KWALA

FIREWALL

LEARN. BUILD. CONNECT

OUR TEAM MEMBER

RONAK THAKUR (TEAM LEADER)

NEERAJ UPADHAYAY

ANJLIKA ANAND

SHREYA

Problem Statement

Existing firewall solutions require heavy backend infrastructure and technical expertise, making them difficult to implement for small teams and individuals. Our project uses Kwala's workflow automation to create a no-backend firewall that is easy to deploy, scalable, and automated for blocking malicious traffic.

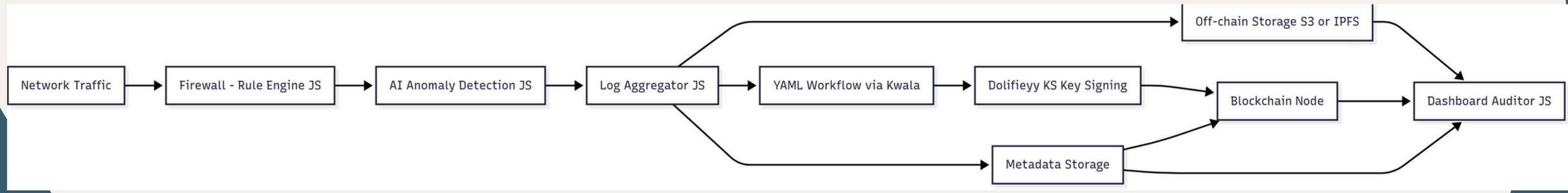
INRODUCTION

- Kwala is a lightweight automation platform that uses simple YAML-based workflows.
- It eliminates the need for a traditional backend by directly connecting triggers and actions.
- By leveraging Kwala, we can design a custom firewall that:
  Monitors incoming requests, Blocks malicious traffic automatically.
- This makes firewall creation faster, simpler, and serverless compared to traditional solutions.

SOLUTION

- **Real-Time Network Monitoring (Kwala)**: Continuously captures incoming and outgoing packets, extracting metadata like IP addresses, ports, protocols, and payload details.
- **Immutable Logging (Solidity Smart Contract)**: Suspicious IPs and attack details are securely stored on the Ethereum blockchain, ensuring tamper-proof records and global transparency.
- **Automated Blocking**: Detected malicious IPs are immediately blocked across all connected applications and systems.
- **Telegram Alerts**: Sends instant messages to administrators on Telegram for every detected threat.

System Architecture



1. Input Layer
   - Network traffic / user requests (HTTP requests) enter the system.
2. Kwala Workflow Layer
   - Kwala YAML Rules define firewall behavior:

     Allow legitimate requests (via POST/GET).

     Block or drop malicious patterns (e.g., SQL injection, XSS, DoS).
   - No backend required → rules execute directly in Kwala.

System Architecture

3. Action Layer
   - ✅ Allowed traffic is forwarded to the application/server.
   - ❌ Blocked traffic is logged and denied access.
4. Monitoring & Logging
   - Logs are stored for every blocked/allowed request.
   - Admin can review logs and update YAML firewall rules as needed.

Tech Stack

1. Core Platform
   - Kwala → Low-code/no-backend platform for workflow automation.
   - YAML → Rule definition language to configure firewall logic.2. Networking & Security
2. HTTP (GET/POST requests) → Traffic monitored by firewall.
   - Regex / Pattern Matching (via Kwala rules) → Detect malicious requests.
3. Monitoring & Logging
   - Kwala Built-in Logs → Store allowed/blocked requests.
   - (Optional) Google Sheets / JSON / Database integration via Kwala for analytics.
4. Feedback / Updates
   - Admin can update rules in Kwala YAML
   - Smart contracts update permissions is used via Solidity.

FUTURE SCOPE

1. **Decentralized Multi-Firewall Network**
   - Extend to a fully distributed firewall system across multiple nodes for enhanced network security.
2. **Advanced AI/ML Threat Detection**
   - Integrate deep learning models to detect complex cyber threats like zero-day attacks or ransomware in real-time.
3. **Cross-Platform Blockchain Support**
   - Add compatibility with multiple blockchains (Ethereum, Polygon, Solana) for better scalability and redundancy.
4. **Automated Compliance Reporting**
   - Generate regulatory reports automatically using blockchain-verified logs.
5. **Integration with IoT & Edge Devices**
   - Apply the firewall + blockchain system to IoT networks, securing smart homes, industrial devices, and sensors.
6. **Smart Contract Enhancements**
   - Implement dynamic smart contracts that adjust logging rules or alert thresholds automatically based on network behavior.
7. **User-Friendly Dashboard Enhancements**
   - Add visual analytics, AI suggestions, and anomaly prediction dashboards for easier monitoring.
8. **Blockchain Cost Optimization**
   - Explore Layer 2 solutions or sidechains to reduce transaction fees for large-scale logging.

Satellite Direct Connection

Global secure communication network

AES-256-GCM | Auto Mode

Active Satellites	Total Bandwidth	Avg Latency	Global Threats
5/6	4750 Mbps	52ms	3
Global coverage active	Combined satellite capacity	Network response time	Detected across network

Satellite Status | Network Map | Security Protocols

CyberGuard Alpha

ACTIVE

Location: North America
Satellite ID: SAT-001
Signal Strength: 100%
Latency: 36.18022422685163ms
Bandwidth: 1215.6527198893148 Mbps
Threats: 2
Connected

CyberGuard Beta

ACTIVE

Location: Europe
Satellite ID: SAT-002
Signal Strength: 93.190314019711 62%
Latency: 56.65676900940754ms
Bandwidth: 1171.315447031906 Mbps
Threats: 3
Connect

CyberGuard Gamma

MAINTENANCE

Location: Asia Pacific
Satellite ID: SAT-003
Signal Strength: 0%
Latency: 0ms
Bandwidth: 0 Mbps
Threats: 0

CyberGuard Delta

ACTIVE

Location: South America
Satellite ID: SAT-004
Signal Strength: 92.725860571234 49%
Latency: 30ms
Bandwidth: 1338.401182694452 Mbps
Threats: 1
Connect

CyberGuard Epsilon

ACTIVE

Location: Africa
Satellite ID: SAT-005
Signal Strength: 85%
Latency
Bandwidth
Threats

CyberGuard Zeta

ACTIVE

Location: Middle East
Satellite ID: SAT-006
Signal Strength: 85.800642145885 77%
Latency
Bandwidth
Threats

Threats/Min

16

Real-time detection

Blocked Requests

1,263

Total blocked today

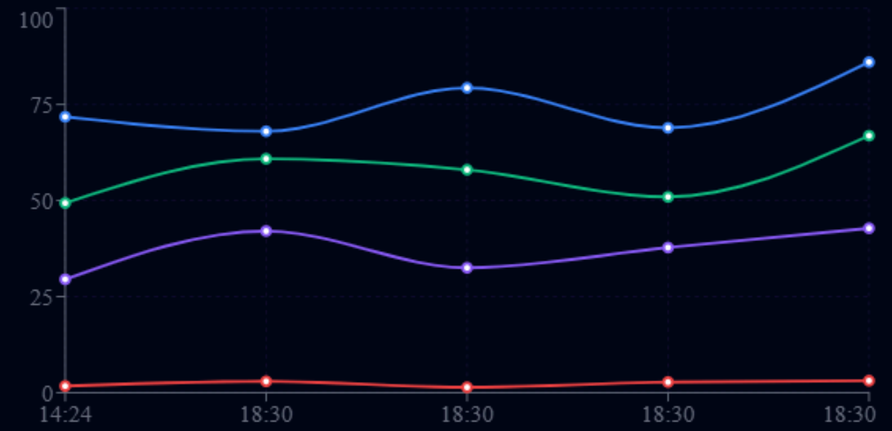
System Status

ONLINE

All systems operational

System Performance

Real-time system metrics



Recent Alerts

System and security alerts

Warning Network Anomaly

Automated detection triggered

18:30:27

Dismiss

Warning Memory Warning

Automated detection triggered

18:30:25

Dismiss

Warning High CPU Usage

CPU usage exceeded 80% threshold

14:23:45

Dismiss

Critical Suspicious Activity

Multiple failed login attempts detected

14:22:12

Dismiss

Active Network Connections

Real-time network activity monitoring

Blocked SSH #CONN-1758978025689

103.26.200.37:443 → 10.0.0.180

240.2 KB transferred • Duration: 00:02:17

Details

CyberGuard AI

Advanced Global Security Platform

SECURE | THREAT: LOW

Active Threats
0
Real-time detection active

Blocked Attacks
1,315
+0 in last hour

AI Processing
98.7%
Neural network efficiency

Network Status
ONLINE
All systems operational

Control Center | Satellite Link | Global Firewall | Live Detection | AI Training | Dashboard

AI Firewall | Threat Monitor | Web3 Security | Live Monitor | Global Detection | 3D Visualization | Reports

Security Score
88.3
Overall security posture

Active Incidents
0
Requiring attention

Vulnerabilities
88
Identified & tracked

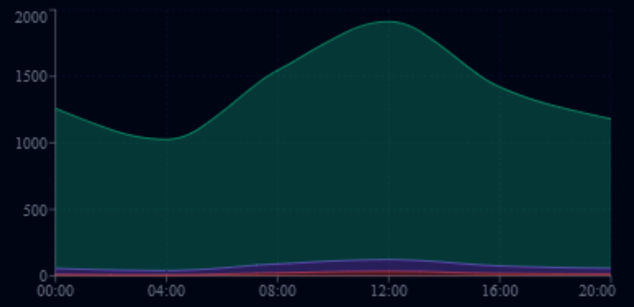
Compliance
97.6%
Regulatory compliance

System Uptime
100.00%

Overview | Threat Intel | Incidents | Vulnerabilities | Compliance

Threat Activity (24h)

Real-time threat detection and blocking



Attack Types Distribution

Most common attack vectors

- SQL Injection 35%
- DDoS 28%
- XSS 18%
- Brute Force 12%
- Other 7%

Geographic Threat Analysis

Attack origins and blocking effectiveness

File Edit Selection View Go Run Terminal Help kwala

EXPLORER

KWALA
- node_modules
- eth.js
- package-lock.json
- package.json

Welcome | eth.js

eth.js > ...

```javascript
const { ethers } = require("ethers");

function createRandomPrivateKey() {
    const wallet = ethers.Wallet.createRandom();
    const privateKey = wallet.privateKey;
    const publicAddress = wallet.address;

    console.log("Private Key:", privateKey);
    console.log("Public Address:", publicAddress);
}
// createRandomPrivateKey();


async function invokeSmartContract() {
    // Connect to the Ethereum network (using Infura or any other provider)
    const provider = new ethers.providers.JsonRpcProvider("https://rpc-amoy.polygon.technology");

    // Define your private key (NEVER hardcode it in production)
    const privateKey = "316aebafb298cd7ff29a947fd7b8b05ed9191820c79503acf95f948e4483b26a"; // Replace with your actual private key
    // 0x0E3710b3167eeE9D5E7E5430d99542A4e31636A7
    const signer = new ethers.Wallet(privateKey, provider);
```

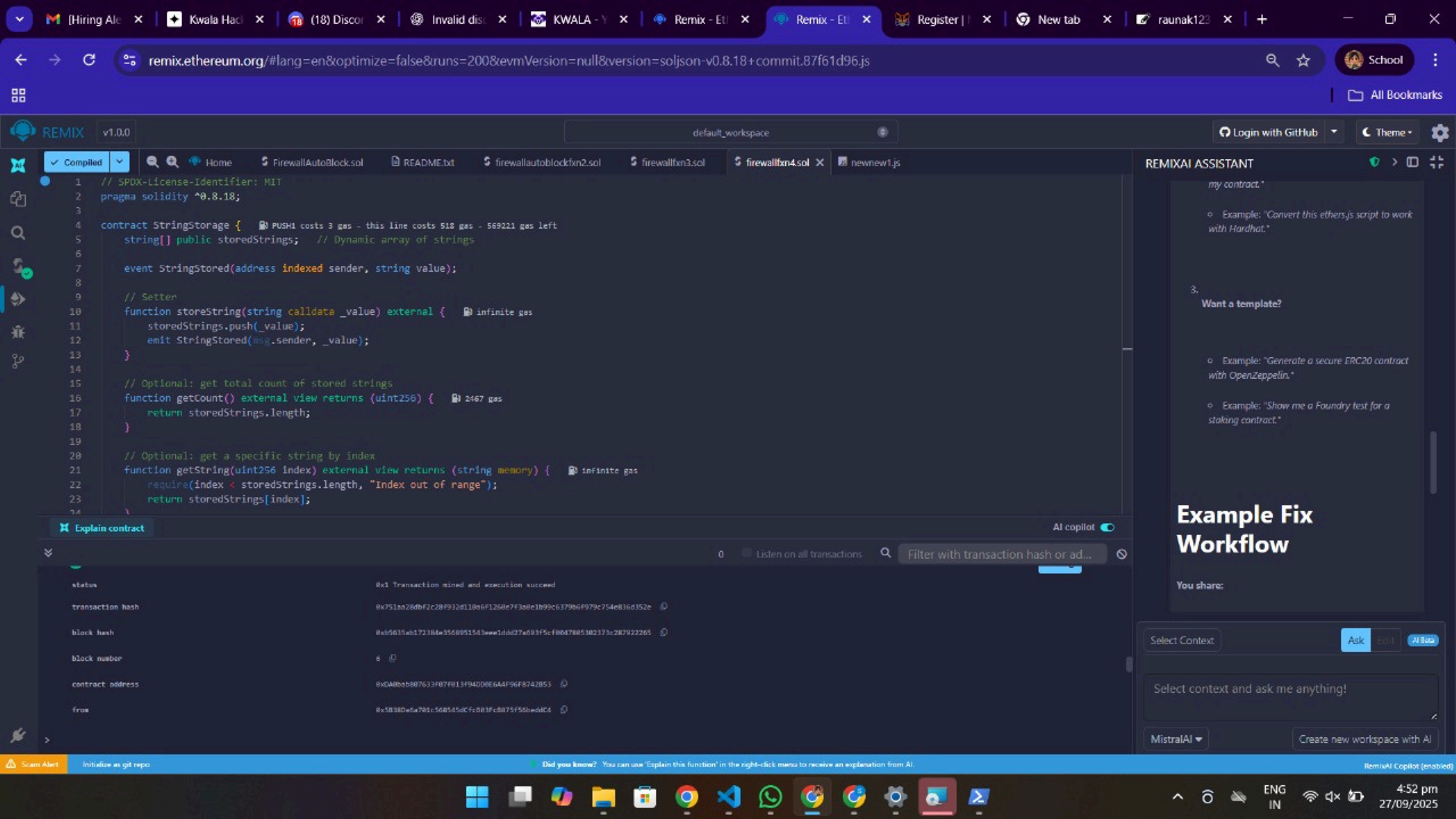
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell

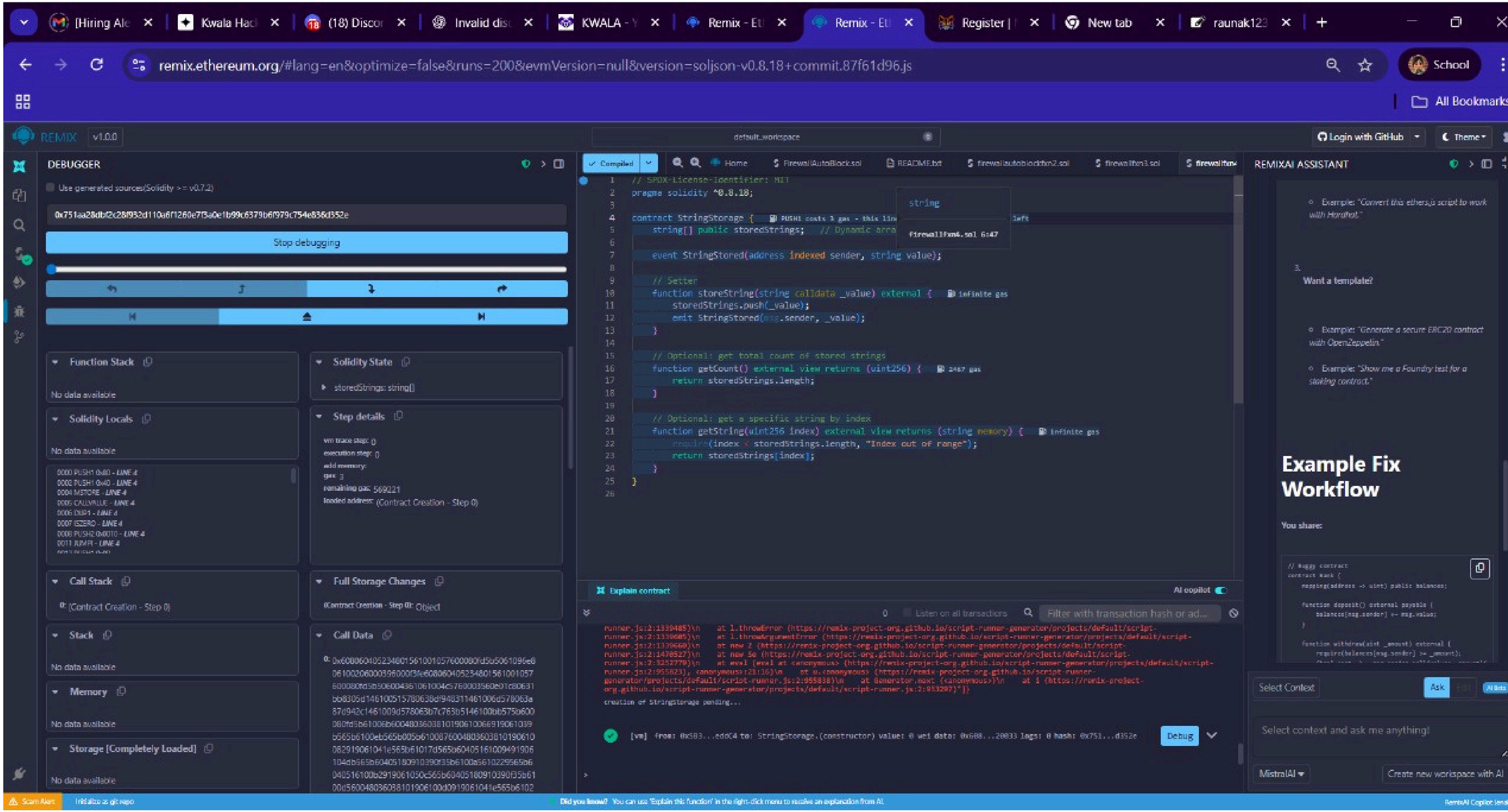
```
  contractAddress: null,
  transactionIndex: 0,
  gasUsed: BigNumber { _hex: '0x010e55', _isBigNumber: true },
  logsBloom: '0x0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000008000000000000000000000000100002000000000000000000000000000000000000000000080000000000000000000000000000020000200400000000400000000000
000080000000080020000000000000000000000000000000000000000000080000000000000000000004000000000000000000200100000000000000000004000000000000100800000000000000000000004000000
000000000000000000000000000000000000000000100000',
  blockHash: '0x47f0ff8e6d37bb96564246c60e9820ac968f4f02e32c62280b5e50dd03dcf6e4',
  transactionHash: '0xf349c5629f2bc4832af628ecb090a1c6458c46f1953234b2b542614bf07412c6',
  logs: [
    {
      transactionIndex: 0,
      blockNumber: 26957054,
      transactionHash: '0xf349c5629f2bc4832af628ecb090a1c6458c46f1953234b2b542614bf07412c6',
```

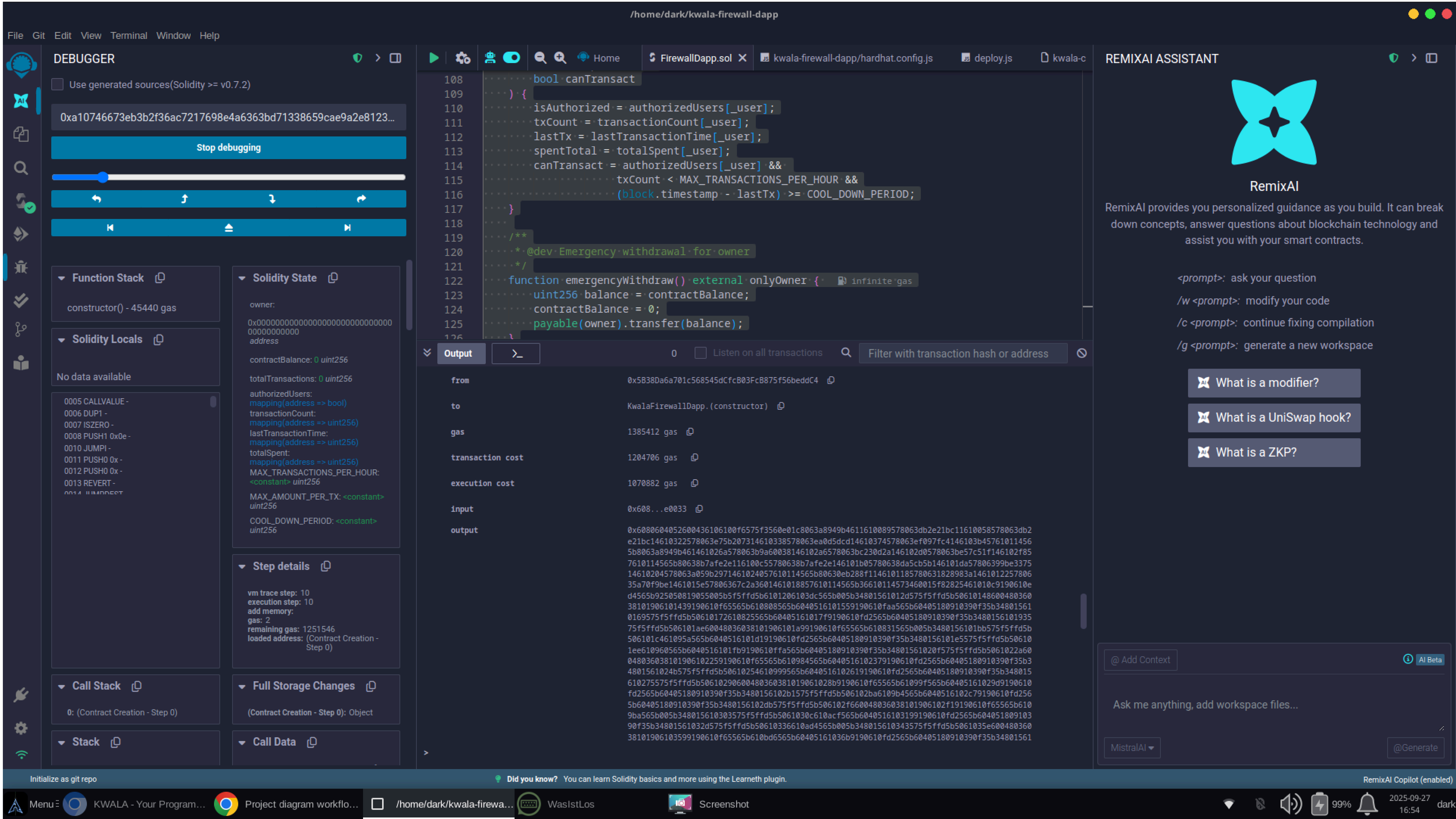
OUTLINE

TIMELINE

0 0 Ln 136, Col 23 Spaces: 4 UTF-8 CRLF {} JavaScript


remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js
REMIX v1.0.0
default_workspace
Login with GitHub
Theme
Compiled
Home
FirewallAutoBlock.sol
README.txt
firewallautoblockfxn2.sol
firewallfxn3.sol
firewallfxn4.sol
newnew1.js
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.18;
contract StringStorage {
PUSH1 costs 3 gas - this line costs 518 gas - 569221 gas left
string[] public storedStrings; // Dynamic array of strings
event StringStored(address indexed sender, string value);
// Setter
function storeString(string calldata _value) external {
infinite gas
storedStrings.push(_value);
emit StringStored(msg.sender, _value);
}
// Optional: get total count of stored strings
function getCount() external view returns (uint256) {
2467 gas
return storedStrings.length;
}
// Optional: get a specific string by index
function getString(uint256 index) external view returns (string memory) {
infinite gas
require(index < storedStrings.length, "Index out of range");
return storedStrings[index];
Explain contract
AI copilot
Listen on all transactions
Filter with transaction hash or ad...
status
0x1 Transaction mined and execution succeed
transaction hash
block hash
block number
6
contract address
from
REMIXAI ASSISTANT
my contract."
Example: "Convert this ethers.js script to work with Hardhat."
3.
Want a template?
Example: "Generate a secure ERC20 contract with OpenZeppelin."
Example: "Show me a Foundry test for a staking contract."
Example Fix Workflow
You share:
Select Context
Ask
Select context and ask me anything!
MistralAI
Create new workspace with AI
Scam Alert
Initialize as git repo
Did you know? You can use 'Explain this function' in the right-click menu to receive an explanation from AI.
RemixAI Copilot (enabled)
ENG IN
4:52 pm
27/09/2025


remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js
REMIX v1.0.0
DEBUGGER
Stop debugging
Function Stack
No data available
Solidity Locals
No data available
Solidity State
Step details
Call Stack
Full Storage Changes
Stack
No data available
Memory
No data available
Storage [Completely Loaded]
No data available
Call Data
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.18;
contract StringStorage {
string[] public storedStrings;
event StringStored(address indexed sender, string value);
function storeString(string calldata _value) external {
storedStrings.push(_value);
emit StringStored(msg.sender, _value);
function getCount() external view returns (uint256) {
return storedStrings.length;
function getString(uint256 index) external view returns (string memory) {
require(index < storedStrings.length, "Index out of range");
return storedStrings[index];
REMIXAI ASSISTANT
Example Fix Workflow
You share:
Explain contract
AI copilot
Select Context
Ask
Select context and ask me anything!
Create new workspace with AI


/home/dark/kwala-firewall-dapp
File Git Edit View Terminal Window Help
DEBUGGER
Use generated sources(Solidity >= v0.7.2)
0xa10746673eb3b2f36ac7217698e4a6363bd71338659cae9a2e8123...
Stop debugging
Function Stack
constructor() - 45440 gas
Solidity Locals
No data available
0005 CALLVALUE -
0006 DUP1 -
0007 ISZERO -
0008 PUSH1 0x0e -
0010 JUMPI -
0011 PUSH0 0x -
0012 PUSH0 0x -
0013 REVERT -
Solidity State
owner:
0x0000000000000000000000000000000000000000
address
contractBalance: 0 uint256
totalTransactions: 0 uint256
authorizedUsers:
mapping(address => bool)
transactionCount:
mapping(address => uint256)
lastTransactionTime:
mapping(address => uint256)
totalSpent:
mapping(address => uint256)
MAX_TRANSACTIONS_PER_HOUR:
<constant> uint256
MAX_AMOUNT_PER_TX: <constant> uint256
COOL_DOWN_PERIOD: <constant> uint256
Step details
vm trace step: 10
execution step: 10
add memory:
gas: 2
remaining gas: 1251546
loaded address: (Contract Creation - Step 0)
Call Stack
0: (Contract Creation - Step 0)
Full Storage Changes
(Contract Creation - Step 0): Object
Stack
Call Data
Home
FirewallDapp.sol
kwala-firewall-dapp/hardhat.config.js
deploy.js
kwala-c
bool canTransact
) {
isAuthorized = authorizedUsers[_user];
txCount = transactionCount[_user];
lastTx = lastTransactionTime[_user];
spentTotal = totalSpent[_user];
canTransact = authorizedUsers[_user] &&
txCount < MAX_TRANSACTIONS_PER_HOUR &&
(block.timestamp - lastTx) >= COOL_DOWN_PERIOD;
}
/**
* @dev Emergency withdrawal for owner
*/
function emergencyWithdraw() external onlyOwner {
infinite gas
uint256 balance = contractBalance;
contractBalance = 0;
payable(owner).transfer(balance);
Output
0
Listen on all transactions
Filter with transaction hash or address
from 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to KwalaFirewallDapp.(constructor)
gas 1385412 gas
transaction cost 1204706 gas
execution cost 1070882 gas
input 0x608...e0033
output
REMIXAI ASSISTANT
RemixAI
RemixAI provides you personalized guidance as you build. It can break down concepts, answer questions about blockchain technology and assist you with your smart contracts.
<prompt>: ask your question
/w <prompt>: modify your code
/c <prompt>: continue fixing compilation
/g <prompt>: generate a new workspace
What is a modifier?
What is a UniSwap hook?
What is a ZKP?
@ Add Context
AI Beta
Ask me anything, add workspace files...
MistralAI
@Generate
Initialize as git repo
Did you know? You can learn Solidity basics and more using the Learneth plugin.
RemixAI Copilot (enabled)
Menu
KWALA - Your Program...
Project diagram workflo...
/home/dark/kwala-firewa...
WasIstLos
Screenshot
99%
2025-09-27
16:54
dark

Workflow Editor

Build your automation workflow with YAML

Import | Workflow Builder

YAML Lines: 39 Editable

```yaml
Name: kwala-simple-monitor_d839
Trigger:
  TriggerSourceContract: NA
  TriggerChainID: NA
  TriggerEventName: NA
  TriggerEventFilter: NA
  TriggerSourceContractABI: NA
  TriggerPrice: NA
  RecurringSourceContract: 0xf8e81D47203A594245E36C48e151709F0C19fBe8
  RecurringChainID: 1
  RecurringEventName: TransactionExecuted
  RecurringEventFilter: NA
  RecurringSourceContractABI: NA
  RecurringPrice: NA
  RepeatEvery: event
  ExecuteAfter: immediate
  ExpiresIn: 1817453880
  Meta: NA
  ActionStatusNotificationPOSTURL: https://workflow-notification-test.kalp
    .network/push_notification
```

Saved | Compile | Deploy | Activate

Validation

Real-time workflow validation

Click Compile to validate your workflow

Get syntax and schema validation from the API

Quick Actions

- Import from File
- Configure Keys
- View Documentation

Workflow Information

Info | Console Logs

Workflow Name
kwala-simple-monitor_d839

Status
Saved · Polling... (5s)

Workflow Address
Not deployed

Need Help?

CONCLUSION

- **Tamper-Proof Security:** Blockchain ensures logs are immutable and verifiable, preventing manipulation.
- **Real-Time Threat Detection:** AI module monitors traffic and flags anomalies instantly.
- **Scalable & Efficient: Detailed logs stored off-chain;** only hashes and metadata on-chain for efficiency.
- **Automated Workflows:** YAML workflows executed via Kwala automate blockchain logging seamlessly.
- **Integrated Dashboard:** JS-based dashboard provides visualization, alerts, and blockchain verification for auditors.
- **Trust & Transparency:** Multi-stakeholder environments can rely on accurate, auditable logs.

THANK YOU