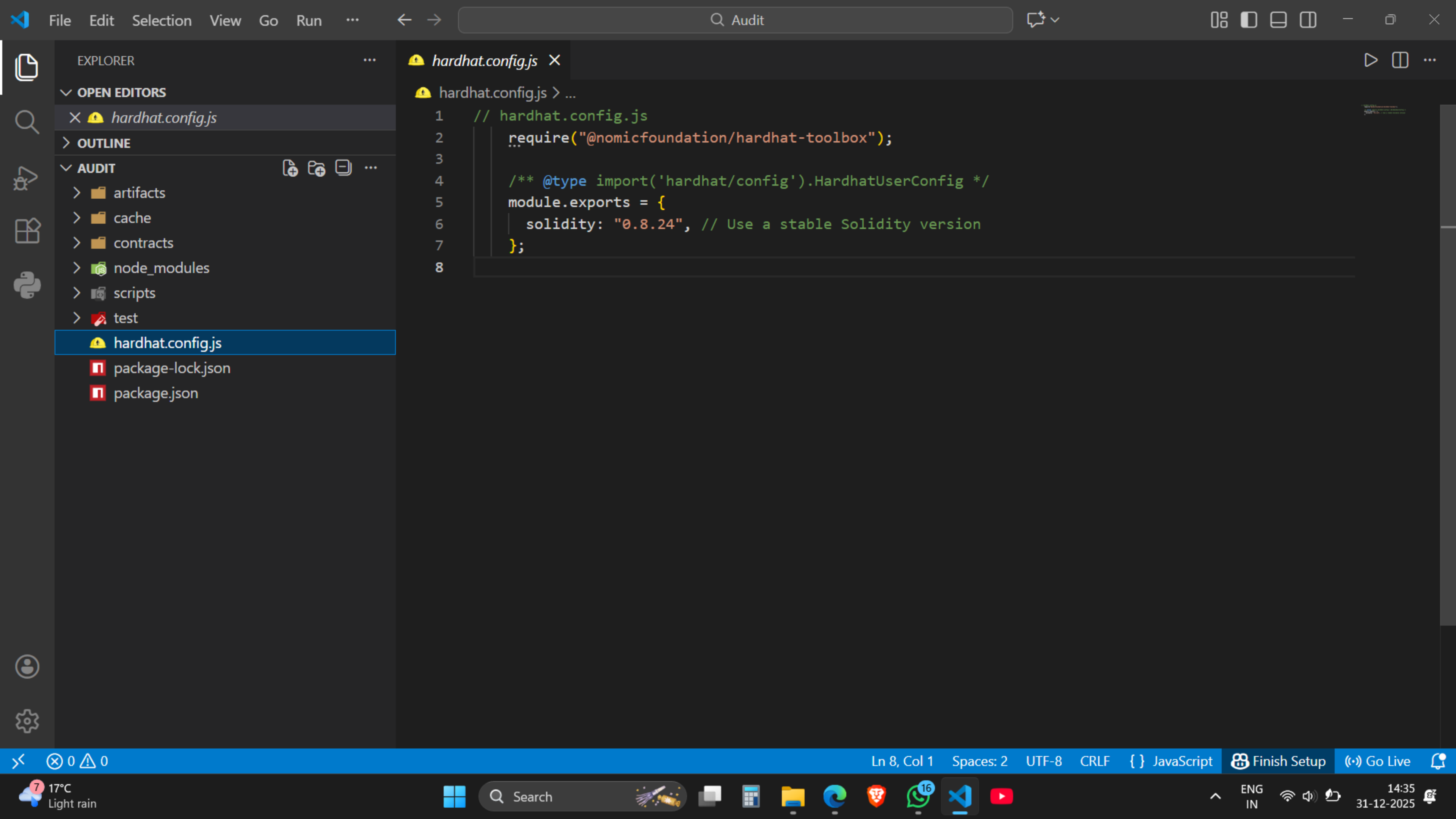


The screenshot displays the Visual Studio Code interface with the Explorer sidebar on the left and the Editor window on the right. The Explorer sidebar shows the project structure with folders like 'artifacts', 'cache', 'contracts', 'node_modules', 'scripts', and 'test'. The 'package.json' file is selected and open in the Editor window. The package.json file contains the following content:

```
{
  "name": "audit",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": "",
  "devDependencies": {
    "@nomicfoundation/hardhat-chai-matchers": "^2.1.0",
    "@nomicfoundation/hardhat-ethers": "^3.1.3",
    "@nomicfoundation/hardhat-ignition": "^0.15.16",
    "@nomicfoundation/hardhat-ignition-ethers": "^0.15.17",
    "@nomicfoundation/hardhat-network-helpers": "^1.1.2",
    "@nomicfoundation/hardhat-toolbox": "^5.0.0",
    "@nomicfoundation/hardhat-verify": "^2.1.3",
    "@nomicfoundation/ignition-core": "^0.15.15",
    "@typechain/ethers-v6": "^0.5.1",
    "@typechain/hardhat": "^9.1.0",
    "@types/chai": "^4.3.20",
    "@types/mocha": "^10.0.10",
    "chai": "^4.5.0",
    "ethers": "^6.16.0",
    "hardhat": "^2.22.10",
    "hardhat-gas-reporter": "^1.0.10",
    "solidity-coverage": "^0.8.17",
```



The screenshot shows the VS Code interface with the Explorer sidebar on the left. The Explorer sidebar displays the project structure, including folders like `artifacts`, `cache`, `contracts`, `node_modules`, `scripts`, and `test`. The `Counter.sol` file is selected in the Explorer sidebar. The main editor area shows the content of `Counter.sol`, which is a Solidity contract. The code is as follows:

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.24;
3
4 contract Counter {
5     uint256 public count;
6
7     event Incremented(uint256 newCount);
8     event Decremented(uint256 newCount);
9
10    function increment() public {
11        count += 1;
12        emit Incremented(count);
13    }
14
15    function decrement() public {
16        require(count > 0, "Counter: cannot decrement below zero");
17        count -= 1;
18        emit Decremented(count);
19    }
20
21    function getCount() public view returns (uint256) {
22        return count;
23    }
24 }
```

The screenshot shows the VS Code Explorer sidebar on a dark theme. The sidebar is titled 'EXPLORER' and contains a list of files and folders. The 'test' folder is expanded, showing its contents. The 'Counter.js' file is selected and highlighted in blue. The file structure is as follows:

- EXPLORER
 - OPEN EDITORS
 - JS Counter.js test
 - OUTLINE
 - AUDIT
 - artifacts
 - cache
 - contracts
 - Counter.sol
 - node_modules
 - scripts
 - test
 - JS Counter.js
 - hardhat.config.js
 - package-lock.json
 - package.json

Hardhat Installation And Counter Example

This document provides a minimal, working example of a Hardhat project with a simple Solidity `Counter` contract and a JavaScript test file that runs without errors.

The setup is configured to use stable versions of Hardhat and its dependencies to ensure compatibility and avoid common installation issues encountered in the sandbox environment.

Setup and Installation

Follow these steps to set up the project environment:

1. Create the project directory and initialize Node.js:**

```
mkdir hardhat-counter
cd hardhat-counter
npm init -y
```

2. Install Hardhat and dependencies:

Due to recent changes in Hardhat's dependency tree, specific stable versions are used, and the `--legacy-peer-deps` flag is required for a clean install.

```
npm install --save-dev hardhat@2.22.10
@nomicfoundation/hardhat-toolbox@5.0.0 --legacy-peer-deps
```

```
# Install remaining dependencies for the toolbox
```

```
npm install --save-dev "@nomicfoundation/hardhat-chai-matchers@^2.0.0"
"@nomicfoundation/hardhat-ethers@^3.0.0"
"@nomicfoundation/hardhat-ignition-ethers@^0.15.0"
"@nomicfoundation/hardhat-network-helpers@^1.0.0"
"@nomicfoundation/hardhat-verify@^2.0.0" "@typechain/ethers-v6@^0.5.0"
"@typechain/hardhat@^9.0.0" "@types/chai@^4.2.0" "@types/mocha@>=9.1.0"
"chai@^4.2.0" "ethers@^6.4.0" "hardhat-gas-reporter@^1.0.8"
"solidity-coverage@^0.8.1" "ts-node@>=8.0.0" "typechain@^8.3.0"
"typescript@>=4.5.0" "@nomicfoundation/hardhat-ignition@^0.15.16"
"@nomicfoundation/ignition-core@^0.15.15" --legacy-peer-deps
```

3. Create the Hardhat Configuration File (`hardhat.config.js`):**

This file uses the CommonJS format for maximum compatibility.

```
\\\\"\\\"javascript
// hardhat.config.js
```

```

require("@nomicfoundation/hardhat-toolbox");

/** @type import('hardhat/config').HardhatUserConfig */
module.exports = {
  solidity: "0.8.24", // Use a stable Solidity version
};
\`\`\`

```

4. Create the necessary directories:

```

mkdir - contracts
mkdir - test
mkdir - scripts

```

```
## Contract Code (`contracts/Counter.sol`)
```

This is a simple Solidity contract that allows incrementing and decrementing a counter, with a check to prevent decrementing below zero.

```

\`\`\`solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.24;

contract Counter {
    uint256 public count;

    event Incremented(uint256 newCount);
    event Decrementd(uint256 newCount);

    function increment() public {
        count += 1;
        emit Incremented(count);
    }

    function decrement() public {
        require(count > 0, "Counter: cannot decrement below zero");
        count -= 1;
        emit Decrementd(count);
    }

    function getCount() public view returns (uint256) {
        return count;
    }
}

```



```
}  
\\\`
```

```
## Test Code (`test/Counter.js`)
```

This JavaScript test file uses Hardhat's built-in environment, Ethers.js, and Chai for assertions. It tests all functionalities of the `Counter` contract, including deployment, state changes, error handling, and event emission.

```
\\\`javascript
```

```
const { expect } = require("chai");  
const { ethers } = require("hardhat");
```

```
describe("Counter Contract", function () {  
  let counter;
```

```
  beforeEach(async function () {  
    // Deploy the contract before each test  
    const Counter = await ethers.getContractFactory("Counter");  
    counter = await Counter.deploy();  
  });
```

```
  it("Should initialize with a count of 0", async function () {  
    expect(await counter.getCount()).to.equal(0);  
  });
```

```
  it("Should increment the count", async function () {  
    await counter.increment();  
    expect(await counter.getCount()).to.equal(1);  
  });
```

```
  it("Should decrement the count", async function () {  
    await counter.increment();  
    await counter.decrement();  
    expect(await counter.getCount()).to.equal(0);  
  });
```

```
  it("Should fail when decrementing below zero", async function () {  
    await expect(counter.decrement()).to.be.revertedWith("Counter: cannot  
decrement below zero");  
  });
```

```
  it("Should emit an event on increment", async function () {  
    await expect(counter.increment())  
      .to.emit(counter, "Incremented")  
      .withArgs(1);  
  });  
});
```

```
\\\`
```


Running the Tests

To compile the tests

```
\\`bash
npx hardhat compile
\\`
```

Running the Tests

To run the tests and see the successful output:

```
\\`bash
npx hardhat test
\\`
```

Expected Output

The output confirms that all 5 tests passed successfully:

```
\\`
Counter Contract
  ✓ Should initialize with a count of 0
  ✓ Should increment the count
  ✓ Should decrement the count
  ✓ Should fail when decrementing below zero
  ✓ Should emit an event on increment
5 passing (XXXms)
\\`
```