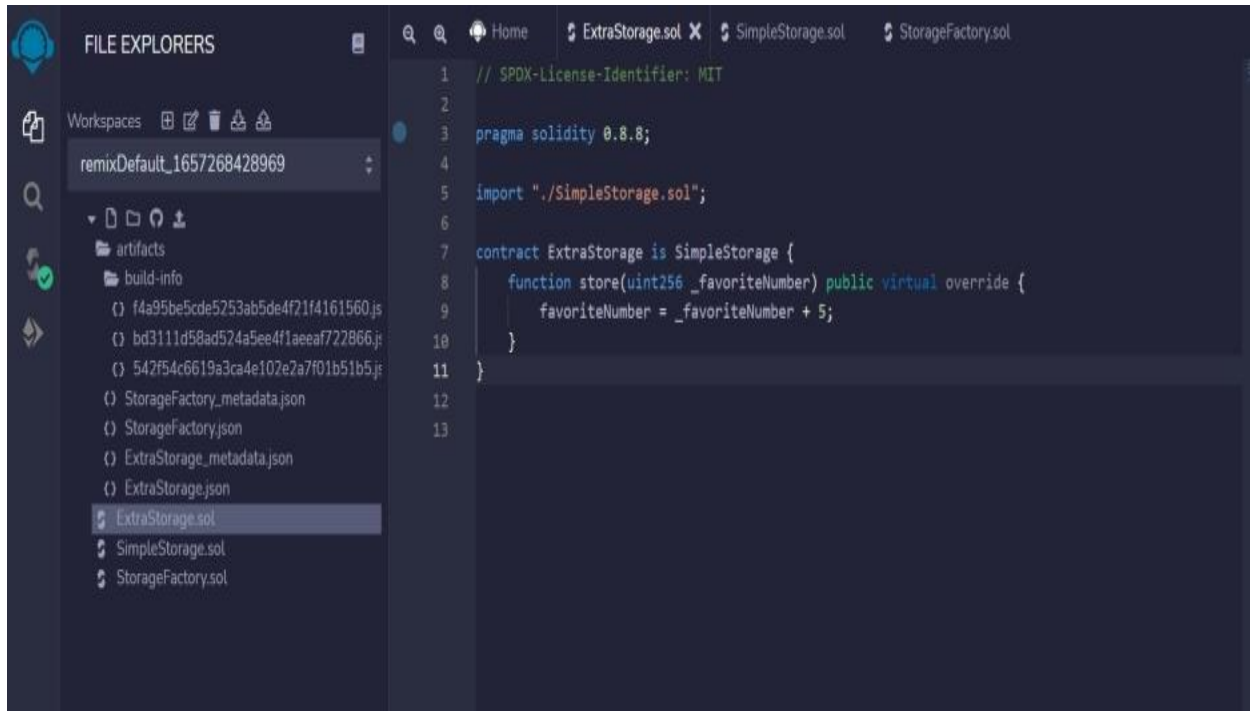
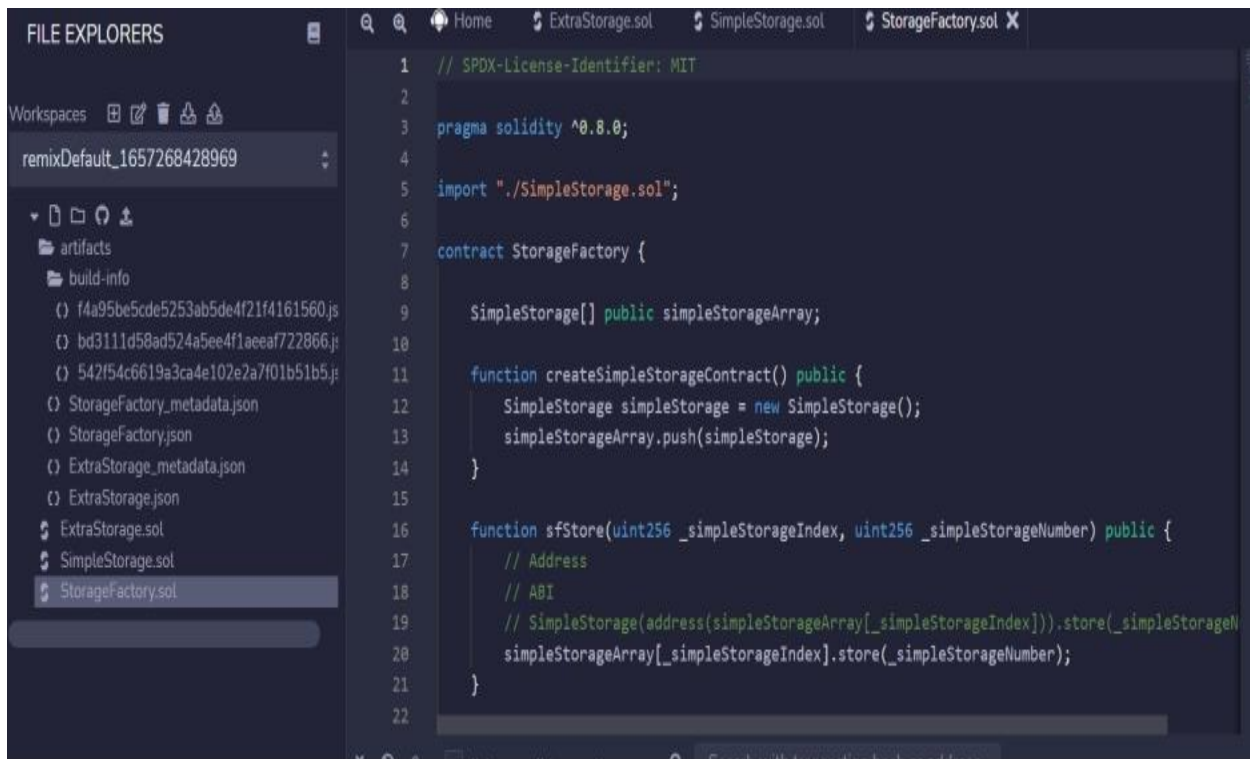


Remix Storage Factory



This screenshot shows the Remix IDE interface. On the left, the 'FILE EXPLORERS' sidebar displays a file tree for the workspace 'remixDefault_1657268428969'. The tree includes folders 'artifacts' and 'build-info', and a list of files: 'f4a95be5cde5253ab5de4f21f4161560.js', 'bd3111d58ad524a5ee4f1aeeaf722866.js', '542f54c6619a3ca4e102e2a7f01b51b5.js', 'StorageFactory_metadata.json', 'StorageFactory.json', 'ExtraStorage_metadata.json', 'ExtraStorage.json', 'ExtraStorage.sol', 'SimpleStorage.sol', and 'StorageFactory.sol'. The 'ExtraStorage.sol' file is selected and highlighted. The main editor area shows the code for 'ExtraStorage.sol'. The code includes a license identifier, a pragma statement for Solidity 0.8.8, an import statement for 'SimpleStorage.sol', and a contract definition for 'ExtraStorage' that inherits from 'SimpleStorage'. The 'store' function in 'ExtraStorage' overrides the one in 'SimpleStorage' by incrementing the favorite number by 5.

```
1 // SPDX-License-Identifier: MIT
2
3 pragma solidity 0.8.8;
4
5 import "./SimpleStorage.sol";
6
7 contract ExtraStorage is SimpleStorage {
8     function store(uint256 _favoriteNumber) public virtual override {
9         favoriteNumber = _favoriteNumber + 5;
10     }
11 }
12
13
```



This screenshot shows the Remix IDE interface with the 'StorageFactory.sol' file selected in the 'FILE EXPLORERS' sidebar. The main editor area displays the code for 'StorageFactory.sol'. The code includes a license identifier, a pragma statement for Solidity ^0.8.0, an import statement for 'SimpleStorage.sol', and a contract definition for 'StorageFactory'. The contract contains a public array 'simpleStorageArray' of type 'SimpleStorage'. It has two functions: 'createSimpleStorageContract', which creates a new 'SimpleStorage' instance and pushes it to the array, and 'sfStore', which calls the 'store' function of the 'SimpleStorage' instance at a specific index in the array with a given number.

```
1 // SPDX-License-Identifier: MIT
2
3 pragma solidity ^0.8.0;
4
5 import "./SimpleStorage.sol";
6
7 contract StorageFactory {
8
9     SimpleStorage[] public simpleStorageArray;
10
11     function createSimpleStorageContract() public {
12         SimpleStorage simpleStorage = new SimpleStorage();
13         simpleStorageArray.push(simpleStorage);
14     }
15
16     function sfStore(uint256 _simpleStorageIndex, uint256 _simpleStorageNumber) public {
17         // Address
18         // ABI
19         // SimpleStorage(address(simpleStorageArray[_simpleStorageIndex])).store(_simpleStorageNumber);
20         simpleStorageArray[_simpleStorageIndex].store(_simpleStorageNumber);
21     }
22
23 }
```