Access NFT Smart Contract

```
Reentrancy in AccessNFTs.mint(string,address) (AccessNFT_flattened.sol#1754-1760):
        External calls:
        - _safeMint(_user,tokenCount) (AccessNFT_flattened.sol#1757)
                - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,data) (AccessNFT_flattened.sol#1619-1632)
        State variables written after the call(s):
        - _setTokenURI(tokenCount,_tokenURI) (AccessNFT_flattened.sol#1758)
                - _tokenURIs[tokenId] = _tokenURI (AccessNFT_flattened.sol#1723)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in AccessNFTs.mint(string,address) (AccessNFT_flattened.sol#1754-1760):
        External calls:
        - _safeMint(_user,tokenCount) (AccessNFT_flattened.sol#1757)
                - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,data) (AccessNFT_flattened.sol#1619-1632)
        Event emitted after the call(s):
        - MetadataUpdate(tokenId) (AccessNFT_flattened.sol#1724)
                - _setTokenURI(tokenCount,_tokenURI) (AccessNFT_flattened.sol#1758)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Parameter AccessNFTs.setRegistrationSC(address)._registrationSC (AccessNFT_flattened.sol#1749) is not in mixedCase
Parameter AccessNFTs.mint(string,address)._tokenURI (AccessNFT_flattened.sol#1754) is not in mixedCase
Parameter AccessNFTs.mint(string,address)._user (AccessNFT_flattened.sol#1754) is not in mixedCase
Parameter AccessNFTs.burn(uint256)._tokenID (AccessNFT_flattened.sol#1762) is not in mixedCase
Variable AccessNFTs.Registration (AccessNFT_flattened.sol#1744) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
AccessNFT_flattened.sol analyzed (19 contracts with 81 detectors), 7 result(s) found
```

Incentive NFT Smart Contract

```
Reentrancy in IncentiveNFTs.mint(string) (NFTIncentives_flattened.sol#1753-1759):
        External calls:
        - _safeMint(msg.sender,tokenCount) (NFTIncentives_flattened.sol#1756)
                - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,data) (NFTIncentives_flattened.sol#1619-1632)
        State variables written after the call(s):
        - _setTokenURI(tokenCount,_tokenURI) (NFTIncentives_flattened.sol#1757)
                - _tokenURIs[tokenId] = _tokenURI (NFTIncentives_flattened.sol#1723)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in IncentiveNFTs.mint(string) (NFTIncentives_flattened.sol#1753-1759):
        External calls:
        - _safeMint(msg.sender,tokenCount) (NFTIncentives_flattened.sol#1756)
                - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,data) (NFTIncentives_flattened.sol#1619-1632)
        Event emitted after the call(s):
        - MetadataUpdate(tokenId) (NFTIncentives_flattened.sol#1724)
                - _setTokenURI(tokenCount,_tokenURI) (NFTIncentives_flattened.sol#1757)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Parameter IncentiveNFTs.setProgressSC(address)._progressSC (NFTIncentives_flattened.sol#1748) is not in mixedCase
Parameter IncentiveNFTs.mint(string)._tokenURI (NFTIncentives_flattened.sol#1753) is not in mixedCase
Variable IncentiveNFTs.Progress (NFTIncentives_flattened.sol#1742) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
NFTIncentives_flattened.sol analyzed (19 contracts with 81 detectors), 5 result(s) found
```

Registration and Access Control Smart Contract

```
Reentrancy in Registration.registerUser(address,string) (RegistrationandAccessControl_flattened.sol#221-228):
        External calls:
        - _tokenId = AccessNFTSC.mint(_IPFSHash,_user) (RegistrationandAccessControl_flattened.sol#225)
        State variables written after the call(s):
        - userAccessNFT[_user] = _tokenId (RegistrationandAccessControl_flattened.sol#226)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in Registration.registerUser(address,string) (RegistrationandAccessControl_flattened.sol#221-228):
        External calls:
        - _tokenId = AccessNFTSC.mint(_IPFSHash,_user) (RegistrationandAccessControl_flattened.sol#225)
        Event emitted after the call(s):
        - UserRegistered(_user,_tokenId) (RegistrationandAccessControl_flattened.sol#227)
Reentrancy in Registration.unregisterUser(address) (RegistrationandAccessControl_flattened.sol#230-235):
        External calls:
        - AccessNFTSC.burn(userAccessNFT[_user]) (RegistrationandAccessControl_flattened.sol#233)
        Event emitted after the call(s):
        - UserUnRegistered(_user) (RegistrationandAccessControl_flattened.sol#234)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Parameter Registration.registerUser(address,string)._user (RegistrationandAccessControl_flattened.sol#221) is not in mixedCase
Parameter Registration.registerUser(address,string)._IPFSHash (RegistrationandAccessControl_flattened.sol#221) is not in mixedCase
Parameter Registration.unregisterUser(address)._user (RegistrationandAccessControl_flattened.sol#230) is not in mixedCase
Parameter Registration.registerProgressOracle(address)._oracle (RegistrationandAccessControl_flattened.sol#237) is not in mixedCase
Variable Registration.AccessNFTSC (RegistrationandAccessControl_flattened.sol#190) is not in mixedCase
Variable Registration.RegisteredUsers (RegistrationandAccessControl_flattened.sol#193) is not in mixedCase
Variable Registration.RegisteredProgressOracles (RegistrationandAccessControl_flattened.sol#194) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
RegistrationandAccessControl_flattened.sol analyzed (4 contracts with 81 detectors), 10 result(s) found
```

Progress Monitoring and Incentives Distribution Smart Contract

```
Reentrancy in Progress.claimNFT() (ProgressMonitoringandIncentivesDistribution_flattened.sol#199-210):
        External calls:
        - IncentiveNFTSC.transferFrom(address(this),msg.sender,nftToClaim) (ProgressMonitoringandIncentivesDistribution_flattened.sol#204)
        State variables written after the call(s):
        - availableNFTs -= 1 (ProgressMonitoringandIncentivesDistribution_flattened.sol#206)
        - claimableINFTs[msg.sender] -= 1 (ProgressMonitoringandIncentivesDistribution_flattened.sol#205)
        - currentIndex += 1 (ProgressMonitoringandIncentivesDistribution_flattened.sol#207)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

Reentrancy in Progress.mintIncentiveNFTs(string) (ProgressMonitoringandIncentivesDistribution_flattened.sol#183-190):
        External calls:
        - _tokenId = IncentiveNFTSC.mint(_ipfshash) (ProgressMonitoringandIncentivesDistribution_flattened.sol#185)
        State variables written after the call(s):
        - availableNFTs += 1 (ProgressMonitoringandIncentivesDistribution_flattened.sol#186)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in Progress.claimNFT() (ProgressMonitoringandIncentivesDistribution_flattened.sol#199-210):
        External calls:
        - IncentiveNFTSC.transferFrom(address(this),msg.sender,nftToClaim) (ProgressMonitoringandIncentivesDistribution_flattened.sol#204)
        Event emitted after the call(s):
        - INFTClaimed(msg.sender,nftToClaim) (ProgressMonitoringandIncentivesDistribution_flattened.sol#209)
Reentrancy in Progress.mintIncentiveNFTs(string) (ProgressMonitoringandIncentivesDistribution_flattened.sol#183-190):
        External calls:
        - _tokenId = IncentiveNFTSC.mint(_ipfshash) (ProgressMonitoringandIncentivesDistribution_flattened.sol#185)
        Event emitted after the call(s):
        - IncentiveNFTMinted(msg.sender,_tokenId,availableNFTs) (ProgressMonitoringandIncentivesDistribution_flattened.sol#188)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

ReentrancyGuard._reentrancyGuardEntered() (ProgressMonitoringandIncentivesDistribution_flattened.sol#109-111) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
ProgressMonitoringandIncentivesDistribution_flattened.sol analyzed (5 contracts with 81 detectors), 5 result(s) found
```