AIXU: Presale Code Audit

June 10th, 2025

https://t.me/coinsult_tg

info@coinsult.net

Preliminary Audit Report

Report is not final and will not be published, awaiting changes.

Issues below

Initialization & Upgradeability

Issue: Missing initialization of PausableUpgradeable and zero-address checks for critical parameters.

```
constructor() {
        _disableInitializers();
}

function initialize(...) external initializer {
        if (aixuPrice_ == 0) revert InvalidPrice();

        __Ownable_init(owner_);
        _pause();
        // Missing: __Pausable_init()
        // Missing: require(treasury_ != address(0), "Invalid treasury");
        // Missing: require(owner_ != address(0), "Invalid owner");
        ...
}
```

Recommendation:

Initialize all inherited contracts and add zero-address guards:

Arithmetic & Rounding

Issue: Hard-coded oracle decimals and rounding-down bias in token calculation.

```
uint256 paidUSD = (msg.value * uint256(nativeCoinPrice)) /
NATIVE_COIN_PRICE_TO_USD_CONVERSION_FACTOR;
 uint256 purchasedAIXU = (paidUSD * AIXU_BASE) / aixuPrice;
```

Recommendation:

Fetch the feed decimals dynamically and document rounding behavior.

External Calls & Reentrancy

Issue: External ETH transfer before state updates and no nonReentrant guard.

```
function purchaseWithNativeCoin(...) external payable whenNotPaused {
    TransferHelper.safeTransferETH(treasury, msg.value);
        _processPurchase(...);
}
```

Recommendation:

Add reentrant guards from openzeppelin.

Payment Token Handling

Issue: Accepts any token solely based on decimals, treating all as $1.

```
function _addPaymentToken(address paymentToken) internal {
        uint256 paymentTokenDecimals = IERC20(paymentToken).decimals();
        paymentTokensBases[paymentToken] = 10 ** paymentTokenDecimals;
}
```

Recommendation:

Require a price oracle for each token or explicitly document that only verified USD-pegged tokens are allowed.

Accounting vs. Distribution

Issue: Balances are recorded but no token mint or transfer, so users cannot claim AIXU.

```
function _processPurchase(...) internal returns (uint256) {
        _balances[recipient] += purchasedAIXU;
        // No mint or transfer of actual AIXU tokens
        return purchasedAIXU;
}
```

Recommendation:

Integrate on-chain minting or transfer within _processPurchase.