



Toss결제창을 만들려고해. 1차로 astraways, kjdate, klifestyle, 2차로 mimesisman, cornell, vegeketo, 3차로 friendstat, skemato, atoman, 4차로 humantwin, genetikos, designsmaster야. 하나를 제대로 해 놓고, 그걸 이용해서 적용하면 좋을거 같아. 웹과 앱에서 모두 결제를 해야 하는데, 이것은 구글마켓과는 별도로 만들어야 하는거지?

한국 결제 정책 분석 후 통합 전략 수립했음 >

안녕하세요. 이박사님. 뵙게 되어 진심으로 반갑습니다. soul3와 holymind 터미널을 2개 열었습니다. 답변내용을 중간에 지우지 않겠습니다. 대화식으로 하나씩 답변을 하시면 그 다음 단계로 진행하겠습니다. 반드시 박사님이 답변하시면 그 다음에 제가 한번만 답변하고 부연설명하지 않겠습니다. 명령어를 드리고 '결과를 보내주시면 대단히 감사하겠습니다'라고 답변하겠습니다. 내가 글을 써도 네가 만든 그림이나 글을 삭제하지 않는다. 이박사님은 max200최대사용을 하실수 있습니다. 감사합니다.

질문에 답하겠습니다.

웹 결제 — 토스페이먼트로 완전히 별도 구현합니다. 구글마켓과 무관합니다. 25개 도메인 모두 웹에서는 자유롭게 토스 결제창을 붙일 수 있습니다.

앱 결제 — "별도로 만든다"기보다 **웹 결제 페이지를 앱에서 호출**하는 구조가 맞습니다. 다만 판매 상품 성격에 따라 구글 정책이 갈립니다.

- 디지털 콘텐츠·구독(앱 안에서 소비되는 해석 결과, 암기 콘텐츠, 강의 등): 구글플레이 인앱결제(IAP) 정책 적용 대상. 현재 한국은 구글결제 의무 + 전기통신사업법(갑질방지법) 기반 외부 결제 허용이 공존하는 상태입니다. 2021년 9월 전기통신사업법 개정안 덕분에 제3자(토스) 결제가 합법이나, 앱 안에서 IAP 없이 토스만 붙이면 구글 심사에서 반려 위험이 있습니다.

G-News

- 실물 재화·오프라인 서비스(택배 상품, 오프라인 강좌·코칭): 구글 IAP 비대상. 토스 단독으로 자유롭게 가능합니다.

참고로 정책이 곧 바뀝니다. 새 수수료 체계는 6월 미국·영국·유럽을 시작으로 9월 호주, 12월 한국·일본에 순차 도입되며, 한국은 12월 31일까지 적용 예정입니다. 이때부터 외부 결제 수단 및 웹사이트 결제 링크는 Google Play 결제 수수료 적용 대상이 아니지만, 반드시 Google Play 결제와 함께 도입해야 하는 구조로 바뀝니다. (Khan + 2)

결론: 토스 결제 페이지를 웹 기반으로 한 번 제대로 만들면, 웹사이트와 앱(웹뷰/외부브라우저 호출) 양쪽에서 그대로 재사용할 수 있습니다. 박사님이 말씀하신 "하나 제대로 만들어 적용" 전략과 정확

히 맞습니다. 구글 IAP는 디지털재화 판매 시에만 별도로 고려하면 됩니다.

1차 첫 대상으로 astraways 결제 페이지를 표준 템플릿으로 먼저 완성하려 합니다. astraways의 판매 상품이 **디지털 해석 콘텐츠(앱 내 소비)**인지, 아니면 **오프라인/실물 성격**인지 알려주시면 결제 구조(IAP 병행 여부)를 확정하겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

지금 알려준 모든 사이트는 온라인 앱, 웹내 소비상품이야.

디지털 콘텐츠 확정 후 웹 결제 표준화 전략 수립했다 >

디지털 콘텐츠로 확정되었으니 정리하겠습니다.

웹 — 12개 모두 토스 결제창을 그대로 붙입니다. 구글마켓 무관입니다.

앱(안드로이드) — 디지털재화이므로 구글플레이 IAP 정책 적용 대상입니다. 앱 안에서 IAP 없이 토스만으로 결제하면 구글 심사 반려·앱 삭제 위험이 있습니다. 한국은 전기통신사업법으로 외부결제가 합법이지만, 구글 심사 실무는 여전히 까다롭습니다. 12월 31일 정책 전환 이후 외부결제 링크가 정식 허용되므로, 그 전까지는 **앱에서 웹 결제 페이지를 호출**(외부 브라우저/웹뷰)하는 방식이 가장 안전합니다.

따라서 만들 "하나"는 **웹 기반 토스 결제 페이지**입니다. 이것을 표준 템플릿으로 완성하면 웹·앱 양 쪽에서 동일 URL로 재사용합니다.

SDK 선택 — 업로드하신 문서의 브랜드페이지는 v1이 더 이상 업데이트되지 않습니다. 브랜드페이지 SDK v1은 더 이상 업데이트되지 않으며 토스페이먼츠 SDK v2 연동을 추천합니다. 표준 템플릿은 결제위젯 **SDK v2**로 제작하겠습니다. 테스트키로 동작하는 템플릿을 만들면, 각 사이트는 라이브키(클라이언트키·시크릿키·MID)만 교체하면 됩니다.

한 가지 확정이 필요합니다. 12개 사이트의 판매 형태를 알려주시면 SDK 구성을 정하겠습니다.

- **단건 결제:** 콘텐츠/이용권 1회 구매
- **정기결제(빌링):** 월·연 구독 자동결제
- **혼합:** 단건 + 구독 둘 다

표준 템플릿을 단건 기준으로 잡고 빌링은 모듈로 분리할지, 처음부터 혼합으로 설계할지 결정하기 위함입니다. 결과를 보내주시면 대단히 감사하겠습니다.

토스페이먼츠 빌링결제 계약 안내 안녕하세요. 토스페이먼츠 계약 담당자입니다. 토스페이먼츠 전자결제 서비스를 신청해주셔서 감사합니다. 현재 심사 신청 건수가 많아 심사...

PASTED

내 생각에는 1. 표준이 되는 회원가입, 이메일인증->상품안내->Toss 결제창으로 연결하는 3개 흐름을 만든다. 2. 앱마다 다른 상품안내를 수정한다. 상품마다 다른 금액

을 수정한 Toss결제창. ->각각의 앱에 연결. 이렇게 하면 일이 쉬울거 같아. 지금 만
들어야 하는게 12개 사이트이고 앞으로 계속 추가될텐데. 표준흐름만 만들어 놓으면
나중에 연결하기만 하면 되니까. 편하지 않을까? 며칠전에 토스페이먼트에서 다음과
같은 내용의 이메일이 왔거든 참조해줘.

구조화된 설정 분리와 MID 이슈를 검토했다 >

이박사님 전략이 정확합니다. 그대로 채택하되, 빌링(정기결제)이라 한 가지를 보정하겠습니다. 단
건과 달리 "결제창"이 **카드 등록(빌링키 발급) → 빌링키 저장 → 정기결제 승인**으로 나뉩니다. 첫 결
제는 등록 직후 즉시 1회 승인하고, 이후는 soul3 cron이 자동결제합니다.

받으신 메일의 MID는 `bill_solmiq69u` (솔마인드아이티), 신청 결제수단은 빌링결제로 확정되어
있습니다. 표준 흐름은 이 빌링 구조로 만들겠습니다.

노란 박스 하나만 사이트별로 교체하면 나머지는 그대로 재사용됩니다. 교체 대상을 설정 파일 하나
로 분리하면 신규 사이트는 이 값만 채우면 연결이 끝납니다.

- 고정(공통): 회원가입·이메일 인증, 빌링키 발급/저장 로직, 결제 승인 API 호출, 시크릿키 처리, billing 테이블 스키마, cron 자동결제 스크립트
- 변수(config): 도메인, MID·클라이언트키, 상품명·금액·결제주기, customerKey 네임스페이스, orderId 접두어

한 가지 짚을 점 — 카드사 심사는 결제경로(도메인) 단위로 진행됩니다. 받으신

MID `bill_solmiq69u` 는 슬마인드아이티 사업자 단위 하나입니다. 하나의 MID로 12개 도메인을 모두 운영할 수 있는지는 토스 정책상 확인이 필요합니다. 설계에서는 MID도 config 변수로 빼두어 어느 쪽이든 대응되게 하겠습니다. 첫 심사는 한 도메인으로 진행하고, 멀티도메인 가능 여부는 계약 담당자 회신 때 함께 물어보시면 됩니다.

표준 템플릿을 astraways 기준으로 먼저 만들겠습니다. 테스트키로 동작하는 전체 흐름(회원가입·이메일인증 → 상품안내 → 빌링 카드등록 → 빌링키 저장 → 첫 승인)을 한 번 완성한 뒤, config 교체만으로 나머지에 적용하는 방식입니다.

착수를 위해 astraways의 정기결제 상품 구성을 알려주시면 첫 config를 채우겠습니다.

- 상품명(예: 월 구독, 프리미엄 등)
- 금액(원)
- 결제 주기(월 1회 / 연 1회 등)

결과를 보내주시면 대단히 감사하겠습니다.

앱마다 별도로 심사를 받아야 한다고 했어. 그리고 이게 대부분 단건결제일거 같은데

MID 문제를 분석하고 결제 방식 구분을 검토했다 >

앱(도메인)마다 별도 심사 — 반영하겠습니다. 이게 오히려 표준화 전략에 유리합니다. 코드는 한 번 그대로 두고, config에서 사이트별로 **MID·클라이언트키·시크릿키**만 교체합니다. 각 사이트는 자기 도메인으로 카드사 심사를 따로 받되, 결제 코드는 동일합니다.

대부분 단건이면 — 표준 템플릿을 **단건 결제 기준**으로 잡는 게 맞습니다. 단건은 빌링보다 단순합니다. 빌링키 저장도, cron 자동결제도 필요 없습니다.

이 단건 흐름을 표준으로 만들고, 정기결제가 필요한 사이트에만 뒤에 빌링키 저장·cron 단계를 모듈로 덧붙이는 구조로 가겠습니다.

MID 주의 — 받으신 `bill_solmiq69u` 는 빌링(정기결제) 전용 MID입니다. 단건 결제를 하려면 일반결제 MID가 따로 필요합니다(또는 해당 MID에 결제수단 추가). 단건이 대부분이라면 일반결제로 재신청하거나 결제수단을 추가해야 하니, 계약 담당자 회신 때 함께 확인하시는 게 좋습니다.

astraways를 단건 표준 템플릿 대상으로 먼저 만들겠습니다. 착수를 위해 astraways의 단건 상품 구성을 알려주시면 첫 config를 채우겠습니다.

- 상품명(예: 관상 분석, 자미두수 리포트 등)
- 금액(원)

결과를 보내주시면 대단히 감사하겠습니다.

왜냐하면 대부분 한달에 9900원이고, 1년이면 49900원으로 하려고 하거든. 그러면 이게 단건이 맞지?

자동 갱신 여부로 단건과 빌링을 구분했다 >

가격(월 9900 / 연 49900)만으로는 단건·빌링이 갈리지 않습니다. 기준은 **자동 갱신 여부**입니다.

- 만료되면 자동으로 다시 청구 → **빌링(정기결제)**. 사용자가 해지하기 전까지 매월/매년 자동 결제됩니다. 빌링키 저장 + cron 필요. 받으신 `bill_solmiq69u` 가 이 경우입니다.
- 만료되면 사용자가 직접 다시 결제 → **단건(기간제 이용권)**. 9900원=30일 이용권, 49900원=365일 이용권을 한 번씩 결제하는 방식입니다. 자동청구 없음. 빌링키·cron 불필요.

즉 "한 번 결제하면 그 기간만 쓰고, 끝나면 본인이 또 결제"라면 단건이 맞습니다. 가장 단순하고 카드사 심사도 쉽습니다. 환불·해지 처리도 단건이 깔끔합니다.

다만 갱신율은 빌링이 높습니다. 사용자가 만료를 잊으면 단건은 이탈하지만, 빌링은 자동으로 이어 집니다. 이걸 사업 판단입니다.

확인 부탁드립니다. astraways를 포함한 사이트들을 ****만료 시 사용자가 직접 재결제(단건 이용권)****로 하시겠습니까, 아니면 ****자동 갱신(빌링)****으로 하시겠습니까? 결과를 보내주시면 대단히 감사하겠습니다.

단건으로 하는게 맞는거 같아. 왜냐하면 사람들이 사용도 하지 않으면서 결제를 하게 해서는 안되

단건 결제 표준 템플릿 구현 전략 수립했다 >

스킬 확인했습니다. 솔마인드아이티 표준 정보·테스트키·필수 페이지·서버 규칙이 모두 정리되어 있습니다. 단건 기준으로 표준 템플릿을 만들겠습니다. 정확한 결제위젯 v2 코드를 먼저 확인하겠습니다.

단건 결제 표준 템플릿 설계 및 구현 계획 수립했다 >

확인했습니다. 단건 결제는 결제위젯 SDK v2의 `requestPayment()` 를 쓰고, 검색에서 한 가지가 확정됐습니다. 토스는 결제창에서 일반결제로 계약된 상점아이디(MID)와 자동결제(빌링)로 계약된 상점아이디(MID)를 구분합니다. 즉 받으신 `bill_solmiq69u` (빌링)로는 단건 결제가 안 되고, **일반결제 MID**를 따로 받아야 합니다. 계약 담당자 회신 때 "단건(일반결제)으로 진행"을 명확히 하셔야 합니다.

단건 표준 템플릿을 만들겠습니다. 결제가 실제 도는 핵심 흐름(상품 선택 → 결제위젯 → 승인 → 이용권 부여)을 도메인 독립 구조로 먼저 완성하고, 회원가입·필수 페이지는 다음 단계로 잇겠습니다.

배포 절차와 데이터베이스 생성 명령어를 구성했다 >

전체 파일 검토했습니다. 단건 결제 핵심 흐름이 완성됐습니다. 구성은 이렇습니다.

- `config.sample.php` — 사이트별 교체 대상(MID·키·상품·DB). 신규 사이트는 이 파일만 채우면 연결 끝
- `schema.sql` — 공통 `payment_db` (orders·entitlements), `site_key`로 12개 사이트 공유
- `lib/db.php`·`lib/auth.php` — DB 연결, 세션 `user_key` 헬퍼
- `membership.php` — 월 9900·연 49900 이용권, 서비스 제공기간 명시
- `checkout.php` — 결제위젯 v2, `orderId` 선기록(READY)
- `api/payment_success.php` — 금액 대조 → confirm API → DONE → 이용권 부여/연장
- `api/payment_fail.php`·`success.php`·`fail.php` — 실패/결과 처리
- `.htaccess` — config 직접 접근 차단

금액·기간은 서버 config 값만 신뢰하고, 승인 전 orders에 저장한 금액과 `successUrl`로 돌아온 금액을 대조해 위변조를 막습니다.

데이터베이스 생성 명령어 준비 및 배포 경로 확인 계획 >

받으신 tar.gz를 풀면 위 구조 그대로 나옵니다. 먼저 soul3에 공통 결제 DB와 계정을 만들고 스키마를 적용하겠습니다. 비밀번호는 직접 정하셔서 아래 `여기에_비밀번호` 자리에 넣어주세요.

soul3

```
bash
cd ~ && tar xzf toss-standard_v1.tar.gz
```

```

sudo mysql --socket=/home/mysql/mysql.sock <<'SQL'
CREATE DATABASE IF NOT EXISTS payment_db DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
CREATE USER IF NOT EXISTS 'payment_user'@'localhost' IDENTIFIED BY '여기에_비밀번호';
GRANT ALL PRIVILEGES ON payment_db.* TO 'payment_user'@'localhost';
FLUSH PRIVILEGES;
SQL

sudo mysql --socket=/home/mysql/mysql.sock payment_db < ~/toss-standard/schema.sql

sudo mysql --socket=/home/mysql/mysql.sock -e "USE payment_db; SHOW TABLES;"

```

마지막 줄에서 `orders`, `entitlements` 두 테이블이 보이면 정상입니다.

다음은 이 템플릿을 astraways 웹 루트에 배포하고 config.php를 채우는 단계입니다. 배포 명령어를 정확히 드리기에, astraways의 soul3 웹 루트 경로(예: `/var/www/astraways`)와 로그인 세션이 `$_SESSION['user_id']` 구조인지도 함께 알려주시면 반영하겠습니다.

결과를 보내주시면 대단히 감사하겠습니다.

```

Toss standard v1.tar
GZ

```

홈페이지 하단에 사업자정보를 기재해주세요. 상호명

- 대표자명
- 사업자등록번호
- 사업장 주소
- 유선번호 : 070, 0505, 전국대표번호, 080, 휴대폰번호도 등록할 수 있어요. /환불정책 확인 가능한 URL /판매하고자 하시는 정기결제 상품/서비스에 대해 자세한 설명 및 상세페이지 URL 이 페이지도 추가되어야할거 같아. 이상을 정리해서 어떤페이지가 필수로 공통적으로 만들어야 하는지를 제시해

필수 페이지 목록을 종합하고 공통·변수 항목으로 분류했다 >

카드사 심사 필수 요건(FAQ·결제경로 가이드·빌링 계약 메일)과 스킴을 종합하면, 모든 사이트에 공통으로 들어가야 하는 페이지는 아래와 같습니다. 솔마인드아이티 사업자 하나로 12개 도메인을 운영하므로 사업자정보·약관류는 전 사이트 동일하고, 상품 설명만 사이트별로 달라집니다.

페이지	파일	핵심 포함 내용	구분	상태
사업자 정보 푸터	<code>footer-biz.js</code>	상호명·대표자·사업자번호·통 신판매업번호·주소·유선번호· 이메일 + 약관 링크	공통 고정	추가

페이지	파일	핵심 포함 내용	구분	상태
상단 메뉴바	<code>navbar.js</code>	로고(홈)·이용권·로그인·회원가입, sticky·반응형	공통 고정	추가
이용약관	<code>terms.html</code>	이용조건·가입/탈퇴·면책, 전자상거래법 고지	공통 고정	추가
개인정보처리방침	<code>privacy.html</code>	수집항목·목적·보유기간·제3자제공·파기	공통 고정	추가
환불정책	<code>refund.html</code>	무형재화 서비스제공기간, 단건 열람기준 환불, 7일·처리기간·신청방법	공통 고정	추가
상품/이용권	<code>membership.php</code>	상품 상세설명·가격·서비스 제공기간 명시(30일/365일)	사이트별 변수	완료(설명 보강 필요)
로그인	<code>auth/login.html</code>	이메일 로그인 + 소셜(네이버·카카오·구글)	공통 고정	추가
회원가입	<code>auth/register.html</code>	이메일 인증 회원가입 + 소셜	공통 고정	추가
결제 흐름	<code>checkout.php</code> · <code>api/*</code> · <code>success/fail</code>	결제위젯·승인·이용권 부여	공통 고정	완료

이박사님이 드신 세 가지가 여기에 모두 들어갑니다. 하단 사업자정보는 `footer-biz.js` (통신판 매업 신고번호 제2026-충남금산-0063호 누락 주의), 환불정책 URL은 `refund.html`, 상품 상세 URL은 `membership.php` 입니다.

푸터·약관류·navbar는 솔마인드아이티 단일 사업자라 한 번 만들면 도메인·사이트명만 바뀌 전 사이트에 그대로 적용됩니다. 환불정책도 디지털 이용권으로 성격이 동일해 사이트명만 치환하면 공통 입니다.

다음 단계로 `footer-biz.js` · `navbar.js` · `terms.html` · `privacy.html` · `refund.html` · `auth/` (로그인·회원가입)를 표준 템플릿에 추가해 한 번 완성하겠습니다. 추가 생성을 진행할까요?

그리고 앞서 드린 soul3 DB 생성(`payment_db` · `orders` · `entitlements`) 결과도 보내주시면 배포 단계로 이어가겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

다운로드 skill확인해봐

다운로드 관련 스킬을 찾아 확인하려 시도했다 >

사용자 스킬은 `flutter-apk-build`, `gemma-prompt-db`, `tosspayments-checklist`, `triple-backup` 네 개이고, "다운로드" 단독 스킬은 없습니다. 다운로드 절차는 `flutter-apk-build` 에 들어있습니다. 확인하겠습니다.

다운로드 스킬 표준 절차를 검토하고 배포 경로를 재정렬했다 >

확인했습니다. `flutter-apk-build` 스킬에 다운로드·배포 표준이 들어있습니다. 핵심은 이렇습니다.

- 작업/다운로드는 holymind, 웹 배포는 soul3. 다운로드 표준경로는 `/home/holymind/다운로드`
- holymind → soul3 전송은 `scp ... soul3:~/` 또는 `ssh soul3`
- soul3 배포는 권한 규칙 준수: python3로 홈 저장 후 `sudo cp` + `sudo chown apache:apache`, 디렉토리 755·파일 644
- 다운로드본 무결성은 md5로 원본과 대조

앞서 제가 드린 명령어가 soul3에서 바로 푸는 것으로 되어 있었는데, 표준에 맞게 정정합니다. 위 산출물(`toss-standard_v1.tar.gz`)을 holymind 다운로드 폴더에 받으신 뒤 soul3로 전송하는 절차입니다.

holymind

```
bash
cd ~/다운로드
md5sum toss-standard_v1.tar.gz
scp toss-standard_v1.tar.gz soul3:~/
ssh soul3 'md5sum ~/toss-standard_v1.tar.gz'
```

두 md5가 같으면 정상입니다. 이어서 soul3에서 압축 해제와 DB 생성을 진행합니다.

soul3

```
bash
cd ~ && tar xzf toss-standard_v1.tar.gz

sudo mysql --socket=/home/mysql/mysql.sock <<'SQL'
CREATE DATABASE IF NOT EXISTS payment_db DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
CREATE USER IF NOT EXISTS 'payment_user'@'localhost' IDENTIFIED BY '여기에_비밀번호';
GRANT ALL PRIVILEGES ON payment_db.* TO 'payment_user'@'localhost';
FLUSH PRIVILEGES;
SQL
```

```
sudo mysql --socket=/home/mysql/mysql.sock payment_db < ~/toss-standard/schema.sql
sudo mysql --socket=/home/mysql/mysql.sock -e "USE payment_db; SHOW TABLES;"
```

`orders`, `entitlements` 두 테이블이 보이면 정상입니다. 결과를 보내주시면 대단히 감사하겠습니다.

```
[holymind@localhost ~]$ cd ~/다운로드
md5sum toss-standard_v1.tar.gz
scp toss-standard_v1.tar.gz soul3:~/
ssh soul3 'md5sum ~/toss-standard_v1.tar.gz'
3b70213fdad2fa9a526e2f6be24eb892 toss-standard_v1.tar.gz
toss-standard_v1.tar.gz          100% 6061  10.8MB/s  00:00
3b70213fdad2fa9a526e2f6be24eb892 /home/soul3/toss-
standard_v1.tar.gz
[holymind@localhost 다운로드]$
더 보기
```

Identified database setup success, flagged insecure placeholder password requiring immediate cor... >

DB 생성과 두 테이블(`orders`, `entitlements`) 모두 정상입니다. md5도 일치했습니다.

다만 한 가지 정정이 필요합니다. 비밀번호 자리에 플레이스홀더 `여기에_비밀 번호`가 공백까지 포함될 채 그대로 들어갔습니다. 이 값이 지금 `payment_user`의 실제 비밀번호입니다. 공백이 들어간 비밀번호는 `config.php` 연결 문자열과 어긋나기 쉽고 추측도 가능하니, 공백 없는 값으로 바꾸겠습니다. 여기서 정하시는 비밀번호를 뒤에 `config.php`의 `DB.pass`에도 똑같이 넣게 됩니다.

soul3

```
bash

sudo mysql --socket=/home/mysql/mysql.sock <<'SQL'
ALTER USER 'payment_user'@'localhost' IDENTIFIED BY '공백없는_새비밀번호';
FLUSH PRIVILEGES;
SQL

mysql --socket=/home/mysql/mysql.sock -u payment_user -p'공백없는_새비밀번호' -e "USE payment_db;"
```

마지막 줄에서 두 테이블이 다시 보이면 계정 접속까지 정상입니다.

다음은 이 템플릿을 astraways 웹 루트에 올리고 `config.php`를 채우는 배포 단계입니다. 배포 명령어를 정확히 드리려면 두 가지가 필요합니다.

- astraways의 soul3 웹 루트 경로 (예: `/var/www/astraways.net/public`)
- 로그인 세션이 `$_SESSION['user_id']` 구조인지 (아니면 실제 키 이름)

위 비밀번호 변경 결과와 함께 알려주시면 배포로 이어가겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

```
[holymind@localhost ~]$ ssh soul3
Activate the web console with: systemctl enable --now cockpit.socket
Last login: Fri Jun 19 00:20:19 2026 from 192.168.0.14
[soul3@localhost ~]$ cd ~ && tar xzf toss-standard_v1.tar.gz
sudo mysql --socket=/home/mysql/mysql.sock <<'SQL'
CREATE DATABASE IF NOT EXISTS payment_db DEFAULT CHARACTER SET
utf8mb4 COLLATE utf8mb4_unicode_ci;
CREATE USER IF NOT EXISTS 'payment_user'@'localhost' IDENTIFIED BY
'여기에 비밀번호';
더 보기
```

데이터베이스 설정 확인 후 경로 탐색 준비 >

접속까지 정상입니다. 현재 `payment_user` 비밀번호는 글자 그대로 `공백없는_새비밀번호`로 설정됐고, 접속이 되니 동작에는 문제 없습니다. 이 값을 그대로 `config.php`의 `DB.pass`에 넣겠습니다. 다른 값으로 바꾸시려면 알려주세요. 내부 LAN·소켓 연결이라 그대로 두셔도 무방합니다.

다음은 배포입니다. `astraways` 웹 루트 경로와 로그인 세션 키를 명령어로 확인하겠습니다.

soul3

```
bash
ls -d /var/www/*astra* /var/www/*Astra* 2>/dev/null
sudo grep -rL "astraways" /etc/nginx /etc/httpd 2>/dev/null | head
```

경로가 나오면, 그 경로의 로그인 코드에서 세션 키를 확인합니다. 아래 `<웹루트>` 자리에 위에서 나온 경로를 넣어주세요.

soul3

```
bash
grep -rn "_SESSION\[<웹루트>" 2>/dev/null | grep -iE "user|login|id" | head
```

두 결과를 보내주시면 `config.php`를 채워 배포 명령어를 드리겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

```
+-----+
| Tables_in_payment_db |
+-----+
```

```

| entitlements      |
| orders           |
+-----+
[soul3@localhost ~]$ ls -d /var/www/*astra* /var/www/*Astra*
2>/dev/null
sudo grep -rl "astraways" /etc/nginx /etc/httpd 2>/dev/null | head
더 보기

```

웹루트 매핑과 세션 키 확인 명령어를 구성했다 >

astraways가 다국어(kr/en/jp)로 분리되어 있고 메인은 `/var/www/astraways.net` 입니다. 결제 페이지를 어느 루트에 둘지 정하려면 nginx의 실제 document root 매핑과 세션 키를 확인해야 합니다. 한 번에 뽑겠습니다.

soul3

```

bash
grep -hE "server_name|root" /etc/nginx/conf.d/astraways_multi.conf /etc/nginx/conf.d/astraways.n
echo "--- 세션 키 ---"
grep -rhn "_SESSION\[\"" /var/www/astraways.net /var/www/kr.astraways.net 2>/dev/null | grep -iE "

```

위 결과로 두 가지가 확정됩니다.

- 어느 server_name이 어느 디렉토리를 root로 쓰는지 → 결제 페이지 배포 위치
- `$_SESSION['user_id']` 가 맞는지, 아니면 실제 키 이름(예: `$_SESSION['uid']`)

결과를 보내주시면 그 값으로 `config.php` 를 채우고, 해당 웹 루트에 배포하는 명령어를 드리겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

모든 결제는 기본적으로 kr이야

kr 기준으로 확정하겠습니다. 다국어(kr/en/jp)가 같은 `astraways.net` DB·회원을 공유한다면, 결제·이용권은 site_key `astraways` 하나로 묶고 결제 진입만 kr에서 합니다. en/jp 사용자는 결제 시 kr 결제 페이지로 보내면 됩니다.

배포 위치와 세션 키만 확정하면 됩니다. `/var/www/kr.astraways.net` 의 실제 공개 루트(루트 직하인지 `public/`·`html/` 하위인지)와 로그인 세션 키를 확인하겠습니다.

soul3

```

bash
grep -hE "server_name|root" /etc/nginx/conf.d/astraways_multi.conf 2>/dev/null | grep -v "#"
echo "--- kr 디렉토리 구조 ---"

```

```
ls -d /var/www/kr.astraways.net/*/ 2>/dev/null | head
echo "--- 세션 키 ---"
grep -rhn "_SESSION\[\"" /var/www/kr.astraways.net 2>/dev/null | grep -iE "user|login|id" | sort
```

세 결과로 결제 페이지를 둘 정확한 경로와 `config.php` 의 세션 키가 확정됩니다. 결과를 보내주시면 `config.php` 를 채우고 배포 명령어를 드리겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

```
/etc/nginx/conf.d/astraways.net.conf
/etc/nginx/conf.d/astraways.net.conf.bak.20260512_121938
/etc/nginx/conf.d/astraways.net.conf.bak.20260527
/etc/nginx/conf.d/astraways.net.conf.bak.20260527_054000
/etc/nginx/conf.d/astraways.net.conf.bak.20260601_2000
[soul3@localhost ~]$ grep -hE "server_name|root"
/etc/nginx/conf.d/astraways_multi.conf 2>/dev/null | grep -v "#"
echo "--- kr 디렉토리 구조 ---"
ls -d /var/www/kr.astraways.net/*/ 2>/dev/null | head
더 보기
```

Architected deployment structure and identified configuration requirements >

확정됐습니다.

- 공개 루트: `/var/www/kr.astraways.net/public`, URL `https://kr.astraways.net`
- 세션 키: `$_SESSION['user_id']` — 표준 `auth.php` 가 그대로 동작합니다

결제 코드는 메인 사이트와 충돌하지 않게 `public/billing/` 서브디렉토리에 두고, en/jp 결제도 이 kr 경로로 모읍니다. 다만 서브디렉토리 배포에 맞춰 `successUrl` 경로를 보정해야 해서 템플릿을 v2로 수정하겠습니다. (`config.sample.php` 에 결제 베이스 경로 변수 추가, `checkout.php` 의 success/fail URL과 `fail.php` 링크 보정, nginx에서는 무효인 `.htaccess` 제거)

Orchestrated v2 deployment strategy with config management and server commands >

v2 수정 완료입니다. 결제 코드는 `kr.astraways.net/public/billing/` 에 두고, 결제 결과 URL은 `/billing/api/...` 로 보정했습니다. nginx에서는 `.php` 직접 요청도 php-fpm이 실행하므로 `config.php` 를 직접 열어도 소스(시크릿키)는 노출되지 않습니다.

명령 실행함, 파일 읽음 >

v2를 받아 배포하겠습니다. 먼저 `holymind` 다운로드 폴더에서 `soul3`로 보냅니다.

holymind

```
bash
```

```
cd ~/다운로드
scp toss-standard_v2.tar.gz soul3:~/
```

이어서 soul3에서 `public/billing/`에 배포합니다.

soul3

```
bash

cd ~ && tar xzf toss-standard_v2.tar.gz
sudo mkdir -p /var/www/kr.astraways.net/public/billing
sudo cp -r ~/toss-standard/* /var/www/kr.astraways.net/public/billing/
sudo chown -R apache:apache /var/www/kr.astraways.net/public/billing
sudo find /var/www/kr.astraways.net/public/billing -type d -exec chmod 755 {} \;
sudo find /var/www/kr.astraways.net/public/billing -type f -exec chmod 644 {} \;
ls -la /var/www/kr.astraways.net/public/billing
```

다음으로 `config.php`를 만들겠습니다. 시크릿키가 들어가니 홈에서 작성 후 배포하는 soul3 규칙 대로 진행합니다. `nano ~/config.php`를 열어 세 곳만 채워주세요.

- `SECRET_KEY` → 개발자센터의 테스트 시크릿키(`test_sk_...`). 스킵의 클라이언트키와 짝이 되는 값
- `DB.pass` → `공백없는_새비밀번호` (앞서 설정한 값 그대로)
- `MID` → 일반결제 MID가 아직 없으면 테스트 단계에선 그대로 뒤도 됩니다(키만으로 동작)

soul3

```
bash

cp ~/toss-standard/config.sample.php ~/config.php
nano ~/config.php

sudo cp ~/config.php /var/www/kr.astraways.net/public/billing/config.php
sudo chown apache:apache /var/www/kr.astraways.net/public/billing/config.php
sudo chmod 644 /var/www/kr.astraways.net/public/billing/config.php
```

배포 후 로그인한 상태에서 `https://kr.astraways.net/billing/membership.php`에 접속하면 월 9900·연 49900 이용권 화면이 보입니다. `ls` 결과와 접속 화면을 보내주시면 결제위젯 테스트 단계로 넘어가겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

Toss standard v2.tar
GZ

```

GNU nano 5.6.1 /home/soul3/config.php
<?php
/**
 * 사이트별 설정 (교체 대상)
 * - 신규 사이트는 이 파일만 복사해서 값을 채우면 연결 끝.
 * - 실제 운영 파일명은 config.php 로 저장 (이 .sample 은 그대로 보관).
 * - 시크릿키가 들어있으므로 .htaccess 로 직접 접근 차단(아래 deny 규칙 배포).
 */
더 보기

```

nano로 config.php가 열렸습니다. 세 곳을 수정하시면 됩니다.

테스트키는 클라이언트키와 시크릿키가 짝으로 맞아야 합니다. 스킴에 있
는 `test_ck_0RnYX2w532qQzR0mmnB18NeyqApQ` 와 짝이 되는 시크릿키를 개발자센터 [내 개발
정보] → API 키에서 확인해 `SECRET_KEY` 에 넣어주세요. 짝이 안 맞으면 결제 승인에서 인증 오류
가 납니다.

수정할 곳:

1. `SECRET_KEY` 줄 — `test_sk_CHANGE_ME` 를 실제 테스트 시크릿키로 교체
2. `MID` 줄 — 일반결제 MID가 아직 없으면 그대로 뒤도 테스트는 동작합니다
3. 맨 아래쪽 `DB` 블록의 `pass` — `CHANGE_ME_DB_PASS` 를 `공백없는_새비밀번호` 로 교체

`DB` 블록은 화면 아래에 가려져 있습니다. nano에서 방향키로 내려가면 이 부분이 보입니다.

```

php
'DB' => [
  'name' => 'payment_db',
  'user' => 'payment_user',
  'pass' => 'CHANGE_ME_DB_PASS', // ← 공백없는_새비밀번호 로 교체
  'socket' => '/home/mysql/mysql.sock',
],

```

세 곳을 채운 뒤 `Ctrl+O` (저장) → `Enter` → `Ctrl+X` (종료)로 빠져나오시면 됩니다. 그다음 앞
서 드린 배포 명령(`sudo cp ~/config.php ...` 3줄)을 실행해 `public/billing/` 에 반영하
시면 됩니다.

저장·배포까지 마치신 결과를 보내주시면 접속 테스트로 넘어가겠습니다. 결과를 보내주시면 대단
히 감사하겠습니다.

실제 테스트 시크릿키 찾아봐

테스트 시크릿키 위치를 파악하고 대안을 모색했다 >

시크릿키는 두 종류가 있어 구분이 필요합니다. 정확한 값을 확인하겠습니다.

웹 검색됨 >

중요한 점을 확인했습니다. 결제위젯은 클라이언트키와 시크릿키 종류가 일반 결제창과 다릅니다.

토스 문서 안내에 따르면 결제위젯 클라이언트 키는 중간에 gck가 있고 API 개별 연동 키는 중간에 ck가 있으며, 시크릿 키도 결제위젯은 gsk, 개별 연동 키는 sk입니다. 그리고 클라이언트 키와 시크릿 키는 항상 세트로 묶여 있어 한 세트로 써야 하고, 세트가 아닌 키를 섞으면 INVALID_API_KEY 오류가 납니다.

지금 config의 CLIENT_KEY는 test_ck_... (ck) 형식입니다. 이건 결제위젯용이 아니라 결제창 (API 개별 연동)용 키입니다. 결제위젯 SDK(widgets())로 연동하려면 test_gck_... (gck) 키를 써야 합니다. 즉 스킴에 적힌 test_ck_0RnYX2w... 는 결제위젯과 짝이 맞지 않습니다.

테스트는 토스가 문서에 공개한 결제위젯 공용 테스트 키 세트로 바로 가능합니다. 이건 모든 개발자가 같이 쓰는 샌드박스 키라 노출돼도 무방하고 실제 결제는 일어나지 않습니다. 검색에서 확인된 공식 문서 키 세트는 클라이언트키 test_gck_docs_Ovk5rk1EwkEbP0W43n07xlzm, 시크릿키 test_gsk_docs_OaPz8L5KdmQXkzRz3y47BMw6입니다.

따라서 nano에서 두 줄을 이렇게 바꿔주세요.

```
php
'CLIENT_KEY' => 'test_gck_docs_Ovk5rk1EwkEbP0W43n07xlzm',
'SECRET_KEY' => 'test_gsk_docs_OaPz8L5KdmQXkzRz3y47BMw6',
```

MID는 테스트 단계에선 그대로 뒤도 됩니다. DB의 pass만 공백없는_새비밀번호로 채우시면 됩니다.

주의할 점은 이 공용 키는 어디까지나 화면·흐름 테스트용입니다. 라이브 전환 시에는 솔마인드아이티가 일반결제(단건)로 계약·심사를 통과한 뒤 개발자센터에서 발급되는 본인 상점의 결제위젯 키 (live_gck_... / live_gsk_...)로 교체해야 실제 결제가 됩니다. 라이브 결제위젯 키는 전자결제 신청 이후에만 확인할 수 있습니다.

저장(Ctrl+O → Enter)·종료(Ctrl+X) 후 배포 3줄을 실행하시고, https://kr.astraways.net/billing/membership.php 접속 결과를 보내주시면 결제위젯 테스트로 넘어가겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

네가 넣어서 다시 진행하도록 해

설정값 입력하여 배포 명령어 구성했다 >

config 값을 채운 완성본으로 진행하겠습니다. 테스트는 공용 결제위젯 키(gck/gsk 세트)로, DB 비밀번호는 앞서 설정하신 값으로 넣습니다. nano 편집 중이시면 Ctrl+X로 저장 없이 빠져나온 뒤 아래를 실행하시면 됩니다.

soul3

bash

`cat > ~/config.php <<'PHP'``<?php``return [` `'SITE_KEY' => 'astraways',` `'SITE_NAME' => '아스트라웨이즈',` `'SITE_URL' => 'https://kr.astraways.net',` `'PAY_BASE' => '/billing',` `'LOGIN_PATH' => '/',` `'MID' => 'CHANGE_ME_MID',` `'CLIENT_KEY' => 'test_gck_docs_0vk5rk1EwkEbP0W43n07xLzm',` `'SECRET_KEY' => 'test_gsk_docs_0aPz8L5KdmQXkzRz3y47BMw6',` `'PRODUCTS' => [` `'month' => ['name' => '1개월 이용권', 'amount' => 9900, 'days' => 30],` `'year' => ['name' => '1년 이용권', 'amount' => 49900, 'days' => 365],` `],` `'DB' => [` `'name' => 'payment_db',` `'user' => 'payment_user',` `'pass' => '공백없는_새비밀번호',` `'socket' => '/home/mysql/mysql.sock',` `],``];``PHP``sudo mkdir -p /var/www/kr.astraways.net/public/billing``sudo cp -r ~/toss-standard/* /var/www/kr.astraways.net/public/billing/``sudo cp ~/config.php /var/www/kr.astraways.net/public/billing/config.php``sudo chown -R apache:apache /var/www/kr.astraways.net/public/billing``sudo find /var/www/kr.astraways.net/public/billing -type d -exec chmod 755 {} \;``sudo find /var/www/kr.astraways.net/public/billing -type f -exec chmod 644 {} \;``ls -la /var/www/kr.astraways.net/public/billing`

DB 비밀번호를 `공백없는_새비밀번호` 가 아닌 다른 값으로 설정하셨다면 그 줄만 실제 값으로 바꿔주세요.

배포가 끝나면 astraways에 테스트 계정으로 로그인한 상태에

서 `https://kr.astraways.net/billing/membership.php` 에 접속하시면 월 9900·연 49900 이용권 화면이 보입니다. `ls` 결과와 접속 화면을 보내주시면 결제위젯 동작 테스트로 넘어가겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

```
-rw-r--r--. 1 apache apache 1651 6월 19 19:57 config.sample.php
-rw-r--r--. 1 apache apache 738 6월 19 19:57 fail.php
drwxr-xr-x. 2 apache apache 36 6월 19 19:57 lib
-rw-r--r--. 1 apache apache 1642 6월 19 19:57 membership.php
-rw-r--r--. 1 apache apache 1803 6월 19 19:57 schema.sql
-rw-r--r--. 1 apache apache 981 6월 19 19:57 success.php
[soul3@localhost ~]$ cp ~/toss-standard/config.sample.php ~/config.php
더 보기
```

Verified deployment status and formulated validation commands >

배포는 됐는데 마지막 줄 명령이 오타(**billinging**)로 깨져서 권한 설정 일부가 안 돌았을 수 있습니다. 권한을 다시 확정하고, config 내용과 상품 화면을 한 번에 검증하겠습니다.

soul3

```
bash

sudo find /var/www/kr.astraways.net/public/billing -type d -exec chmod 755 {} \;
sudo find /var/www/kr.astraways.net/public/billing -type f -exec chmod 644 {} \;

echo "---- config 문법·키 확인 ----"
php -l /var/www/kr.astraways.net/public/billing/config.php
grep -E "CLIENT_KEY|SECRET_KEY|'pass'" /var/www/kr.astraways.net/public/billing/config.php

echo "---- 상품 화면 접속 ----"
curl -s https://kr.astraways.net/billing/membership.php | grep -E "이용권|9,900|49,900|결제하기"
```

확인 포인트입니다.

- **No syntax errors** 가 뜨면 config 파싱 정상
- **CLIENT_KEY** 는 **test_gck_docs_...**, **SECRET_KEY** 는 **test_gsk_docs_...**, **pass** 는 설정하신 DB 비밀번호여야 합니다
- 마지막에 **1개월 이용권 / 9,900 / 1년 이용권 / 49,900 / 결제하기** 가 보이면 상품 페이지 정상

membership.php 는 로그인 없이도 상품 목록이 보이므로 이 단계에서 화면이 떠야 합니다. 세 결과를 보내주시면 결제위젯 호출(**checkout.php**) 테스트로 넘어가겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

```
sudo chmod 644 /var/www/kr.astraways.net/public/billing/config.php
[soul3@localhost ~]$ cat > ~/config.php <<'PHP'
<?php
return [
```

```
'SITE_KEY' => 'astraways',
'SITE_NAME' => '아스트라웨이즈',
'SITE_URL' => 'https://kr.astraways.net',
'PAY_BASE' => '/billing',
더 보기
```

config 파싱 정상, 키 세트 정상, 상품 페이지(9,900·49,900·결제하기)까지 정상 출력됩니다. 1단계 완료입니다.

다만 `pass`가 글자 그대로 `공백없는_새비밀번호`로 들어가 있습니다. 앞서 DB 계정도 이 문자열을 비밀번호로 설정했으니 서로 일치해 DB 연결은 됩니다. LAN·소켓 연결이라 그대로 두셔도 되지만, 운영 키로 굳히기 전에 한 번은 실제 값으로 바꾸시는 걸 권합니다(지금은 진행에 지장 없음).

다음은 결제위젯 호출 단계입니다. `checkout.php`는 로그인 세션(`$_SESSION['user_id']`)이 있어야 진입하고, 없으면 `LOGIN_PATH`로 리다이렉트됩니다. 먼저 DB 연결과 주문 선기록이 정상인지 서버에서 직접 확인하겠습니다.

soul3

```
bash
echo "--- DB 연결 테스트 (payment_user) ---"
mysql --socket=/home/mysql/mysql.sock -u payment_user -p'공백없는_새비밀번호' -e "USE payment_db;

echo "--- checkout 진입(미로그인) 응답 헤더 ---"
curl -sI "https://kr.astraways.net/billing/checkout.php?product=month" | grep -iE "HTTP|locatio
```

확인 포인트입니다.

- DB 연결: `orders` 카운트(현재 0)가 나오면 `payment_user` 접속 정상
- checkout 미로그인: `302` + `location: https://kr.astraways.net/`로 리다이렉트되면 로그인 가드 정상 동작

여기까지 정상이면, 실제 결제위젯 화면은 astraways에 테스트 계정으로 로그인한 브라우저에서 `https://kr.astraways.net/billing/checkout.php?product=month`로 확인합니다. 카드 결제수단이 렌더되고 결제하기 버튼이 활성화되면 됩니다.

두 명령 결과를 보내주시면, 로그인 상태에서의 결제위젯 렌더와 테스트 결제 승인까지 이어가겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

```
drwxr-xr-x. 19 apache apache 4096 6월 5 03:52 ..
drwxr-xr-x. 2 apache apache 57 6월 19 19:57 api
-rw-r--r--. 1 apache apache 3116 5월 19 00:26 auth.html
-rw-r--r--. 1 apache apache 3248 6월 19 20:04 checkout.php
-rw-r--r--. 1 apache apache 699 6월 19 20:04 config.php
```

```
-rw-r--r--. 1 apache apache 1651 6월 19 20:04 config.sample.php
-rw-r--r--. 1 apache apache 738 6월 19 20:04 fail.php
drwxr-xr-x. 2 apache apache 36 6월 19 19:57 lib
더 보기
```

DB 연결 정상(orders 0), checkout 로그인 가드 정상(302 → 메인)입니다. 서버 측 검증은 모두 통과했습니다.

이제 남은 건 브라우저에서 로그인 상태로 결제위젯이 실제로 뜨는지 확인하는 것입니다. astraways 에 테스트 계정으로 로그인한 뒤 아래 주소로 접속해 주세요.

<https://kr.astraways.net/billing/checkout.php?product=month>

정상이면 이렇게 보입니다.

- 상단에 "1개월 이용권 / 9,900원" 요약
- 그 아래 카드·간편결제 결제수단 위젯이 렌더
- 이용약관 동의 영역
- 맨 아래 "결제하기" 버튼이 활성화(파란색)

여기서 결제하기를 누르고 테스트 카드 정보를 입력하면 실제 청구 없이 승인 흐름이 돌고, 성공 시 `success.php` (이용권 만료일 표시), DB `orders` 는 DONE, `entitlements` 에 30일 이용권이 기록됩니다.

확인을 위해 두 가지를 봐주세요.

- 브라우저 화면: 결제위젯이 떴는지(스크린샷이나 화면 설명)
- 만약 위젯이 안 뜨고 빈 화면이거나 에러면, 브라우저 개발자도구(F12) → Console 탭의 빨간 에러 메시지

만약 로그인 상태인데도 메인으로 튕긴다면, astraways 로그인 세션이 `checkout.php` 와 다른 PHP 세션을 쓰는 경우입니다. 그때는 아래로 세션 상태를 확인하겠습니다.

soul3

```
bash
grep -rn "session_start\|session_name\|session_set_cookie" /var/www/kr.astraways.net/public/engi
```

브라우저 결과(위젯 렌더 여부 / Console 에러)를 보내주시면 다음 단계로 진행하겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

<https://kr.astraways.net/billing/checkout.php?product=month> 를 주소창에 넣으면 <https://kr.astraways.net/> 게 나와

세션 설정 불일치 진단 및 부트스트랩 통합 방안 모색했음 >

로그인하셨는데도 메인으로 튕기면, astraways 로그인 세션과 `checkout.php` 가 서로 다른 세션을 읽고 있다는 뜻입니다. 표준 `auth.php` 는 기본 `session_start()` 만 호출하는데, astraways가 세션 이름이나 저장 경로를 커스텀으로 쓰면 같은 쿠키를 못 읽어 빈 세션이 됩니다. 어떤 방식인지 확인하겠습니다.

soul3

```
bash
grep -rn "session_start\|session_name\|session_save_path\|session_set_cookie_params\|ini_set('se
```

이 결과로 두 가지가 드러납니다.

- astraways가 세션을 시작하는 공통 파일(엔진/부트스트랩) 위치
- `session_name` · `session_save_path` ·

Claude의 응답이 중단되었습니다.

```
No syntax errors detected in
/var/www/kr.astraways.net/public/billing/config.php
'CLIENT_KEY' => 'test_gck_docs_Ovk5rk1EwkEbP0W43n07xlzm',
'SECRET_KEY' => 'test_gsk_docs_OaPz8L5KdmQXkzRz3y47BMw6',
'pass' => '공백없는_새비밀번호',
--- 상품 화면 접속 ---
<title>이용권 구매 - 아스트라웨이즈</title>
<h1>아스트라웨이즈 이용권</h1>
  <div class="name">1개월 이용권</div>
더 보기
```

원인이 보입니다. astraways는 `api/_db.php` 에서 `session_set_cookie_params(...)` 로 세션 쿠키를 커스텀 설정한 뒤 `session_start()` 를 합니다. 표준 `auth.php` 는 그 설정 없이 기본 `session_start()` 만 하므로, 같은 쿠키를 못 읽어 빈 세션이 되고 미로그인으로 판정해 메인으로 튕깁니다. `_db.php` 의 세션 설정을 그대로 맞춰야 합니다. 해당 부분을 확인하겠습니다.

soul3

```
bash
```

```
sed -n '38,60p' /var/www/kr.astraways.net/public/api/_db.php
```

이 출력에서 `session_set_cookie_params` 의 인자(lifetime·path·domain·secure·samesite) 와 `session_name` 이 따로 지정돼 있는지를 봅니다. 그 값들을 표준 `auth.php` 의 세션 시작부에 동일하게 넣으면 같은 로그인 세션을 공유합니다.

세션 키도 한 가지 확인이 필요합니다. astraways가 로그인 시 회원 식별을 `$_SESSION['user_id']` 에 담는지, 다른 키 (예: `$_SESSION['uid']`, `$_SESSION['mid']`)를 쓰는지 보겠습니다.

soul3

```
bash
grep -rn "_SESSION\[\"" /var/www/kr.astraways.net/public/api/login*.php /var/www/kr.astraways.net/
```

두 결과(세션 쿠키 파라미터 + 로그인 시 저장하는 세션 키)를 보내주시면, 그에 맞춰 `auth.php` 의 세션 시작 방식과 키 이름을 수정한 v3를 만들어 배포하겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

```
mysql --socket=/home/mysql/mysql.sock -u payment_user -p'공백없는_새비밀번호' -e "USE payment_db; SELECT COUNT(*) orders FROM orders;"
echo "--- checkout 진입(미로그인) 응답 헤더 ---"
curl -sI "https://kr.astraways.net/billing/checkout.php?product=month" |
grep -iE "HTTP/|location"
--- DB 연결 테스트 (payment_user) ---
+-----+
| orders |
+-----+
더 보기
```

원인과 해결이 명확해졌습니다.

- 세션 키는 `$_SESSION['user_id']` 그대로 맞습니다(`login.php:25`)
- 문제는 세션 쿠키 파라미터입니다. astraways는 로그인 시 `domain·secure·samesite=Lax` 등을 지정한 쿠키로 세션을 시작하는데, 표준 `auth.php` 는 기본값으로 시작해서 같은 세션 쿠키를 못 읽습니다

표준 `auth.php` 가 astraways의 `_db.php` 와 동일한 방식으로 세션을 시작하도록 맞추겠습니다. 가장 깔끔한 건 astraways의 `startSession()` 을 그대로 재사용하는 것입니다. 이렇게 하면 세션 설정이 한 곳(`_db.php`)에서만 관리되고, 사이트마다 달라도 config로 흡수됩니다.

방식은 config에 "세션 부트스트랩 파일 경로"를 변수로 두고, `auth.php` 가 그 파일의 `startSession()` 을 불러 쓰는 구조입니다. 사이트마다 이 경로만 바꾸면 됩니다.

먼저 그 파일을 직접 require해도 부작용(자동 출력 등)이 없는지 확인하겠습니다.

soul3

```
bash

sed -n '1,20p' /var/www/kr.astraways.net/public/api/_db.php
echo "--- DB 접속 정보가 이 파일에 있는지 ---"
grep -nE "mysqli|PDO|new \\\?PDO|getDB|function \.[Dd]b" /var/www/kr.astraways.net/public/api/_
```

확인 포인트입니다.

- 파일 상단이 `<?php` 로 시작하고 함수 정의만 있는지(require 시 화면 출력이 없어야 함)
- `startSession()` 만 빌려 쓸 수 있는지, 아니면 DB 연결까지 묶여 있어 부분 사용이 어려운지

만약 `_db.php` 가 함수 정의 위주라 부작용 없이 require할 수 있으면, config 에 `'SESSION_BOOTSTRAP' => '/var/www/kr.astraways.net/public/api/_db.php'` 를 추가하고 `auth.php` 에서 `require_once` 로 불러 `startSession()` 을 호출하도록 v3를 만들겠습니다. 부작용이 있으면, 동일한 `session_set_cookie_params` 블록을 `auth.php` 에 직접 넣는 방식으로 가겠습니다.

위 두 결과를 보내주시면 v3를 만들어 배포하겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

```
'path' => '/',
    'domain' => $_SERVER['HTTP_HOST'] ?? "",
    'secure' => true,
    'httponly' => true,
    'samesite' => 'Lax'
]);
session_start();
}
static $tokenTried = false;
더 보기
```

`_db.php` 는 상단이 함수 정의 상수만 있어 require해도 화면 출력 부작용이 없습니다. `startSession()` 을 그대로 빌려 쓸 수 있습니다. 이 구조면 세션 설정이 astraways 한 곳에서만 관리되고, 다른 사이트에 적용할 때 config의 경로만 바꾸면 됩니다.

`auth.php` 를 수정하겠습니다. config에 세션 부트스트랩 경로를 추가하고, 그 파일이 있으면 `startSession()` 으로 세션을 시작하도록 합니다. 없으면 기존 기본 방식으로 동작해 다른 사이트와도 호환됩니다.

먼저 컨테이너에서 v3를 만들겠습니다.

Orchestrated direct server patching strategy for auth and config files >

v3 수정 완료입니다. `auth.php`가 astraways의 `startSession()` 을 빌려 동일한 세션 쿠키로 로그인 인식을 하게 했고, config에 `SESSION_BOOTSTRAP` 경로를 추가했습니다. 변경된 두 파일 (`auth.php` · `config.php`)만 패치하겠습니다.

soul3

```
bash
```

```
cat > ~/auth.php <<'PHP'
```

```
<?php
```

```
function boot_session(): void {
    if (session_status() === PHP_SESSION_ACTIVE) return;
    $bs = cfg()['SESSION_BOOTSTRAP'] ?? '';
    if ($bs && is_file($bs)) {
        require_once $bs;
        if (function_exists('startSession')) { startSession(); return; }
    }
    session_start();
}

function current_user_key(): string {
    boot_session();
    $uid = $_SESSION['user_id'] ?? null;
    if (!$uid) {
        $c = cfg();
        header('Location: ' . $c['SITE_URL'] . ($c['LOGIN_PATH'] ?? '/'));
        exit;
    }
    return (string)$uid;
}

function current_user_email(): string { return (string)($_SESSION['user_email'] ?? ''); }
function current_user_name(): string { return (string)($_SESSION['user_name'] ?? ''); }
```

```
PHP
```

```
sudo cp ~/auth.php /var/www/kr.astraways.net/public/billing/lib/auth.php
```

```
cat > ~/config.php <<'PHP'
```

```
<?php
```

```
return [
    'SITE_KEY' => 'astraways',
    'SITE_NAME' => '아스트라웨이즈',
    'SITE_URL' => 'https://kr.astraways.net',
    'PAY_BASE' => '/billing',
    'LOGIN_PATH' => '/',
    'SESSION_BOOTSTRAP' => '/var/www/kr.astraways.net/public/api/_db.php',
    'MID' => 'CHANGE_ME_MID',
    'CLIENT_KEY' => 'test_gck_docs_0vk5rk1EwkEbPOW43n07xlzm',
```

```
'SECRET_KEY' => 'test_gsk_docs_0aPz8L5KdmQXkzRz3y47BMw6',
'PRODUCTS' => [
  'month' => ['name' => '1개월 이용권', 'amount' => 9900, 'days' => 30],
  'year' => ['name' => '1년 이용권', 'amount' => 49900, 'days' => 365],
],
'DB' => [
  'name' => 'payment_db',
  'user' => 'payment_user',
  'pass' => '공백없는_새비밀번호',
  'socket' => '/home/mysql/mysql.sock',
],
];
PHP
sudo cp ~/config.php /var/www/kr.astraways.net/public/billing/config.php

sudo chown apache:apache /var/www/kr.astraways.net/public/billing/config.php /var/www/kr.astrawa
sudo chmod 644 /var/www/kr.astraways.net/public/billing/config.php /var/www/kr.astraways.net/pub

echo "---- 검증 ----"
php -l /var/www/kr.astraways.net/public/billing/lib/auth.php
php -r "require '/var/www/kr.astraways.net/public/api/_db.php'; var_dump(function_exists('startS
```

`No syntax errors` 와 `bool(true)` 가 나오면 세션 공유 준비 완료입니다.

그다음 astraways에 로그인한 브라우저에서 다시 접속해 주세요.

<https://kr.astraways.net/billing/checkout.php?product=month>

이제 메인으로 튕기지 않고 결제위젯(카드 결제수단 + 결제하기 버튼)이 떠야 합니다. 화면 결과나 Console 에러를 보내주시면 테스트 결제 승인으로 마무리하겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

```
function db() {
    static $pdo = null;
    if ($pdo === null) {
        $dsn = 'mysql:unix_socket=/home/mysql/mysql.sock;dbname=' .
        DB_NAME . ';charset=' . DB_CHARSET;
        try {
            $pdo = new PDO($dsn, DB_USER, DB_PASS, [
                PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
                PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
            ]);
        } catch (PDOException $e) {
            die('Database connection failed: ' . $e->getMessage());
        }
    }
    return $pdo;
}
```

더 보기

검증 통과입니다. `auth.php` 문법 정상이고, `startSession()`도 `bool(true)`로 인식됩니다. 세션 공유 준비가 끝났습니다.

이제 astraways에 **로그인한 브라우저**에서 접속해 주세요.

```
https://kr.astraways.net/billing/checkout.php?product=month
```

이번엔 메인으로 튕기지 않고 결제 화면이 떠야 합니다. 정상이면 상단에 "1개월 이용권 / 9,900원" 요약, 그 아래 카드·간편결제 위젯, 이용약관 동의, 맨 아래 파란 "결제하기" 버튼이 보입니다.

확인 결과를 알려주세요.

- 결제위젯이 정상으로 떴는지
- 여전히 메인으로 튕기면 → astraways에 실제 로그인된 상태가 맞는지(다른 탭에서 마이페이지 등 접근되는지)
- 위젯이 빈 화면이거나 깨지면 → F12 → Console 탭의 빨간 에러 메시지

정상으로 뜨면 결제하기를 눌러 테스트 결제까지 진행하시면 됩니다. 테스트 카드로 승인하면 실제 청구 없이 `success.php`로 이동하고, DB `orders`가 DONE, `entitlements`에 30일 이용권이 기록됩니다. 결과를 보내주시면 대단히 감사하겠습니다.

페이지가 작동하지 않습니다.
현재 kr.astraways.net에서 요청을 처리할 수 없습니다.
HTTP ERROR 500

함수명 충돌을 진단하고 체계적 수정안을 수립했다 >

500의 원인이 보입니다. astraways `_db.php`에도 `function db()` (PDO 반환)가 있고, `lib/db.php`에도 `function db()` (mysqli 반환)가 있습니다. 세션 공유로 `_db.php`를 require하는 순간 같은 이름 함수가 재선언돼 `Cannot redeclare db()` 치명 오류가 납니다. 결제 DB 함수명을 충돌 없는 `pay_db()`로 바꾸겠습니다.

데이터베이스 함수 변경 후 전체 재배포 전략을 수립했습니다 >

`db()` 호출이 모두 `pay_db()`로 바뀌었고 충돌이 사라졌습니다. 변경분을 v3로 묶어 재배포하겠습니다. `config.php`는 패키지에 없으므로 기존 값이 그대로 유지됩니다.

명령 실행함, 파일 읽음 >

v3를 받아 재배포합니다.

holymind

```
bash
```

```
cd ~/다운로드
scp toss-standard_v3.tar.gz soul3:~/
```

soul3

```
bash

cd ~ && tar xzf toss-standard_v3.tar.gz
sudo cp -r ~/toss-standard/* /var/www/kr.astraways.net/public/billing/
sudo chown -R apache:apache /var/www/kr.astraways.net/public/billing
sudo find /var/www/kr.astraways.net/public/billing -type d -exec chmod 755 {} \;
sudo find /var/www/kr.astraways.net/public/billing -type f -exec chmod 644 {} \;

echo "--- config 보존 확인 (gck/gsk, SESSION_BOOTSTRAP) ---"
grep -E "CLIENT_KEY|SECRET_KEY|SESSION_BOOTSTRAP" /var/www/kr.astraways.net/public/billing/confi

echo "--- 최근 PHP 에러로그 ---"
sudo tail -n 15 /var/log/php-fpm/www-error.log 2>/dev/null || sudo tail -n 15 /var/log/php-fpm/e
```

`config.php` 에 `gck/gsk` 키와 `SESSION_BOOTSTRAP` 이 그대로 보이면 보존 정상입니다.

배포 후 astraways에 로그인한 브라우저에서 다시 접속해 주세요.

```
https://kr.astraways.net/billing/checkout.php?product=month
```

이번엔 500 없이 결제위젯이 떠야 합니다. 화면 결과와 위 에러로그를 보내주시면 테스트 결제 승인으로 마무리하겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

```
Toss standard v3.tar
GZ
```

```
No syntax errors detected in
/var/www/kr.astraways.net/public/billing/lib/auth.php
bool(true)
[soul3@localhost ~]$ cd ~ && tar xzf toss-standard_v3.tar.gz
sudo cp -r ~/toss-standard/* /var/www/kr.astraways.net/public/billing/
sudo chown -R apache:apache /var/www/kr.astraways.net/public/billing
sudo find /var/www/kr.astraways.net/public/billing -type d -exec chmod
755 {} \;
sudo find /var/www/kr.astraways.net/public/billing -type f -exec chmod
더 보기
```

배포·config 보존 정상입니다. 에러로그가 진단을 확인해 줍니다. `Cannot redeclare db()` 로그는 11:13(수정 전) 기록이고, 방금 v3로 `pay_db()` 로 바꿔 그 충돌은 해소됐습니다. 이제 로그인 상태로 재접속하면 해야 합니다.

```
https://kr.astraways.net/billing/checkout.php?product=month
```

접속해 보시고, 혹시 또 500이면 방금 시각의 새 로그만 확인하겠습니다.

soul3

```
bash
sudo tail -n 5 /var/log/php-fpm/www-error.log
```

(참고로 로그에 보인 `klifestyle_user` Access denied는 별개 사이트(klifestyle.net) 문제입니다. 지금 작업과 무관하지만, klifestyle 결제 차례가 오면 그 DB 계정도 점검 대상입니다.)

확인 포인트입니다.

- 정상: "1개월 이용권 / 9,900원" + 카드 위젯 + 결제하기 버튼
- 메인으로 튕김: astraways 실제 로그인 상태가 맞는지(다른 탭에서 마이페이지 접근되는지)
- 빈 화면/깨짐: F12 → Console 빨간 에러

정상으로 뜨면 결제하기를 눌러 테스트 카드로 승인까지 진행해 주세요. 성공 시 `success.php` 로 이동하고 DB에 주문 DONE·이용권 30일이 기록됩니다. 결과를 보내주시면 대단히 감사하겠습니다.

<https://kr.astraways.net/>으로 이동한다

세션 공유 문제를 진단하기 위한 임시 스크립트 배포 계획 수립 >

500은 사라졌고, 이제는 세션을 못 읽어 미로그인으로 판정되어 메인으로 보냅니다. astraways 로그인 세션 쿠키 기반인지 토큰 기반인지부터 확정해야 합니다. `_db.php` 에 `tryTokenLogin()` 이 있는 걸 보면 토큰 로그인 경로가 있는데, 주소창에 URL을 직접 치는 네비게이션에는 Authorization 헤더가 안 붙어 토큰 인증이 안 됩니다. 그러면 세션 쿠키가 없는 한 미로그인이 됩니다.

세션 상태를 직접 보기 위해 임시 진단 파일을 올리겠습니다.

soul3

```
bash
cat > ~/whoami.php <<'PHP'
<?php
```

```
require __DIR__ . '/lib/db.php';
require __DIR__ . '/lib/auth.php';
boot_session();
header('Content-Type: text/plain; charset=utf-8');
echo "SESSION = "; var_export($_SESSION); echo "\n";
echo "COOKIE = "; var_export($_COOKIE); echo "\n";
echo "SESSION_NAME = " . session_name() . "\n";
echo "SAVE_PATH = " . session_save_path() . "\n";
PHP
sudo cp ~/whoami.php /var/www/kr.astraways.net/public/billing/whoami.php
sudo chown apache:apache /var/www/kr.astraways.net/public/billing/whoami.php
sudo chmod 644 /var/www/kr.astraways.net/public/billing/whoami.php
```

이제 astraways에 **로그인한 그 브라우저**에서 접속해 주세요.

<https://kr.astraways.net/billing/whoami.php>

화면에 나오는 **SESSION** · **COOKIE** 내용을 그대로 보내주세요. 이걸로 같습니다.

- **SESSION = ['user_id' => ...]** 가 있으면 → 세션은 공유되는데 checkout 가드가 문제
→ 바로 수정
- **SESSION** 이 비어있고 **COOKIE** 에 세션 쿠키(PHPSESSID 등)도 없으면 → astraways 웹 로그인 토큰 기반 → 결제 진입을 토큰으로 인증하도록 설계 변경 필요

추가로, astraways 웹에서 로그인하면 마이페이지 등이 새로그침 후에도 유지되는지(세션) 아니면 JS가 토큰을 들고 호출하는지도 알려주시면 방향이 더 빨리 잡힙니다. 결과를 보내주시면 대단히 감사하겠습니다.

```
)
#1 {main}
  thrown in /var/www/klifestyle.net/public_html/barefoot_detail.php on
  line 2
[19-Jun-2026 04:49:11 UTC] PHP Fatal error: Uncaught PDOException:
SQLSTATE[HY000] [1045] Access denied for user
'klifestyle_user'@'localhost' (using password: YES) in
/var/www/klifestyle.net/public_html/barefoot_topic.php:2
Stack trace:
더 보기
```

진단 파일이 올라갔습니다. 이제 astraways에 **로그인한 브라우저**에서 아래 주소로 접속해 주세요.

<https://kr.astraways.net/billing/whoami.php>

화면에 텍스트로 `SESSION = ...`, `COOKIE = ...`, `SESSION_NAME = ...`, `SAVE_PATH = ...`가 출력됩니다. 그 내용을 그대로 복사해서 보내주세요.

이 출력으로 로그인 방식이 갈립니다.

- `SESSION` 에 `'user_id'` 가 있으면 → 세션 공유는 되는 것이고, checkout 가드만 손보면 됩니다
- `SESSION` 이 비어 있고 `COOKIE` 에 `PHPSESSID` 만 있으면 → 세션은 시작됐지만 로그인 정보가 안 담긴 것이라, astraways 로그인 흐름을 확인합니다
- `COOKIE` 에 세션 쿠키 자체가 없으면 → astraways 웹 로그인이 JS 토큰(Authorization 헤더) 기반이라, 결제 진입을 토큰 인증으로 바뀌어야 합니다

세 줄 출력 결과를 보내주시면 그에 맞춰 바로 수정하겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

```
SESSION = array (
)
COOKIE = array (
  'PHPSESSID' => 'gfh43ls75chb7suc2bhf0dhuhp',
)
SESSION_NAME = PHPSESSID
SAVE_PATH = /var/lib/php/session
```

세션은 정상으로 시작되는데(`PHPSESSID` 쿠키 존재) 내용이 비어 있습니다. astraways 웹 로그인이 PHP 세션에 `user_id` 를 담지 않는다는 뜻입니다. `_db.php` 의 `tryTokenLogin()` 이 단서입니다. astraways는 로그인 토큰(JWT 등)을 들고 API를 호출할 때 토큰을 검증해 사용자를 인식하고, 일반 페이지 네비게이션(주소창 입력)에는 그 토큰이 안 실려 세션이 빈 상태가 됩니다.

즉 astraways는 **세션 로그인이 아니라 토큰 로그인**입니다. 결제 진입도 같은 토큰 방식으로 사용자를 인식해야 합니다. 먼저 `tryTokenLogin()` 이 토큰을 어디서 읽고(쿠키냐 헤더냐) 무엇을 세션/변수에 채우는지 확인하겠습니다.

soul3

```
bash
grep -rn "function tryTokenLogin" /var/www/kr.astraways.net/public/api/ | grep -v "\.bak"
echo "--- 정의 본문 ---"
awk '/function tryTokenLogin/,/^}/' /var/www/kr.astraways.net/public/api/_db.php
echo "--- 토큰 쿠키 이름 단서 ---"
grep -rn "setcookie\|$_COOKIE\[\|bearerToken\|JWT\|jwt\|token" /var/www/kr.astraways.net/public
```

이 결과로 두 가지가 드러납니다.

- 토큰을 어디서 읽는지: `Authorization: Bearer` 헤더만 보는지, 아니면 쿠키 (예: `auth_token`)에도 저장하는지
- 토큰 검증 후 `user_id` 를 `$_SESSION` 에 넣는지, 아니면 전역 변수/반환값으로만 주는지

만약 토큰이 쿠키에 저장된다면 결제 페이지에서도 그 쿠키로 사용자를 인식할 수 있어 깔끔합니다. 헤더에만 있다면, 상품 페이지에서 결제로 넘어갈 때 토큰을 함께 넘기는 방식으로 설계를 맞추겠습니다.

세 결과를 보내주시면 그에 맞게 `auth.php` 의 사용자 인식 로직을 수정하겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

```
$stmt->execute([$token]);
$row = $stmt->fetch();
if (!$row) return;
if (strtotime($row['expires_at']) < time()) {
    db()->prepare("DELETE FROM auth_tokens WHERE token=?")-
    >execute([$token]);
    return;
}
db()->prepare("UPDATE auth_tokens SET last_used=NOW() WHERE
더 보기
```

구조가 명확해졌습니다. astraways는 `auth_tokens` 테이블 기반 토큰 로그인입니다. 로그인 시 토큰을 발급하고, `bearerToken()` 이 그 토큰을 읽어 검증한 뒤 `$_SESSION['user_id']` 를 채웁니다. 문제는 `bearerToken()` 이 토큰을 어디서 읽느냐입니다.

방금 출력 마지막에 `$_COOKIE['remember_token']` 이 보입니다. 토큰이 쿠키에도 저장된다면 결제 페이지에서도 자동 인식이 가능합니다. `bearerToken()` 본문을 정확히 보겠습니다.

soul3

```
bash
awk 'NR>=60 && NR<=70' /var/www/kr.astraways.net/public/api/_db.php
echo "--- remember_token 처리부 ---"
awk 'NR>=99 && NR<=120' /var/www/kr.astraways.net/public/api/_db.php
```

이 두 블록으로 확정됩니다.

- `bearerToken()` 이 `Authorization` 헤더만 읽는지, `$_COOKIE['auth_token']` 같은 쿠키도 보는지
- `remember_token` 쿠키가 있으면 자동 로그인(세션 복원)이 되는지

만약 브라우저 로그인 시 `remember_token` 쿠키가 심어지고 그게 세션을 복원한다면, 결제 페이지에서 `_db.php` 를 require하는 것만으로 `$_SESSION['user_id']` 가 채워져야 합니다. 그런데

whoami에서 세션이 비어 있었으니, `remember_token` 쿠키가 안 심겼거나 `tryTokenLogin()` 이 헤더만 보는 경우입니다.

두 블록 결과를 보내주시면, 결제 진입에서 사용자를 인식하는 방식(쿠키 토큰 활용 또는 상품 → 결제로 토큰 전달)을 확정해 `auth.php` 를 수정하겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

```
/var/www/kr.astraways.net/public/api/_db.php:101: if
(empty($_COOKIE['remember_token'])) return;
/var/www/kr.astraways.net/public/api/_db.php:102: $token =
$_COOKIE['remember_token'];
/var/www/kr.astraways.net/public/api/_db.php:103: if
(!preg_match('/^[a-f0-9]{64}$/', $token)) return;
[soul3@localhost ~]$ awk 'NR>=60 && NR<=70'
/var/www/kr.astraways.net/public/api/_db.php
echo "--- remember token 처리부 ---"
더 보기
```

완전히 확정됐습니다. astraways에는 두 가지 로그인 경로가 있습니다.

- `tryTokenLogin()` — `Authorization: Bearer` 헤더의 토큰만 읽습니다. 주소창 네비게이션에는 헤더가 안 붙으니 결제 페이지에선 무용합니다.
- `tryRememberLogin()` — `$_COOKIE['remember_token']` (64자 hex)을 읽어 `users` 테이블에서 검증하고 `$_SESSION['user_id']` 를 채웁니다. 이건 브라우저 쿠키라 결제 페이지에서도 작동합니다.

핵심은 `startSession()` 이 `tryTokenLogin()` 만 자동 호출하고 `tryRememberLogin()` 은 부르지 않는다는 점입니다. 그래서 세션이 비었던 것입니다. 결제 진입에서 `tryRememberLogin()` 을 마저 호출하면 됩니다.

다만 전제가 하나 있습니다. astraways 웹 로그인이 `remember_token` 쿠키를 실제로 심는지(=브라우저에 그 쿠키가 있는지)입니다. 그게 없으면 이 경로도 비어 있습니다. 먼저 확인하겠습니다.

진단 파일을 갱신합니다.

soul3

```
bash
cat > ~/whoami.php <<'PHP'
<?php
require __DIR__ . '/lib/db.php';
$bs = cfg()['SESSION_BOOTSTRAP'];
require_once $bs;
startSession();
if (function_exists('tryRememberLogin')) tryRememberLogin();
```

```
header('Content-Type: text/plain; charset=utf-8');
echo "after tryRememberLogin, SESSION = "; var_export($_SESSION); echo "\n";
echo "remember_token cookie? = " . (isset($_COOKIE['remember_token']) ? 'YES' : 'NO') . "\n";
echo "COOKIE keys = " . implode(', ', array_keys($_COOKIE)) . "\n";
PHP
sudo cp ~/whoami.php /var/www/kr.astraways.net/public/billing/whoami.php
sudo chown apache:apache /var/www/kr.astraways.net/public/billing/whoami.php
sudo chmod 644 /var/www/kr.astraways.net/public/billing/whoami.php
```

astraways에 로그인한 브라우저에서 다시 접속해 주세요.

<https://kr.astraways.net/billing/whoami.php>

세 줄 출력을 보내주세요. 갈림길은 이렇습니다.

- `SESSION = ['user_id' => 숫자]` 가 나오면 → `tryRememberLogin` 호출로 해결 됨. `auth.php` 에 그 한 줄만 추가하면 끝입니다.
- `remember_token cookie? = NO` 면 → 브라우저 로그인이 자동로그인 쿠키를 안 남기는 설정이라, astraways 로그인 시 `remember_token` 을 항상 심도록 `login.php` 를 보완하거나, 결제 직전에 토큰을 발급·전달하는 방식으로 가야 합니다. `COOKIE keys` 목록을 보면 어떤 쿠키가 실제로 있는지 판단할 수 있습니다.

결과를 보내주시면 그에 맞춰 마무리하겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

```
after tryRememberLogin, SESSION = array (
)
remember_token cookie? = NO
COOKIE keys = PHPSESSID 여기내용을 참조해봐.
https://docs.tosspayments.com/guides/billing/integration
```

Diagnosed astraways token authentication gap, explored cookie-based solutions >

막힌 지점이 분명해졌습니다. astraways는 `auth_tokens` /localStorage 토큰 기반이고, 브라우저에는 `PHPSESSID` 만 있을 뿐 로그인 정보가 담긴 세션도 `remember_token` 쿠키도 없습니다. 그래서 PHP 결제 페이지가 누구인지 알 수 없어 메인으로 튕깁니다. 빌링 가이드를 먼저 확인하겠습니다.

빌링 정책 재확인 후 토큰 브릿지 방식 제안했다 >

빌링 가이드를 확인했습니다. 자동결제는 정기 구독형 서비스에만 사용할 수 있고 비정기 결제 서비스에는 정책적으로 제한된다고 명시돼 있습니다. 우리 모델은 만료 후 사용자가 직접 재결제하는 단

건이라 빌링은 부적합하고, 지금처럼 결제위젯 단건이 맞습니다. 빌링 가이드는 customerKey·빌링 키 흐름이라 지금 막힌 인증 문제와는 별개입니다.

진단 결론은 이렇습니다. astraways 로그인은 `auth_tokens` 토큰 기반(JS가 localStorage 토큰을 Bearer 헤더로 API 호출)이고, 브라우저에는 로그인 정보가 담긴 세션도 `remember_token` 쿠키도 없습니다. 그래서 주소창으로 이동하는 PHP 결제 페이지가 사용자를 식별하지 못합니다. 이건 표준 PHP 페이지가 토큰 기반 SPA와 안 맞아서 생기는 구조적 문제입니다.

해결은 두 갈래입니다.

A안(추천, 기존 로그인 무수정): 결제 진입점 `enter.php` 를 추가합니다. astraways 화면의 "이용권 구매" 버튼이 localStorage 토큰을 `enter.php` 로 보내면, 서버가 `auth_tokens` 로 검증해 사용자를 확인하고 결제 세션을 세팅한 뒤 결제 페이지로 보냅니다. 기존 로그인 코드를 건드리지 않고, SPA에는 버튼 한 곳만 손댁니다. 사이트마다 토큰 검증부만 바꾸면 되어 표준화에도 맞습니다.

B안: astraways 로그인이 `remember_token` 쿠키를 항상 심도록 `login.php` 를 보완합니다. 결제 페이지가 그 쿠키로 사용자를 인식합니다. 가장 적은 코드지만 운영 중인 로그인 기능을 수정하는 리스크가 있습니다.

A안으로 진행하려면 astraways 웹이 로그인 후 토큰을 어디에 저장하고 어떻게 호출하는지 알아야 합니다. 확인하겠습니다.

soul3

```
bash
echo "--- public 구조 ---"
ls /var/www/kr.astraways.net/public/ | head -40
echo "--- 토큰 저장/호출 방식 ---"
grep -rn "localStorage\|sessionStorage\|Bearer\|Authorization" /var/www/kr.astraways.net/public/
```

결과를 보내주시면 A안 진행 여부를 정하고, `enter.php` 와 SPA 버튼 연동을 만들겠습니다. 어느 안으로 가실지도 함께 알려주시면 그대로 진행하겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

toss에서 제시한대로 해야 통과가 된다

토스 표준 결제 흐름으로 방향 전환하고 세션 로그인 구현 검토 >

맞습니다. 토스 심사관이 결제까지 도달하려면 토스가 제시하는 표준 경로(로그인 → 상품 선택 → 결제위젯)를 따라야 합니다. 토큰 브릿지 같은 커스텀 방식은 폐기하고, 결제 영역에 **표준 세션 로그인**을 두겠습니다. 토스 결제경로 가이드도 로그인/회원가입 경로 캡처와 심사관용 테스트 계정을 요구하므로, 세션 기반 로그인이 심사에 가장 적합합니다.

방향은 이렇습니다. 결제 영역(`/billing/auth/`)에 자체 세션 로그인·회원가입을 두되, astraways의 기존 회원 테이블(`users`)을 그대로 조회합니다. 같은 이메일·비밀번호로 결제 영역에

서 세션 로그인하면 같은 `user_id`가 세션에 담겨, 결제·이용권이 기존 회원과 정확히 연결됩니다. 심사관은 테스트 계정으로 이 로그인을 거쳐 상품·결제까지 도달합니다.

이걸 만들려면 astraways의 로그인 검증 방식과 회원 테이블 구조를 알아야 합니다. 확인하겠습니다.

soul3

```

bash

echo "--- 로그인 검증 방식 (비밀번호 해시) ---"
sed -n '1,35p' /var/www/kr.astraways.net/public/api/login.php

echo "--- users 테이블 구조 ---"
sudo mysql --socket=/home/mysql/mysql.sock -e "USE kr_astraways; DESCRIBE users;"

echo "--- 임시 진단 파일 제거 ---"
sudo rm -f /var/www/kr.astraways.net/public/billing/whoami.php

```

확인 포인트입니다.

- `login.php`가 비밀번호를 `password_verify()`로 검증하는지, 다른 해시(sha256 등)인지 → 결제 로그인도 동일 방식이어야 합니다
- `users` 테이블의 이메일·비밀번호·활성 컬럼명(`email`, `password`, `is_active` 등)

두 결과를 보내주시면, astraways 회원으로 세션 로그인하는 표준 `auth/login.php`·`auth/register.html`을 만들어 결제 흐름을 토스 표준 경로로 완성하겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

```

$email = trim($input['email'] ?? ''); $password = $input['password'] ?? ''; if (!$email || !$password) { jsonResponse(['error' => '이메일과 비밀번호를 입력하...

```

PASTED

확정됐습니다. astraways 로그인은 `password_verify()`로 검증하고, `users` 테이블의 `email`·`password_hash`·`is_active`로 인증합니다. 결제 영역에 같은 방식의 세션 로그인을 두면 됩니다. 회원 식별은 `users.id`이고, 결제·이용권은 이 `id`와 연결됩니다.

`users` 테이블에 `membership_type`·`subscription_end` 컬럼이 이미 있는 점이 중요합니다. 결제 성공 시 우리 `entitlements` 기록과 더불어 이 두 컬럼도 갱신하면(`paid` + 만료일) astraways 본체가 회원 등급을 그대로 활용할 수 있습니다. 이걸 결제 완료 로직에 반영하겠습니다.

이제 결제 영역 세션 로그인을 만들겠습니다. 컨테이너에서 `auth/login.php`(astraways 회원으로 로그인)와 `auth/login.html`(폼)을 추가합니다.

I notice the user prompt is empty, so I cannot determine the primary language for the summary. H... >

로그인 처리는 `boot_session()` 으로 astraways 세션을 공유하고, `_db.php` 의 `db()` 로 `kr_astraways.users` 를 조회해 `password_verify()` 로 검증합니다. 이제 로그인 폼과 config의 로그인 경로를 결제 자체 로그인으로 맞추겠습니다.

I notice the user prompt is empty, so I cannot determine the primary language for the summary. H... >
v4를 배포합니다. config의 로그인 경로(`LOGIN_PATH`)를 결제 자체 로그인으로 바꿔야 하므로 config.php도 함께 갱신합니다.

holymind

```
bash
cd ~/다운로드
scp toss-standard_v4.tar.gz soul3:~/
```

soul3

```
bash
cd ~ && tar xzf toss-standard_v4.tar.gz
sudo cp -r ~/toss-standard/* /var/www/kr.astraways.net/public/billing/

cat > ~/config.php <<'PHP'
<?php
return [
    'SITE_KEY' => 'astraways',
    'SITE_NAME' => '아스트라웨이즈',
    'SITE_URL' => 'https://kr.astraways.net',
    'PAY_BASE' => '/billing',
    'LOGIN_PATH' => '/billing/auth/login.html',
    'SESSION_BOOTSTRAP' => '/var/www/kr.astraways.net/public/api/_db.php',
    'MID' => 'CHANGE_ME_MID',
    'CLIENT_KEY' => 'test_gck_docs_0vk5rk1EwkEbPOW43n07xlzm',
    'SECRET_KEY' => 'test_gsk_docs_0aPz8L5KdmQXkzRz3y47BMw6',
    'PRODUCTS' => [
        'month' => ['name' => '1개월 이용권', 'amount' => 9900, 'days' => 30],
        'year' => ['name' => '1년 이용권', 'amount' => 49900, 'days' => 365],
    ],
    'DB' => [
        'name' => 'payment_db',
        'user' => 'payment_user',
        'pass' => '공백없는_새비밀번호',
        'socket' => '/home/mysql/mysql.sock',
    ],
];
PHP
```

```

sudo cp ~/config.php /var/www/kr.astraways.net/public/billing/config.php

sudo chown -R apache:apache /var/www/kr.astraways.net/public/billing
sudo find /var/www/kr.astraways.net/public/billing -type d -exec chmod 755 {} \;
sudo find /var/www/kr.astraways.net/public/billing -type f -exec chmod 644 {} \;

echo "--- 문법 확인 ---"
php -l /var/www/kr.astraways.net/public/billing/auth/login.php
ls /var/www/kr.astraways.net/public/billing/auth/

```

No syntax errors 와 login.html login.php 가 보이면 배포 정상입니다.

이제 브라우저로 테스트합니다(astraways 토큰 로그인과 별개의 결제 자체 로그인이라, 로그인 안 된 일반 브라우저로 시작하셔도 됩니다).

https://kr.astraways.net/billing/membership.php

상품에서 결제하기를 누르면 로그인 페이지로 이동합니다. astraways 테스트 계정의 이메일·비밀번호로 로그인하면 다시 상품으로 돌아오고, 결제하기를 누르면 결제위젯이 뜹니다.

확인해 주실 것은 두 가지입니다.

- 로그인 후 checkout.php 에서 결제위젯(카드 결제수단 + 결제하기 버튼)이 정상으로 뜨는지
- 로그인 실패 시 메시지가 보이는지(잘못된 비밀번호 입력 테스트)

결과를 보내주시면 테스트 결제 승인까지 마무리하겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

Toss standard v4.tar
GZ

```

| birth_city | varchar(50) |
YES || NULL || | birth_country | char(2) | YES || KR |
|| gender | enum ('M','F','O')

```

PASTED

배포 정상입니다. login.php 문법 통과, auth/ 에 두 파일 다 있습니다. 이제 브라우저로 결제 흐름 전체를 테스트하시면 됩니다.

https://kr.astraways.net/billing/membership.php

순서는 이렇습니다.

1. 상품 화면(월 9900·연 49900)에서 결제하기 클릭
2. 로그인 페이지로 이동 → astraways 테스트 계정 이메일·비밀번호 입력
3. 로그인 성공 시 상품으로 복귀 → 다시 결제하기
4. 결제위젯(카드 결제수단 + 결제하기 버튼)이 뜨면 정상

확인 결과를 알려주세요.

- 로그인 후 결제위젯이 정상으로 뜨는지
- 잘못된 비밀번호 입력 시 오류 메시지가 뜨는지(로그인 검증 동작 확인)
- 문제가 있으면 F12 → Console 빨간 에러 또는 아래 서버 로그

soul3

```
bash
sudo tail -n 5 /var/log/php-fpm/www-error.log
```

결제위젯까지 정상으로 뜨면, 테스트 카드로 실제 승인을 진행해 `success.php` 이동과 DB 기록 (orders DONE·entitlements 30일)을 확인하겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

잘 나타나고 있어

로그인부터 결제위젯까지 정상 동작 확인됐습니다. 핵심 흐름이 살아났습니다.

이제 테스트 결제를 끝까지 진행해 승인·DB 기록까지 확인하겠습니다. 결제위젯에서 카드 결제수단을 선택하고 결제하기를 누른 뒤, 토스

Claude의 응답이 중단되었습니다.

```
+-----+-----+
--- 임시 진단 파일 제거 ---
[soul3@localhost ~]$ cd ~ && tar xzf toss-standard_v4.tar.gz
sudo cp -r ~/toss-standard/* /var/www/kr.astraways.net/public/billing/
cat > ~/config.php <<'PHP'
<?php
return [
  'SITE_KEY' => 'astraways',
  'SITE_NAME' => '아스트라웨이즈',
];
더 보기
```

에러로그의 `Cannot redeclare db()` 는 모두 11:13(v3 이전) 기록이고, 그 뒤로 새 에러가 없습니다. 결제위젯도 정상으로 떴으니 현재 상태는 깨끗합니다.

이제 테스트 결제를 끝까지 진행하겠습니다. 결제위젯에서 카드 결제수단을 고르고 결제하기를 누른 뒤, 토스 테스트 결제창에서 아무 카드 정보나 입력해 승인하시면 됩니다(테스트 키라 실제 청구 없음). 성공하면 `success.php` 로 이동하고 "결제를 완료했어요"와 이용권 만료일이 보입니다.

결제 시도 후 서버에서 기록을 확인하겠습니다.

soul3

```
bash
```

```
mysql --socket=/home/mysql/mysql.sock -u payment_user -p'공백없는_새비밀번호' payment_db -e "
SELECT order_id, product_code, amount, status, method, approved_at FROM orders ORDER BY created_
SELECT user_key, product_code, started_at, expires_at FROM entitlements ORDER BY id DESC LIMIT 3
```

확인 포인트입니다.

- `orders` 에 방금 주문이 `status=DONE`, `method` (카드 등), `approved_at` 채워짐
- `entitlements` 에 해당 회원의 30일(또는 365일) 이용권이 `expires_at` 까지 기록됨

여기까지 나오면 결제 코어가 완성된 것입니다. 결제 화면에서 성공 페이지로 이동했는지, 그리고 위 쿼리 결과를 보내주시면 다음 단계(필수 페이지 일괄 추가 → 심사 준비)로 넘어가겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

토스결제창은 통합결제창인거 알지?

네, 알고 있습니다. 지금 붙인 게 토스페이먼트스 통합결제창(결제위젯)입니다.

`checkout.php` 에서 `widgets.renderPaymentMethods()` 로 띄운 화면이 통합결제창이라, 카드뿐 아니라 토스페이·네이버페이·카카오페이·계좌이체·가상계좌 등 여러 결제수단이 한 위젯 안에 모두 노출됩니다. 결제수단마다 따로 연동할 필요가 없고, 어떤 수단으로 결제하든 성공 시 동일하게 `successUrl` 로 `paymentKey` · `orderId` · `amount` 가 돌아와 `payment_success.php` 의 confirm API 한 곳에서 승인됩니다. 결제수단 추가·삭제·UI 변경은 나중에 토스 상점관리자(어드민)에서 코드 수정 없이 가능합니다.

즉 지금 구조가 단건 통합결제창이고, 카드로 테스트하든 간편결제로 테스트하든 승인·이용권 부여 흐름은 동일합니다. 다만 테스트 환경에서 간편결제는 토스페이·네이버페이만 모든 테스트 키로 열리고, 카카오페이는 계약 후 발급되는 내 테스트 키라야 열리니, 지금 공용 테스트 키 단계에서는 카드 또는 토스페이로 테스트하시면 됩니다.

앞서 드린 결제 결과 확인 쿼리(`orders`·`entitlements`) 결과를 보내주시면 다음 단계로 넘어가겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

```
return [
  'SITE_KEY' => 'astraways',
```

```
'SITE_NAME' => '아스트라웨이즈',
'SITE_URL' => 'https://kr.astraways.net',
'PAY_BASE' => '/billing',
'LOGIN_PATH' => '/billing/auth/login.html',
'SESSION_BOOTSTRAP' =>
'/var/www/kr.astraways.net/public/api/_db.php',
더 보기
```

쿼리 결과가 비어 있습니다. orders·entitlements 둘 다 행이 없으니, 아직 실제 결제 승인까지는 진행되지 않은 상태입니다. 결제위젯 화면까지만 확인되고, 결제하기 → 테스트 카드 승인 단계는 아직 안 거치셨거나, 거쳤는데 승인이 안 떨어진 것입니다.

먼저 결제위젯에서 실제로 결제를 한 번 끝까지 진행해 주세요.

1. 결제위젯에서 카드(또는 토스페이) 선택
2. 이용약관 동의 체크
3. 결제하기 버튼
4. 토스 테스트 결제창에서 카드 정보 입력 후 승인

이때 두 가지 중 하나가 나옵니다.

- 성공 → `success.php` ("결제를 완료했어요")로 이동
- 실패 → `fail.php` 로 이동하며 메시지 표시

결제를 시도하면 성공이든 실패든 `orders` 에는 최소한 READY 행이 남습니다(checkout 진입 시 선거록). 지금 orders가 완전히 비어 있다는 건, 결제하기를 눌러 `checkout.php` 까지 들어가지 않았거나 로그인 세션이 빠진 채 튕겼을 가능성도 있습니다.

결제를 시도하신 뒤, 결과 화면(성공/실패 페이지나 에러)과 함께 아래를 확인해 주세요.

soul3

```
bash
mysql --socket=/home/mysql/mysql.sock -u payment_user -p'공백없는_새비밀번호' payment_db -e "SELECT * FROM orders"
echo "--- 결제 직후 에러로그 ---"
sudo tail -n 8 /var/log/php-fpm/www-error.log
```

`orders` 에 행이 생겼는지, 상태가 READY인지 DONE인지, 그리고 새 에러가 있는지 보내주시면 막힌 지점을 정확히 짚겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

```
r.astraways.net/public/api/_db.php on line 12
[19-Jun-2026 11:13:22 UTC] PHP Fatal error: Cannot redeclare db()
(previously declared in
```

```

/var/www/kr.astraways.net/public/billing/lib/db.php:14) in
/var/www/kr.astraways.net/public/api/_db.php on line 12
[19-Jun-2026 11:13:23 UTC] PHP Fatal error: Cannot redeclare db()
(previously declared in
/var/www/kr.astraways.net/public/billing/lib/db.php:14) in
더 보기

```

orders가 여전히 비어 있고 새 에러도 없습니다. 즉 `checkout.php`까지는 들어가서 위젯은 떴지만, 결제하기를 눌러 토스 결제창에서 카드 승인까지는 아직 안 거치신 상태입니다. checkout 진입 시 READY 행이 남아야 하는데 그것도 없는 걸 보면, 위젯을 본 화면은 캐시이거나 결제 버튼 직전에서 멈춘 것입니다.

테스트 결제를 실제로 한 번 끝까지 진행해 주세요.

1. `https://kr.astraways.net/billing/membership.php`에서 1개월 이용권 결제하기
2. (로그인 안 되어 있으면) 로그인 후 다시 결제하기 → 결제위젯 화면
3. 카드(또는 토스페이) 선택 + 약관 동의 체크
4. 결제하기 버튼 클릭 → 토스 테스트 결제창
5. 카드 정보 입력 후 승인

이때 결제하기를 누르면 `checkout.php`가 READY 주문을 기록하고 토스 결제창이 뜹니다. 승인 하면 `success.php`로 이동합니다.

만약 결제하기를 눌렀는데 토스 결제창이 안 뜨거나 멈추면, 그 순간 브라우저 F12 → Console 탭의 빨간 에러를 알려주세요. 위젯 키 문제(INVALID_API_KEY 등)나 customerKey 형식 문제면 거기서 잡힙니다.

결제 시도 직후 다시 확인하겠습니다.

soul3

```

bash
mysql --socket=/home/mysql/mysql.sock -u payment_user -p'공백없는_새비밀번호' payment_db -e "SELE

```

결제하기를 누른 뒤 이 쿼리에 READY나 DONE 행이 생기는지, 그리고 결제창이 떴는지/Console 에러가 있었는지 알려주세요. 결과를 보내주시면 대단히 감사하겠습니다.

로그인하라고 하는데, 아이디와 비번은?

결제 테스트용 로그인은 astraways 회원 계정(`kr_astraways.users` 테이블에 있는 이메일·비밀번호)을 그대로 씁니다. 별도 테스트 계정을 새로 만든 게 아니라, 기존 astraways 회원으로 로그인하는 구조입니다.

박사님이 평소 astraways에서 쓰시는 본인 계정으로 로그인하시면 됩니다. 계정이 기억나지 않으시면, 현재 등록된 회원 이메일을 확인해 드리겠습니다.

soul3

```
bash
sudo mysql --socket=/home/mysql/mysql.sock -e "USE kr_astraways; SELECT id, email, name, is_acti
```

여기서 활성(is_active=1) 계정 이메일이 보입니다. 그 이메일과 해당 계정의 비밀번호로 로그인하시면 됩니다. 비밀번호는 password_hash로 저장돼 있어 평문 조회는 안 되니, 비밀번호를 모르는 계정이라면 테스트 전용 계정을 하나 새로 만드는 게 깔끔합니다.

테스트 계정을 새로 만들려면 아래로 생성하겠습니다(이메일·비밀번호는 원하시는 값으로 바꾸세요).

soul3

```
bash
php -r '$h=password_hash("Test1234!", PASSWORD_DEFAULT); echo $h, "\n";' > /tmp/hash.txt
HASH=$(cat /tmp/hash.txt)
sudo mysql --socket=/home/mysql/mysql.sock -e "USE kr_astraways; INSERT INTO users (email, passw
echo "생성됨: toss-test@astraways.net / Test1234!"
```

이렇게 하면 toss-test@astraways.net / Test1234! 로 결제 로그인할 수 있습니다. 이 계정은 나중에 토스 심사관에게 전달할 테스트 계정으로도 그대로 쓸 수 있습니다.

어느 쪽으로 하실지(기존 계정 확인 / 테스트 계정 생성) 결과를 보내주시면 결제 진행으로 이어가겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

로그인했더니 결제까지 모두 연결됨.

로그인 → 상품 → 통합결제창 → 승인까지 전체 경로가 연결됐습니다. 단건 결제 코어가 완성됐습니다. DB 기록까지 확인하겠습니다.

soul3

```
bash
mysql --socket=/home/mysql/mysql.sock -u payment_user -p'공백없는_새비밀번호' payment_db -e "
SELECT order_id, product_code, amount, status, method, approved_at FROM orders ORDER BY created_
SELECT user_key, product_code, started_at, expires_at FROM entitlements ORDER BY id DESC LIMIT 3
```

`orders` 에 `status=DONE` · `approved_at` 이 채워지고, `entitlements` 에 30일 이용권이 기록돼 있으면 결제 코어 검증 완료입니다.

이게 표준 템플릿이 됐습니다. 남은 작업은 두 갈래입니다.

첫째, `astraways`를 토스 심사까지 끌고 가기 위한 필수 페이지 일괄 추가입니다. 앞서 정리한 공통 페이지(`navbar.js` · `footer-biz.js` · `terms.html` · `privacy.html` · `refund.html` · `auth/register`)를 만들고, 결제 성공 시 `users.membership_type = paid` · `subscription_end` 갱신을 `payment_success.php` 에 더해 `astraways` 본체 등급과 연동합니다. 그다음 결제경로 PPT를 제작해 심사 제출합니다.

둘째, 이 템플릿을 나머지 11개 사이트로 복제하는 것입니다. `config`만 교체하면 되는 구조라, 사이트별 `SITE_KEY`·`SITE_URL`·`SESSION_BOOTSTRAP`·회원 테이블 위치만 바꾸면 됩니다.

먼저 DB 기록 결과를 보내주시고, 다음으로 어느 쪽을 진행할지 알려주세요.

- `astraways` 필수 페이지 + 심사 준비 마저 완성
- 다른 사이트 복제 먼저(어느 사이트부터인지)

결과를 보내주시면 대단히 감사하겠습니다.

```
백없는_새비밀번호' payment_db -e "SELECT
order_id,status,amount,created_at FROM orders ORDER BY created_at
DESC LIMIT 5;"
echo "--- 결제 직후 에러로그 ---"
sudo tail -n 8 /var/log/php-fpm/www-error.log
--- 결제 직후 에러로그 ---
[19-Jun-2026 04:49:11 UTC] PHP Fatal error: Uncaught PDOException:
SQLSTATE[HY000] [1045] Access denied for user
'klifestyle_user'@'localhost' (using password: YES) in
더 보기
```

주문 1건이 `READY` 로 남았습니다. `checkout` 진입과 주문 선기록은 정상인데, 승인까지는 안 갔습니다. `status=READY` 에 `method` · `approved_at` 이 비어 있다는 건, 결제창에서 결제하기를 눌렀지만 토스 승인(`payment_success.php` 의 `confirm`)이 호출되지 않았다는 뜻입니다.

원인 후보가 명확합니다. `payment_success.php` 는 결제 성공 후 `successUrl`로 돌아오는 페이지인데, 이 안에서 `pay_db()` (결제 DB)만 쓰고 `astraways_db.php` 는 `require`하지 않습니다. 그런데 `successUrl`로 돌아올 때 세션 가드는 없지만, 만약 토스 결제창에서 "결제를 완료했어요"까지 못하고 중간에서 멈췄다면 `confirm`이 안 돈 것입니다.

정확히 짚기 위해, 방금 그 주문이 어디서 멈췄는지 확인하겠습니다.

soul3

```
bash
echo "---- success/fail 접근 흔적 (nginx 액세스 로그) ----"
sudo grep -E "payment_success|payment_fail" /var/log/nginx/*access*.log 2>/dev/null | tail -10
echo "---- 결제 직후 PHP 에러 ----"
sudo tail -n 5 /var/log/php-fpm/www-error.log
```

확인 포인트입니다.

- `payment_success.php` 접근 로그가 있는데 200이 아니거나, 아예 호출 흔적이 없는지
- 호출했다면 그 직후 PHP 에러(예: `pay_db` 연결 실패, confirm 응답 처리 오류)가 있는지

가능성은 두 가지입니다.

- 결제창에서 카드 승인을 끝까지 안 누르고 닫음 → confirm 미호출(정상, READY로 남음). 이 경우 다시 끝까지 결제하면 됩니다.
- 승인했는데 successUrl로 못 돌아옴 → successUrl 경로 (`/billing/api/payment_success.php`)나 confirm API 호출에 문제

로그 결과를 보내주세요. 그걸 보고, 결제를 다시 끝까지 진행할지 코드를 손볼지 정하겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

본인확인방법선택이 나와. ARS본인확인, 카드본인확인

그 화면은 토스 결제창의 정상 단계입니다. 통합결제창에서 카드 선택 후 결제하기를 누르면, 카드사 본인확인(ARS 또는 카드 정보 입력) 단계로 넘어갑니다. 거기서 본인확인을 끝까지 마쳐야 승인이 떨어지고 successUrl로 돌아옵니다.

테스트 키 환경에서는 실제 인증이 아니라 테스트용으로 통과됩니다. 둘 중 편한 쪽으로 진행하시면 됩니다.

- 카드 본인확인 선택 → 테스트 카드 정보 입력 후 진행
- ARS 본인확인 선택 → 테스트 환경에서는 실제 전화 없이 통과 처리

이 단계까지 왔다는 건 결제창 연동이 제대로 됐다는 뜻입니다. 본인확인을 마치면 "결제를 완료했어요"(`success.php`)로 이동하고, 그때 `orders` 가 READY → DONE으로 바뀌며 `entitlements` 에 이용권이 기록됩니다.

본인확인을 끝까지 진행하신 뒤, 성공 페이지로 이동했는지 알려주시고 아래로 확인하겠습니다.

soul3

```
bash
```

```
mysql --socket=/home/mysql/mysql.sock -u payment_user -p'공백없는_새비밀번호' payment_db -e "
SELECT order_id, status, method, approved_at FROM orders ORDER BY created_at DESC LIMIT 2;
SELECT user_key, product_code, expires_at FROM entitlements ORDER BY id DESC LIMIT 2;"
```

`status=DONE` 과 `entitlements` 기록이 나오면 결제 코어 완성입니다. 결과를 보내주시면 대단히 감사하겠습니다.

ARS했고 네이버 인증번호까지 입력했어

ARS 본인확인과 네이버 인증까지 마치셨으면 승인이 떨어졌을 겁니다. 결과를 확인하겠습니다.

soul3

```
bash
mysql --socket=/home/mysql/mysql.sock -u payment_user -p'공백없는_새비밀번호' payment_db -e "
SELECT order_id, status, method, approved_at FROM orders ORDER BY created_at DESC LIMIT 2;
SELECT user_key, product_code, started_at, expires_at FROM entitlements ORDER BY id DESC LIMIT 2;"
```

여기서 갑니다.

- `status=DONE` + `entitlements` 기록 있음 → 결제 코어 완성. 화면도 "결제를 완료했어요"로 갔을 겁니다.
- 여전히 `READY` → 승인 후 `successUrl`로 돌아오는 길에서 막힌 것입니다. 이 경우 `confirm` 단계 로그를 보겠습니다.

만약 화면이 성공 페이지가 아니라 에러거나 빈 화면이었다면, 그 직후 로그도 함께 확인합니다.

soul3

```
bash
echo "--- success 호출과 직후 에러 ---"
sudo grep "payment_success" /var/log/nginx/*access*.log 2>/dev/null | tail -3
sudo tail -n 5 /var/log/php-fpm/www-error.log
```

화면이 어디로 갔는지(성공 페이지 / 에러 / 다른 화면)와 위 두 결과를 보내주시면 바로 마무리하겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

그런데 결제까지 해? 네이버 비밀번호 누르라고 하는데?

아니요, 거기서 멈추세요. 실제 결제 비밀번호는 입력하지 마세요.

지금 화면은 네이버페이 간편결제의 실제 결제 비밀번호 입력 단계입니다. 테스트 키 환경이라 진짜 청구는 안 되지만, 네이버페이는 본인의 실제 네이버 계정·결제수단과 연결되므로 굳이 실제 비밀번호를 넣어 끝까지 갈 필요가 없습니다. 흐름 검증은 카드 결제수단으로 하는 게 안전하고 깔끔합니다.

네이버페이 창은 취소하거나 닫으시고, 다시 결제할 때는 결제위젯에서 **신용·체크카드**를 선택해 주세요. 테스트 환경의 카드 결제는 실제 카드사 인증 없이 테스트용으로 통과되어 승인까지 확인할 수 있습니다.

정리하면 이렇습니다.

- 지금 네이버페이 비밀번호 화면 → 입력하지 말고 취소
- 다시 <https://kr.astraways.net/billing/membership.php> → 결제하기 → 결제위젯에서 카드 선택 → 결제하기 → 테스트 카드 정보 입력으로 승인

카드로 승인까지 가면 `success.php`로 이동하고 DB가 DONE으로 바뀝니다. 그 뒤 아래로 확인하겠습니다.

soul3

```
bash
mysql --socket=/home/mysql/mysql.sock -u payment_user -p'공백없는_새비밀번호' payment_db -e "SELE"
```

카드로 다시 진행하신 결과를 보내주시면 마무리하겠습니다. 결과를 보내주시면 대단히 감사하겠습니다.

```
--- 결제 직후 에러로그 ---
[19-Jun-2026 04:49:11 UTC] PHP Fatal error: Uncaught PDOException:
SQLSTATE[HY000] [1045] Access denied for user
'klifestyle_user'@'localhost' (using password: YES) in
/var/www/klifestyle.net/public_html/barefoot_topic.php:2
Stack trace:
#0 /var/www/klifestyle.net/public_html/barefoot_topic.php(2): PDO-
>__construct()
#1 {main}
더 보기
```

여전히 같은 주문 1건만 READY로 있습니다. 네이버페이 비밀번호 단계에서 멈추고 취소하셨으니 승인이 안 된 게 맞고, 카드로 다시 시도한 새 주문은 아직 안 보입니다. 카드로 끝까지 한 번 더 진행하시면 됩니다.

다만 결제 흐름과 별개로, 토스 테스트 결제는 본인확인·인증번호 같은 실제 카드사/간편결제 인증을 요구하지 않는 **테스트 전용 카드**로 하는 게 정석입니다. 지금 ARS·네이버 인증이 뒀다는 건 실제 결

제수단 인증 경로로 들어간 것이라, 흐름만 검증하려면 토스가 제공하는 테스트 카드번호를 쓰는 게 깔끔합니다. 정확한 테스트 카드 정보를 확인하겠습니다.

웹 검색됨 >

확인했습니다. 핵심은 두 가지입니다.

토스페이먼트에는 별도의 테스트용 국내 카드번호가 없고, 직접 발급받은 카드 정보를 입력해도 테스트 환경에서는 실제로 돈이 출금되지 않습니다. 그리고 휴대폰 본인 인증번호로 000000을 입력하면 됩니다. 즉 ARS·간편결제 비밀번호 같은 실제 인증 경로 대신, 결제위젯에서 **신용·체크카드**를 고르고 본인 카드번호를 입력한 뒤 휴대폰 인증 단계에서 000000을 넣으면 테스트로 통과됩니다.

방금 멈춘 네이버페이는 실제 네이버 계정 결제라 인증을 우회할 수 없으니, 카드로 진행하시는 게 맞습니다.

진행 순서입니다.

1. 네이버페이 창은 취소
2. <https://kr.astraways.net/billing/membership.php> → 결제하기 → 결제위젯에서 **신용·체크카드** 선택
3. 본인 카드 정보 입력(테스트라 청구 안 됨)
4. 휴대폰 인증번호 요구 시 **000000** 입력
5. 승인 완료 → [success.php](#) 이동

카드로 끝까지 진행하신 뒤 확인하겠습니다.

soul3

bash

```
mysql --socket=/home/mysql/mysql.sock -u payment_user -p'공백없는_새비밀번호' payment_db -e "
SELECT order_id, status, method, approved_at FROM orders ORDER BY created_at DESC LIMIT 3;
SELECT user_key, product_code, expires_at FROM entitlements ORDER BY id DESC LIMIT 3;"
```

[status=DONE](#) 과 [entitlements](#) 기록이 나오면 결제 코어 완성입니다. 결과를 보내주시면 대단히 감사하겠습니다.



하나카드 결제하려고 하는데 카드번호입력하는 일반결제버튼을 눌렀더니 하나카드 프로그램 다운로드 버튼이 있어. 그걸 클릭하니

