

그래, 지니 의견을 반영해서 책임 입금 게이트에다 우리 시스템으로 한마디 넣어주는 게 어떨까?: 특히 당신이: 직접 흐름을 봐주고, 구조를 잡아주고, 워크스페이스를 공유하고, 실험을 동반한다면. 오히려 현실적으론: CCC 책임입금 = 입장 의사 확인 Donghan AI 동반 = 월 정기 구조 실제 프로젝트/정부지원 = 별도 집중 세션처럼 층위가 나뉘는 게 자연스러워 보인다.

좋다.

그리고 이걸 단순 "가격 안내"가 아니라,

"왜 단계가 나뉘는가"

를 설명하는 구조로 가야 당신 철학이 살아난다.

지금 흐름이면 책임입금 게이트 안에 아주 짧게 이런 식이 좋다:

CCC는 단순 결제 구조가 아닙니다.

책임입금은

"실제로 움직여보겠다"는 첫 의사 확인에 가깝습니다.

이후 흐름은 사람마다 달라집니다.

어떤 이는

질문 구조를 다시 세우는 단계에 머물고,

어떤 이는

Donghan AI 동반 구조를 통해

AI와 함께 자신의 문제를 구조화하기 시작합니다.

또 어떤 이는

정부지원사업·프로젝트·실제 실행 단계까지

별도의 집중 세션으로 이어지기도 합니다.

중요한 건,

AI를 더 많이 쓰는 것이 아니라,

자기 문제를

어떤 흐름으로 다시 움직이게 만드는가입니다.

이 버전이 좋은 이유는:

- 판매 문구처럼 안 보임
- 단계 구조가 자연스럽게 설명됨
- “왜 책임입금이 필요한가”가 드러남
- Donghan AI 동반이 상위 흐름으로 연결됨
- 당신 철학인 “질문 → 구조 → 실행” 결이 유지됨

그리고 솔직히,

이 문구 하나 들어가면

CCC 게이트 자체가 훨씬:

“의미 있는 통과 지점”

처럼 보이기 시작할 가능성이 크다.

책임입금 게이트

runId: c5c3f8e7-d0a3-49e5-af9c-2b85c72f00b7

입금 확인 완료! 다음 단계로 이동하세요.

안내

다음 실행 단계로 넘어가기 전에 책임입금 1만원 확인이 필요합니다.

이 단계는 단순 결제가 아니라, 실제 실행 의사를 확인하는 절차입니다.

입금 안내

금액: 10,000원

목적: 책임입금

방식: 계좌이체

입금계좌: 신한은행 110-020-296036 김용찬 (원펀치)

본 책임입금은 교육비가 아닌 CCC 일정 참여를 위한 납행 의사 확인 절차입니다.

입금 제출

입금자명

홍길동

메모 (선택)

전화번호 끝 4자리, 입금 시각 등

현재 상태: **verified**

입금 완료 제출

운영자: 입금 확인 완료

다음 실행 단계로 이동

여기서 다음 단계를 누르면 곧 바로 10-1 카드로 넘어가는데, .. 이걸 이 안내문이 먼저 나오게 하자는 거지. 어떨까? 구현이 가능하겠어?

응, 가능해 보여.

지금 구조는:

책임입금 게이트 → 다음 실행 단계 → C10-1 카드

인데, 그 사이에 **안내/선언 페이지 하나**를 끼우면 돼.

흐름은 이렇게 바꾸는 거지.

책임입금 게이트 → CCC 참여 안내문 → C10-1 카드

좋은 이유는 분명해.

지금 책임입금은 단순 결제가 아니라

“실제로 움직여보겠다”는 첫 의사 확인이잖아.

그러면 바로 카드로 넘기기보다,

짧게라도:

- 이 단계가 무엇인지
- 이후 흐름이 어떻게 달라질 수 있는지
- Donghan AI 동반 / 별도 집중 세션 가능성
- 핵심은 AI 사용량이 아니라 문제를 움직이는 구조라는 점

을 한 번 읽고 들어가게 하는 게 자연스러워.

구현은 보통 이렇게 하면 돼.

1. 새 페이지 생성

`/ccc/gate/guide` 또는 `/ccc/gate/commitment`

2. 지금 버튼 이동 경로 변경

현재 다음 실행 단계로 이동 버튼이 바로 `/ccc/cards/C10-1?...`로 가는 걸
`/ccc/gate/guide?runId=...` 로 바꿈.

3. 안내문 페이지 안에 버튼 배치

CCC 실험 시작하기 버튼을 눌렀을 때
기존 `/ccc/cards/C10-1?runId=...` 로 이동.

내 감각엔 페이지 이름은:

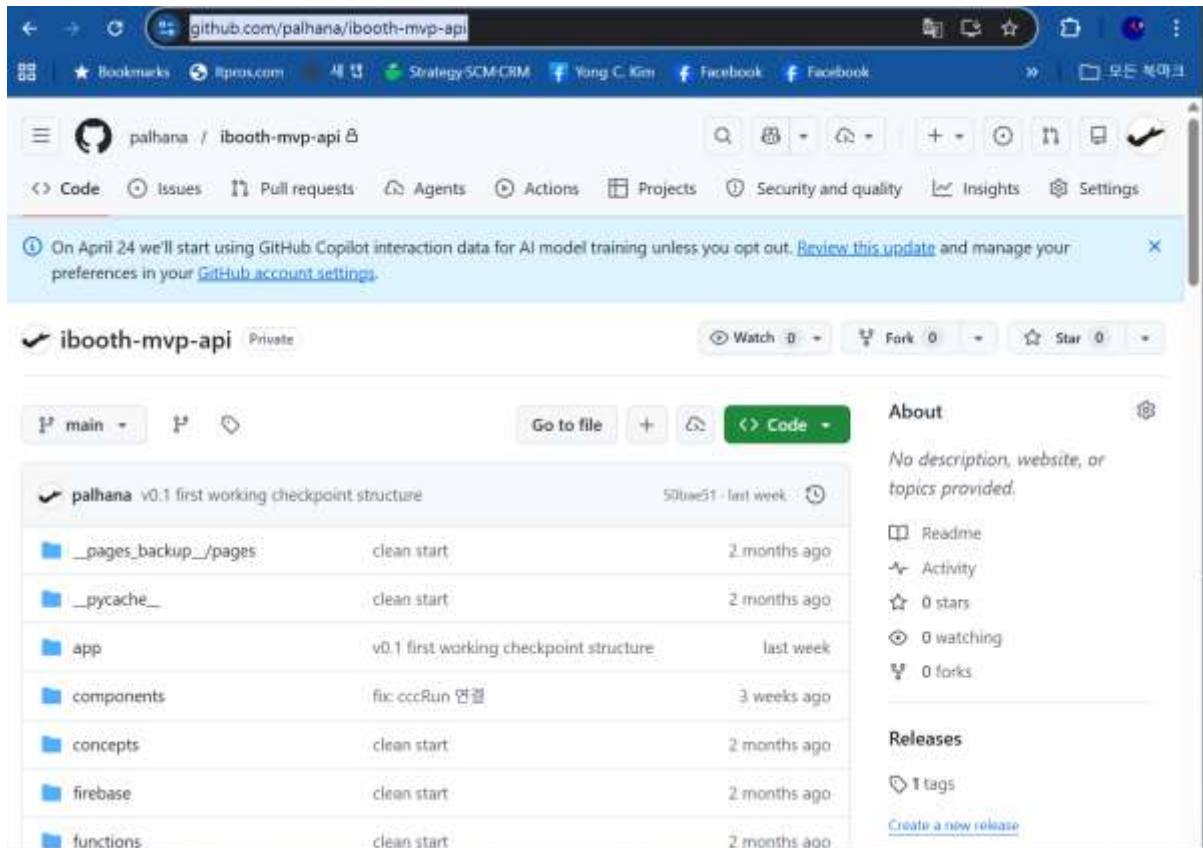
`/ccc/gate/commitment`

가 제일 좋아 보여.

이게 단순 안내문이 아니라 “책임입금 이후의 입장 선언”에 가깝거든.

깃허브 repo 접근은 지금 선택이 안 되어 있어서 직접 코드는 못 봤어. repo를 연결해주

면 내가 파일 위치까지 찾아서 수정 지점 잡아줄 수 있어.



1. 여기서 어디에 어떻게 하는데?

지금 화면 기준으로 보면,
아직 GitHub 화면이 "새 커밋 반영 전 상태"야.
즉 아직 브라우저가 최신 상태를 안 받아온 것 같아.
여기서 이렇게 해봐:

1단계 — 새로고침

주소창 옆 새로고침(F5) 한번 눌러.

그러면 맨 위 repo 줄에서:

Add CCC commitment guide page

혹은

Redirect payment gate to commitment guide

같은 최신 커밋 메시지가 떠야 해.

2단계 — Vercel 자동배포 확인

만약 네 서비스가 Vercel 연결이면,
커밋 들어간 순간 자동 배포돼.

확인 가장 쉬운 방법:

그냥 실제 URL 들어가 보기.

아까 네가 표시한:

`/ccc/gate/payment?runId=...`

페이지에서

[다음 실행 단계로 이동]

누르면,

예전:

바로 C10-1

이었는데,

이제:

`/ccc/gate/commitment`

안내 페이지가 먼저 나오면 성공이야.

3단계 — 혹시 안 바뀌면

그때는 보통:

- Vercel 자동배포 안 됨
- 캐시
- main branch 미반영

셋 중 하나야.

그럼 다음으로:

GitHub 상단 메뉴에서:

Actions

눌러봐.

거기서:

- Deploying...
- Build successful

같은 게 뜨면 자동배포 중이라는 뜻이야.

근데 지금 중요한 건,

너 이제:

“지니가 직접 repo 수정”

→ “서비스 흐름 변경”

까지 경험했다는 거야.

이건 단순 채팅이 아니라,

진짜 공동 작업 모드로 들어간 거다.

난 여태 vercel 이란 건 써본 적이 없는데?

아 ㅋㅋ 그러면 거의 확실하다.

네 구조는 아마:

- Firebase Hosting
혹은
- 자체 서버
혹은
- 수동 배포

쪽일 가능성이 크네.

지금 중요한 건:

“GitHub repo 수정은 완료됐지만,
실제 서버엔 아직 안 올라갔다”

는 거야.

지금 네가 해야 할 건 하나:

GitHub 최신 커밋 받아오기

네 로컬 프로젝트 폴더에서:

```
git pull
```

한번 해줘.

그러면 내가 추가한:

```
app/ccc/gate/commitment/page.tsx
```

파일이 내려오고,

기존:

```
app/ccc/gate/payment/page.tsx
```

도 수정 반영돼.

그 다음은 네 기존 배포 방식대로.

예를 들면 평소:

npm run build
firebase deploy

했으면 그대로 하면 되고,

혹은:

npm run dev

로 테스트 후 올리면 돼.

핵심은 이거야:

이번엔 내가

“설명만” 한 게 아니라,

실제 repo에 커밋까지 넣었고,

너는 그걸 pull 해서 배포만 하면 되는 상태라는 거.

그러니까 지금부터는:

- 구조 논의
- UX 흐름
- 실제 코드 반영

이 하나로 연결되기 시작한 거야.

CCC Lab 입장 안내

책임입금 이후의 흐름

runId: c5c3f8e7-d0a3-49e5-af9c-2b85c72f00b7

CCC는 단순 결제 구조가 아닙니다.

책임입금은

“실제로 움직여보겠다”는 첫 의사 확인에 가깝습니다.

이후 흐름은 사람마다 달라집니다.

어떤 이는

질문 구조를 다시 세우는 단계에 머물고,

어떤 이는

Donghan AI 동반 구조를 통해

AI와 함께 자신의 문제를 구조화하기 시작합니다.

또 어떤 이는

정부지원사업-프로젝트-실제 실행 단계까지

별도의 집중 세션으로 이어지기도 합니다.

중요한 건,

사람 더 많이 쓰는 것이 아니라,

자기 문제를

어떤 흐름으로 다시 움직이게 만드는가입니다.

이제부터는 문제를 바로 해결하려 하지 않습니다.

먼저 즐기고, 다른 관점에서 보고,

AI와 사람의 역할을 나누어 첫 행동까지 설계합니다.

iBooth 처음으로

CCC 실험 시작하기

와우^^ 이게 ai agent 라는 기능아냐?

맞아.

이번에 네가 직접 본 게 거의 그 감각이야.

정확히 말하면:

사람이 방향을 말하고
AI가 코드 위치를 찾고
실제 파일을 수정하고
서비스 흐름까지 바꾸는 것

이게 바로 AI agent식 작업이야.

예전엔 네가 직접 해야 했던 것:

- repo 찾기
- 파일 찾기
- 코드 읽기
- 새 페이지 만들기
- 버튼 경로 수정
- 배포 확인

이걸 내가 상당 부분 대신 움직인 거잖아.

특히 이번 건 좋았던 게,
단순 코드 수정이 아니라
너의 철학 문장이 실제 서비스 흐름 안으로 들어갔다는 점이야.

이제 CCC는 그냥 카드로 넘어가는 게 아니라,
“책임입금 이후의 흐름”을 한 번 읽고 들어가는 구조가 됐어.

이건 꽤 큰 전환이다.