md의 1단계만 먼저 만들어줘. 다 되면 멈추고 나한테 확인받아.
```

화면을 확인하고 잘 되면 저장한 뒤 다음 단계로 가요.

복사해서 쓰기

```
잘 됐. 여기까지 저장하고 계획.md에 1단계 체크한 다음, 2단계 만들어줘. 다 되면 멈추고 확인받아.
```

5단계. 다 끝나면 정리하기

복사해서 쓰기

```
계획.md가 다 끝났어. 무엇을 만들었는지 세 줄로 요약하고, 나중에할것.md에서 끝난 건 지워줘.
```

막히면

이럴 때	이렇게 하세요
계획 없이 바로 만들어 버림	"멈춰. 방금 만든 거 취소하고, 계획부터 보여줘"
한 단계가 너무 큼	"그 단계를 더 작게 나눠줘"
중간에 방향을 바꾸고 싶음	"계획.md의 남은 단계를 이렇게 바꿔줘: OO"

이 장의 문장

상황	붙여 넣을 문장
계획부터 받기	OO 기능을 넣고 싶어. 아직 만들지 말고, 화면이 어떻게 바뀌는지랑 해야 할 일을 단계별로 정리해줘. 내가 OK하면 그때 만들어.
계획 고치기	OO는 빼고, OO는 OO했으면 좋겠어. 이렇게 고쳐서 계획 다시 보여줘.
계획 파일	지금 정한 계획을 계획.md 파일로 저장해줘. 끝난 단계는 체크 표시해줘.

상황	붙여 넣을 문장
한 단계씩	계획.md의 1단계만 먼저 만들어줘. 다 되면 멈추고 나한테 확인받아.
다음 단계	잘 돼. 여기까지 저장하고 계획.md에 체크한 다음, 다음 단계 만들어줘. 다 되면 멈추고 확인받아.
마무리 정리	계획.md가 다 끝났어. 무엇을 만들었는지 세 줄로 요약하고, 나중에할것.md에서 끝난 건 지워줘.

PART 03

바이브코딩을 잘하는 사람들의 노하우

시간과 비용을 아끼고 실패를 줄이는 실전 기술

10장

AI가 내 말을 제대로 이해하게 만드는 요청법

이 장을 마치면 한 번에 원하는 결과가 나오는 요청을 쓸 수 있어요.

AI는 내 화면을 보고 있지 않아요. 내가 말해준 것만 가지고 일해요. 그래서 말하는 방식만 바꿔도 결과가 크게 달라져요.

원칙 1. 한 번에 하나만

이렇게 말하면	이렇게 바꿔요
색 바꾸고 버튼 키우고 소리도 넣어줘	먼저 배경색만 바꿔줘 (확인 후 다음 것)

하나가 잘못됐을 때 어느 것 때문인지 알 수 있어요.

원칙 2. 보이는 것을 말하기

"안 돼"는 판단이고, "눌러도 아무 반응이 없어"는 관찰이에요. AI는 관찰을 들어야 원인을 찾아요.

이렇게 말하면	이렇게 바꿔요
버튼이 안 돼	계산 버튼을 눌러도 아무 반응이 없어
화면이 이상해	휴대폰으로 열었더니 버튼이 화면 오른쪽 밖으로 잘려

원칙 3. 원하는 결과까지 말하기

이렇게 말하면	이렇게 바꿔요
더 예쁘게 해줘	배경을 밝은 민트색으로, 글자는 더 크게 해줘
버튼 좀 고쳐줘	버튼이 화면 가운데에 있었으면 좋겠어

기본 틀

뭔가 이상하면 이 틀에 넣어요.

복사해서 쓰기

```
OO를 했더니 OO가 보여. 원래는 OO가 돼야 해.
```

새로 만들어 달라고 할 때는 이 틀이에요.

복사해서 쓰기

```
[누가] 쓰는 [무엇을] 하는 기능 만들어줘. 화면은 [어떤 느낌]이었으면 좋겠어.
```

예시와 사진 보여주기

말로 설명하기 어려우면 보여주는 게 가장 빨라요.

- 마음에 드는 앱 화면을 캡처해서 붙여 넣고 "이런 느낌으로 바꿔줘"
- 넣을 데이터 예시를 직접 적어 주기: "예를 들면 우유, 10월 28일"

컴퓨터	화면 찍기	붙여 넣기
Mac	`Control + Shift + Command + 4` 누르고 드래그	대화창에서 `Command + V`
Windows	`Windows 키 + Shift + S` 누르고 드래그	대화창에서 `Ctrl + V`

모를 때는 물어보기

복사해서 쓰기

```
이걸 하려면 어떤 방법들이 있어? 쉬운 말로, 추천하는 것 하나를 먼저 이유와 함께 말해줘.
```

이 장의 문장

상황	붙여 넣을 문장
문제 설명 틀	OO를 했더니 OO가 보여. 원래는 OO가 돼야 해.
새 기능 틀	[누가] 쓰는 [무엇을] 하는 기능 만들어줘. 화면은 [어떤 느낌]이었으면 좋겠어.

상황	붙여 넣을 문장
참고 화면	(참고 화면 붙여 넣기) 이런 느낌으로 바꿔줘
방법 묻기	이걸 하려면 어떤 방법들이 있어? 쉬운 말로, 추천하는 것 하나를 먼저 이유와 함께 말해줘.

11장

기존 기능을 망가뜨리지 않고 수정하는 법

이 장을 마치면 AI가 내가 말한 곳만 고치고, 그 약속을 매번 말하지 않아도 지켜요.

기능이 많아지면 이런 일이 생겨요. 복사 버튼 글자만 바꿔 달라고 했는데 계산하는 부분까지 손대서 금액이 틀어지는 거예요. AI는 고치면서 "이왕 하는 김에" 주변도 정리하려고 할 때가 있어요.

1단계. 고치기 전에 저장

복사해서 쓰기

```
여기까지 저장해줘. 이름은 "수정 전"으로.
```

2단계. 고칠 곳만 고치라고 말하기

복사해서 쓰기

```
OO만 고쳐줘. 다른 기능은 건드리지 말고, 바꾸기 전에 어디를 바꿀지 먼저 말해줘. 내가 OK하면 그때 바꿔.
```

나: 복사 버튼 글자만 "정산 문구 복사"로 고쳐줘. 다른 기능은 건드리지 말고, 바꾸기 전에 어디를 바꿀지 먼저 말해줘. 내가 OK하면 그때 바꿔.

AI: 버튼 글자 한 곳만 바꿀게요. 계산 부분은 건드리지 않아요. 진행할까요?

나: OK

예시 대화예요.

3단계. 계획이 이상하면 되묻기

아래가 보이면 OK하지 마세요.

- 작은 수정인데 여러 곳을 바꾼다고 해요
- "정리", "개선"처럼 내가 말하지 않은 일이 있어요
- 무언가를 지운다고 해요

복사해서 쓰기

```
왜 그것까지 바꿔야 해? 꼭 필요한 것만 남기고 다시 말해줘.
```

4단계. 규칙 파일로 한 번만 말하기

2단계 문장을 매번 붙이기 귀찮으면 규칙 파일에 저장해요. AI는 작업을 시작할 때마다 이 파일을 먼저 읽어요.

복사해서 쓰기

```
아래 규칙을 CLAUDE.md 파일에 저장해줘.
- 내가 말한 부분만 고친다. 다른 기능은 건드리지 않는다.
- 바꾸기 전에 어디를 바꿀지 먼저 말하고, 내가 OK하면 바꾼다.
- 나는 비개발자다. 쉬운 말로 설명한다.
- 잘 되는 상태가 되면 저장할지 물어본다.
```

새 대화를 열고 확인해요.

복사해서 쓰기

```
이 폴더의 규칙 파일에 뭐라고 적혀 있어? 한 줄씩 요약해줘.
```

같은 말을 두 번 하게 되면 규칙으로 옮겨요.

복사해서 쓰기

```
방금 내가 한 말을 규칙 파일에 한 줄로 추가해줘.
```

막히면

이럴 때	이렇게 하세요
이미 다른 게 고장 남	"'수정 전'으로 저장한 때로 되돌려줘. 저장 기록은 지우지 말고."
규칙이 있는데 안 지킴	"규칙 파일 다시 읽고, 방금 한 작업이 규칙을 지켰는지 확인해줘"
규칙이 없다고 함	"CLAUDE.md가 어느 폴더에 있어? 작업 폴더 맨 위로 옮겨줘"

이 장의 문장

상황	붙여 넣을 문장
고칠 곳만	OO만 고쳐줘. 다른 기능은 건드리지 말고, 바꾸기 전에 어디를 바꿀지 먼저 말해줘. 내가 OK하면 그때 바꿔.
되묻기	왜 그것까지 바꿔야 해? 꼭 필요한 것만 남기고 다시 말해줘.
규칙 파일 만들기	아래 규칙을 CLAUDE.md 파일에 저장해줘. (규칙 붙여 넣기)
규칙 확인	이 폴더의 규칙 파일에 뭐라고 적혀 있어? 한 줄씩 요약해줘.
규칙 더하기	방금 내가 한 말을 규칙 파일에 한 줄로 추가해줘.

12장

오류가 반복될 때 해결하는 방법

이 장을 마치면 같은 오류가 계속 날 때 빙빙 돌지 않고 빠져나올 수 있어요.

같은 문제를 세 번 말했는데도 안 고쳐지면, 네 번째도 같은 방식으로 말하지 마세요. AI가 같은 추측을 반복하면서 점점 더 꼬이는 경우가 많아요.

1단계. 에러를 통째로 보여주기

에러 메시지는 읽을 수 없어도 괜찮아요. 일부만 옮기지 말고 처음부터 끝까지 복사해서 보내요.

복사해서 쓰기

```
00를 했더니 아래 에러가 떠.
(에러 메시지 통째로 붙여 넣기)
```

화면에 에러가 안 보이고 그냥 안 될 때는 이렇게 물어요.

복사해서 쓰기

```
화면에는 에러가 안 보이는데 00가 안 돼. 숨어 있는 에러가 있는지 직접 확인해줘.
```

2단계. 고치지 말고 원인부터 묻기

복사해서 쓰기

```
아직 고치지 말고, 왜 이런 일이 생기는지 가능한 원인을 쉬운 말로 3가지 설명해줘. 가장 가능성 높은 것부터.
```

AI가 설명하는 과정에서 진짜 원인을 찾는 경우가 많아요.

3단계. 하나씩 확인하기

복사해서 쓰기

```
1번 원인이 맞는지 먼저 확인해줘. 맞으면 그것만 고치고, 아니면 2번으로 넘어가.
```

4단계. 다른 방법으로 다시 만들기

같은 방식으로 계속 안 되면 그 부분만 새로 만들게 해요.

복사해서 쓰기

```
지금 방식은 그만하고, 그 기능만 아예 다른 방법으로 다시 만들어줘. 다른 기능은 건드리지 말고.
```

5단계. 새 대화로 시작하기

대화가 길어지면 AI가 앞의 잘못된 시도에 끌려다녀요.

복사해서 쓰기

```
지금 문제와 지금까지 시도한 방법을 문제정리.md에 정리해줘.
```

새 대화를 열고 같은 폴더를 고른 다음 보내요.

복사해서 쓰기

```
문제정리.md를 읽고, 이미 시도한 방법 말고 다른 방법으로 해결해줘.
```

6단계. 그래도 안 되면 되돌리기

복사해서 쓰기

```
마지막으로 잘 되던 저장으로 되돌려줘. 저장 기록은 지우지 말고.
```

되돌린 다음 기능을 더 작게 나눠서 다시 시작해요. (9장)

멈춰야 할 신호

- AI가 "고쳤습니다"라는데 화면은 그대로예요
- 하나를 고치면 다른 게 고장 나요
- AI가 앞에서 했던 수정을 다시 되돌리고 있어요

이 중 하나라도 보이면 2단계로 돌아가요.

이 장의 문장

상황	붙여 넣을 문장
에러 보여주기	OO를 했더니 아래 에러가 떠. (에러 메시지 통째로 붙여 넣기)
숨은 에러 찾기	화면에는 에러가 안 보이는데 OO가 안 돼. 숨어 있는 에러가 있는지 직접 확인해줘.
원인부터	아직 고치지 말고, 왜 이런 일이 생기는지 가능한 원인을 쉬운 말로 3가지 설명해줘. 가장 가능성 높은 것부터.
하나씩 확인	1번 원인이 맞는지 먼저 확인해줘. 맞으면 그것만 고치고, 아니면 2번으로 넘어가.
다른 방법	지금 방식은 그만하고, 그 기능만 아예 다른 방법으로 다시 만들어줘. 다른 기능은 건드리지 말고.
문제 정리	지금 문제와 지금까지 시도한 방법을 문제정리.md에 정리해줘.
새 대화에서	문제정리.md를 읽고, 이미 시도한 방법 말고 다른 방법으로 해결해줘.

13장

AI가 만든 기능이 정말 작동하는지 검증하는 법

이 장을 마치면 고칠 때마다 내가 하나하나 눌러보지 않아도, AI가 직접 확인하고 결과를 알려줘요.

AI가 "다 됐어요"라고 해도 정말 되는지는 따로 확인해야 해요. 기능이 늘수록 매번 손으로 다 눌러보기는 힘들어요.

1단계. 확인할 목록 만들기

복사해서 쓰기

```
이 앱에서 항상 맞아야 하는 것들을 확인 목록으로 만들어서 확인목록.md에 저장해줘.
```

정산기라면 이런 항목이 들어가요.

- 1차 120,000원 4명, 2차 27,000원 3명(민준 빠짐) → 민준 30,000원, 나머지 39,000원
- 새로고침해도 입력한 내용이 남는다
- 복사 버튼을 누르면 문구가 복사된다

예시예요.

2단계. 직접 열어서 확인시키기

복사해서 쓰기

```
확인목록.md에 있는 걸 하나씩 직접 앱을 열어서 넣어보고 결과가 맞는지 확인해줘. 확인하지 못한 건 확인 못 했다고 말해줘.
```

틀린 게 있으면 그것만 고치게 해요.

복사해서 쓰기

```
틀린 것만 고쳐줘. 다른 건 건드리지 말고. 고친 다음 확인목록을 처음부터 다시 확인해줘.
```

3단계. 결과를 증거로 받기

"확인했어요"만으로는 부족해요. 무엇을 넣고 무엇이 나왔는지 받아요.

복사해서 쓰기

```
어떤 값을 넣었고 화면에 뭐가 나왔는지 항목마다 그대로 보여줘.
```

4단계. 내가 직접 보는 것도 남기기

AI가 확인할 수 없는 것도 있어요. 내 휴대폰에서 보이는 모습, 실제 송금 화면 같은 것들이요.

복사해서 쓰기

```
확인목록.md 중에서 네가 직접 확인할 수 없는 건 "직접 확인"이라고 따로 표시해줘.
```

5단계. 고칠 때마다 확인 붙이기

복사해서 쓰기

```
"고친 다음에는 확인목록.md로 다시 확인한다"를 규칙 파일에 추가해줘.
```

새 기능을 만들면 확인 목록에도 추가해요.

복사해서 쓰기

```
방금 만든 기능도 확인목록.md에 추가해줘.
```

이 장의 문장

상황	붙여 넣을 문장
확인 목록 만들기	이 앱에서 항상 맞아야 하는 것들을 확인 목록으로 만들어서 확인목록.md에 저장해줘.
직접 확인시키기	확인목록.md에 있는 걸 하나씩 직접 앱을 열어서 넣어보고 결과가 맞는지 확인해줘. 확인하지 못한 건 확인 못 했다고 말해줘.

상황	붙여 넣을 문장
틀린 것만 고치기	틀린 것만 고쳐줘. 다른 건 건드리지 말고. 고친 다음 확인목록을 처음부터 다시 확인해줘.
증거 받기	어떤 값을 넣었고 화면에 뭐가 나왔는지 항목마다 그대로 보여줘.
직접 확인 표시	확인목록.md 중에서 네가 직접 확인할 수 없는 건 "직접 확인"이라고 따로 표시해줘.
목록에 추가	방금 만든 기능도 확인목록.md에 추가해줘.

14장

AI가 이전 작업을 잊어버리지 않게 관리하는 법

이 장을 마치면 다음 날 새 대화를 열어도 AI가 어제 하던 일을 바로 이어서 해요.

AI는 새 대화를 열면 앞의 대화를 기억하지 못해요. 대화가 아주 길어져도 앞부분을 헷갈리기 시작해요. 그래서 기억할 것을 **파일**로 남겨요.

남겨둘 파일 네 가지

파일	무엇을 적나	어디서 배웠나
`CLAUDE.md`	지켜야 할 규칙	11장
`기획.md`	이 앱이 무엇인지, 기능·화면·흐름	6~7장
`계획.md`	지금 만드는 기능의 단계와 진행 상황	9장
`진행상황.md`	오늘 한 일, 다음에 할 일, 막힌 것	이 장

1단계. 하루 끝에 정리하기

복사해서 쓰기

```
오늘 한 일, 다음에 할 일, 아직 해결 못 한 문제를 진행상황.md에 정리해줘. 원래 내용이 있으면 맨 위에 오늘 날짜로 추가해.
```

2단계. 다음 날 이어서 시작하기

새 대화를 열고 같은 폴더를 고른 다음 보내요.

복사해서 쓰기

```
CLAUDE.md, 기획.md, 진행상황.md를 읽고, 다음에 할 일부터 이어서 하자. 시작하기 전에 지금 상황을 세 줄로 요약해줘.
```

요약이 맞는지 확인하고 시작해요.

3단계. 대화가 길어지면 갈아타기

이런 신호가 보이면 새 대화로 옮겨요.

- 아까 정한 걸 AI가 다시 물어봐요
- 이미 고친 문제를 또 고치려고 해요
- 답이 눈에 띄게 느려져요

복사해서 쓰기

```
지금까지 이 대화에서 한 일을 진행상황.md에 정리해줘. 새 대화로 옮길 거야.
```

4단계. 결정한 것 남기기

중요한 결정은 이유와 함께 남겨요. 나중에 "왜 이렇게 했지?" 할 때 도움이 돼요.

복사해서 쓰기

```
방금 정한 것(OO)을 이유와 함께 기획.md의 "결정한 것" 아래에 적어줘.
```

이 장의 문장

상황	붙여 넣을 문장
하루 정리	오늘 한 일, 다음에 할 일, 아직 해결 못 한 문제를 진행상황.md에 정리해줘. 원래 내용이 있으면 맨 위에 오늘 날짜로 추가해.
이어서 시작	CLAUDE.md, 기획.md, 진행상황.md를 읽고, 다음에 할 일부터 이어서 하자. 시작하기 전에 지금 상황을 세 줄로 요약해줘.
대화 갈아타기	지금까지 이 대화에서 한 일을 진행상황.md에 정리해줘. 새 대화로 옮길 거야.
결정 남기기	방금 정한 것(OO)을 이유와 함께 기획.md의 "결정한 것" 아래에 적어줘.

15장

프로젝트가 복잡해져도 흔들리지 않는 작업 방식

이 장을 마치면 기능이 많아져도 어디에 뭐가 있는지 알고, 하나를 고쳐도 다른 게 무너지지 않게 일할 수 있어요.

습관 1. 한 대화에는 한 가지 일만

"버튼 고치기"와 "로그인 붙이기"를 한 대화에서 섞지 않아요. 일이 끝나면 저장하고, 다음 일은 새 대화에서 시작해요. (14장)

습관 2. 작업 순서 고정하기

어떤 기능이든 이 순서로 해요.

1. 저장 (4장)
2. 계획 받기 (9장)
3. 한 단계씩 만들기
4. 확인 목록으로 확인 (13장)
5. 저장하고 올리기

이 순서를 규칙 파일에 넣어 두면 AI도 따라와요.

복사해서 쓰기

```
"새 기능을 만들 때는 저장 → 계획 → 한 단계씩 → 확인목록 확인 → 저장 순서로 한다"를 규칙 파일에 추가해줘.
```

습관 3. 파일 정리 맡기기

기능이 늘면 파일 하나가 너무 길어져요. 그러면 AI도 고치다 실수하기 쉬워요.

복사해서 쓰기

```
지금 파일 구조를 쉬운 말로 설명해줘. 파일이 너무 길거나 섞여 있으면, 기능은 그대로 두고 정리하는 계획부터 보여줘.
```

정리하기 전에 꼭 저장하고, 정리한 뒤에는 확인 목록으로 전부 확인해요.

습관 4. 지도 만들기

복사해서 쓰기

```
이 앱의 화면과 기능이 어느 파일에 있는지 지도.md에 표로 정리해줘. 앞으로 파일이 바뀌면 같이 고쳐줘.
```

뭘 고치고 싶을 때 지도를 보면 "OO 파일의 OO만 고쳐줘"라고 정확히 말할 수 있어요.

습관 5. 실험은 따로 해보기

크게 바꿔 보고 싶은데 지금 잘 되는 걸 망칠까 걱정되면 따로 갈래를 만들어 실험해요.

복사해서 쓰기

```
지금 상태는 그대로 두고, 실험용 갈래를 따로 만들어서 거기서 00를 해봐줘. 마음에 들면 합치고, 아니면 버릴 거야.
```

복사해서 쓰기

```
실험한 거 마음에 들어. 원래 쪽에 합쳐줘.
```

복사해서 쓰기

```
실험한 거 별로야. 실험 갈래는 버리고 원래 상태로 돌아가줘.
```

이 장의 문장

상황	붙여 넣을 문장
작업 순서 규칙	"새 기능을 만들 때는 저장 → 계획 → 한 단계씩 → 확인목록 확인 → 저장 순서로 한다"를 규칙 파일에 추가해줘.
파일 정리	지금 파일 구조를 쉬운 말로 설명해줘. 파일이 너무 길거나 섞여 있으면, 기능은 그대로 두고 정리하는 계획부터 보여줘.
지도 만들기	이 앱의 화면과 기능이 어느 파일에 있는지 지도.md에 표로 정리해줘. 앞으로 파일이 바뀌면 같이 고쳐줘.

상황	붙여 넣을 문장
실험 갈래	지금 상태는 그대로 두고, 실험용 갈래를 따로 만들어서 거기서 OO를 해봐줘. 마음에 들면 합치고, 아니면 버릴 거야.
실험 합치기	실험한 거 마음에 들어. 원래 쪽에 합쳐줘.
실험 버리기	실험한 거 별로야. 실험 갈래는 버리고 원래 상태로 돌아가줘.

16장

AI에게 반복 작업을 맡기고 사용 비용 줄이기

이 장을 마치면 자주 하는 부탁을 한 줄로 시키고, 사용량 한도에 덜 걸리게 일할 수 있어요.

반복 작업 맡기기

1단계. 자주 하는 부탁 모아두기

매번 길게 쓰는 부탁이 있다면 파일로 저장해요.

복사해서 쓰기

```
내가 자주 하는 부탁을 요청모음.md에 번호를 붙여서 저장해줘.
1. 저장하고 Vercel에 다시 올리기
2. 확인목록.md로 전부 확인하고 결과 보여주기
3. 오늘 한 일을 진행상황.md에 정리하기
```

다음부터는 한 줄이면 돼요.

복사해서 쓰기

```
요청모음.md의 1번 해줘
```

2단계. 마무리 한 번에 시키기

복사해서 쓰기

```
요청모음.md의 2번, 3번, 1번을 순서대로 해줘. 중간에 틀린 게 나오면 멈추고 알려줘.
```

3단계. 반복되는 입력은 AI에게

같은 형식의 내용을 여러 개 넣어야 할 때도 AI에게 맡겨요.

복사해서 쓰기

```
아래 목록을 앱에 한 번에 넣어줘.
우유 10월 28일, 두부 10월 26일, 계란 11월 3일
```

사용 비용 줄이기

AI 도구는 쓴 만큼 사용량이 닳아요. 아래만 지켜도 한도에 덜 걸려요.

습관	이유
대화가 길어지면 새 대화로 (14장)	긴 대화는 매번 앞 내용을 다시 읽느라 사용량이 커요
한 번에 하나만 부탁 (10장)	엉뚱한 결과를 다시 고치는 비용이 줄어요
계획부터 받기 (9장)	잘못 만들고 되돌리는 일이 줄어요
필요한 파일만 읽게 하기	모든 파일을 읽지 않아도 돼요
에러는 통째로 한 번에 (12장)	여러 번 주고받지 않아도 돼요

필요한 파일만 읽게 할 때는 이렇게 말해요.

복사해서 쓰기

```
지도.md를 보고, 이 문제와 관련된 파일만 읽어서 고쳐줘.
```

한도에 걸렸다는 안내가 뜨면 안내에 나온 시간만큼 기다리거나, 그사이 `진행상황.md` 를 정리해 두면 다시 시작할 때 사용량을 아낄 수 있어요.

이 장의 문장

상황	붙여 넣을 문장
요청 모음 만들기	내가 자주 하는 부탁을 요청모음.md에 번호를 붙여서 저장해줘. (부탁 목록 붙여 넣기)
모음에서 시키기	요청모음.md의 OO번 해줘
마무리 한 번에	요청모음.md의 2번, 3번, 1번을 순서대로 해줘. 중간에 틀린 게 나오면 멈추고 알려줘.
여러 개 한 번에 넣기	아래 목록을 앱에 한 번에 넣어줘. (목록 붙여 넣기)

상황	붙여 넣을 문장
필요한 파일만	지도.md를 보고, 이 문제와 관련된 파일만 읽어서 고쳐줘.

PART 04

진짜 서비스처럼 만들어보자

필요한 기능을 골라 붙이는 실전 레시피

17장

데이터를 저장하고 다시 불러오기

이 장을 마치면 창을 닫았다 열어도 입력한 내용이 남고, 지난 기록을 다시 불러올 수 있어요.

PART 04는 필요한 기능을 골라 붙이는 레시피예요. 예시는 PART 01의 정산기로 들어요. 내 앱에 붙일 때는 "정산"을 내 앱의 내용으로 바꾸면 돼요. 시작 전에 늘 저장해요.

1단계. 새로고침해 보기

정산기에 이름과 금액을 넣고 새로고침을 눌러 봐요. 입력한 게 다 사라지죠? 이걸 고쳐요.

2단계. 새로고침해도 남게

복사해서 쓰기

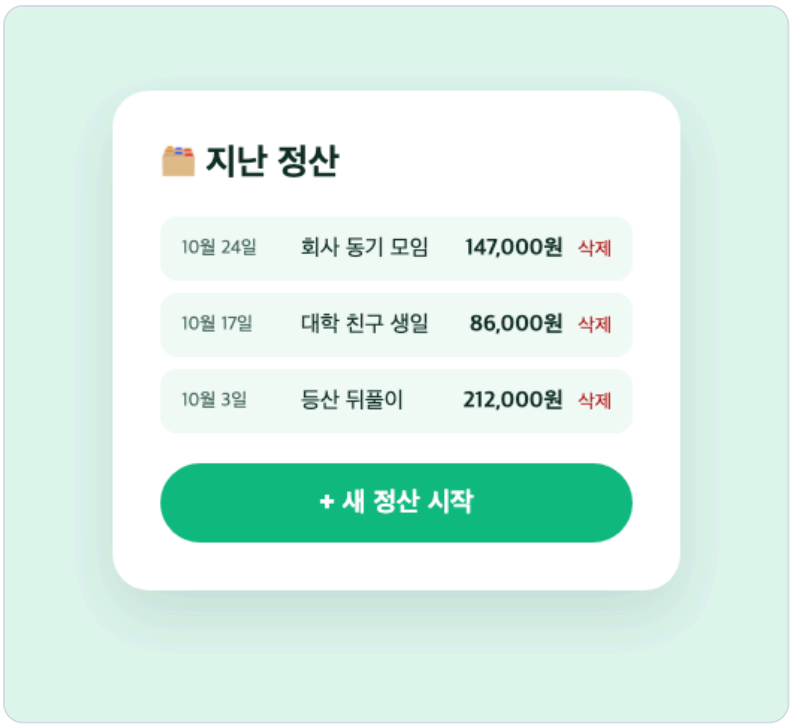
```
새로고침하거나 창을 닫았다 열어도 입력한 내용이 그대로 남게 해줘. 이 브라우저에만 저장하면 돼.
```

다시 넣고 새로고침해서 남아 있으면 성공이에요.

3단계. 지난 기록 목록

복사해서 쓰기

```
정산이 끝나면 "기록에 저장" 버튼으로 모임 이름이랑 날짜를 붙여서 남기고, 지난 정산 목록을 최신순으로 보여줘.
목록을 누르면 그때 정산을 다시 볼 수 있게.
```



예시 화면이에요.

4단계. 지우기

복사해서 쓰기

```
기록마다 삭제 버튼을 넣어줘. 누르면 정말 지울지 한 번 물어보게.
```

5단계. 파일로 내보내고 불러오기

브라우저 기록을 지우거나 휴대폰을 바꾸면 기록이 사라져요. 파일로 내보내 둘 수 있게 해요.

복사해서 쓰기

```
기록 전체를 파일 하나로 내려받는 "내보내기" 버튼이랑, 그 파일을 다시 넣으면 기록이 돌아오는 "불러오기" 버튼을 만들어줘.
```

알아둘 것

이 방법으로 저장한 기록은 **지금 쓰는 브라우저에만** 있어요. 다른 사람과 같은 데이터를 보려면 18장으로 가요.

막히면

이럴 때	이렇게 하세요
새로고침하면 여전히 사라짐	"새로고침했더니 입력한 게 사라져. 원래는 남아 있어야 해"
기록을 눌러도 안 열림	"지난 정산 목록을 눌렀더니 아무 일도 안 일어나. 원래는 그때 정산이 보여야 해"
불러오기 했더니 엉킴	저장한 때로 되돌리고 (4장) "불러오기는 기존 기록을 지우지 말고 뒤에 추가되게 해줘"

이 장의 문장

상황	붙여 넣을 문장
새로고침해도 남게	새로고침하거나 창을 닫았다 열어도 입력한 내용이 그대로 남게 해줘. 이 브라우저에만 저장하면 돼.
기록 목록	정산이 끝나면 "기록에 저장" 버튼으로 이름이랑 날짜를 붙여서 남기고, 지난 목록을 최신순으로 보여줘. 목록을 누르면 다시 볼 수 있게.
지우기 버튼	기록마다 삭제 버튼을 넣어줘. 누르면 정말 지울지 한 번 물어보게.
내보내기·불러오기	기록 전체를 파일 하나로 내려받는 "내보내기" 버튼이랑, 그 파일을 다시 넣으면 기록이 돌아오는 "불러오기" 버튼을 만들어줘.

18장

여러 사용자가 같은 데이터를 공유하기

이 장을 마치면 링크 하나로 모임 사람 모두가 같은 정산을 보고, 각자 "송금 완료"를 체크할 수 있어요.

17장 기록은 내 휴대폰에만 있어요. 여럿이 같은 데이터를 보려면 인터넷 한곳에 모아 둬야 해요. 그 보관함이 **데이터베이스**예요. 무료로 쓸 수 있는 **Supabase**를 써요.

1단계. Supabase 프로젝트 만들기 (직접)

1. supabase.com에 가입하고 **New project**를 눌러요.
2. 이름은 `moim-jungsan`, 지역(Region)은 **Seoul**을 골라요. 무료 요금제면 충분해요.
3. 데이터베이스 비밀번호는 따로 적어 두고 아무에게도 알려주지 마세요. AI에게도요.
4. 만들어지면 **Project Settings → API**에서 **Project URL**과 **publishable 키**를 복사해요. 이 둘은 앱에 넣어도 되는 값이에요. (24장)

2단계. 계획부터 받기

복사해서 쓰기

```
정산을 Supabase에 저장해서, 링크를 가진 사람 모두가 같은 정산을 보고 각자 "송금 완료"를 체크할 수 있게 하고 싶어. Project URL은 OO, publishable 키는 OO야. 아직 만들지 말고 계획부터 보여줘. 나 비개발자야.
```

계획에 아래 세 가지가 들어 있는지 확인해요.

복사해서 쓰기

```
계획에 이 세 가지를 넣어줘. 정산마다 추측할 수 없는 링크, 다른 정산은 못 보게 보안 설정 켜기, 표 만드는 문장은 내가 Supabase SQL Editor에 붙여 넣게 따로 주기.
```

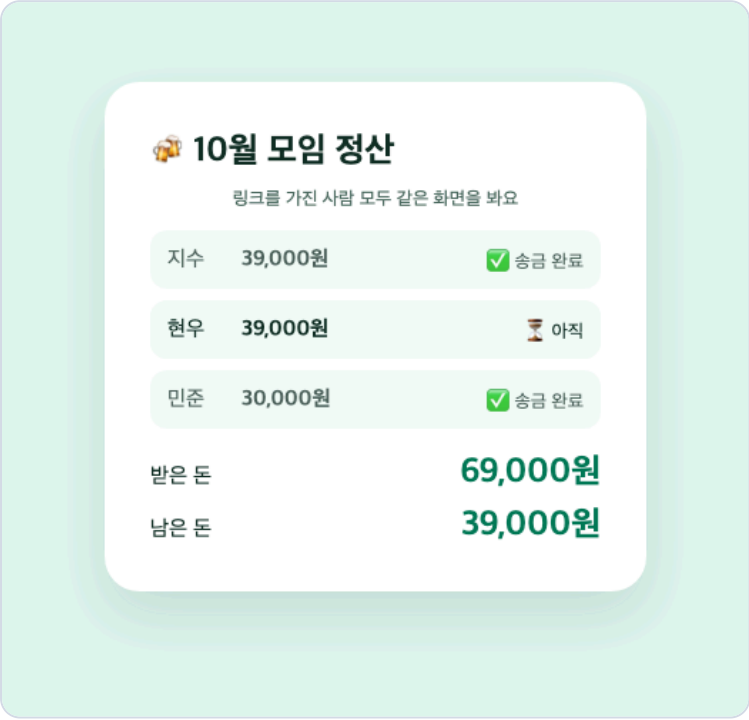
3단계. 표 만들기 (직접 붙여 넣기)

AI가 준 문장을 복사해서 Supabase의 **SQL Editor**에 붙여 넣고 **Run**을 눌러요. "Success"가 나오면 돼요.

4단계. 만들고 올리기

복사해서 쓰기

```
표 만들었어. 계획대로 앱을 만들고 Vercel에 다시 올려줘. 다 되면 확인목록.md에 "두 사람이 같은 링크로 들어가서 송금 완료를 체크하면 서로 보인다"를 추가해줘.
```



예시 화면이에요.

5단계. 친구와 확인하기

정산을 하나 만들고 링크를 친구에게 보내요. 친구가 "송금 완료"를 누르면 내 화면에도 보이는지 확인해요.

복사해서 쓰기

```
친구가 송금 완료를 누르면 내 화면에 새로고침 없이 바로 보이게 해줘
```

알아둘 것

- 링크를 아는 사람은 그 정산을 보고 체크할 수 있어요. 누가 누구인지 확인하려면 19장 로그인을 붙여요.
- Supabase 무료 요금제는 오래 안 쓰면 잠들어요. 대시보드에서 다시 켤 수 있어요.

막히면

이럴 때	이렇게 하세요
SQL Editor에서 빨간 에러	에러 문구를 통째로 복사해서 "표 만들다가 이 에러가 났어"와 함께 보내요
저장이 안 된다는 에러	"저장이 안 돼. 보안 설정 때문에 막힌 건지 확인하고, 필요한 설정 문장을 줘"
다른 사람 정산도 보임	바로 "다른 정산이 보여. 보안 설정을 다시 확인해줘"

이 장의 문장

상황	붙여 넣을 문장
같이 보기 계획	OO를 Supabase에 저장해서, 링크를 가진 사람 모두가 같은 내용을 보게 하고 싶어. Project URL은 OO, publishable 키는 OO야. 아직 만들지 말고 계획부터 보여줘. 나 비개발자야.
계획에 꼭 넣기	계획에 이 세 가지를 넣어줘. 추측할 수 없는 링크, 다른 데이터는 못 보게 보안 설정 켜기, 표 만드는 문장은 내가 Supabase SQL Editor에 붙여 넣게 따로 주기.
만들고 올리기	표 만들었어. 계획대로 앱을 만들고 Vercel에 다시 올려줘.
바로 보이게	다른 사람이 바꾸면 내 화면에 새로고침 없이 바로 보이게 해줘

19장

회원가입과 로그인 기능 붙이기

이 장을 마치면 이메일로 로그인해서 내가 만든 정산만 모아 보고, 다른 사람은 내 정산을 고칠 수 없어요.

18장에서는 링크만 있으면 누구나 체크할 수 있었어요. 로그인을 붙이면 "누가" 만들고 "누가" 체크했는지 알 수 있어요. 18장의 Supabase를 그대로 써요.

먼저 생각해 보기 로그인이 정말 필요한지 확인해요. 친구끼리 쓰는 정산기라면 18장만으로 충분할 때가 많아요. 로그인을 붙이면 쓰는 사람이 한 단계 더 거쳐야 해요.

1단계. 로그인 방식 고르기

처음에는 **이메일 로그인**을 추천해요. 이메일을 넣으면 로그인 링크가 메일로 와서, 비밀번호를 만들 필요가 없어요. 구글·카카오 로그인은 설정할 게 많아서 익숙해진 뒤에 붙여요.

2단계. 계획부터 받기

복사해서 쓰기

```
Supabase로 이메일 로그인을 붙이고 싶어. 이메일을 넣으면 로그인 링크가 메일로 오는 방식으로. 로그인하면 내가 만든 정산만 목록에 보이고, 정산은 만든 사람만 고칠 수 있게. 아직 만들지 말고 계획부터 보여줘. Supabase 대시보드에서 내가 직접 해야 하는 설정도 따로 알려줘.
```

3단계. 대시보드 설정 (직접)

AI가 알려준 대로 Supabase 대시보드에서 설정해요. 보통 이런 것들이에요.

- **Authentication**에서 이메일 로그인 켜기
- **URL Configuration**에 내 앱 주소(`https://이름.vercel.app`) 넣기

메뉴 이름은 바뀔 수 있으니, 안 보이면 화면을 캡처해서 "어디를 눌러야 해?"라고 물어요.

4단계. 만들고 올리기

복사해서 쓰기

```
설정 끝났어. 계획대로 만들고 Vercel에 다시 올려줘. 다 되면 확인목록.md에 "다른 계정으로 로그인하면 내 정산이 안 보인다"를 추가하고 확인해줘.
```

5단계. 두 계정으로 확인하기

내 이메일과 다른 이메일(가족 것)로 각각 로그인해서, 서로의 정산이 안 보이는지 확인해요.

알아둘 것

- 로그인 메일은 받은편지함이 아니라 스팸함에 들어갈 수 있어요.
- Supabase가 기본으로 보내주는 로그인 메일은 보낼 수 있는 양이 적어요. 친구 몇 명이 쓰는 정도에 맞아요.
- 로그인을 붙이면 이메일이라는 개인정보를 모으게 돼요. 24장의 개인정보 안내를 꼭 함께 해요.

막히면

이럴 때	이렇게 하세요
로그인 링크를 눌렀더니 이상한 주소로 감	"로그인 링크를 누르면 OO로 가. 내 앱 주소로 돌아오게 하려면 대시보드에서 뭘 바꿔야 해?"
메일이 안 옴	스팸함을 보고, 몇 분 기다렸다가 다시 시도해요
로그인했는데 정산이 안 보임	"로그인했는데 내가 만든 정산이 안 보여. 보안 설정 때문인지 확인해줘"

이 장의 문장

상황	붙여 넣을 문장
로그인 계획	Supabase로 이메일 로그인을 붙이고 싶어. 이메일을 넣으면 로그인 링크가 메일로 오는 방식으로. 로그인하면 내 것만 보이고, 만든 사람만 고칠 수 있게. 아직 만들지 말고 계획부터 보여줘. 대시보드에서 내가 직접 해야 하는 설정도 따로 알려줘.
만들고 올리기	설정 끝났어. 계획대로 만들고 Vercel에 다시 올려줘. 다 되면 확인목록.md에 "다른 계정으로 로그인하면 내 것이 안 보인다"를 추가하고 확인해줘.

상황	붙여 넣을 문장
이상한 주소로 갈 때	로그인 링크를 누르면 OO로 가. 내 앱 주소로 돌아오게 하려면 대시보드에서 뭘 바꿔야 해?

20장

외부 API와 AI 기능 연결하기

이 장을 마치면 다른 서비스의 정보를 가져오고, 영수증 사진을 올리면 AI가 금액을 읽어서 채워줘요.

API는 다른 서비스에 정보를 달라고 부탁하는 창구예요. 환율, 날씨 같은 정보를 가져오거나, 내 앱 안에서 Claude가 일하게 할 수 있어요.

레시피 1. 키가 필요 없는 정보 가져오기

해외여행 정산이라면 환율이 필요해요.

복사해서 쓰기

```
정산 금액을 달러나 엔으로 넣으면 오늘 환율로 원화로 바꿔주는 기능을 넣고 싶어. 무료로 쓸 수 있고 키가 필요 없는 환율 정보를 골라서 써줘. 아직 만들지 말고 어떤 걸 쓸지랑 계획부터 보여줘.
```

가져온 정보가 맞는지는 포털에서 환율을 검색해 한 번 비교해요.

레시피 2. 앱 안에 AI 넣기 (영수증 인식)

이번에는 앱 **안에서** Claude가 일해요. 내 앱이 Claude를 부를 열쇠, **API 키**가 필요해요.

1단계. Claude API 키 만들기 (직접)

1. console.anthropic.com에 가입해요. Claude 앱 구독과 별도로, 쓴 만큼 요금이 나가요.
2. 결제 수단을 등록하고 **한 달 사용 한도**를 작게 정해요. (25장)
3. **API Keys**에서 새 키를 만들고 복사해요.

중요 이 키는 비밀번호와 같아요. AI 채팅창에도 붙여 넣지 마세요. Vercel에만 넣어요.

2단계. 키를 Vercel에 넣기 (직접)

vercel.com에서 프로젝트 → **Settings → Environment Variables**로 가서 이름 `ANTHROPIC_API_KEY`, 값에 키를 넣고 저장해요.

3단계. 계획부터 받기

복사해서 쓰기

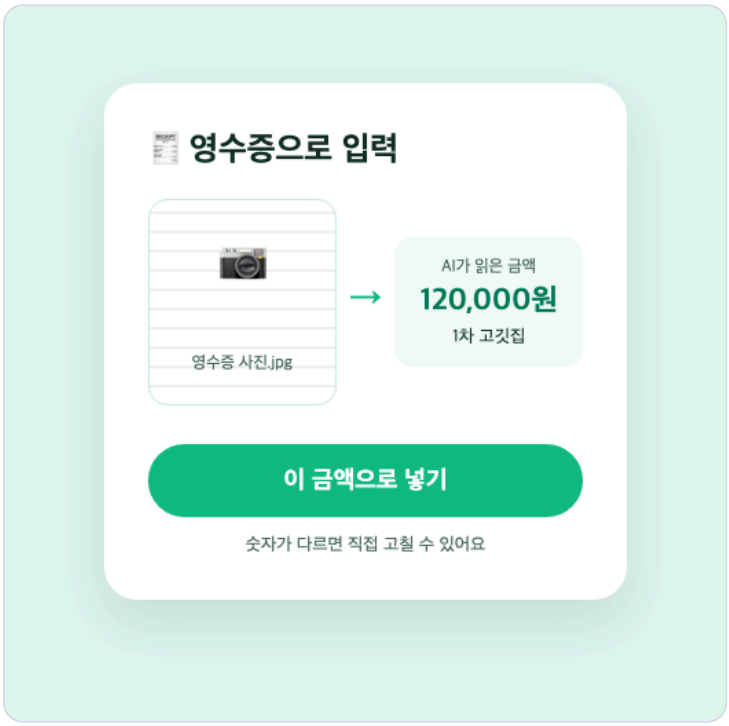
```
영수증 사진을 올리면 Claude API로 총 금액을 읽어서 금액 칸에 채워주는 기능을 넣고 싶어. API 키는 Vercel 환경변수 ANTHROPIC_API_KEY에 넣어 뒀어. 키가 화면 쪽 코드에 절대 들어가지 않게 서버 쪽에서 처리해줘.
아직 만들지 말고 계획부터 보여줘.
```

계획에 "키를 화면 쪽 코드에 넣는다"는 말이 있으면 OK하지 말고 다시 짜 달라고 해요.

4단계. 만들고 확인하기

복사해서 쓰기

```
좋아. 계획대로 만들고 Vercel에 올려줘. 사진에서 금액을 못 읽으면 "직접 입력해 주세요"라고 보여주게.
```



예시 화면이에요.

휴대폰에서 진짜 영수증을 찍어 올려 봐요. AI가 읽은 숫자는 틀릴 수 있으니, 넣기 전에 한 번 보고 고칠 수 있게 되어 있는지 확인해요.

5단계. 횟수 제한 걸기

링크를 아는 사람은 누구나 이 기능을 쓸 수 있고, 요금은 내 키로 나가요.

복사해서 쓰기

```
영수증 인식은 한 사람이 하루 10번까지만 쓸 수 있게 제한해줘. 넘으면 "오늘은 직접 입력해 주세요"라고 보여주게.
```

막히면

이럴 때	이렇게 하세요
"키가 없다"는 에러	Vercel 환경변수 이름이 `ANTHROPIC_API_KEY` 인지 확인하고 "다시 올려줘"
금액이 엉뚱하게 나옴	"영수증에서 '합계' 또는 '총액' 옆 숫자만 읽게 해줘"
키를 실수로 채팅에 붙여 넣음	콘솔에서 그 키를 바로 삭제하고 새 키를 만들어 다시 넣어요 (24장)

이 장의 문장

상황	붙여 넣을 문장
외부 정보 가져오기	OO 정보를 가져와서 OO하는 기능을 넣고 싶어. 무료로 쓸 수 있고 키가 필요 없는 걸 골라서 써줘. 아직 만들지 말고 어떤 걸 쓸지랑 계획부터 보여줘.
앱 안 AI 계획	OO하면 Claude API로 OO하는 기능을 넣고 싶어. API 키는 Vercel 환경변수 ANTHROPIC_API_KEY에 넣어 뒀어. 키가 화면 쪽 코드에 절대 들어가지 않게 서버 쪽에서 처리해줘. 아직 만들지 말고 계획부터 보여줘.
실패 처리	사진에서 금액을 못 읽으면 "직접 입력해 주세요"라고 보여주게
횟수 제한	이 기능은 한 사람이 하루 10번까지만 쓸 수 있게 제한해줘. 넘으면 안내 문구를 보여주게.

21장

결제·알림·파일 업로드 기능 추가하기

이 장을 마치면 송금 버튼, 내 메일로 오는 알림, 영수증 사진 보관을 정산기에 붙일 수 있어요.

레시피 1. 송금 버튼 (돈 받기)

1단계. 내 토스 송금 링크 만들기 (직접)

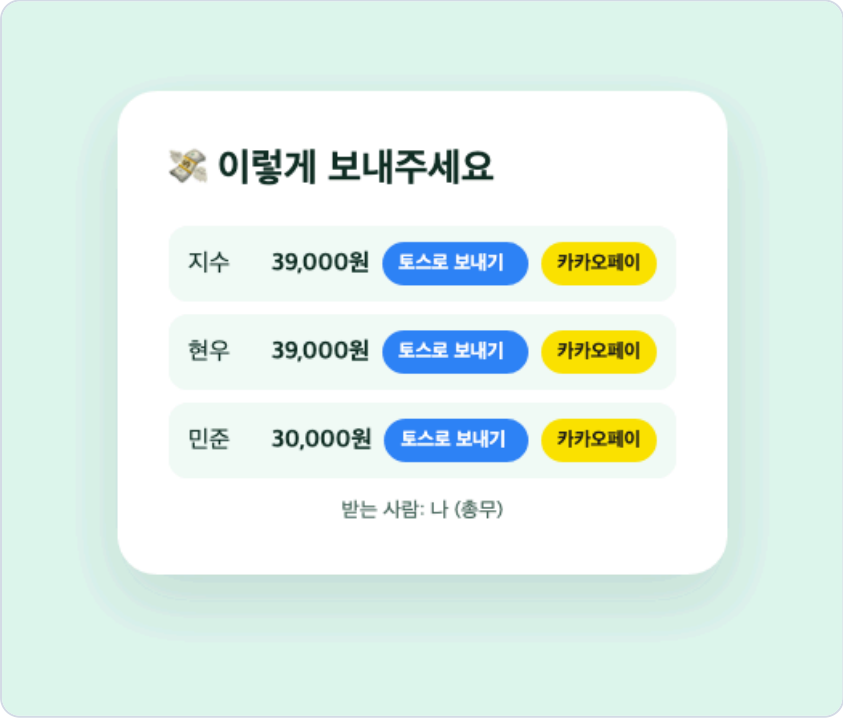
1. 토스 앱 → 전체 → **송금** 메뉴에서 **토스아이디**를 골라요.
2. 아이디를 정하면 `toss.me/내아이디` 주소가 생겨요. 링크로 쓸 거라서 영문 아이디가 좋아요.

카카오페이 송금 링크가 있다면 같은 방법으로 함께 넣을 수 있어요.

2단계. 버튼 붙이기

복사해서 쓰기

```
정산 결과에 사람마다 "토스로 보내기" 버튼을 넣고 싶어. 받는 사람은 나(총무)고, 내 송금 링크는 toss.me/내아이디야. 아직 만들지 말고 화면이 어떻게 바뀌는지 계획부터 보여줘.
```



휴대폰에서 버튼을 눌러 토스 송금 화면이 열리면 성공이에요. 금액은 송금 화면에서 직접 입력하게 될 수 있어서, 버튼 옆에 금액이 크게 보이는 게 좋아요.

카드 결제를 받으려면 상품을 팔고 카드로 돈을 받는 진짜 결제는 사업자등록과 결제 대행사(토스페이먼츠 같은 곳) 계약이 필요해요. 이 책의 범위를 넘어서, 친구끼리 돈을 주고받는 송금 링크까지만 다뤄요.

레시피 2. 내 메일로 알림 받기

친구가 "송금 완료"를 누르면 총무인 나에게 메일이 오게 해요. 18장의 Supabase를 쓰고 있어야 해요.

1. resend.com에 **내 이메일로** 가입하고 API 키를 만들어요. (직접) 무료 요금제는 내 도메인을 연결하기 전까지 가입한 이메일로만 보낼 수 있어서, 나에게 오는 알림에 딱 맞아요.
2. 키는 채팅에 붙여 넣지 말고, AI가 알려주는 곳(Supabase나 Vercel의 비밀 설정)에 직접 넣어요.

복사해서 쓰기

```
친구가 송금 완료를 누르면 내 이메일로 "00님 송금 완료" 알림이 오게 하고 싶어. 메일은 Resend로 보내고, 키는 코드에 넣지 말고 비밀 설정에 넣을 거야. 내가 직접 넣어야 할 이름이랑 위치를 알려주고, 계획부터 보여줘.
```

레시피 3. 영수증 사진 보관하기 (파일 업로드)

20장의 영수증 사진을 정산에 같이 남겨 둬요. Supabase의 파일 보관함(**Storage**)을 써요.

복사해서 쓰기

```
정산마다 영수증 사진을 올려서 보관하고, 나중에 다시 볼 수 있게 하고 싶어. Supabase Storage를 쓰고, 그 정산을 볼 수 있는 사람만 사진도 볼 수 있게 보안 설정을 해줘. 사진은 올리기 전에 작게 줄여줘. 아직 만들지 말고 계획부터 보여줘.
```

사진을 줄여서 올리면 무료 저장 공간을 아낄 수 있어요.

막히면

이럴 때	이렇게 하세요
송금 버튼을 눌러도 아무 일도 없음	휴대폰에서 눌러 봐요. 컴퓨터에서는 토스가 안 열릴 수 있어요
알림 메일이 안 옴	스팸함을 보고, "알림이 안 와. 보낸 기록이 있는지 확인해줘"
사진이 안 보임	"올린 사진이 안 보여. Storage 보안 설정 때문인지 확인해줘"

이 장의 문장

상황	붙여 넣을 문장
송금 버튼	정산 결과에 사람마다 "토스로 보내기" 버튼을 넣고 싶어. 받는 사람은 나(총무)고, 내 송금 링크는 toss.me/내아이디야. 아직 만들지 말고 계획부터 보여줘.
메일 알림	OO하면 내 이메일로 알림이 오게 하고 싶어. 메일은 Resend로 보내고, 키는 코드에 넣지 말고 비밀 설정에 넣을 거야. 내가 직접 넣어야 할 이름이랑 위치를 알려주고, 계획부터 보여줘.
사진 보관	OO마다 사진을 올려서 보관하고 다시 볼 수 있게 하고 싶어. Supabase Storage를 쓰고, 볼 수 있는 사람만 사진도 보게 보안 설정을 해줘. 사진은 올리기 전에 작게 줄여줘. 아직 만들지 말고 계획부터 보여줘.

22장

모바일에서도 앱처럼 사용할 수 있게 만들기

이 장을 마치면 휴대폰 홈 화면에 진짜 앱처럼 아이콘이 생기고, 주소창 없이 꽉 찬 화면으로 열려요.

앱스토어에 올리지 않아도 웹앱을 휴대폰에 앱처럼 설치할 수 있어요.

1단계. 휴대폰 화면부터 점검하기

복사해서 쓰기

```
휴대폰 화면에서 잘리거나 너무 작은 곳이 있는지 직접 확인하고 고쳐줘. 버튼은 손가락으로 누르기 편한 크기로.
```

2단계. 앱처럼 보이게 하기

복사해서 쓰기

```
휴대폰 홈 화면에 추가하면 진짜 앱처럼 보이게 해줘. 앱 이름은 "모임 정산", 아이콘은 초록색 배경에 💸 모양으로 만들어줘. 열면 주소창 없이 꽉 찬 화면으로 보이게.
```

3단계. 인터넷이 약해도 열리게

복사해서 쓰기

```
인터넷이 약한 곳에서도 마지막으로 열었던 화면은 보이게 해줘.
```

4단계. 올리고 설치해 보기

복사해서 쓰기

```
여기까지 저장하고 Vercel에도 다시 올려줘
```

휴대폰에서 주소를 열고 홈 화면에 추가해요.

- **아이폰 (Safari):** 아래 가운데 공유 버튼 → **홈 화면에 추가**

- **안드로이드 (Chrome):** 오른쪽 위 점 세 개 → **홈 화면에 추가**

예전에 추가한 아이콘이 있으면 지우고 다시 추가해요.

5단계. 고친 내용이 바로 반영되게

앱처럼 설치하면 예전 화면이 남아 있을 때가 있어요.

복사해서 쓰기

```
고친 내용이 휴대폰에 설치한 앱에도 바로 반영되게 해줘
```

막히면

이럴 때	이렇게 하세요
홈 화면 아이콘이 그대로	기존 아이콘을 지우고 다시 추가해요
주소창이 계속 보임	"홈 화면에서 열어도 주소창이 보여. 앱처럼 꽉 찬 화면으로 열리게 해줘"
새 버전이 안 보임	앱을 완전히 닫았다 다시 열어요. 그래도 그대로면 5단계 문장을 보내요

이 장의 문장

상황	붙여 넣을 문장
휴대폰 점검	휴대폰 화면에서 잘리거나 너무 작은 곳이 있는지 직접 확인하고 고쳐줘. 버튼은 손가락으로 누르기 편한 크기로.
앱처럼 설치	휴대폰 홈 화면에 추가하면 진짜 앱처럼 보이게 해줘. 앱 이름은 "OO", 아이콘은 OO 모양으로 만들어줘. 열면 주소창 없이 꽉 찬 화면으로 보이게.
인터넷 약할 때	인터넷이 약한 곳에서도 마지막으로 열었던 화면은 보이게 해줘.
바로 반영	고친 내용이 휴대폰에 설치한 앱에도 바로 반영되게 해줘

PART 05

출시하고 나서도 문제없이 운영하기

앱을 만들고 끝내지 않는 사람을 위한 노하우

23장

배포 전에 반드시 확인해야 할 것들

이 장을 마치면 링크를 사람들에게 보내기 전에 10분 만에 점검을 끝낼 수 있어요.

친구 몇 명에게 보내는 것과 단톡방·SNS에 올리는 것은 달라요. 많은 사람에게 공개하기 전에 이 장의 목록을 한 번 돌려요.

1단계. AI에게 점검 시키기

복사해서 쓰기

```
이 앱을 많은 사람에게 공개하려고 해. 아래 항목을 하나씩 직접 확인하고, 문제가 있는 것만 알려줘. 고치지는 말고.
1. 확인목록.md 전부 통과하는지
2. 휴대폰 화면에서 잘리거나 안 눌리는 곳이 없는지
3. 코드에 비밀 키가 들어간 곳이 없는지 (실제 값은 출력하지 말고 파일 이름만)
4. 에러가 나면 사용하는 사람에게 알아들을 수 있는 안내가 나오는지
5. 개인정보(이름, 이메일, 사진)를 모으는 곳이 있는지
```

문제가 나오면 하나씩 고쳐요. (11장)

2단계. 내가 직접 보는 것

확인할 것	어떻게
내 휴대폰에서 처음부터 끝까지 써보기	실제로 정산 하나를 만들어 봐요
다른 사람 휴대폰에서 열어보기	가족이나 친구 한 명에게 먼저 보내요
카톡 미리보기	나에게 링크를 보내서 제목과 그림을 확인해요 (5장)
주소 이름	이상하거나 헷갈리는 이름이 아닌지

3단계. 개인정보를 모은다면

1단계 5번에서 개인정보를 모은다고 나오면, 무엇을 왜 모으는지 화면에 알려야 해요.

복사해서 쓰기

```
이 앱에서 모으는 개인정보를 정리하고, 화면 맨 아래에 "무엇을, 왜, 언제까지 보관하는지" 짧은 안내를 넣어줘. 필요 없는 정보는 모으지 않게 바꿔줘.
```

자세한 건 24장에서 다뤄요.

4단계. 요금 한도 확인

API 키를 쓰는 기능(20장)이 있으면, 공개 전에 사용 한도를 꼭 걸어요. (25장)

5단계. 공개 직전 저장

복사해서 쓰기

```
공개 직전 상태로 저장해줘. 이름은 "첫 공개 버전"으로. 그리고 GitHub랑 Vercel에도 올려줘.
```

문제가 생기면 이 저장으로 돌아올 수 있어요. (26장)

이 장의 문장

상황	붙여 넣을 문장
공개 전 점검	이 앱을 많은 사람에게 공개하려고 해. 아래 항목을 하나씩 직접 확인하고, 문제가 있는 것만 알려줘. 고치지는 말고. (항목 붙여 넣기)
개인정보 안내	이 앱에서 모으는 개인정보를 정리하고, 화면 맨 아래에 "무엇을, 왜, 언제까지 보관하는지" 짧은 안내를 넣어줘. 필요 없는 정보는 모으지 않게 바꿔줘.
공개 직전 저장	공개 직전 상태로 저장해줘. 이름은 "첫 공개 버전"으로. 그리고 GitHub랑 Vercel에도 올려줘.

24장

개인정보·API 키·보안 사고 예방하기

이 장을 마치면 내 앱에 비밀 키가 새어 있지 않은지 점검하고, 개인정보를 꼭 필요한 만큼만 다루고, 사고가 나면 무엇부터 할지 알아요.

1단계. 공개돼도 되는 것, 안 되는 것

값	어디에 둬도 되나	이 책에서
송금 링크 (`toss.me/…`)	화면에 보여도 돼요	21장
Supabase **Project URL**, **publishable** 키	화면 쪽 코드에 넣어도 돼요 (보안 설정을 켰을 때)	18장
Claude **API 키**, Resend 키	Vercel·Supabase의 비밀 설정에만	20·21장
Supabase **secret** 키, 데이터베이스 비밀번호	앱에 넣지 않아요. 채팅에도 붙여 넣지 않아요	

헷갈리면 이것만 기억해요. **"secret", "API key", "password"가 붙은 값은 채팅에도, 코드에도 넣지 않는다.**

2단계. 새어 있는 키 찾기

복사해서 쓰기

```
이 프로젝트에서 비밀 키(API 키, secret, token, password)가 코드나 설정 파일에 그대로 들어간 곳 찾아줘. 실제 값은 출력하지 말고 파일 이름이랑 줄 번호만 알려줘.
```

뭔가 나오면 이렇게 고쳐요.

복사해서 쓰기

```
찾은 비밀 키를 코드에서 빼고 환경변수에서 읽게 바꿔줘. 실제 값은 출력하지 마. 내가 Vercel에 넣어야 할 이름만 알려줘.
```

올린 앱에서도 확인해요.

복사해서 쓰기

```
Vercel에 올린 주소의 화면 쪽 코드에 비밀 키가 보이지 않는지 확인해줘. 실제 값은 출력하지 말고 있다/없다만 알려줘.
```

3단계. 데이터 보안 설정 점검

18~21장에서 Supabase를 쓰고 있다면 꼭 확인해요.

복사해서 쓰기

```
Supabase의 모든 표와 파일 보관함에 보안 설정(RLS)이 켜져 있는지, 다른 사람 데이터를 읽거나 지울 수 있는 구멍이 없는지 점검해줘. 고치지는 말고 결과만 알려줘.
```

4단계. 개인정보는 꼭 필요한 만큼만

- 필요 없는 정보는 아예 모으지 않아요. 이름 대신 별명, 전화번호는 받지 않기.
- 모은다면 무엇을, 왜, 언제까지 보관하는지 화면에 알려요. (23장)
- 그만 쓰고 싶은 사람이 지울 수 있게 해요.

복사해서 쓰기

```
사용자가 자기 데이터를 직접 지울 수 있는 "내 정보 삭제" 버튼을 만들어줘. 누르면 한 번 더 확인하고, 지운 다음에는 되돌릴 수 없다고 알려주게.
```

5단계. 규칙 파일에 넣기

복사해서 쓰기

```
"비밀 키는 코드에 직접 쓰지 않고 채팅에도 출력하지 않는다. 필요하면 환경변수에서 읽는다"를 규칙 파일에 추가해줘.
```

사고가 났다면

키가 새어 나갔을 때는 코드에서 지우는 것만으로는 부족해요.

1. 키를 만든 곳(Claude 콘솔, Supabase, Resend)에서 그 키를 **삭제**해요.

2. 새 키를 만들어 비밀 설정에 넣고 "다시 올려줘"라고 보내요.

3. 콘솔에서 사용량을 확인해 이상하게 많이 쓰인 기록이 없는지 봐요.

4. GitHub에 올라간 적이 있다면 저장소를 공개로 바꾸지 말고, 키부터 바꿔요. 저장 기록에 남아 있을 수 있어요.

이 장의 문장

상황	붙여 넣을 문장
새어 있는 키 찾기	이 프로젝트에서 비밀 키(API 키, secret, token, password)가 코드나 설정 파일에 그대로 들어간 곳 찾아줘. 실제 값은 출력하지 말고 파일 이름이랑 줄 번호만 알려줘.
환경변수로 옮기기	찾은 비밀 키를 코드에서 빼고 환경변수에서 읽게 바꿔줘. 실제 값은 출력하지 마. 내가 Vercel에 넣어야 할 이름만 알려줘.
올린 앱 확인	Vercel에 올린 주소의 화면 쪽 코드에 비밀 키가 보이지 않는지 확인해줘. 실제 값은 출력하지 말고 있다/없다만 알려줘.
보안 설정 점검	Supabase의 모든 표와 파일 보관함에 보안 설정(RLS)이 켜져 있는지, 다른 사람 데이터를 읽거나 지울 수 있는 구멍이 없는지 점검해줘. 고치지는 말고 결과만 알려줘.
내 정보 삭제	사용자가 자기 데이터를 직접 지울 수 있는 "내 정보 삭제" 버튼을 만들어줘. 누르면 한 번 더 확인하고, 지운 다음에는 되돌릴 수 없다고 알려주게.

25장

예상치 못한 요금 폭탄 막기

이 장을 마치면 내 앱에 돈이 나가는 곳을 알고, 한도를 걸어서 갑자기 큰 요금이 나오지 않게 할 수 있어요.

돈이 나갈 수 있는 곳

서비스	무엇에 돈이 드나	이 책에서
Claude 앱 구독	앱을 만드는 동안의 구독료 (정해진 금액)	1장
Claude API	앱 안의 AI 기능을 쓸 때마다	20장
Vercel	무료 한도를 넘거나 돈을 받는 서비스로 쓸 때	5장
Supabase	무료 저장 공간이나 사용량을 넘을 때	18장
Resend	무료로 보낼 수 있는 메일 수를 넘을 때	21장

가장 위험한 건 API처럼 "쓴 만큼" 나가는 곳이에요. 링크를 아는 누군가가 반복해서 쓰면 요금이 내 카드로 나가요.

1단계. 사용 한도 걸기 (직접)

console.anthropic.com에서 한 달에 쓸 수 있는 최대 금액을 작게 정해요. 한도에 닿으면 AI 기능이 멈출 뿐 요금이 더 나가지 않아요.

2단계. 앱 안에서 횟수 제한

복사해서 쓰기

```
돈이 드는 기능(AI 호출, 메일 발송)이 있는지 찾아서 목록으로 보여줘. 각각 한 사람이 하루에 쓸 수 있는 횟수를 제한하는 계획부터 보여줘.
```

3단계. 줄일 수 있는 건 줄이기

복사해서 쓰기

```
AI 기능에서 같은 사진을 두 번 보내면 다시 부르지 말고 저번 결과를 쓰게 해줘. 사진은 보내기 전에 작게 줄여줘.
```

사진을 작게 줄이면 AI 요금과 저장 공간을 둘 다 아낄 수 있어요.

4단계. 한 달에 한 번 점검

어디서	무엇을
Claude 콘솔	이번 달 사용 금액, 한도 설정
Supabase 대시보드	저장 용량, 사용량, 프로젝트가 잠들지 않았는지
Vercel 대시보드	사용량 경고가 있는지

복사해서 쓰기

```
돈이 드는 서비스들의 이번 달 사용량을 확인하려면 어디를 봐야 하는지 서비스마다 정리해줘. 정리한 걸 운영점검.md에 저장해줘.
```

알아둘 것

Vercel 무료 요금제(Hobby)는 개인·비상업용이에요. 돈을 받고 파는 서비스로 바꾸면 유료 요금제가 필요할 수 있어요.

이 장의 문장

상황	붙여 넣을 문장
돈 드는 기능 찾기	돈이 드는 기능(AI 호출, 메일 발송)이 있는지 찾아서 목록으로 보여줘. 각각 한 사람이 하루에 쓸 수 있는 횟수를 제한하는 계획부터 보여줘.
중복 호출 줄이기	AI 기능에서 같은 입력을 두 번 보내면 다시 부르지 말고 저번 결과를 쓰게 해줘. 사진은 보내기 전에 작게 줄여줘.

상황	붙여 넣을 문장
점검 목록 만들기	돈이 드는 서비스들의 이번 달 사용량을 확인하려면 어디를 봐야 하는지 서비스마다 정리해줘. 정리한 걸 운영점검.md에 저장해줘.

26장

업데이트와 백업, 오류 복구하기

이 장을 마치면 사람들이 쓰는 중에도 안전하게 고치고, 문제가 생기면 몇 분 안에 잘 되던 버전으로 되돌릴 수 있어요.

1단계. GitHub에 백업하기

4장의 저장은 내 컴퓨터 안에만 있어요. 컴퓨터가 고장 나면 같이 사라져요. **GitHub**에 올려 두면 어느 컴퓨터에서든 이어서 할 수 있어요.

1. github.com에 직접 가입해요.
2. 대화창에 보내요.

```
복사해서 쓰기
```

```
이 프로젝트를 GitHub 비공개 저장소에 올려줘. 올리기 전에 비밀 키나 비밀번호가 든 파일은 올라가지 않게 막아줘. 내가 직접 해야 하는 단계가 있으면 하나씩 알려줘.
```

3. GitHub 로그인 승인 화면이 뜨면 내가 로그인하고 승인을 눌러요.
4. 내 저장소에 **Private** 표시가 있으면 끝이에요.

다른 컴퓨터에서 이어서 하려면 그 컴퓨터에서 이렇게 보내요.

```
복사해서 쓰기
```

```
GitHub의 OO 저장소를 이 컴퓨터로 받아줘
```

2단계. 안전하게 업데이트하는 순서

사람들이 쓰고 있는 앱은 이 순서로 고쳐요.

1. 저장하고 GitHub에 올리기 (지금 잘 되는 상태 보관)
2. 고치기 (9·11장 방식)
3. 확인목록으로 확인 (13장)
4. 내 휴대폰에서 한 번 써보기
5. 저장하고 Vercel에 올리기

복사해서 쓰기

```
여기까지 저장하고 GitHub랑 Vercel에도 올려줘
```

3단계. 올렸는데 문제가 생기면 (빠른 복구)

먼저 사람들이 쓰는 앱부터 되돌려요. 원인은 그다음에 찾아요.

방법 1. Vercel에서 되돌리기 (직접) vercel.com에서 프로젝트 → **Deployments**로 가면 지금까지 올린 버전이 쭉 보여요. 잘 되던 버전을 골라 되돌리기(Rollback/Promote)를 눌러요. 메뉴 이름은 바뀔 수 있으니 안 보이면 화면을 캡처해서 물어봐요.

방법 2. AI에게 되돌려 달라고 하기

복사해서 쓰기

```
방금 올린 버전에 문제가 있어. "첫 공개 버전"으로 저장한 때로 되돌려서 Vercel에 다시 올려줘. 저장 기록은 지우지 말고.
```

4단계. 원인 찾고 다시 고치기

앱이 안정되면 12장 방법으로 원인을 찾아요.

복사해서 쓰기

```
되돌리기 전 버전에서 생긴 문제를 문제정리.md에 정리해줘. 무엇을 바꿨고, 어떤 문제가 생겼는지.
```

5단계. 데이터 백업

Supabase에 쌓인 데이터도 가끔 받아 둬요.

복사해서 쓰기

```
Supabase에 있는 데이터를 파일로 내려받아 백업하는 방법을 알려줘. 내가 대시보드에서 직접 하는 방법으로.
```

이 장의 문장

상황	붙여 넣을 문장
GitHub 백업	이 프로젝트를 GitHub 비공개 저장소에 올려줘. 올리기 전에 비밀 키나 비밀번호가 든 파일은 올라가지 않게 막아줘. 내가 직접 해야 하는 단계가 있으면 하나씩 알려줘.
다른 컴퓨터로 받기	GitHub의 OO 저장소를 이 컴퓨터로 받아줘
모두 올리기	여기까지 저장하고 GitHub랑 Vercel에도 올려줘
빠른 복구	방금 올린 버전에 문제가 있어. "OO"로 저장한 때로 되돌려서 Vercel에 다시 올려줘. 저장 기록은 지우지 말고.
문제 기록	되돌리기 전 버전에서 생긴 문제를 문제정리.md에 정리해줘. 무엇을 바꿨고, 어떤 문제가 생겼는지.
데이터 백업	Supabase에 있는 데이터를 파일로 내려받아 백업하는 방법을 알려줘. 내가 대시보드에서 직접 하는 방법으로.

27장

사용자 피드백으로 서비스 개선하기

이 장을 마치면 몇 명이 어떻게 쓰는지 보고, 쓰는 사람의 의견을 받아서 다음에 고칠 것을 정할 수 있어요.

1단계. 몇 명이 쓰는지 보기

Vercel에는 방문자 수를 보는 기능이 있어요.

1. vercel.com에서 프로젝트 → **Analytics** 탭 → **Enable**을 눌러요. (직접)
2. 대화창에 보내요.

복사해서 쓰기

```
Vercel Web Analytics를 켰어. 앱에 연결하고 다시 올려줘.
```

하루쯤 지나면 방문자 수와 많이 본 화면을 볼 수 있어요.

2단계. 어디서 멈추는지 보기

방문자 수만으로는 부족해요. 사람들이 어디까지 쓰고 그만두는지 알고 싶다면 이렇게 해요.

복사해서 쓰기

```
사람들이 정산을 시작했다가 어디서 그만두는지 알고 싶어. "정산 시작", "사람 추가", "문구 복사" 버튼을 몇 번 눌렀는지 Vercel Analytics에 기록되게 해줘. 개인정보는 기록하지 말고.
```

3단계. 의견 받기

쓰는 사람이 "이거 안 돼요", "이런 거 있으면 좋겠어요"를 남길 곳을 만들어요. 18장의 Supabase를 써요.

복사해서 쓰기

```
화면 맨 아래에 "의견 보내기" 버튼을 넣어줘. 누르면 한 줄 의견을 적어서 Supabase에 저장되게. 다른 사람 의견은 못 보게 보안 설정 켜고, 표 만드는 문장은 내가 SQL Editor에 붙여 넣게 따로 줘.
```

들어온 의견은 Supabase 대시보드의 **Table Editor**에서 볼 수 있어요. 21장처럼 새 의견이 오면 내 메일로 알림이 오게 할 수도 있어요.

4단계. 다음에 고칠 것 정하기

의견과 숫자가 모이면 AI와 함께 정리해요.

복사해서 쓰기

```
아래는 이번 주에 받은 의견이야. 비슷한 것끼리 묶고, 많이 나온 순서로 정리해줘. 각각 고치는 데 드는 수고가 큰지 작은지도 같이.
(의견 붙여 넣기)
```

많이 나왔는데 고치기 쉬운 것부터 해요. 고칠 것은 나중에할것.md에 옮기고 9장 방식으로 하나씩 만들어요.

5단계. 고쳤다고 알리기

의견을 준 사람에게 고쳤다고 알려주면 다시 써줘요.

복사해서 쓰기

```
이번에 고친 것들을 단톡방에 올릴 짧은 안내 문구로 만들어줘. 세 줄 이내로.
```

이 장의 문장

상황	붙여 넣을 문장
방문자 보기	Vercel Web Analytics를 켰어. 앱에 연결하고 다시 올려줘.
버튼 기록	사람들이 어디서 그만두는지 알고 싶어. "OO", "OO" 버튼을 몇 번 눌렀는지 Vercel Analytics에 기록되게 해줘. 개인정보는 기록하지 말고.
의견 받기	화면 맨 아래에 "의견 보내기" 버튼을 넣어줘. 누르면 한 줄 의견을 적어서 Supabase에 저장되게. 다른 사람 의견은 못 보게 보안 설정 켜고, 표 만드는 문장은 내가 SQL Editor에 붙여 넣게 따로 줘.
의견 정리	아래는 이번 주에 받은 의견이야. 비슷한 것끼리 묶고, 많이 나온 순서로 정리해줘. 각각 고치는 데 드는 수고가 큰지 작은지도 같이.

상황	붙여 넣을 문장
고쳤다고 알리기	이번에 고친 것들을 단톡방에 올릴 짧은 안내 문구로 만들어줘. 세 줄 이내로.

특별 부록

복사해서 사용하는 실전 자료

A. 상황별 실전 프롬프트 모음

각 장에서 쓴 문장을 모았어요. `OO` 만 내 상황에 맞게 바꿔 보내요.

상황	붙여 넣을 문장	장
계산기 만들기	총 금액이랑 인원을 넣으면 1인당 얼마인지 알려주는 계산기 앱 만들어줘. 나 비개발자야. 브라우저에서 바로 열리는 파일 하나로 만들고, 다 만들면 화면에 띄워줘.	2
허용 창이 떴을 때	방금 허용해 달라고 한 거, 무슨 일 하는 건지 쉬운 말로 설명해줘.	2
화면이 안 보일 때	만든 거 어떻게 열어? 열어줘	2
결과가 이상할 때	OO을 넣었더니 OO이 나와. 원래는 OO이 나와야 해.	2
색 바꾸기	배경을 연한 민트색으로 바꿔줘	3
결과 크게	1인당 금액을 아주 크고 굵게 보여줘	3
휴대폰에 맞추기	휴대폰 화면에서도 잘 보이게 바꿔줘. 글자랑 버튼이 손가락으로 누르기 편하게.	3
차수 나누기	1차, 2차처럼 여러 번 정산할 수 있게 해줘. 차수마다 이름이랑 금액을 따로 넣고, 차수를 추가하는 버튼도 만들어줘.	3
사람마다 계산	같이 간 사람 이름을 넣고, 차수마다 누가 참석했는지 체크할 수 있게 해줘. 맨 아래에 사람마다 내야 할 금액을 보여줘.	3
문구 복사 버튼	정산 결과를 단톡방에 보낼 문구로 만들어서, 버튼을 누르면 복사되게 해줘.	3
방금 것 취소	방금 바꾼 거 취소하고 그 전으로 돌려줘	3
내 마음대로	OO을 OO처럼 바꿔줘	3

상황	붙여 넣을 문장	장
처음 저장	지금 상태를 저장해줘. 이름은 "OO"으로. 저장 기능이 아직 없으면 켜고 해줘.	4
이름 붙여 저장	여기까지 저장해줘. 이름은 "OO"로.	4
되돌리기	"OO"로 저장한 때로 되돌려줘. 저장 기록은 지우지 말고.	4
저장 목록	지금까지 저장한 거 목록 보여줘. 날짜랑 이름만 최신순으로.	4
처음 올리기	이 정산기를 인터넷 주소로 열 수 있게 Vercel에 올려줘. 나 비개발자야. 내가 직접 해야 하는 단계가 있으면 하나씩 알려줘.	5
주소 이름 바꾸기	주소를 OO.vercel.app처럼 알아보기 쉬운 이름으로 바꿔줘	5
미리보기 카드	카톡이나 문자로 링크를 보내면 미리보기 카드가 예쁘게 뜨게 해줘. 제목은 "OO", 설명은 "OO", 그림은 분위기에 맞게 만들어줘.	5
다시 올리기	고친 거 저장하고 Vercel에도 다시 올려줘	5
아이디어 다듬기	앱 아이디어가 있어. 아직 만들지 말고 같이 다듬어줘. 나 비개발자야. 불편한 점: OO. 쓰는 사람: OO. 나한테 질문 5개만 해서 이 앱이 꼭 해야 할 일을 정리해줘.	6
한 문단 정리	지금까지 이야기한 걸 앱 설명 한 문단으로 정리해줘. 누가, 언제, 무엇을 하려고 쓰는지가 들어가게.	6
비슷한 앱 찾기	이 아이디어랑 비슷한 앱이나 서비스가 이미 있는지 알려줘. 있다면 내 앱이 다르게 할 수 있는 점 2개만.	6
파일로 남기기	지금까지 정리한 앱 설명을 기획.md 파일로 저장해줘.	6
아이디어 추천	나는 OO 일을 하고, 평소에 OO가 불편해. 화면 하나로 만들 수 있는 앱 아이디어 5개 추천해줘	6
기능 목록	기획.md를 읽고, 이 앱에 들어갈 기능을 목록으로 뽑아줘. 꼭 필요한 것과 있으면 좋은 것으로 나눠서.	7
화면 정하기	꼭 필요한 기능만으로 화면을 몇 개 만들면 될지 정해줘. 화면마다 무엇이 보이는지 그림 그리듯 글로 설명해줘.	7
사용 흐름	OO가 OO하는 순서를 한 단계씩 적어줘. 누르는 버튼까지.	7

상황	붙여 넣을 문장	장
화면 그림 먼저	이 화면을 아주 간단한 그림으로 먼저 만들어서 보여줘. 기능은 안 돼도 괜찮아	7
기획 정리	지금까지 정한 기능 목록, 화면 설명, 사용 흐름을 기획.md에 이어서 정리해줘.	7
기능 하나 고르기	기획.md를 읽고, 이 앱이 쓸모 있으려면 꼭 있어야 하는 기능 딱 하나만 골라줘. 나머지는 "나중에"로 옮겨줘.	8
나중 목록 만들기	"나중에" 기능들을 나중에할것.md 파일로 저장해줘. 첫 버전을 다 만들기 전에는 손대지 않을 거야.	8
나중 목록에 추가	"OO"를 나중에할것.md에 추가해줘. 지금은 만들지 마.	8
첫 버전 만들기	기획.md의 첫 버전 기능만으로 앱을 만들어줘. 나 비개발자야. 브라우저에서 바로 열리는 파일 하나로 만들고, 다 만들면 화면에 띄워줘.	8
다음 기능 고르기	며칠 써보니까 이런 점이 불편했어: OO. 나중에할것.md를 보고 다음에 만들 기능 하나를 추천해줘.	8
계획부터 받기	OO 기능을 넣고 싶어. 아직 만들지 말고, 화면이 어떻게 바뀌는지랑 해야 할 일을 단계별로 정리해줘. 내가 OK하면 그때 만들어.	9
계획 고치기	OO는 빼고, OO는 OO했으면 좋겠어. 이렇게 고쳐서 계획 다시 보여줘.	9
계획 파일	지금 정한 계획을 계획.md 파일로 저장해줘. 끝난 단계는 체크 표시해줘.	9
한 단계씩	계획.md의 1단계만 먼저 만들어줘. 다 되면 멈추고 나한테 확인받아.	9
다음 단계	잘 돼. 여기까지 저장하고 계획.md에 체크한 다음, 다음 단계 만들어줘. 다 되면 멈추고 확인받아.	9
마무리 정리	계획.md가 다 끝났어. 무엇을 만들었는지 세 줄로 요약하고, 나중에할것.md에서 끝난 건 지워줘.	9
문제 설명 틀	OO를 했더니 OO가 보여. 원래는 OO가 돼야 해.	10
새 기능 틀	[누가] 쓰는 [무엇을] 하는 기능 만들어줘. 화면은 [어떤 느낌]이었으면 좋겠어.	10

상황	붙여 넣을 문장	장
참고 화면	(참고 화면 붙여 넣기) 이런 느낌으로 바꿔줘	10
방법 묻기	이걸 하려면 어떤 방법들이 있어? 쉬운 말로, 추천하는 것 하나를 먼저 이유와 함께 말해줘.	10
고칠 곳만	OO만 고쳐줘. 다른 기능은 건드리지 말고, 바꾸기 전에 어디를 바꿀지 먼저 말해줘. 내가 OK하면 그때 바꿔.	11
되묻기	왜 그것까지 바꿔야 해? 꼭 필요한 것만 남기고 다시 말해줘.	11
규칙 파일 만들기	아래 규칙을 CLAUDE.md 파일에 저장해줘. (규칙 붙여 넣기)	11
규칙 확인	이 폴더의 규칙 파일에 뭐라고 적혀 있어? 한 줄씩 요약해줘.	11
규칙 더하기	방금 내가 한 말을 규칙 파일에 한 줄로 추가해줘.	11
에러 보여주기	OO를 했더니 아래 에러가 떠. (에러 메시지 통째로 붙여 넣기)	12
숨은 에러 찾기	화면에는 에러가 안 보이는데 OO가 안 돼. 숨어 있는 에러가 있는지 직접 확인해줘.	12
원인부터	아직 고치지 말고, 왜 이런 일이 생기는지 가능한 원인을 쉬운 말로 3가지 설명해줘. 가장 가능성 높은 것부터.	12
하나씩 확인	1번 원인이 맞는지 먼저 확인해줘. 맞으면 그것만 고치고, 아니면 2번으로 넘어가.	12
다른 방법	지금 방식은 그만하고, 그 기능만 아예 다른 방법으로 다시 만들어줘. 다른 기능은 건드리지 말고.	12
문제 정리	지금 문제와 지금까지 시도한 방법을 문제정리.md에 정리해줘.	12
새 대화에서	문제정리.md를 읽고, 이미 시도한 방법 말고 다른 방법으로 해결해줘.	12

상황	붙여 넣을 문장	장
확인 목록 만들기	이 앱에서 항상 맞아야 하는 것들을 확인 목록으로 만들어서 확인목록.md에 저장해줘.	13
직접 확인시키기	확인목록.md에 있는 걸 하나씩 직접 앱을 열어서 넣어보고 결과가 맞는지 확인해줘. 확인하지 못한 건 확인 못 했다고 말해줘.	13
틀린 것만 고치기	틀린 것만 고쳐줘. 다른 건 건드리지 말고. 고친 다음 확인목록을 처음부터 다시 확인해줘.	13
증거 받기	어떤 값을 넣었고 화면에 뭐가 나왔는지 항목마다 그대로 보여줘.	13
직접 확인 표시	확인목록.md 중에서 네가 직접 확인할 수 없는 건 "직접 확인"이라고 따로 표시해줘.	13
목록에 추가	방금 만든 기능도 확인목록.md에 추가해줘.	13
하루 정리	오늘 한 일, 다음에 할 일, 아직 해결 못 한 문제를 진행상황.md에 정리해줘. 원래 내용이 있으면 맨 위에 오늘 날짜로 추가해.	14
이어서 시작	CLAUDE.md, 기획.md, 진행상황.md를 읽고, 다음에 할 일부터 이어서 하자. 시작하기 전에 지금 상황을 세 줄로 요약해줘.	14
대화 갈아타기	지금까지 이 대화에서 한 일을 진행상황.md에 정리해줘. 새 대화로 옮길 거야.	14
결정 남기기	방금 정한 것(OO)을 이유와 함께 기획.md의 "결정한 것" 아래에 적어줘.	14
작업 순서 규칙	"새 기능을 만들 때는 저장 → 계획 → 한 단계씩 → 확인목록 확인 → 저장 순서로 한다"를 규칙 파일에 추가해줘.	15
파일 정리	지금 파일 구조를 쉬운 말로 설명해줘. 파일이 너무 길거나 섞여 있으면, 기능은 그대로 두고 정리하는 계획부터 보여줘.	15
지도 만들기	이 앱의 화면과 기능이 어느 파일에 있는지 지도.md에 표로 정리해줘. 앞으로 파일이 바뀌면 같이 고쳐줘.	15
실험 갈래	지금 상태는 그대로 두고, 실험용 갈래를 따로 만들어서 거기서 OO를 해봐줘. 마음에 들면 합치고, 아니면 버릴 거야.	15

상황	붙여 넣을 문장	장
실험 합치기	실험한 거 마음에 들어. 원래 쪽에 합쳐줘.	15
실험 버리기	실험한 거 별로야. 실험 갈래는 버리고 원래 상태로 돌아가줘.	15
요청 모음 만들기	내가 자주 하는 부탁을 요청모음.md에 번호를 붙여서 저장해줘. (부탁 목록 붙여 넣기)	16
모음에서 시키기	요청모음.md의 OO번 해줘	16
마무리 한 번에	요청모음.md의 2번, 3번, 1번을 순서대로 해줘. 중간에 틀린 게 나오면 멈추고 알려줘.	16
여러 개 한 번에 넣기	아래 목록을 앱에 한 번에 넣어줘. (목록 붙여 넣기)	16
필요한 파일만	지도.md를 보고, 이 문제와 관련된 파일만 읽어서 고쳐줘.	16
새로고침 해도 남게	새로고침하거나 창을 닫았다 열어도 입력한 내용이 그대로 남게 해줘. 이 브라우저에만 저장하면 돼.	17
기록 목록	정산이 끝나면 "기록에 저장" 버튼으로 이름이랑 날짜를 붙여서 남기고, 지난 목록을 최신순으로 보여줘. 목록을 누르면 다시 볼 수 있게.	17
지우기 버튼	기록마다 삭제 버튼을 넣어줘. 누르면 정말 지울지 한 번 물어보게.	17
내보내기·불러오기	기록 전체를 파일 하나로 내려받는 "내보내기" 버튼이랑, 그 파일을 다시 넣으면 기록이 돌아오는 "불러오기" 버튼을 만들어줘.	17
같이 보기 계획	OO를 Supabase에 저장해서, 링크를 가진 사람 모두가 같은 내용을 보게 하고 싶어. Project URL은 OO, publishable 키는 OO야. 아직 만들지 말고 계획부터 보여줘. 나 비개발자야.	18
계획에 꼭 넣기	계획에 이 세 가지를 넣어줘. 추측할 수 없는 링크, 다른 데이터는 못 보게 보안 설정 켜기, 표 만드는 문장은 내가 Supabase SQL Editor에 붙여 넣게 따로 주기.	18
만들고 올리기	표 만들었어. 계획대로 앱을 만들고 Vercel에 다시 올려줘.	18

상황	붙여 넣을 문장	장
바로 보이게	다른 사람이 바꾸면 내 화면에 새로고침 없이 바로 보이게 해줘	18
로그인 계획	Supabase로 이메일 로그인을 붙이고 싶어. 이메일을 넣으면 로그인 링크가 메일로 오는 방식으로. 로그인하면 내 것만 보이고, 만든 사람만 고칠 수 있게. 아직 만들지 말고 계획부터 보여줘. 대시보드에서 내가 직접 해야 하는 설정도 따로 알려줘.	19
만들고 올리기	설정 끝났어. 계획대로 만들고 Vercel에 다시 올려줘. 다 되면 확인목록.md에 "다른 계정으로 로그인하면 내 것이 안 보인다"를 추가하고 확인해줘.	19
이상한 주소로 갈 때	로그인 링크를 누르면 OO로 가. 내 앱 주소로 돌아오게 하려면 대시보드에서 뭘 바꿔야 해?	19
외부 정보 가져오기	OO 정보를 가져와서 OO하는 기능을 넣고 싶어. 무료로 쓸 수 있고 키가 필요 없는 걸 골라서 써줘. 아직 만들지 말고 어떤 걸 쓸지랑 계획부터 보여줘.	20
앱 안 AI 계획	OO하면 Claude API로 OO하는 기능을 넣고 싶어. API 키는 Vercel 환경변수 ANTHROPIC_API_KEY에 넣어 뒀어. 키가 화면 쪽 코드에 절대 들어가지 않게 서버 쪽에서 처리해줘. 아직 만들지 말고 계획부터 보여줘.	20
실패 처리	사진에서 금액을 못 읽으면 "직접 입력해 주세요"라고 보여주게	20
횟수 제한	이 기능은 한 사람이 하루 10번까지만 쓸 수 있게 제한해줘. 넘으면 안내 문구를 보여주게.	20
송금 버튼	정산 결과에 사람마다 "토스로 보내기" 버튼을 넣고 싶어. 받는 사람은 나(총무)고, 내 송금 링크는 toss.me/내아이디야. 아직 만들지 말고 계획부터 보여줘.	21
메일 알림	OO하면 내 이메일로 알림이 오게 하고 싶어. 메일은 Resend로 보내고, 키는 코드에 넣지 말고 비밀 설정에 넣을 거야. 내가 직접 넣어야 할 이름이랑 위치를 알려주고, 계획부터 보여줘.	21
사진 보관	OO마다 사진을 올려서 보관하고 다시 볼 수 있게 하고 싶어. Supabase Storage를 쓰고, 볼 수 있는 사람만 사진도 보게 보안 설정을 해줘. 사진은 올리기 전에 작게 줄여줘. 아직 만들지 말고 계획부터 보여줘.	21
휴대폰 점검	휴대폰 화면에서 잘리거나 너무 작은 곳이 있는지 직접 확인하고 고쳐줘. 버튼은 손가락으로 누르기 편한 크기로.	22
앱처럼 설치	휴대폰 홈 화면에 추가하면 진짜 앱처럼 보이게 해줘. 앱 이름은 "OO", 아이콘은 OO 모양으로 만들어줘. 열면 주소창 없이 꽉 찬 화면으로 보이게.	22

상황	붙여 넣을 문장	장
인터넷 약할 때	인터넷이 약한 곳에서도 마지막으로 열었던 화면은 보이게 해줘.	22
바로 반영	고친 내용이 휴대폰에 설치한 앱에도 바로 반영되게 해줘	22
공개 전 점검	이 앱을 많은 사람에게 공개하려고 해. 아래 항목을 하나씩 직접 확인하고, 문제가 있는 것만 알려줘. 고치지는 말고. (항목 붙여 넣기)	23
개인정보 안내	이 앱에서 모으는 개인정보를 정리하고, 화면 맨 아래에 "무엇을, 왜, 언제까지 보관하는지" 짧은 안내를 넣어줘. 필요 없는 정보는 모으지 않게 바꿔줘.	23
공개 직전 저장	공개 직전 상태로 저장해줘. 이름은 "첫 공개 버전"으로. 그리고 GitHub랑 Vercel에도 올려줘.	23
새어 있는 키 찾기	이 프로젝트에서 비밀 키(API 키, secret, token, password)가 코드나 설정 파일에 그대로 들어간 곳 찾아줘. 실제 값은 출력하지 말고 파일 이름이랑 줄 번호만 알려줘.	24
환경변수로 옮기기	찾은 비밀 키를 코드에서 빼고 환경변수에서 읽게 바꿔줘. 실제 값은 출력하지 마. 내가 Vercel에 넣어야 할 이름만 알려줘.	24
올린 앱 확인	Vercel에 올린 주소의 화면 쪽 코드에 비밀 키가 보이지 않는지 확인해줘. 실제 값은 출력하지 말고 있다/없다만 알려줘.	24
보안 설정 점검	Supabase의 모든 표와 파일 보관함에 보안 설정(RLS)이 켜져 있는지, 다른 사람 데이터를 읽거나 지울 수 있는 구멍이 없는지 점검해줘. 고치지는 말고 결과만 알려줘.	24
내 정보 삭제	사용자가 자기 데이터를 직접 지울 수 있는 "내 정보 삭제" 버튼을 만들어줘. 누르면 한 번 더 확인하고, 지운 다음에는 되돌릴 수 없다고 알려주게.	24
돈 드는 기능 찾기	돈이 드는 기능(AI 호출, 메일 발송)이 있는지 찾아서 목록으로 보여줘. 각각 한 사람이 하루에 쓸 수 있는 횟수를 제한하는 계획부터 보여줘.	25
중복 호출 줄이기	AI 기능에서 같은 입력을 두 번 보내면 다시 부르지 말고 저번 결과를 쓰게 해줘. 사진은 보내기 전에 작게 줄여줘.	25
점검 목록 만들기	돈이 드는 서비스들의 이번 달 사용량을 확인하려면 어디를 봐야 하는지 서비스마다 정리해줘. 정리한 걸 운영점검.md에 저장해줘.	25
GitHub 백업	이 프로젝트를 GitHub 비공개 저장소에 올려줘. 올리기 전에 비밀 키나 비밀번호가 든 파일은 올라가지 않게 막아줘. 내가 직접 해야 하는 단계가 있으면 하나씩 알려줘.	26

상황	붙여 넣을 문장	장
다른 컴퓨터로 받기	GitHub의 OO 저장소를 이 컴퓨터로 받아줘	26
모두 올리기	여기까지 저장하고 GitHub랑 Vercel에도 올려줘	26
빠른 복구	방금 올린 버전에 문제가 있어. "OO"로 저장한 때로 되돌려서 Vercel에 다시 올려줘. 저장 기록은 지우지 말고.	26
문제 기록	되돌리기 전 버전에서 생긴 문제를 문제정리.md에 정리해줘. 무엇을 바꿨고, 어떤 문제가 생겼는지.	26
데이터 백업	Supabase에 있는 데이터를 파일로 내려받아 백업하는 방법을 알려줘. 내가 대시보드에서 직접 하는 방법으로.	26
방문자 보기	Vercel Web Analytics를 켰어. 앱에 연결하고 다시 올려줘.	27
버튼 기록	사람들이 어디서 그만두는지 알고 싶어. "OO", "OO" 버튼을 몇 번 눌렀는지 Vercel Analytics에 기록되게 해줘. 개인정보는 기록하지 말고.	27
의견 받기	화면 맨 아래에 "의견 보내기" 버튼을 넣어줘. 누르면 한 줄 의견을 적어서 Supabase에 저장되게. 다른 사람 의견은 못 보게 보안 설정 켜고, 표 만드는 문장은 내가 SQL Editor에 붙여 넣게 따로 줘.	27
의견 정리	아래는 이번 주에 받은 의견이야. 비슷한 것끼리 묶고, 많이 나온 순서로 정리해줘. 각각 고치는 데 드는 수고가 큰지 작은지도 같이.	27
고쳤다고 알리기	이번에 고친 것들을 단톡방에 올릴 짧은 안내 문구로 만들어줘. 세 줄 이내로.	27

B. AI가 말을 안 들을 때 사용하는 해결 문장

이럴 때	붙여 넣을 문장
시키지 않은 것까지 바꿈	내가 말한 것만 해줘. 방금 바꾼 것 중에 내가 말하지 않은 건 전부 되돌려줘.
계획 없이 바로 만들어 버림	멈춰. 방금 만든 거 취소하고, 계획부터 보여줘.
"고쳤어요"라는데 그대로	어떤 값을 넣었고 화면에 뭐가 나왔는지 그대로 보여줘. 직접 확인 못 했으면 못 했다고 말해줘.

이럴 때	붙여 넣을 문장
같은 실수를 반복	아직 고치지 말고, 왜 이런 일이 생기는지 가능한 원인을 쉬운 말로 3가지 설명해줘.
어려운 말만 함	나 비개발자야. 중학생도 알아듣게 다시 설명해줘.
선택지만 잔뜩 늘어놓음	하나만 골라줘. 추천하는 것 하나를 이유와 함께.
규칙 파일을 안 지킴	규칙 파일 다시 읽고, 방금 한 작업이 규칙을 지켰는지 확인해줘.
앞에서 정한 걸 잊음	진행상황.md랑 기획.md 다시 읽고 이어서 해줘.
너무 길게 대답함	세 줄로 줄여서 다시 말해줘.
비밀 키를 달라고 함	비밀 키는 채팅에 안 줄 거야. 내가 직접 넣어야 할 곳과 이름만 알려줘.

C. 앱 기획부터 출시까지 체크리스트

기획 (6~8장)

- [ ] 불편한 순간을 세 줄로 적었다
- [ ] 앱 설명 한 문단을 기획.md에 남겼다
- [ ] 기능을 "꼭 필요한 것 / 있으면 좋은 것"으로 나눴다
- [ ] 첫 버전 기능을 하나만 골랐고, 나머지는 나중에할것.md에 옮겼다

만들기 (9~16장)

- [ ] 큰 기능은 계획부터 받고 한 단계씩 만들었다
- [ ] 규칙 파일(CLAUDE.md)을 만들었다
- [ ] 확인목록.md를 만들고 고칠 때마다 확인했다
- [ ] 하루 끝에 진행상황.md를 정리했다

출시 전 (23~25장)

- [ ] 확인목록을 전부 통과했다
- [ ] 다른 사람 휴대폰에서 처음부터 끝까지 써봤다
- [ ] 코드에 비밀 키가 없다
- [ ] 데이터 보안 설정을 점검했다
- [ ] 개인정보 안내를 넣었다 (모으는 경우)
- [ ] API 사용 한도와 횟수 제한을 걸었다

- [ ] "첫 공개 버전"으로 저장하고 GitHub에 백업했다

출시 후 (26~27장)

- [ ] 문제가 생기면 되돌리는 방법을 알고 있다
- [ ] 방문자 수와 의견을 받을 곳이 있다
- [ ] 한 달에 한 번 사용량을 점검한다

D. 바이브코딩에서 자주 하는 실수와 해결법

실수	왜 문제인가	이렇게 해요
한 번에 여러 개를 시킴	뭐가 잘못됐는지 알 수 없어요	하나씩 시키고 확인해요 (10장)
저장 없이 크게 고침	망가지면 돌아갈 곳이 없어요	고치기 전에 저장해요 (4장)
"안 돼요"라고만 말함	AI가 원인을 추측해야 해요	"했더니 / 보여 / 원래는" 틀로 말해요 (10장)
처음부터 다 만들려고 함	반쯤 되다 꼬여요	기능 하나로 첫 버전을 만들어요 (8장)
같은 대화를 며칠씩 씀	AI가 앞 내용을 헷갈려요	진행상황.md에 정리하고 새 대화로 (14장)
AI의 "다 됐어요"를 그대로 믿음	실제로는 안 될 수 있어요	확인목록으로 증거를 받아요 (13장)
비밀 키를 채팅에 붙여 넣음	키가 새어 나갈 수 있어요	비밀 설정에만 직접 넣어요 (24장)
사용 한도 없이 AI 기능 공개	요금 폭탄이 올 수 있어요	한도와 횟수 제한부터 (25장)
같은 오류에 같은 말을 반복	점점 더 꼬여요	원인부터 묻고, 안 되면 다른 방법으로 (12장)
에러 메시지를 요약해서 보냄	중요한 단서가 빠져요	통째로 복사해서 보내요 (12장)

E. 목적별 추천 AI 도구와 활용법

하고 싶은 것	도구	이 책에서
앱 만들기·고치기	Claude 데스크톱 앱의 Code 모드 (또는 ChatGPT의 Codex)	1장~
아이디어 다듬기·기획	같은 앱에서 "아직 만들지 말고"로 대화만	6~7장
저장·되돌리기	Git (AI가 대신 써줘요)	4장

하고 싶은 것	도구	이 책에서
인터넷에 공개	Vercel	5장
백업	GitHub	26장
여럿이 같은 데이터 보기·로그인·파일 보관	Supabase	18·19·21장
앱 안에 AI 기능	Claude API	20장
메일 알림	Resend	21장
방문자 보기	Vercel Web Analytics	27장

Claude 요금 (2026년 10월 24일 공식 요금 페이지 기준, 달러 기준이고 세금이 따로 붙어요)

플랜	월 요금	코딩 기능
Free	$0	Claude Code 없음
Pro	$20 (연간 결제 시 월 $17)	Claude Code 포함
Max	$100부터	Claude Code 포함, 사용량 5배·20배

Vercel, GitHub, Supabase, Resend는 이 책을 따라 하는 범위에서는 무료 요금제로 충분해요. 요금은 자주 바뀌니 결제 전에 공식 페이지를 한 번 더 확인하세요.

F. 실제 앱 프로젝트 예제와 응용 방법

이 책에서 만든 두 가지

예제	어디서	배운 것
모임 정산기	PART 01, PART 04	첫 앱, 꾸미기, 저장, 공개, 데이터·공유·로그인·AI·송금
가족 냉장고 유통기한 앱	PART 02	아이디어 구체화, 기획, 작은 첫 버전, 단계별 계획

정산기 방식으로 만들 수 있는 앱

앱	첫 문장	붙이면 좋은 레시피
모임 날짜 투표	날짜 후보를 넣고 사람마다 되는 날을 체크하는 앱 만들어줘	18장 (같이 보기)
운동 기록	오늘 한 운동과 횟수를 넣으면 이번 주 합계를 보여주는 앱 만들어줘	17장 (저장), 22장 (앱처럼)
우리 집 할 일 당번	할 일과 가족 이름을 넣으면 이번 주 당번을 정해주는 앱 만들어줘	18장, 21장 (알림)
명함 대신 내 소개 페이지	내 소개, 연락처, 링크를 보여주는 한 장짜리 페이지 만들어줘	5장 (공개), 22장
독서 기록	책 제목과 한 줄 감상을 남기고 월별로 모아 보는 앱 만들어줘	17장, 20장 (AI 요약)

응용하는 법

1. 첫 문장은 10장의 틀로 써요: "[누가] 쓰는 [무엇을] 하는 앱 만들어줘."
2. 첫 버전은 기능 하나로 만들어요. (8장)
3. 필요한 레시피만 PART 04에서 골라 붙여요.
4. 공개 전에는 부록 C 체크리스트를 돌려요.