Audit

git : https://github.com/pluto1011/-Lending-DEX-_solidity/blob/main/src/DEX.sol

commit : `d22ce57192079037f493b5eee3cad624fdfc2617`

ID	요약	위험도
ICARUS-001	`addLiquidity` 함수에서 유동성 비율 계산 없이 유동성 풀에 추가됩니다.	High

Description

`addLiquidity` 함수는 사용자가 두 개의 토큰(`amountX` 와 `amountY`)을 제공하여 유동성 풀에 추가할 때, 현재 유동성 풀의 비율에 따라 `amountX` 와 `amountY` 를 계산하지 않고 바로 추가합니다. 이는 유동성 풀의 균형을 고려하지 않은 방식입니다.

Impact

High

공격자가 풀의 비율을 조작하려면 **상당한 자산이 필요**할 수 있으며, 공격 비용이 높을 수 있습니다. 그러나 **풀 비율을 충분히 조작시킬 수 있는 경우**, 공격자는 큰 이익을 취할 수 있습니다.

또한 공격 자체가 단순하지 않지만, 성공했을 때 **피해가 크고, 서비스 운영에 미치는 영향이 매우 큽니다**.

Recommendation

- 유동성 풀 비율 검증 로직 추가

ID	요약	위험도
ICARUS-002	`addLiquidity` 함수 호출 실패시 불필요한 가스비가 발생하게 됩니다.	Information

Description

`addLiquidity` 함수를 호출할 때, 복잡한 수학적 연산 후에 토큰의 `allowance`와 `balanceOf` 조건을 확인합니다. 이로 인해 조건에 부합하지 않는 경우 불필요한 가스비가 발생합니다.

Impact

Information

조건에 부합하지 않는 경우 불필요한 가스비가 발생합니다.

Recommendation

- `require` 문을 코드 앞쪽에 배치시켜 함수 호출시 바로 token에 대한 조건들을 확인하도록 구현

ID	요약	위험도
ICARUS-003	`removeLiquidity` 함수는 유동성 공급자 개인의 LP 토큰량을 검증하지 않고 전체 LP 토큰량만 검증합니다.	information

Description

`removeLiquidity` 함수는 유동성 공급자 개인의 LP 토큰량을 검증하지 않고 전체 LP 토큰량만 검증합니다.

이로 인해 사용자가 개인의 LP 토큰량을 초과하여 함수 호출을 시도할 경우, 불필요한 가스비를 추가로 소모하게 됩니다.

Impact

information

함수 호출을 실패하는 경우 불필요한 가스비를 소모합니다.

Recommendation

- 유동성 풀에서 제거하려는 개인 LP 토큰량에 대한 검증 로직을 구현