Cross-Chain Asset Transfer System with Gasless Transaction Support

Project Overview

This project requires the development of a comprehensive cross-chain asset transfer system with integrated gasless transaction functionality. The system will enable users to transfer ERC-20 and ERC-721 tokens across different blockchain networks without requiring ETH for gas fees, while maintaining security, transparency, and verifiability throughout the transfer process.

Technical Requirements

Gasless Transaction Forwarder

- Implement a forwarder contract that allows users to send transactions without holding ETH
- Support both ERC-20 and ERC-721 token standards
- Create a mechanism to accept user transactions and forward them on-chain
- Implement proper signature verification and meta-transaction handling

Cross-Chain Bridge Functionality

- Deploy gateway smart contracts on two different testnets (e.g., Amoy & Sepolia)
- Implement a lock & mint mechanism:
  - Assets locked on Chain A result in equivalent tokens minted on Chain B
  - Burning/locking assets on Chain B enables unlocking on Chain A
- Provide proof-of-transfer for all transactions including:
  - Transaction hash
  - Timestamp
  - Sender and receiver addresses

  - Asset amount/identification
- No third-party bridge integration allowed (custom implementation required)
- Allow use of standard libraries like OpenZeppelin for security best practices

Security Implementation
- Implement comprehensive security measures:
  - Replay attack protection
  - Secure contract access controls
  - Reentrancy guards
  - Input validation
  - Protection against unauthorized minting
- Optimize for gas efficiency
- Ensure cross-chain validation mechanisms

User Interface
- Develop an intuitive frontend interface using React, Next.js, or Vue
- Implement wallet connectivity supporting both MetaMask AND WalletConnect
- Display key information:
  - Asset balances across chains
  - Pending transfers
  - Transaction history and logs
  - Transfer receipts with verification capability
- Create a dedicated interface for gasless transfers where users can enter transaction details

Bonus Features

- Implement on-chain verification using Merkle proofs or zk-SNARKs
- Develop a decentralized relayer network to remove centralized dependencies
- Advanced gas optimization techniques
- Additional UI enhancements:
  - Asset filtering
  - Notification system
  - Transfer progress indicators
  - Detailed transaction history

Evaluation Metrics
- Smart contract implementation quality
- Gas optimization effectiveness
- Security measures implementation:
  - Protection against known attack vectors
  - Access control implementation
  - Input validation thoroughness
- User experience and interface quality:
  - Wallet connection seamlessness
  - Transaction flow clarity
  - Error handling
- Solidity Shield audit score for all smart contracts
- Bridge safety and verification mechanisms

Deliverables
1. **Smart Contracts**:

- Deployed and verified on-chain (addresses must be included in README)
- Full source code with comprehensive documentation
- Passing all security checks

2. **Frontend Application**:
   - Live hosted demo URL
   - Source code repository
   - Support for all required wallet connections
   - Complete implementation of gasless transaction interface

3. **Documentation**:
   - Comprehensive README with setup instructions
   - Technical architecture diagram
   - Contract interaction flow documentation

4. **Demo Materials**:
   - Recorded demo video (LOOM link)
   - Test scripts for verification

5. **Repository**:
   - GitHub repository with complete source code
   - Clear commit history
   - Properly structured project

All implementations must be original (newly deployed contracts) with no reliance on existing third-party bridge solutions, while maintaining the highest standards of security, efficiency, and user experience.