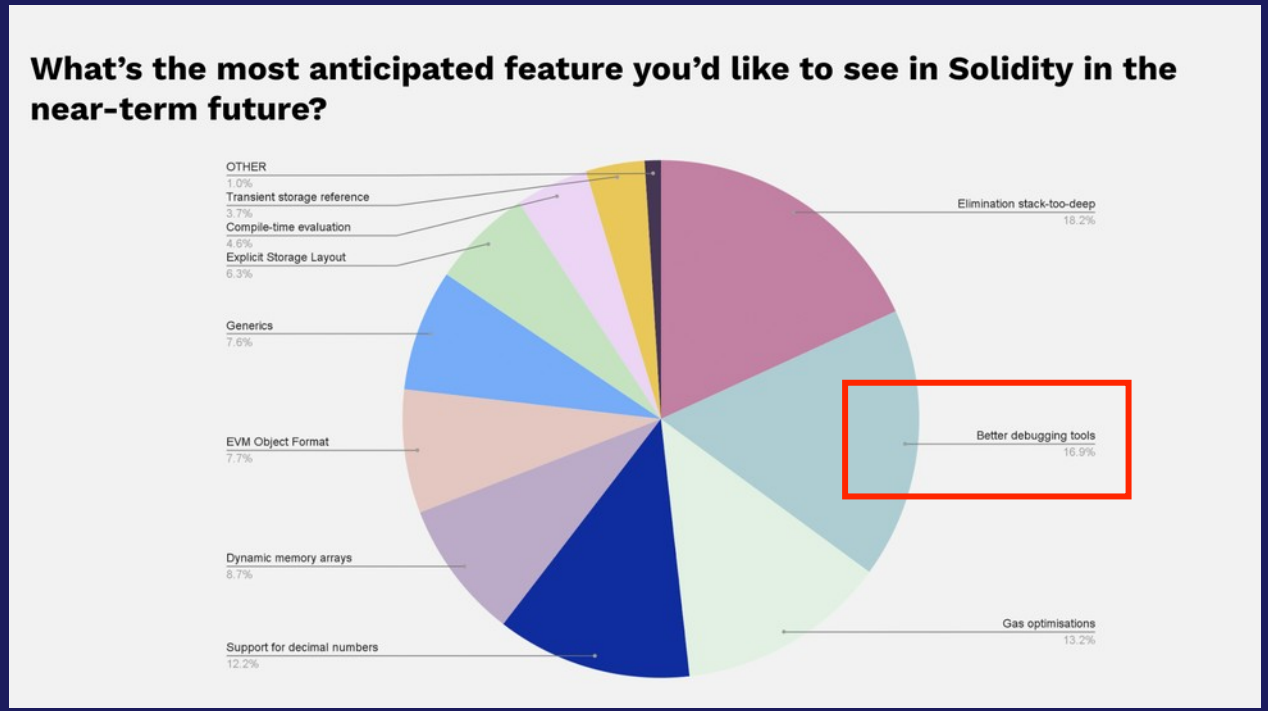
28 November 2025

debugging solidity

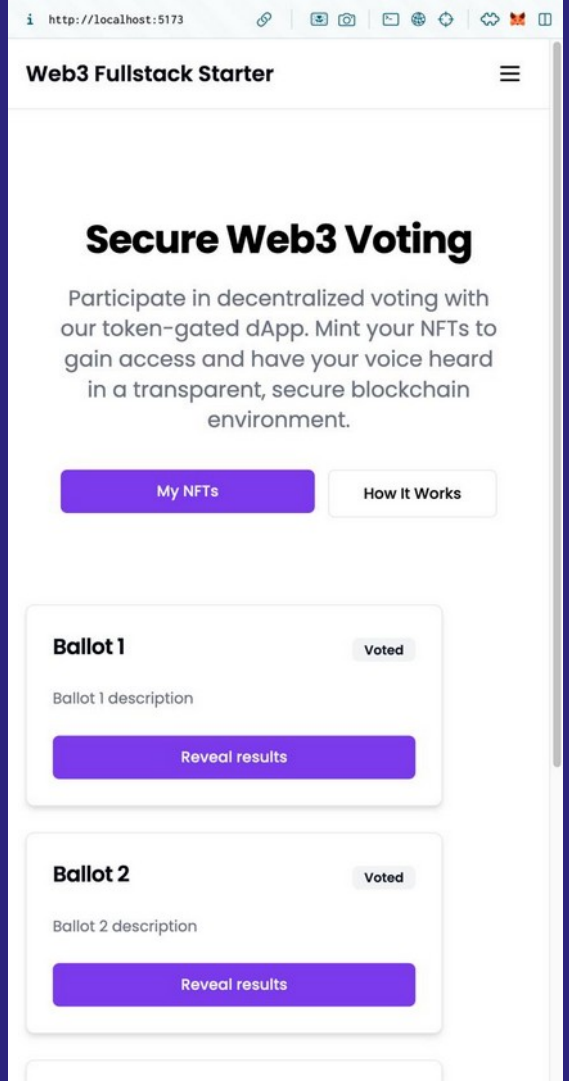
roman from walnut.dev

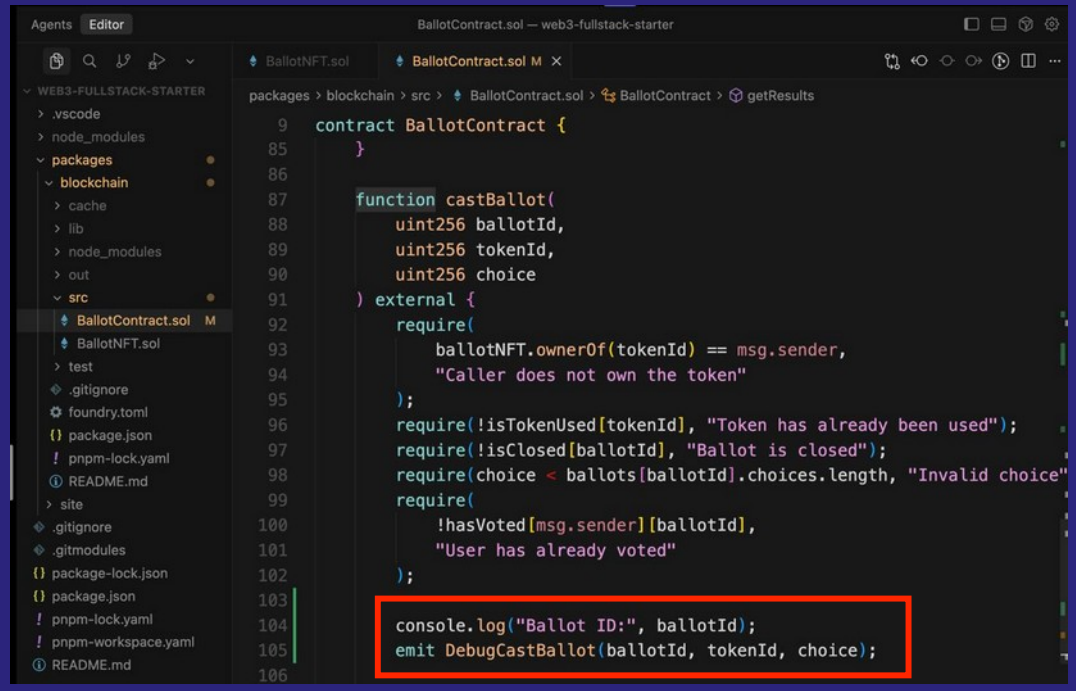


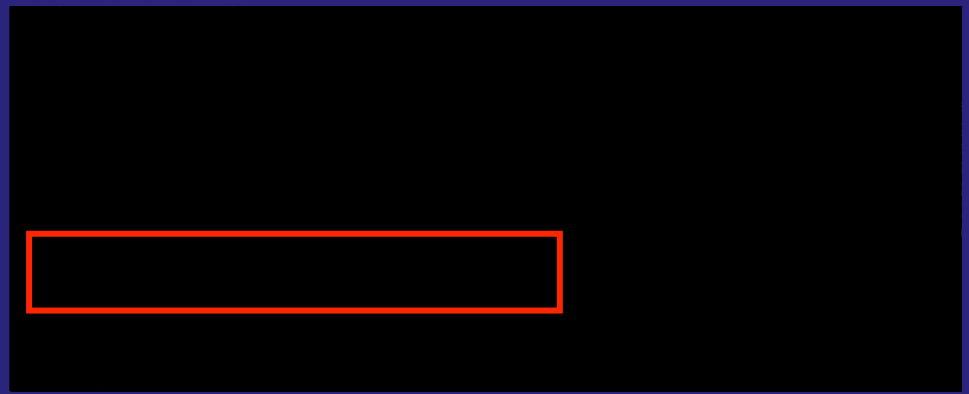
Source: https://www.soliditylang.org/blog/2025/04/25/solidity-developer-survey-2024-results/

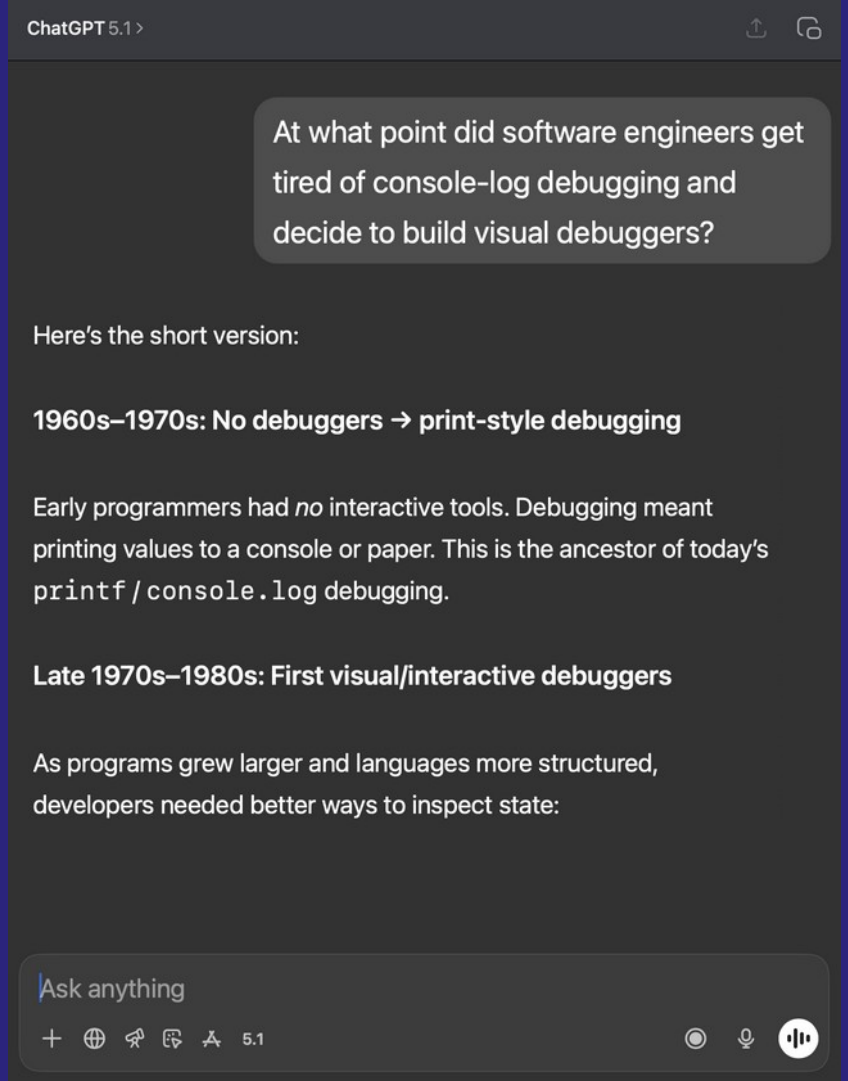
Agenda

1. Demo of the best Solidity Debugger with most amazing UX 💅
2. Challenges
3. Proposed Improvements


http://localhost:5173
Web3 Fullstack Starter
Secure Web3 Voting
Participate in decentralized voting with our token-gated dApp. Mint your NFTs to gain access and have your voice heard in a transparent, secure blockchain environment.
My NFTs
How It Works
Ballot 1
Voted
Ballot 1 description
Reveal results
Ballot 2
Voted
Ballot 2 description
Reveal results

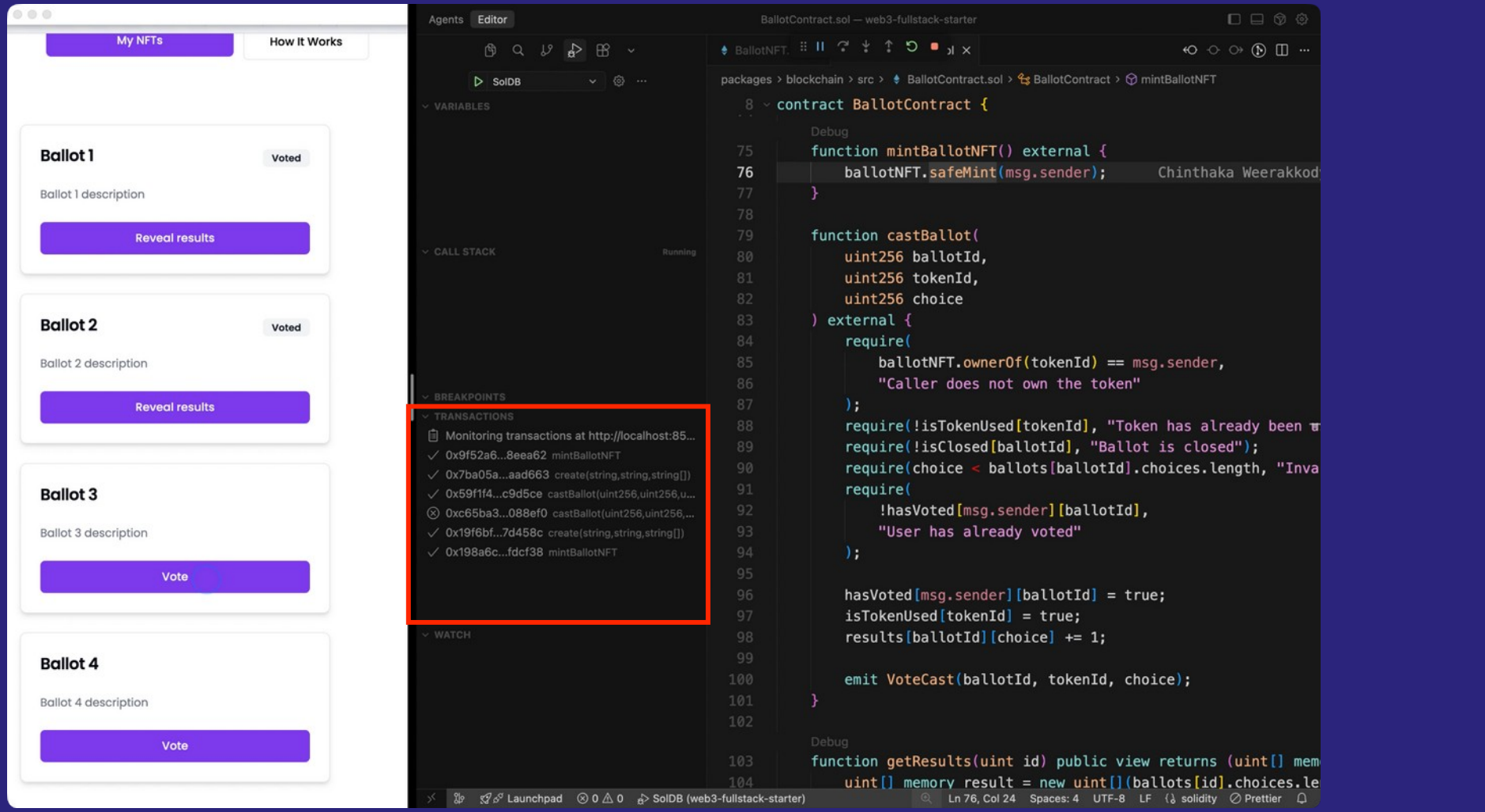

Agents
Editor
BallotContract.sol — web3-fullstack-starter
BallotNFT.sol
BallotContract.sol M
packages › blockchain › src › BallotContract.sol › BallotContract › getResults
WEB3-FULLSTACK-STARTER
.vscode
node_modules
packages
blockchain
cache
lib
node_modules
out
src
BallotContract.sol M
BallotNFT.sol
test
.gitignore
foundry.toml
package.json
pnpm-lock.yaml
README.md
site
.gitignore
.gitmodules
package-lock.json
package.json
pnpm-lock.yaml
pnpm-workspace.yaml
README.md
9 contract BallotContract {
85 }
86
87 function castBallot(
88 uint256 ballotId,
89 uint256 tokenId,
90 uint256 choice
91) external {
92 require(
93 ballotNFT.ownerOf(tokenId) == msg.sender,
94 "Caller does not own the token"
95);
96 require(!isTokenUsed[tokenId], "Token has already been used");
97 require(!isClosed[ballotId], "Ballot is closed");
98 require(choice < ballots[ballotId].choices.length, "Invalid choice"
99 require(
100 !hasVoted[msg.sender][ballotId],
101 "User has already voted"
102);
103
104 console.log("Ballot ID:", ballotId);
105 emit DebugCastBallot(ballotId, tokenId, choice);
106

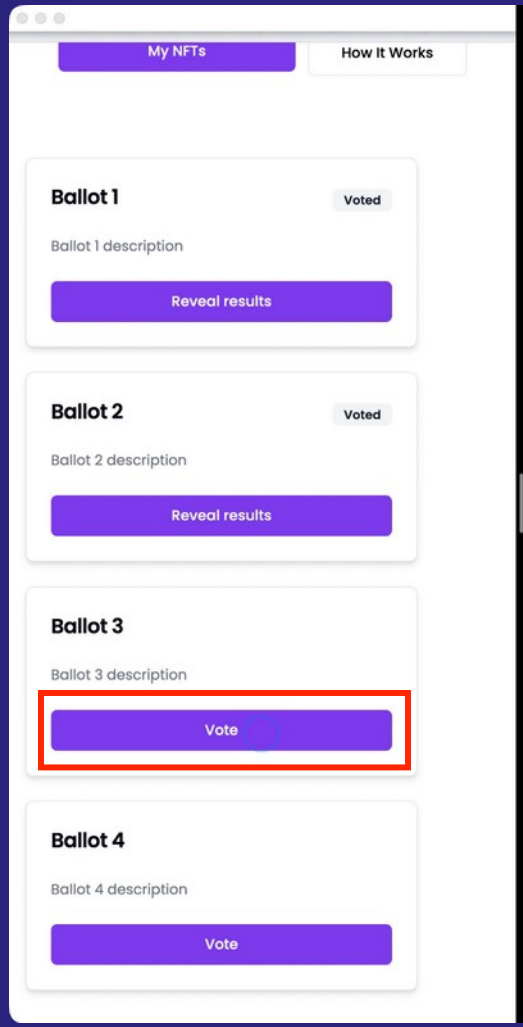


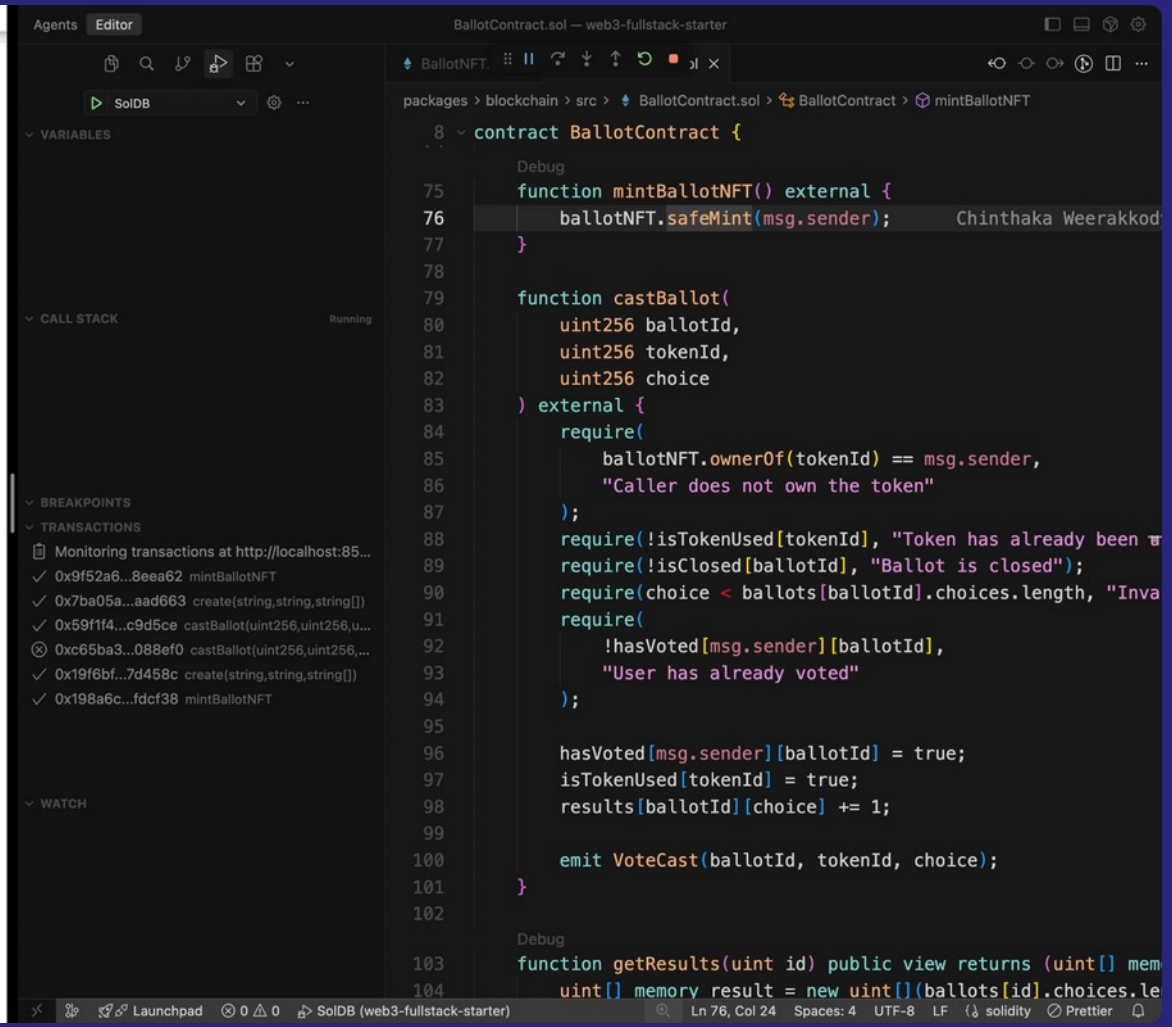

ChatGPT 5.1 >
At what point did software engineers get tired of console-log debugging and decide to build visual debuggers?
Here's the short version:
1960s–1970s: No debuggers → print-style debugging
Early programmers had no interactive tools. Debugging meant printing values to a console or paper. This is the ancestor of today's printf / console.log debugging.
Late 1970s–1980s: First visual/interactive debuggers
As programs grew larger and languages more structured, developers needed better ways to inspect state:
Ask anything
5.1

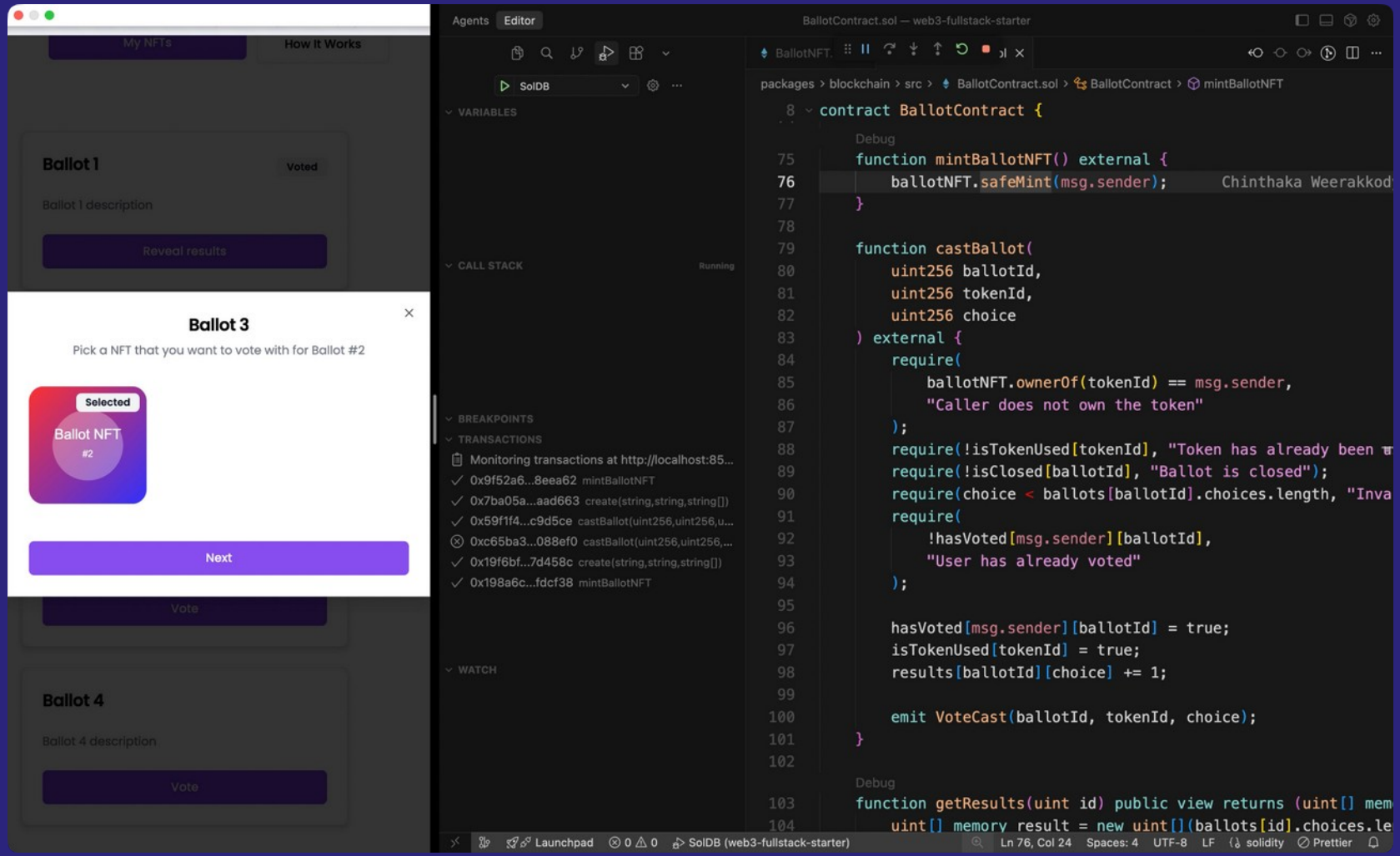
Demo:

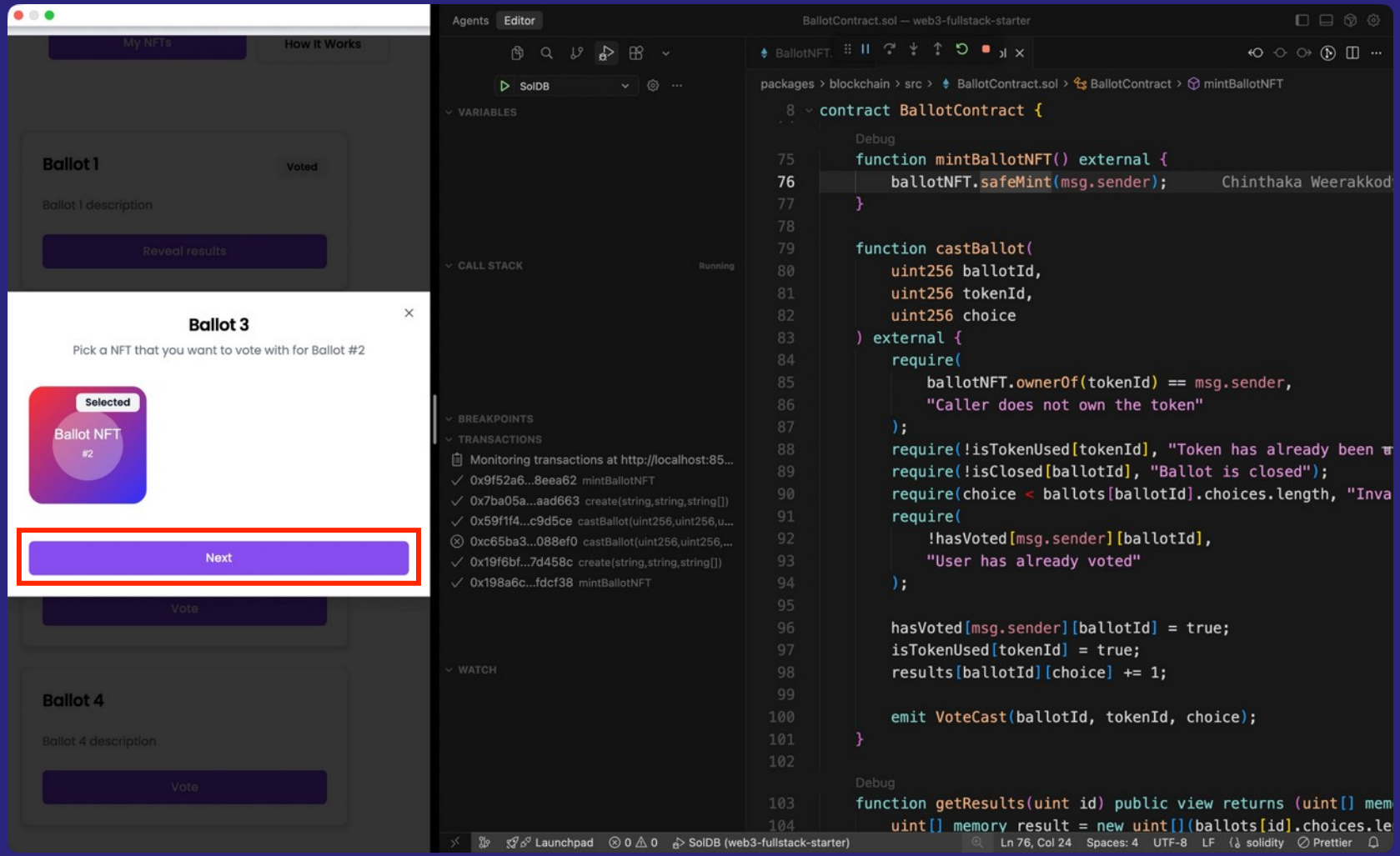
The best Solidity Debugger with most amazing UX 

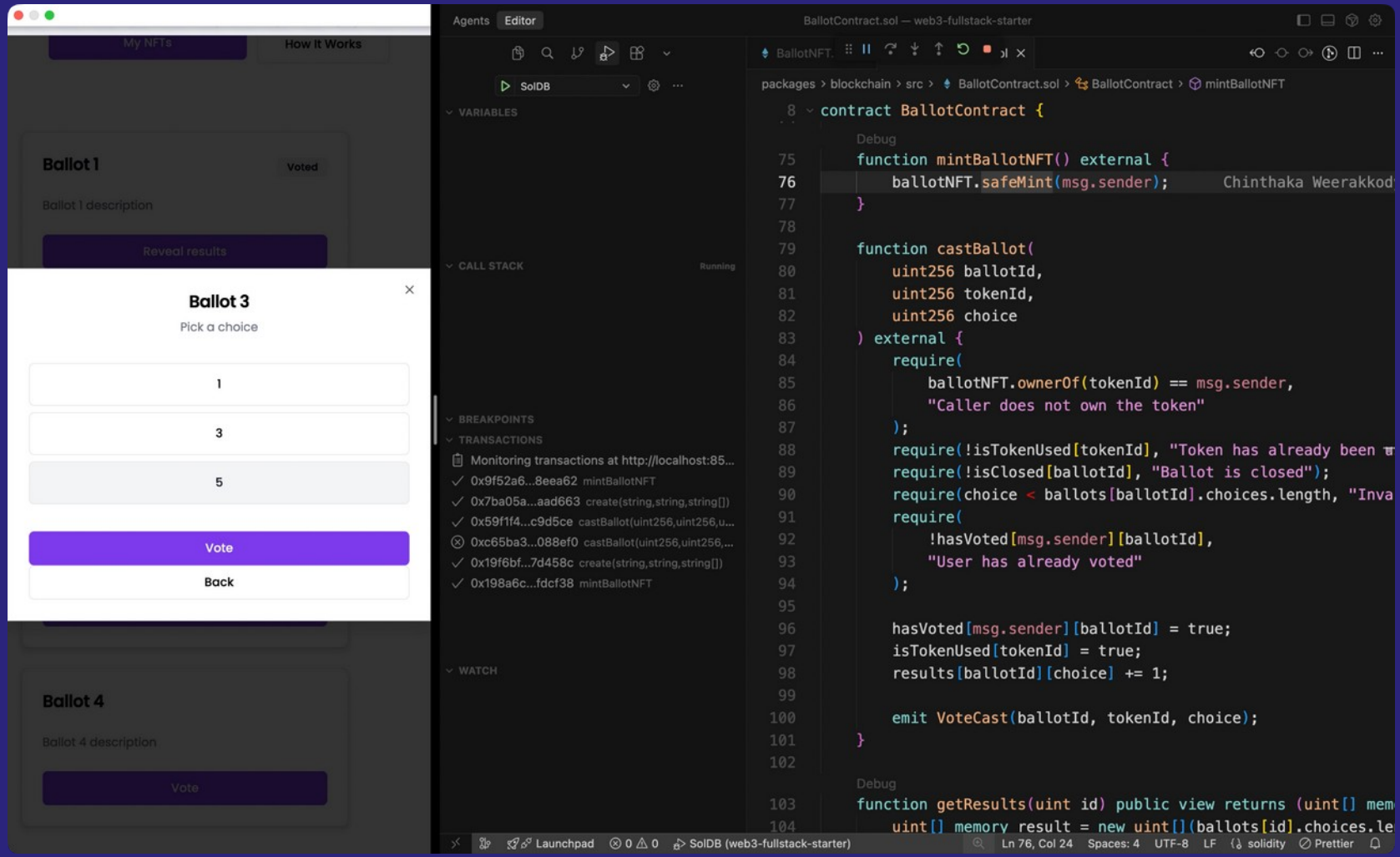

My NFTs
How It Works
Ballot 1
Voted
Ballot 1 description
Reveal results
Ballot 2
Voted
Ballot 2 description
Reveal results
Ballot 3
Ballot 3 description
Vote
Ballot 4
Ballot 4 description
Vote
Agents
Editor
BallotContract.sol — web3-fullstack-starter
SolDB
VARIABLES
CALL STACK
Running
BREAKPOINTS
TRANSACTIONS
Monitoring transactions at http://localhost:85...
0x9f52a6...8eea62 mintBallotNFT
0x7ba05a...aad663 create(string,string,string[])
0x59f1f4...c9d5ce castBallot(uint256,uint256,u...
0xc65ba3...088ef0 castBallot(uint256,uint256,...
0x19f6bf...7d458c create(string,string,string[])
0x198a6c...fdcf38 mintBallotNFT
WATCH
packages › blockchain › src › BallotContract.sol › BallotContract › mintBallotNFT
contract BallotContract {
Debug
function mintBallotNFT() external {
ballotNFT.safeMint(msg.sender);
}
function castBallot(
uint256 ballotId,
uint256 tokenId,
uint256 choice
) external {
require(
ballotNFT.ownerOf(tokenId) == msg.sender,
"Caller does not own the token"
);
require(!isTokenUsed[tokenId], "Token has already been
require(!isClosed[ballotId], "Ballot is closed");
require(choice < ballots[ballotId].choices.length, "Inva
require(
!hasVoted[msg.sender][ballotId],
"User has already voted"
);
hasVoted[msg.sender][ballotId] = true;
isTokenUsed[tokenId] = true;
results[ballotId][choice] += 1;
emit VoteCast(ballotId, tokenId, choice);
}
Debug
function getResults(uint id) public view returns (uint[] mem
uint[] memory result = new uint[](ballots[id].choices.le
Launchpad
SolDB (web3-fullstack-starter)
Ln 76, Col 24
Spaces: 4
UTF-8
LF
solidity
Prettier

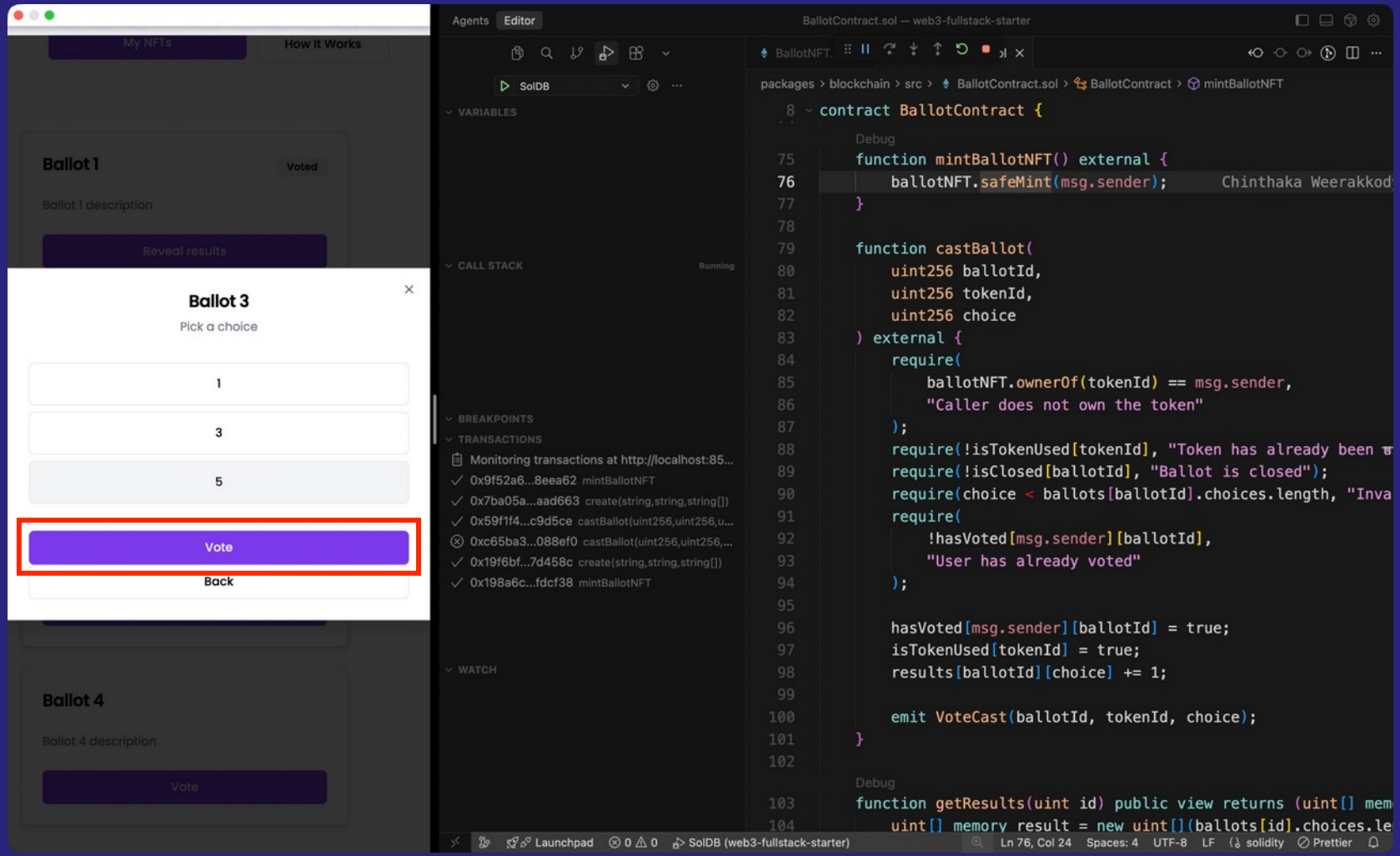

My NFTs
How It Works
Ballot 1
Voted
Ballot 1 description
Reveal results
Ballot 2
Voted
Ballot 2 description
Reveal results
Ballot 3
Ballot 3 description
Vote
Ballot 4
Ballot 4 description
Vote

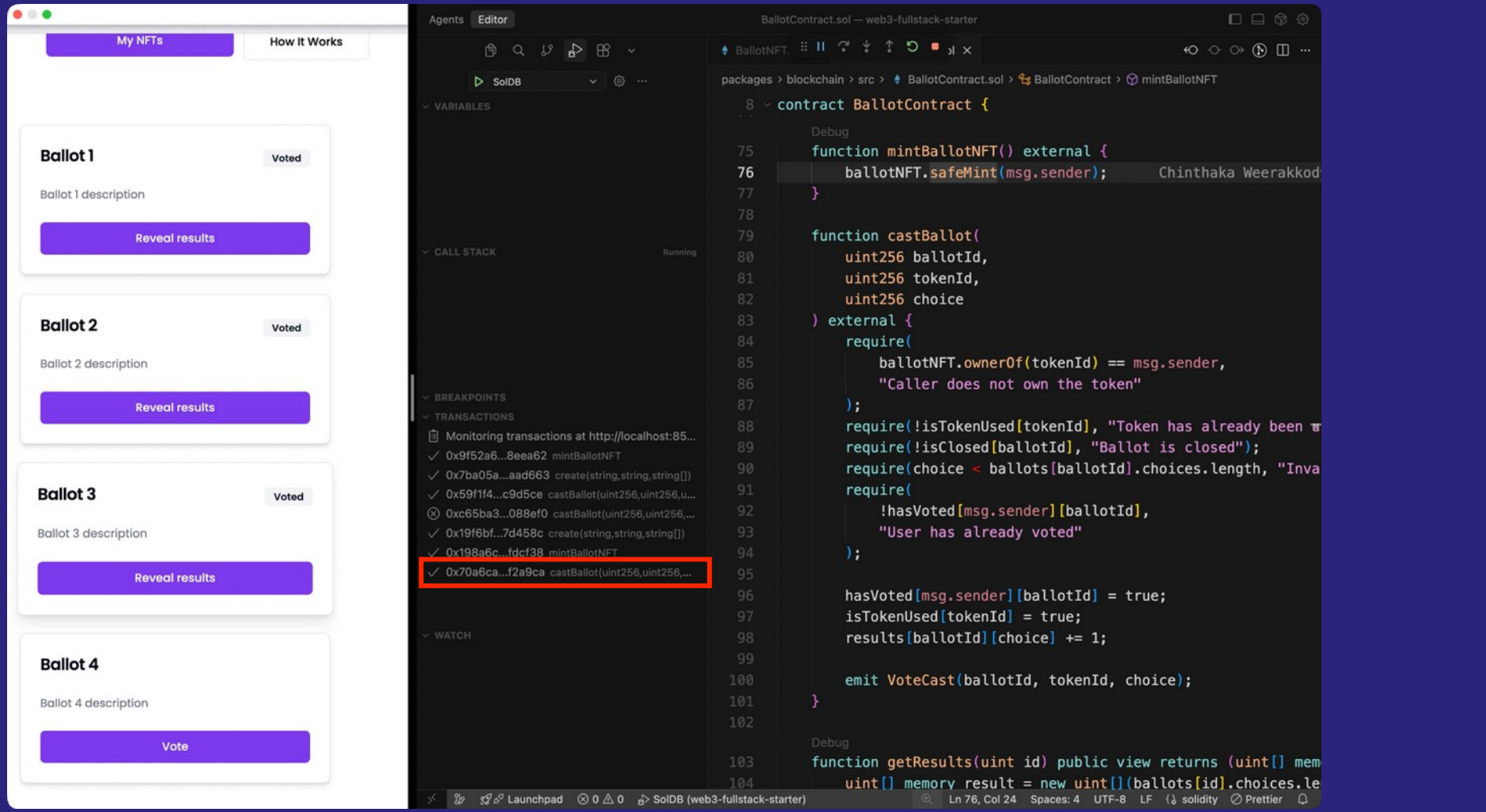

Agents
Editor
BallotContract.sol — web3-fullstack-starter
SolDB
VARIABLES
CALL STACK
Running
BREAKPOINTS
TRANSACTIONS
Monitoring transactions at http://localhost:85...
0x9f52a6...8eea62 mintBallotNFT
0x7ba05a...aad663 create(string,string,string[])
0x59f1f4...c9d5ce castBallot(uint256,uint256,u...
0xc65ba3...088ef0 castBallot(uint256,uint256,...
0x19f6bf...7d458c create(string,string,string[])
0x198a6c...fdcf38 mintBallotNFT
WATCH
packages › blockchain › src › BallotContract.sol › BallotContract › mintBallotNFT
contract BallotContract {
Debug
function mintBallotNFT() external {
ballotNFT.safeMint(msg.sender);
}
function castBallot(
uint256 ballotId,
uint256 tokenId,
uint256 choice
) external {
require(
ballotNFT.ownerOf(tokenId) == msg.sender,
"Caller does not own the token"
);
require(!isTokenUsed[tokenId], "Token has already been
require(!isClosed[ballotId], "Ballot is closed");
require(choice < ballots[ballotId].choices.length, "Inva
require(
!hasVoted[msg.sender][ballotId],
"User has already voted"
);
hasVoted[msg.sender][ballotId] = true;
isTokenUsed[tokenId] = true;
results[ballotId][choice] += 1;
emit VoteCast(ballotId, tokenId, choice);
}
Debug
function getResults(uint id) public view returns (uint[] mem
uint[] memory result = new uint[](ballots[id].choices.le
Launchpad
SolDB (web3-fullstack-starter)
Ln 76, Col 24
Spaces: 4
UTF-8
LF
solidity
Prettier

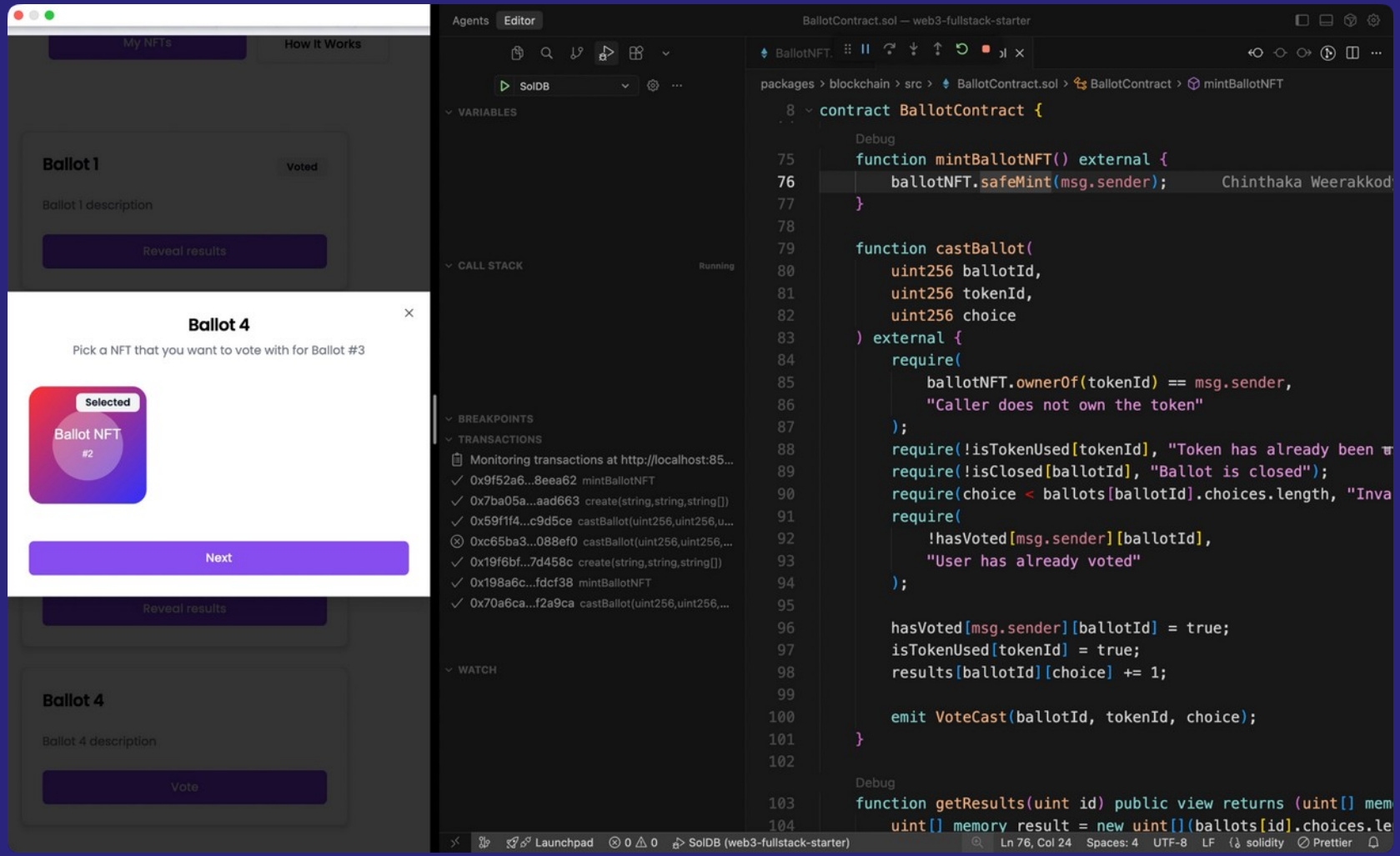

My NFTs
How It Works
Ballot 1
Voted
Ballot 1 description
Reveal results
Ballot 3
Pick a NFT that you want to vote with for Ballot #2
Selected
Ballot NFT
#2
Next
Vote
Ballot 4
Ballot 4 description
Vote
Agents
Editor
BallotContract.sol — web3-fullstack-starter
SolDB
VARIABLES
CALL STACK
Running
BREAKPOINTS
TRANSACTIONS
Monitoring transactions at http://localhost:85...
0x9f52a6...8eea62 mintBallotNFT
0x7ba05a...aad663 create(string,string,string[])
0x59f1f4...c9d5ce castBallot(uint256,uint256,u...
0xc65ba3...088ef0 castBallot(uint256,uint256,...
0x19f6bf...7d458c create(string,string,string[])
0x198a6c...fdcf38 mintBallotNFT
WATCH
packages › blockchain › src › BallotContract.sol › BallotContract › mintBallotNFT
contract BallotContract {
Debug
function mintBallotNFT() external {
ballotNFT.safeMint(msg.sender);
Chinthaka Weerakkod
}
function castBallot(
uint256 ballotId,
uint256 tokenId,
uint256 choice
) external {
require(
ballotNFT.ownerOf(tokenId) == msg.sender,
"Caller does not own the token"
);
require(!isTokenUsed[tokenId], "Token has already been
require(!isClosed[ballotId], "Ballot is closed");
require(choice < ballots[ballotId].choices.length, "Inva
require(
!hasVoted[msg.sender][ballotId],
"User has already voted"
);
hasVoted[msg.sender][ballotId] = true;
isTokenUsed[tokenId] = true;
results[ballotId][choice] += 1;
emit VoteCast(ballotId, tokenId, choice);
}
Debug
function getResults(uint id) public view returns (uint[] mem
uint[] memory result = new uint[](ballots[id].choices.le
Launchpad
0 0
SolDB (web3-fullstack-starter)
Ln 76, Col 24
Spaces: 4
UTF-8
LF
solidity
Prettier

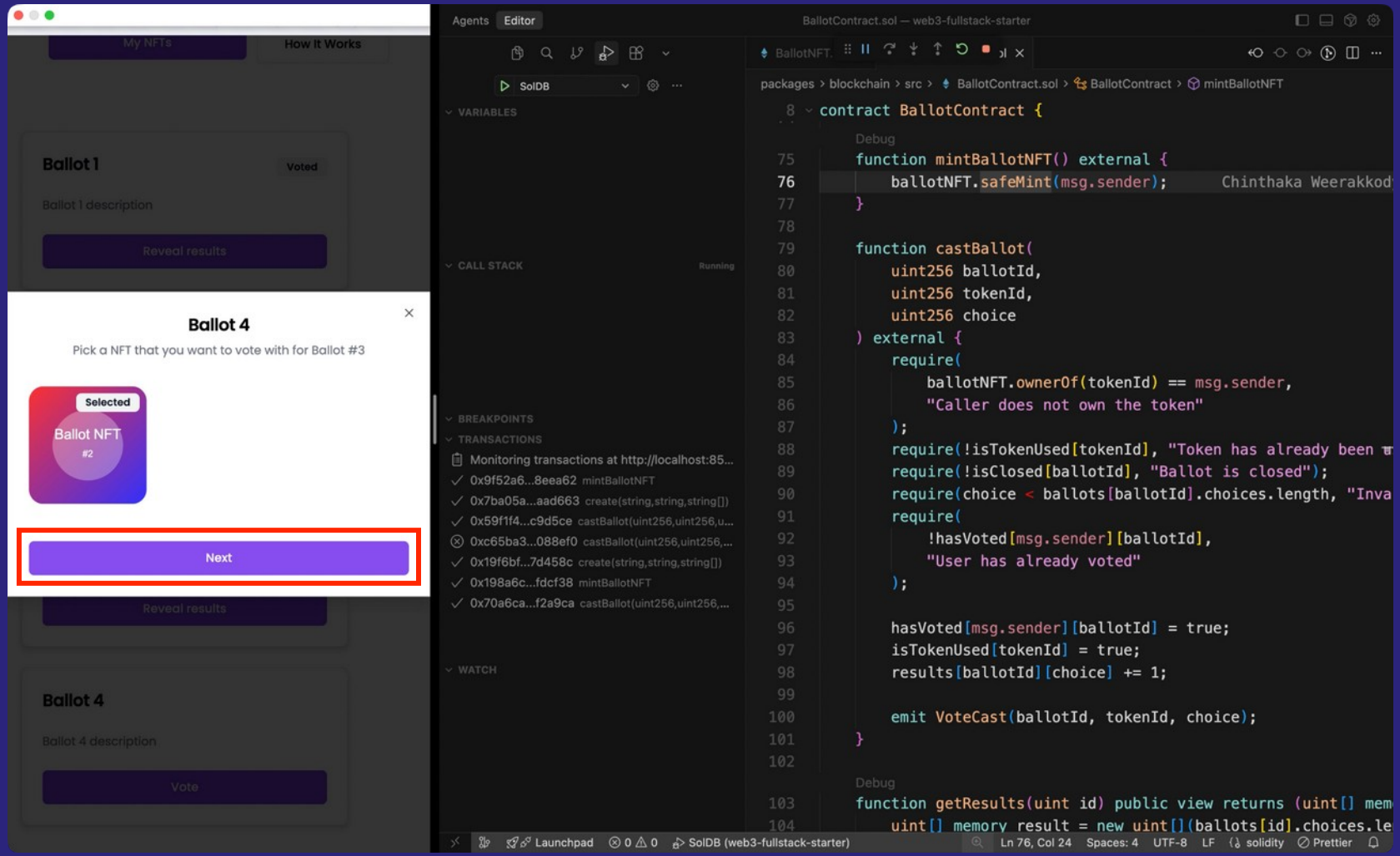

My NFTs
How It Works
Ballot 1
Voted
Ballot 1 description
Reveal results
Ballot 3
Pick a NFT that you want to vote with for Ballot #2
Selected
Ballot NFT
#2
Next
Vote
Ballot 4
Ballot 4 description
Vote
Agents
Editor
BallotContract.sol — web3-fullstack-starter
SolDB
VARIABLES
CALL STACK
Running
BREAKPOINTS
TRANSACTIONS
Monitoring transactions at http://localhost:85...
0x9f52a6...8eea62 mintBallotNFT
0x7ba05a...aad663 create(string,string,string[])
0x59f1f4...c9d5ce castBallot(uint256,uint256,u...
0xc65ba3...088ef0 castBallot(uint256,uint256,...
0x19f6bf...7d458c create(string,string,string[])
0x198a6c...fdcf38 mintBallotNFT
WATCH
packages › blockchain › src › BallotContract.sol › BallotContract › mintBallotNFT
contract BallotContract {
Debug
function mintBallotNFT() external {
ballotNFT.safeMint(msg.sender);
Chinthaka Weerakkod
}
function castBallot(
uint256 ballotId,
uint256 tokenId,
uint256 choice
) external {
require(
ballotNFT.ownerOf(tokenId) == msg.sender,
"Caller does not own the token"
);
require(!isTokenUsed[tokenId], "Token has already been
require(!isClosed[ballotId], "Ballot is closed");
require(choice < ballots[ballotId].choices.length, "Inva
require(
!hasVoted[msg.sender][ballotId],
"User has already voted"
);
hasVoted[msg.sender][ballotId] = true;
isTokenUsed[tokenId] = true;
results[ballotId][choice] += 1;
emit VoteCast(ballotId, tokenId, choice);
}
Debug
function getResults(uint id) public view returns (uint[] mem
uint[] memory result = new uint[](ballots[id].choices.le
Launchpad
0 0
SolDB (web3-fullstack-starter)
Ln 76, Col 24
Spaces: 4
UTF-8
LF
solidity
Prettier

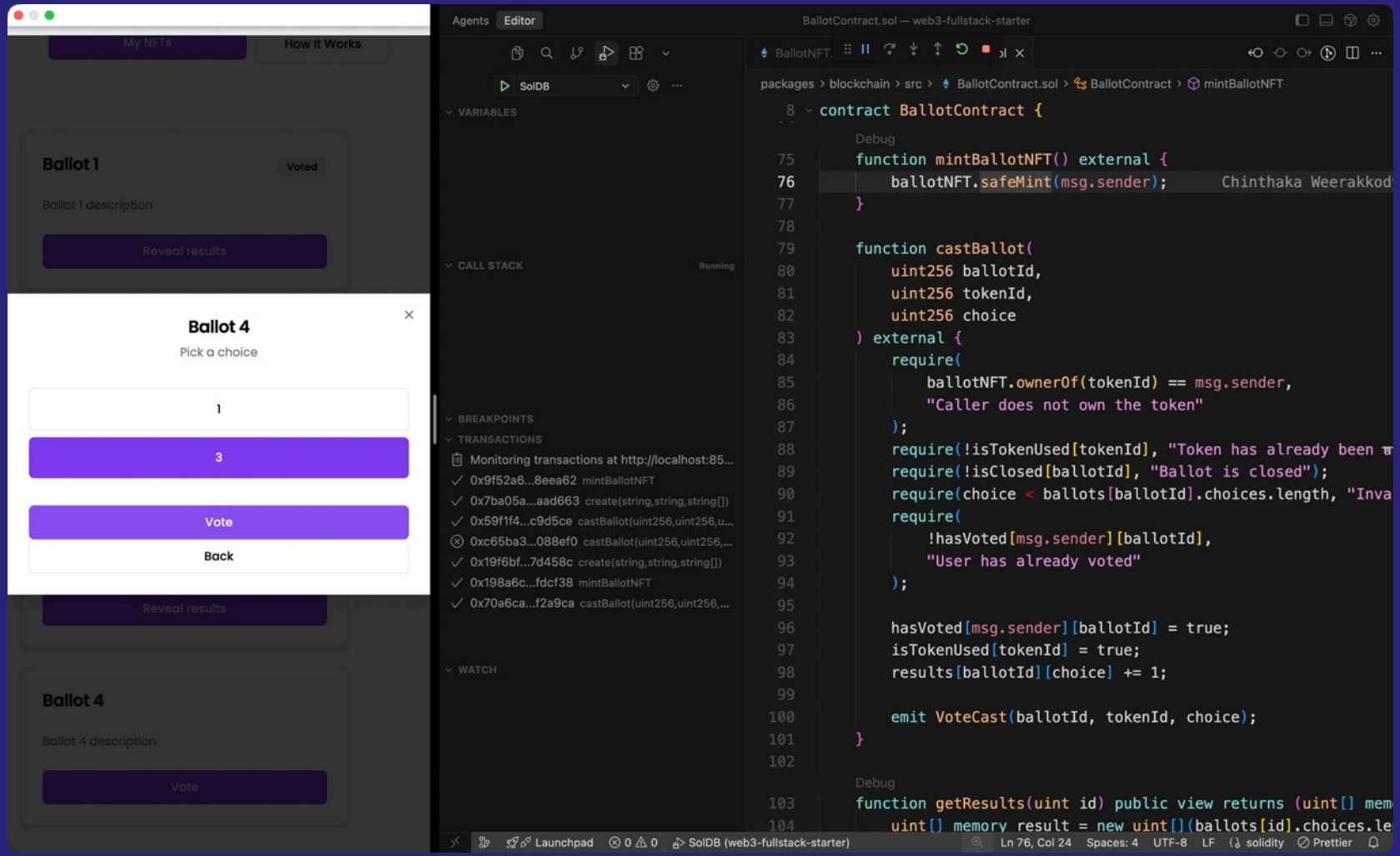

My NFTs
How It Works
Ballot 1
Voted
Ballot 1 description
Reveal results
Ballot 3
Pick a choice
1
3
5
Vote
Back
Ballot 4
Ballot 4 description
Vote
Agents
Editor
BallotContract.sol — web3-fullstack-starter
SolDB
VARIABLES
CALL STACK
Running
BREAKPOINTS
TRANSACTIONS
Monitoring transactions at http://localhost:85...
0x9f52a6...8eea62 mintBallotNFT
0x7ba05a...aad663 create(string,string,string[])
0x59f1f4...c9d5ce castBallot(uint256,uint256,u...
0xc65ba3...088ef0 castBallot(uint256,uint256,...
0x19f6bf...7d458c create(string,string,string[])
0x198a6c...fdcf38 mintBallotNFT
WATCH
packages › blockchain › src › BallotContract.sol › BallotContract › mintBallotNFT
contract BallotContract {
Debug
function mintBallotNFT() external {
ballotNFT.safeMint(msg.sender);
Chinthaka Weerakkod
}
function castBallot(
uint256 ballotId,
uint256 tokenId,
uint256 choice
) external {
require(
ballotNFT.ownerOf(tokenId) == msg.sender,
"Caller does not own the token"
);
require(!isTokenUsed[tokenId], "Token has already been
require(!isClosed[ballotId], "Ballot is closed");
require(choice < ballots[ballotId].choices.length, "Inva
require(
!hasVoted[msg.sender][ballotId],
"User has already voted"
);
hasVoted[msg.sender][ballotId] = true;
isTokenUsed[tokenId] = true;
results[ballotId][choice] += 1;
emit VoteCast(ballotId, tokenId, choice);
}
Debug
function getResults(uint id) public view returns (uint[] mem
uint[] memory result = new uint[](ballots[id].choices.le
Launchpad
0 0
SolDB (web3-fullstack-starter)
Ln 76, Col 24
Spaces: 4
UTF-8
LF
solidity
Prettier

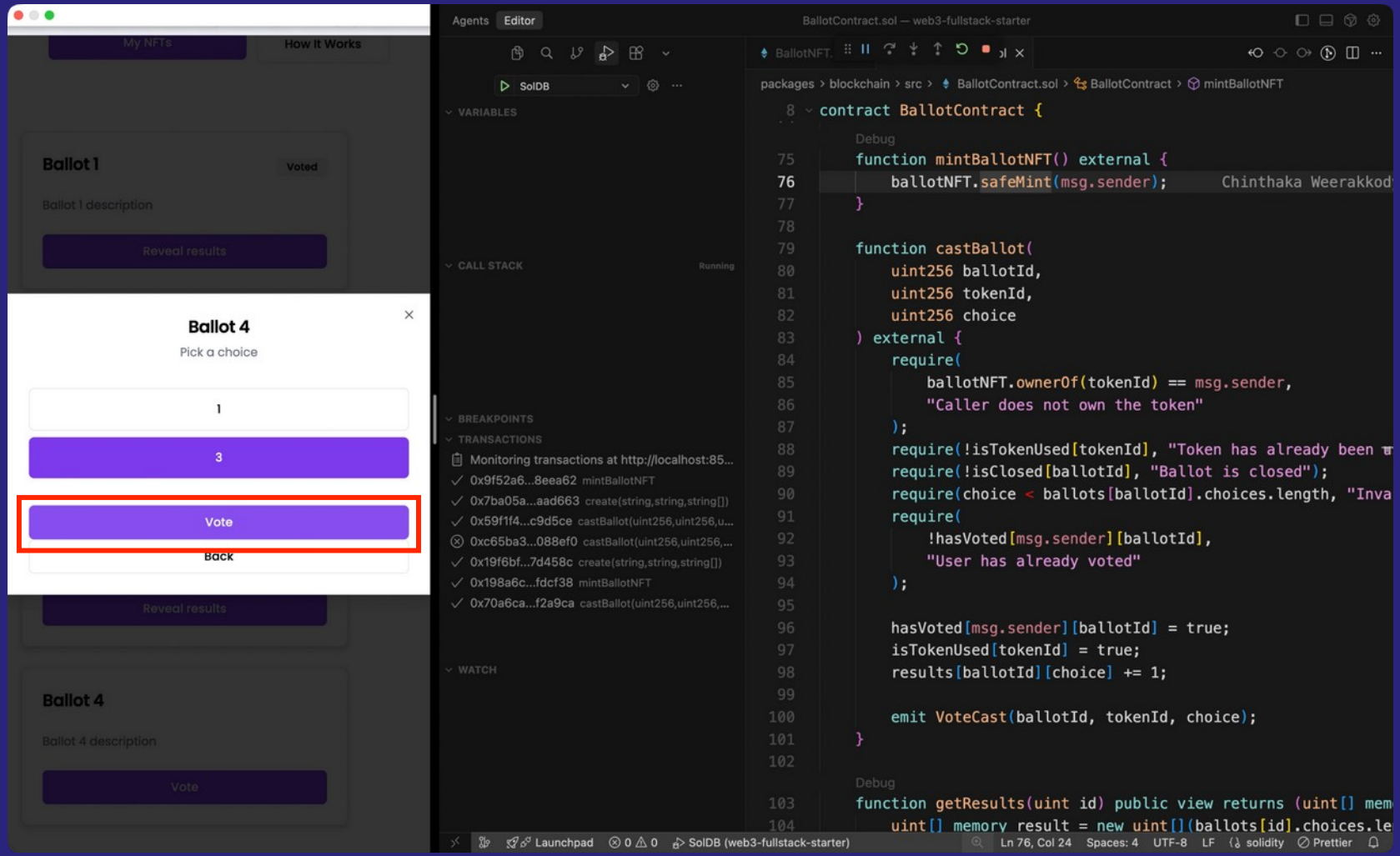

My NFTs
How It Works
Ballot 1
Voted
Ballot 1 description
Reveal results
Ballot 3
Pick a choice
1
3
5
Vote
Back
Ballot 4
Ballot 4 description
Vote
Agents
Editor
BallotContract.sol — web3-fullstack-starter
SolDB
VARIABLES
CALL STACK
Running
BREAKPOINTS
TRANSACTIONS
Monitoring transactions at http://localhost:85...
0x9f52a6...8eea62 mintBallotNFT
0x7ba05a...aad663 create(string,string,string[])
0x59f1f4...c9d5ce castBallot(uint256,uint256,u...
0xc65ba3...088ef0 castBallot(uint256,uint256,...
0x19f6bf...7d458c create(string,string,string[])
0x198a6c...fdcf38 mintBallotNFT
WATCH
packages › blockchain › src › BallotContract.sol › BallotContract › mintBallotNFT
contract BallotContract {
Debug
function mintBallotNFT() external {
ballotNFT.safeMint(msg.sender);
Chinthaka Weerakkod
}
function castBallot(
uint256 ballotId,
uint256 tokenId,
uint256 choice
) external {
require(
ballotNFT.ownerOf(tokenId) == msg.sender,
"Caller does not own the token"
);
require(!isTokenUsed[tokenId], "Token has already been
require(!isClosed[ballotId], "Ballot is closed");
require(choice < ballots[ballotId].choices.length, "Inva
require(
!hasVoted[msg.sender][ballotId],
"User has already voted"
);
hasVoted[msg.sender][ballotId] = true;
isTokenUsed[tokenId] = true;
results[ballotId][choice] += 1;
emit VoteCast(ballotId, tokenId, choice);
}
Debug
function getResults(uint id) public view returns (uint[] mem
uint[] memory result = new uint[](ballots[id].choices.le
Launchpad
0 0
SolDB (web3-fullstack-starter)
Ln 76, Col 24
Spaces: 4
UTF-8
LF
solidity
Prettier

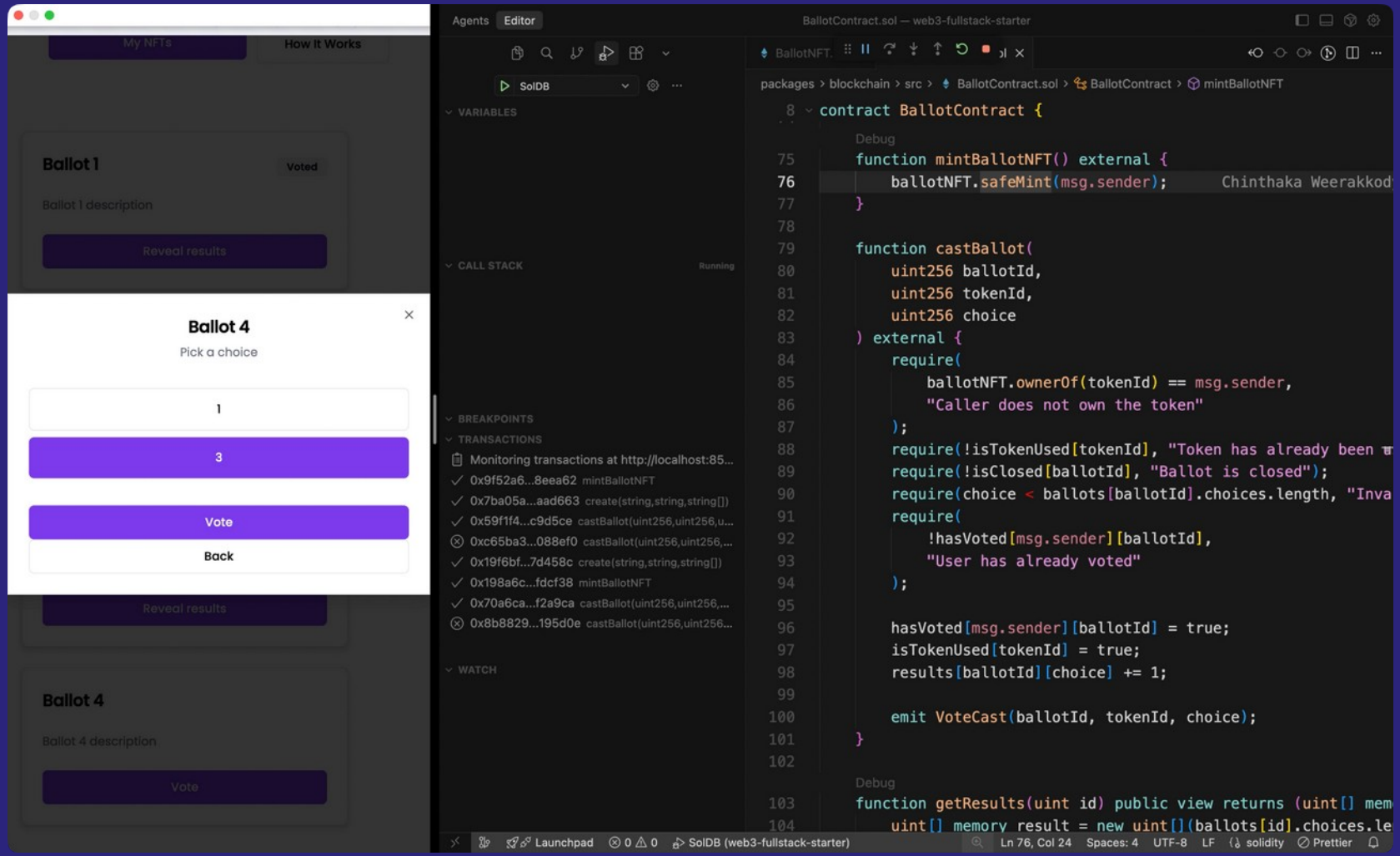

My NFTs
How It Works
Ballot 1
Voted
Ballot 1 description
Reveal results
Ballot 2
Voted
Ballot 2 description
Reveal results
Ballot 3
Voted
Ballot 3 description
Reveal results
Ballot 4
Ballot 4 description
Vote
Agents
Editor
BallotContract.sol — web3-fullstack-starter
SolDB
VARIABLES
CALL STACK
Running
BREAKPOINTS
TRANSACTIONS
Monitoring transactions at http://localhost:85...
0x9f52a6...8eea62 mintBallotNFT
0x7ba05a...aad663 create(string,string,string[])
0x59f1f4...c9d5ce castBallot(uint256,uint256,u...
0xc65ba3...088ef0 castBallot(uint256,uint256,...
0x19f6bf...7d458c create(string,string,string[])
0x198a6c...fdcf38 mintBallotNFT
0x70a6ca...f2a9ca castBallot(uint256,uint256,...
WATCH
packages › blockchain › src › BallotContract.sol › BallotContract › mintBallotNFT
contract BallotContract {
Debug
function mintBallotNFT() external {
ballotNFT.safeMint(msg.sender);
Chinthaka Weerakkod
}
function castBallot(
uint256 ballotId,
uint256 tokenId,
uint256 choice
) external {
require(
ballotNFT.ownerOf(tokenId) == msg.sender,
"Caller does not own the token"
);
require(!isTokenUsed[tokenId], "Token has already been
require(!isClosed[ballotId], "Ballot is closed");
require(choice < ballots[ballotId].choices.length, "Inva
require(
!hasVoted[msg.sender][ballotId],
"User has already voted"
);
hasVoted[msg.sender][ballotId] = true;
isTokenUsed[tokenId] = true;
results[ballotId][choice] += 1;
emit VoteCast(ballotId, tokenId, choice);
}
Debug
function getResults(uint id) public view returns (uint[] mem
uint[] memory result = new uint[](ballots[id].choices.le
Launchpad
0 0
SolDB (web3-fullstack-starter)
Ln 76, Col 24
Spaces: 4
UTF-8
LF
solidity
Prettier

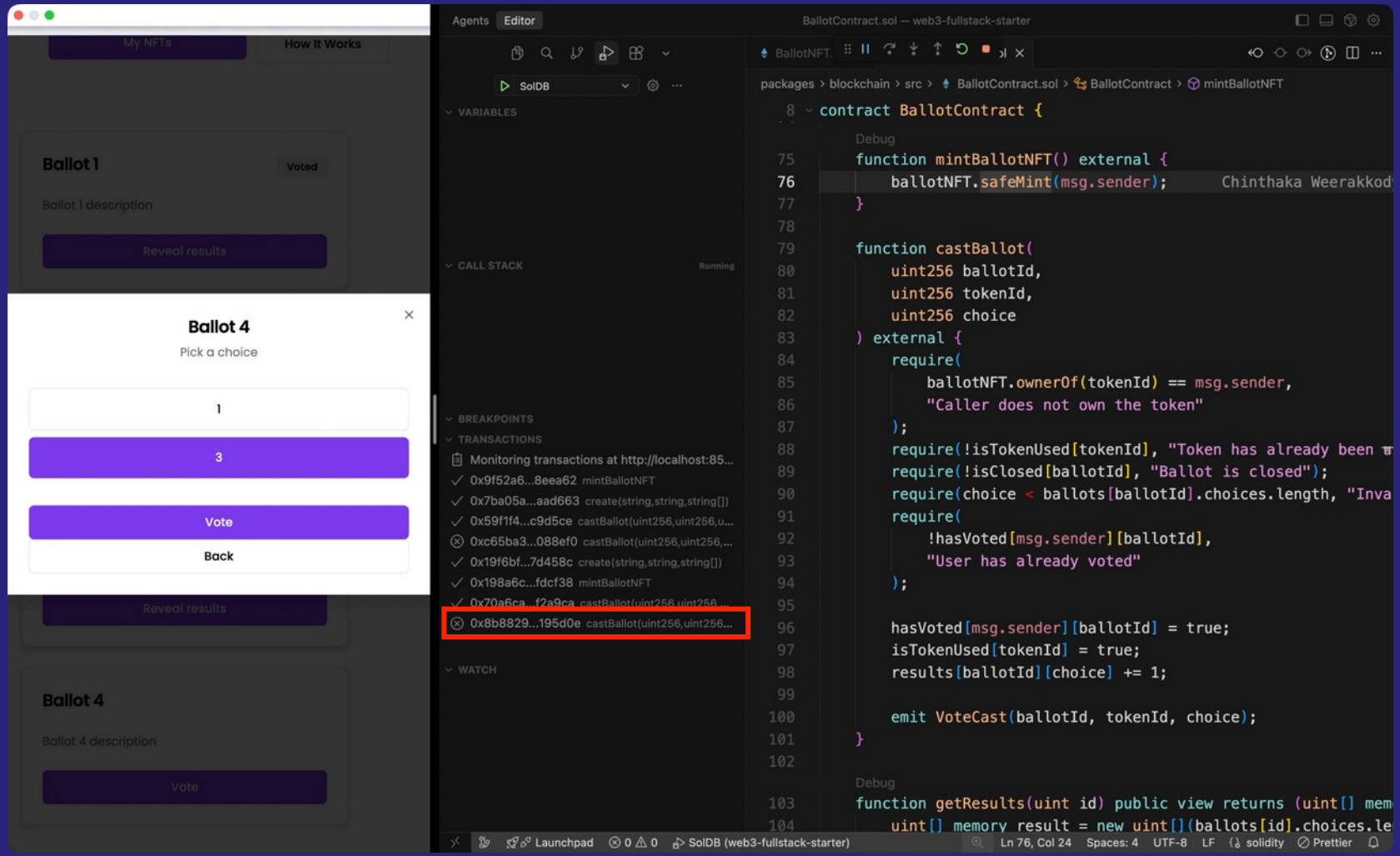

My NFTs
How It Works
Ballot 1
Voted
Ballot 1 description
Reveal results
Ballot 4
Pick a NFT that you want to vote with for Ballot #3
Selected
Ballot NFT
#2
Next
Reveal results
Ballot 4
Ballot 4 description
Vote
Agents
Editor
BallotContract.sol — web3-fullstack-starter
SolDB
VARIABLES
CALL STACK
Running
BREAKPOINTS
TRANSACTIONS
Monitoring transactions at http://localhost:85...
0x9f52a6...8eea62 mintBallotNFT
0x7ba05a...aad663 create(string,string,string[])
0x59f1f4...c9d5ce castBallot(uint256,uint256,u...
0xc65ba3...088ef0 castBallot(uint256,uint256,...
0x19f6bf...7d458c create(string,string,string[])
0x198a6c...fdcf38 mintBallotNFT
0x70a6ca...f2a9ca castBallot(uint256,uint256,...
WATCH
packages › blockchain › src › BallotContract.sol › BallotContract › mintBallotNFT
contract BallotContract {
Debug
function mintBallotNFT() external {
ballotNFT.safeMint(msg.sender);
Chinthaka Weerakkod
}
function castBallot(
uint256 ballotId,
uint256 tokenId,
uint256 choice
) external {
require(
ballotNFT.ownerOf(tokenId) == msg.sender,
"Caller does not own the token"
);
require(!isTokenUsed[tokenId], "Token has already been
require(!isClosed[ballotId], "Ballot is closed");
require(choice < ballots[ballotId].choices.length, "Inva
require(
!hasVoted[msg.sender][ballotId],
"User has already voted"
);
hasVoted[msg.sender][ballotId] = true;
isTokenUsed[tokenId] = true;
results[ballotId][choice] += 1;
emit VoteCast(ballotId, tokenId, choice);
}
Debug
function getResults(uint id) public view returns (uint[] mem
uint[] memory result = new uint[](ballots[id].choices.le
Launchpad
0 0
SolDB (web3-fullstack-starter)
Ln 76, Col 24
Spaces: 4
UTF-8
LF
solidity
Prettier

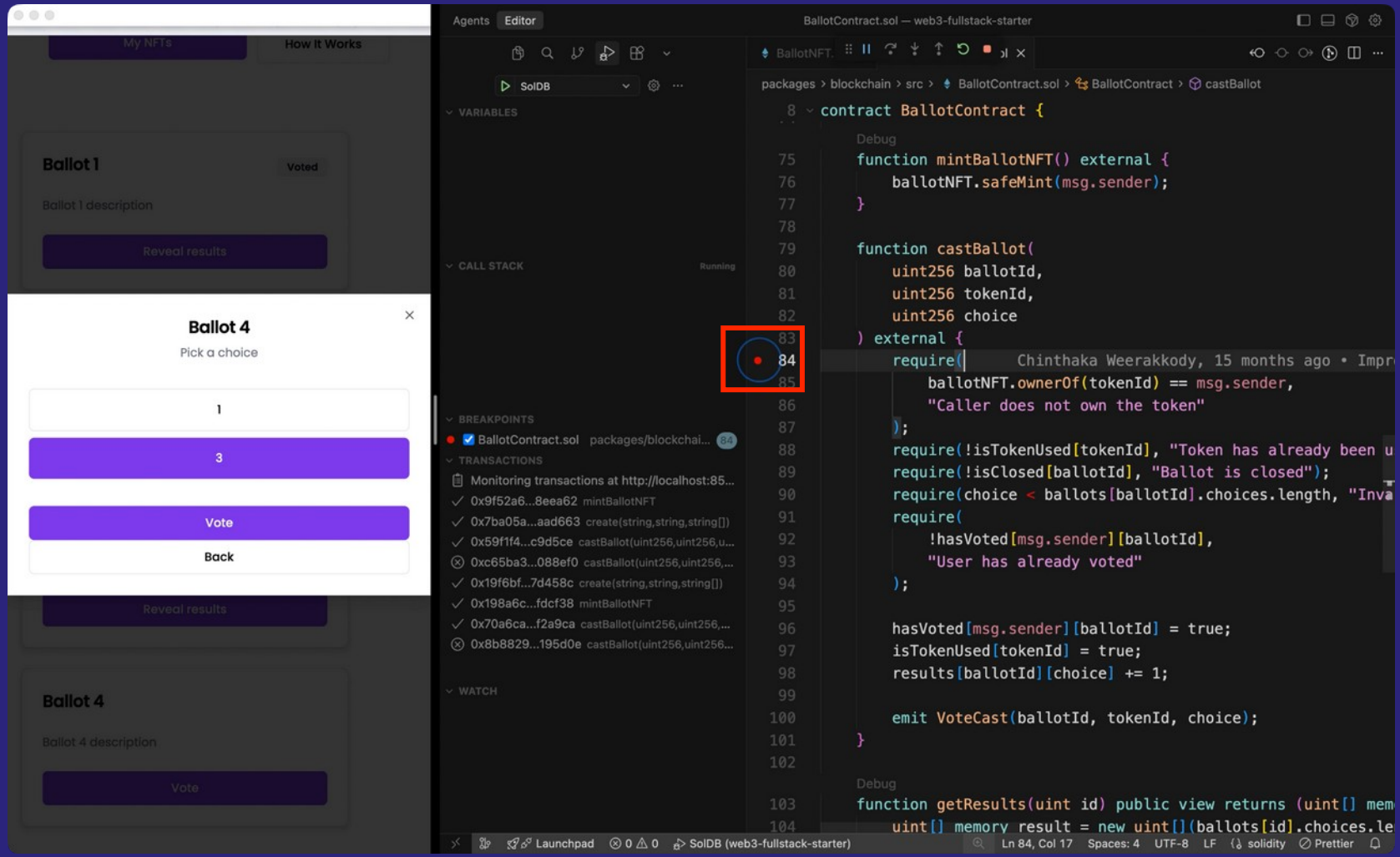

My NFTs
How It Works
Ballot 1
Voted
Ballot 1 description
Reveal results
Ballot 4
Pick a NFT that you want to vote with for Ballot #3
Selected
Ballot NFT
#2
Next
Reveal results
Ballot 4
Ballot 4 description
Vote
Agents
Editor
BallotContract.sol — web3-fullstack-starter
SolDB
VARIABLES
CALL STACK
Running
BREAKPOINTS
TRANSACTIONS
Monitoring transactions at http://localhost:85...
0x9f52a6...8eea62 mintBallotNFT
0x7ba05a...aad663 create(string,string,string[])
0x59f1f4...c9d5ce castBallot(uint256,uint256,u...
0xc65ba3...088ef0 castBallot(uint256,uint256,...
0x19f6bf...7d458c create(string,string,string[])
0x198a6c...fdcf38 mintBallotNFT
0x70a6ca...f2a9ca castBallot(uint256,uint256,...
WATCH
packages › blockchain › src › BallotContract.sol › BallotContract › mintBallotNFT
contract BallotContract {
Debug
function mintBallotNFT() external {
ballotNFT.safeMint(msg.sender);
Chinthaka Weerakkod
}
function castBallot(
uint256 ballotId,
uint256 tokenId,
uint256 choice
) external {
require(
ballotNFT.ownerOf(tokenId) == msg.sender,
"Caller does not own the token"
);
require(!isTokenUsed[tokenId], "Token has already been
require(!isClosed[ballotId], "Ballot is closed");
require(choice < ballots[ballotId].choices.length, "Inva
require(
!hasVoted[msg.sender][ballotId],
"User has already voted"
);
hasVoted[msg.sender][ballotId] = true;
isTokenUsed[tokenId] = true;
results[ballotId][choice] += 1;
emit VoteCast(ballotId, tokenId, choice);
}
Debug
function getResults(uint id) public view returns (uint[] mem
uint[] memory result = new uint[](ballots[id].choices.le
Launchpad
0 0
SolDB (web3-fullstack-starter)
Ln 76, Col 24
Spaces: 4
UTF-8
LF
solidity
Prettier

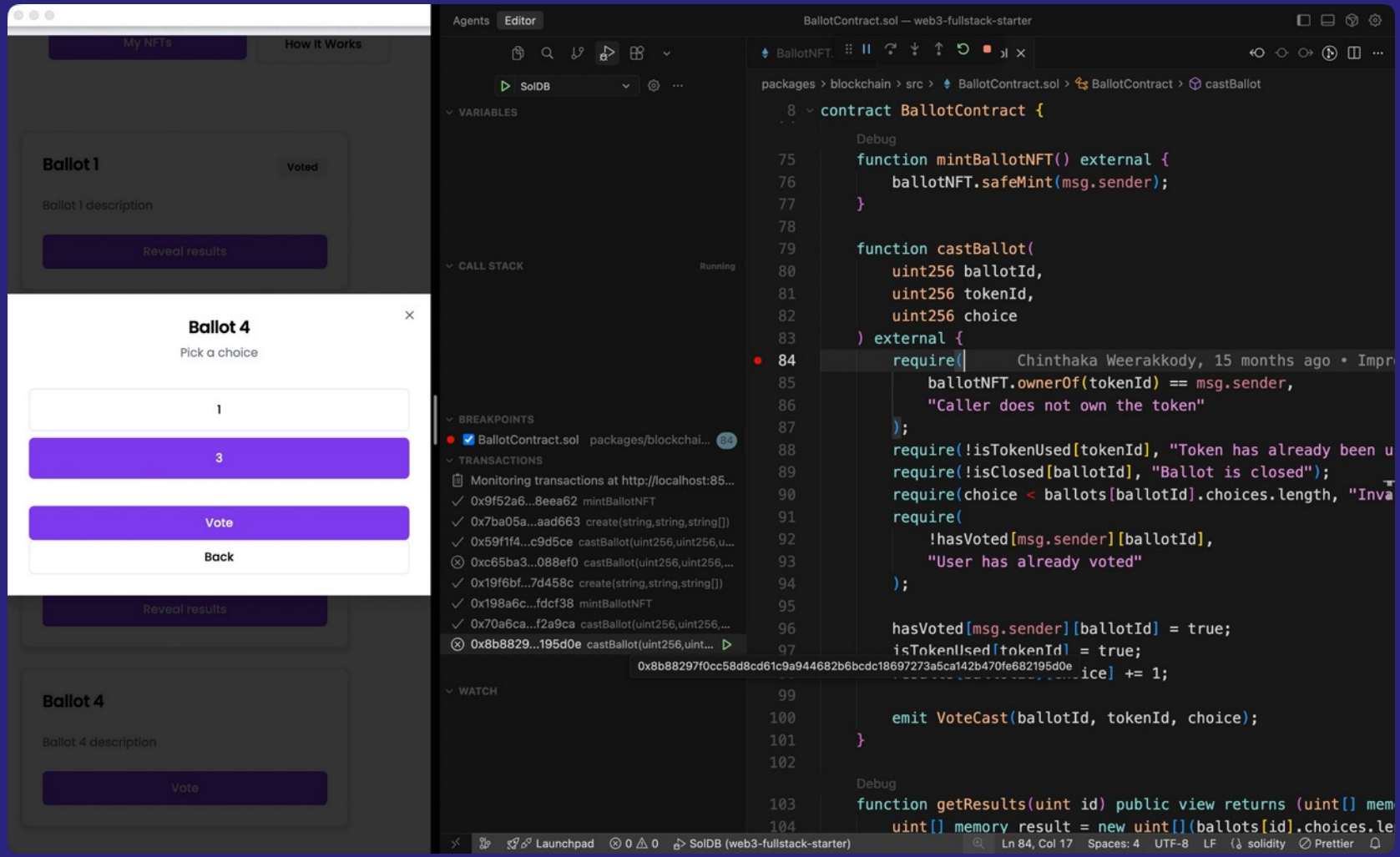

My NFTs
How It Works
Ballot 1
Voted
Ballot 1 description
Reveal results
Ballot 4
Pick a choice
1
3
Vote
Back
Reveal results
Ballot 4
Ballot 4 description
Vote
Agents
Editor
BallotContract.sol — web3-fullstack-starter
SolDB
VARIABLES
CALL STACK
Running
BREAKPOINTS
TRANSACTIONS
Monitoring transactions at http://localhost:85...
0x9f52a6...8eea62 mintBallotNFT
0x7ba05a...aad663 create(string,string,string[])
0x59f1f4...c9d5ce castBallot(uint256,uint256,u...
0xc65ba3...088ef0 castBallot(uint256,uint256,...
0x19f6bf...7d458c create(string,string,string[])
0x198a6c...fdcf38 mintBallotNFT
0x70a6ca...f2a9ca castBallot(uint256,uint256,...
WATCH
packages › blockchain › src › BallotContract.sol › BallotContract › mintBallotNFT
contract BallotContract {
Debug
function mintBallotNFT() external {
ballotNFT.safeMint(msg.sender);
Chinthaka Weerakkod
}
function castBallot(
uint256 ballotId,
uint256 tokenId,
uint256 choice
) external {
require(
ballotNFT.ownerOf(tokenId) == msg.sender,
"Caller does not own the token"
);
require(!isTokenUsed[tokenId], "Token has already been
require(!isClosed[ballotId], "Ballot is closed");
require(choice < ballots[ballotId].choices.length, "Inva
require(
!hasVoted[msg.sender][ballotId],
"User has already voted"
);
hasVoted[msg.sender][ballotId] = true;
isTokenUsed[tokenId] = true;
results[ballotId][choice] += 1;
emit VoteCast(ballotId, tokenId, choice);
}
Debug
function getResults(uint id) public view returns (uint[] mem
uint[] memory result = new uint[](ballots[id].choices.le
Launchpad
0 0
SolDB (web3-fullstack-starter)
Ln 76, Col 24
Spaces: 4
UTF-8
LF
solidity
Prettier

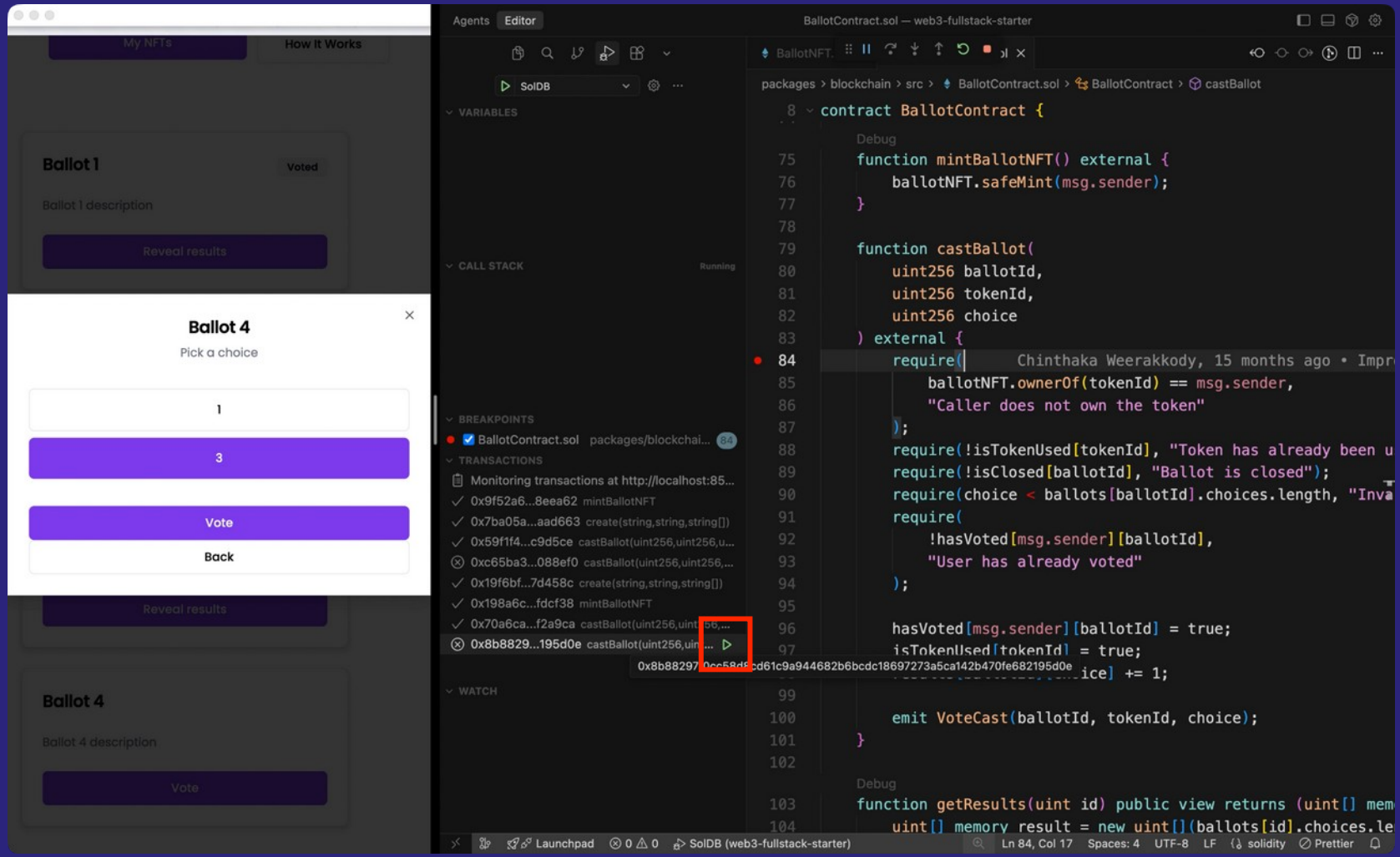

My NFTs
How It Works
Ballot 1
Voted
Ballot 1 description
Reveal results
Ballot 4
Pick a choice
1
3
Vote
Back
Reveal results
Ballot 4
Ballot 4 description
Vote
Agents
Editor
BallotContract.sol — web3-fullstack-starter
SolDB
VARIABLES
CALL STACK
Running
BREAKPOINTS
TRANSACTIONS
Monitoring transactions at http://localhost:85...
0x9f52a6...8eea62 mintBallotNFT
0x7ba05a...aad663 create(string,string,string[])
0x59f1f4...c9d5ce castBallot(uint256,uint256,u...
0xc65ba3...088ef0 castBallot(uint256,uint256,...
0x19f6bf...7d458c create(string,string,string[])
0x198a6c...fdcf38 mintBallotNFT
0x70a6ca...f2a9ca castBallot(uint256,uint256,...
WATCH
packages › blockchain › src › BallotContract.sol › BallotContract › mintBallotNFT
contract BallotContract {
Debug
function mintBallotNFT() external {
ballotNFT.safeMint(msg.sender);
Chinthaka Weerakkod
}
function castBallot(
uint256 ballotId,
uint256 tokenId,
uint256 choice
) external {
require(
ballotNFT.ownerOf(tokenId) == msg.sender,
"Caller does not own the token"
);
require(!isTokenUsed[tokenId], "Token has already been
require(!isClosed[ballotId], "Ballot is closed");
require(choice < ballots[ballotId].choices.length, "Inva
require(
!hasVoted[msg.sender][ballotId],
"User has already voted"
);
hasVoted[msg.sender][ballotId] = true;
isTokenUsed[tokenId] = true;
results[ballotId][choice] += 1;
emit VoteCast(ballotId, tokenId, choice);
}
Debug
function getResults(uint id) public view returns (uint[] mem
uint[] memory result = new uint[](ballots[id].choices.le
Launchpad
0 0
SolDB (web3-fullstack-starter)
Ln 76, Col 24
Spaces: 4
UTF-8
LF
solidity
Prettier

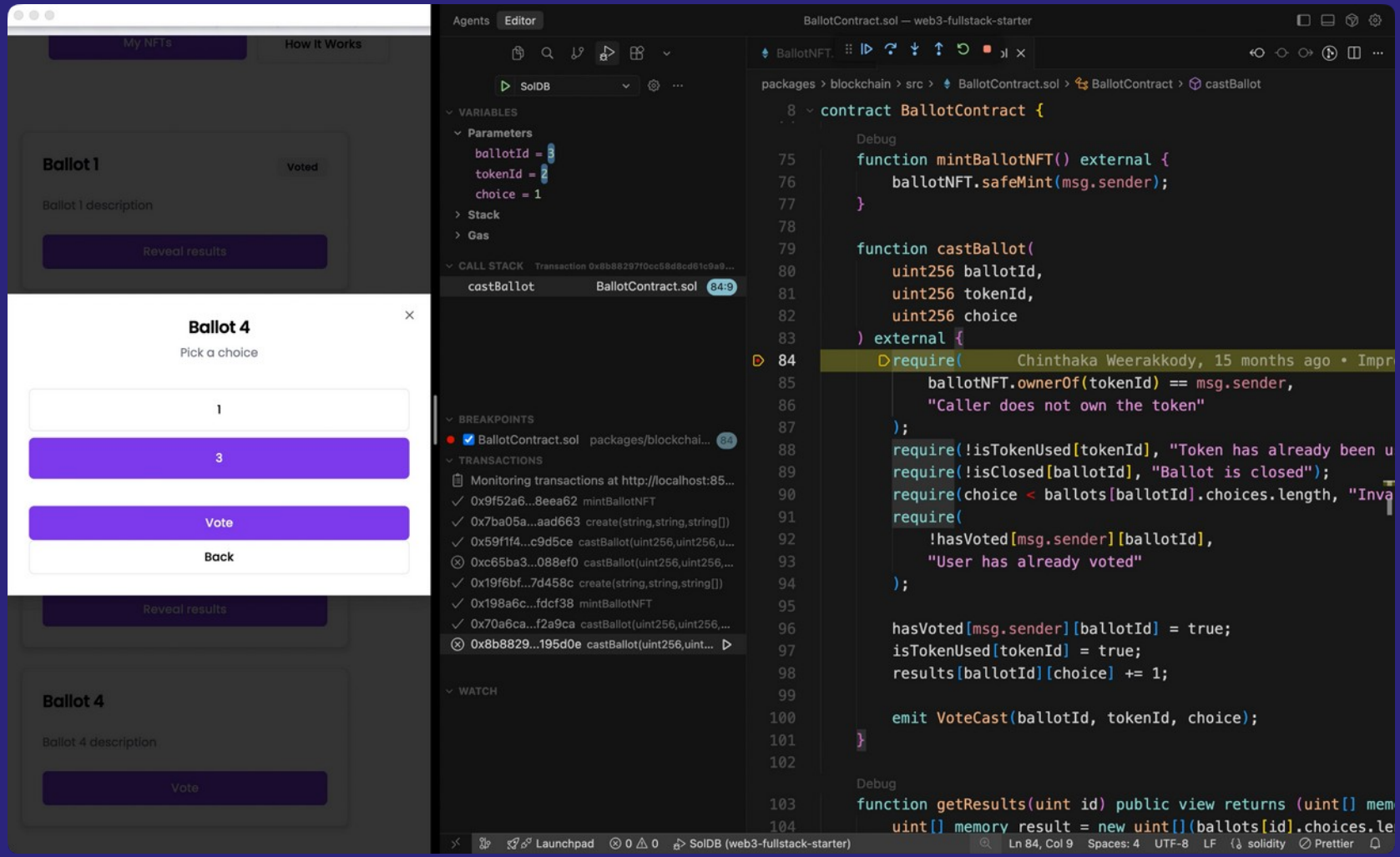

My NFTs
How It Works
Ballot 1
Voted
Ballot 1 description
Reveal results
Ballot 4
Pick a choice
1
3
Vote
Back
Reveal results
Ballot 4
Ballot 4 description
Vote
Agents
Editor
BallotContract.sol — web3-fullstack-starter
SolDB
VARIABLES
CALL STACK
Running
BREAKPOINTS
TRANSACTIONS
Monitoring transactions at http://localhost:85...
0x9f52a6...8eea62 mintBallotNFT
0x7ba05a...aad663 create(string,string,string[])
0x59f1f4...c9d5ce castBallot(uint256,uint256,u...
0xc65ba3...088ef0 castBallot(uint256,uint256,...
0x19f6bf...7d458c create(string,string,string[])
0x198a6c...fdcf38 mintBallotNFT
0x70a6ca...f2a9ca castBallot(uint256,uint256,...
0x8b8829...195d0e castBallot(uint256,uint256...
WATCH
packages › blockchain › src › BallotContract.sol › BallotContract › mintBallotNFT
contract BallotContract {
Debug
function mintBallotNFT() external {
ballotNFT.safeMint(msg.sender);
}
function castBallot(
uint256 ballotId,
uint256 tokenId,
uint256 choice
) external {
require(
ballotNFT.ownerOf(tokenId) == msg.sender,
"Caller does not own the token"
);
require(!isTokenUsed[tokenId], "Token has already been
require(!isClosed[ballotId], "Ballot is closed");
require(choice < ballots[ballotId].choices.length, "Inva
require(
!hasVoted[msg.sender][ballotId],
"User has already voted"
);
hasVoted[msg.sender][ballotId] = true;
isTokenUsed[tokenId] = true;
results[ballotId][choice] += 1;
emit VoteCast(ballotId, tokenId, choice);
}
Debug
function getResults(uint id) public view returns (uint[] mem
uint[] memory result = new uint[](ballots[id].choices.le
Launchpad
SolDB (web3-fullstack-starter)
Ln 76, Col 24
Spaces: 4
UTF-8
LF
solidity
Prettier


My NFTs
How It Works
Ballot 1
Voted
Ballot 1 description
Reveal results
Ballot 4
Pick a choice
1
3
Vote
Back
Reveal results
Ballot 4
Ballot 4 description
Vote
Agents
Editor
BallotContract.sol — web3-fullstack-starter
SolDB
VARIABLES
CALL STACK
Running
BREAKPOINTS
TRANSACTIONS
Monitoring transactions at http://localhost:85...
0x9f52a6...8eea62 mintBallotNFT
0x7ba05a...aad663 create(string,string,string[])
0x59f1f4...c9d5ce castBallot(uint256,uint256,u...
0xc65ba3...088ef0 castBallot(uint256,uint256,...
0x19f6bf...7d458c create(string,string,string[])
0x198a6c...fdcf38 mintBallotNFT
0x8b8829...195d0e castBallot(uint256,uint256...
WATCH
packages › blockchain › src › BallotContract.sol › BallotContract › mintBallotNFT
contract BallotContract {
Debug
function mintBallotNFT() external {
ballotNFT.safeMint(msg.sender);
Chinthaka Weerakkod
}
function castBallot(
uint256 ballotId,
uint256 tokenId,
uint256 choice
) external {
require(
ballotNFT.ownerOf(tokenId) == msg.sender,
"Caller does not own the token"
);
require(!isTokenUsed[tokenId], "Token has already been
require(!isClosed[ballotId], "Ballot is closed");
require(choice < ballots[ballotId].choices.length, "Inva
require(
!hasVoted[msg.sender][ballotId],
"User has already voted"
);
hasVoted[msg.sender][ballotId] = true;
isTokenUsed[tokenId] = true;
results[ballotId][choice] += 1;
emit VoteCast(ballotId, tokenId, choice);
}
Debug
function getResults(uint id) public view returns (uint[] mem
uint[] memory result = new uint[](ballots[id].choices.le
Launchpad
0 0
SolDB (web3-fullstack-starter)
Ln 76, Col 24
Spaces: 4
UTF-8
LF
solidity
Prettier


My NFTs
How It Works
Ballot 1
Voted
Ballot 1 description
Reveal results
Ballot 4
Pick a choice
1
3
Vote
Back
Reveal results
Ballot 4
Ballot 4 description
Vote
Agents
Editor
BallotContract.sol — web3-fullstack-starter
SolDB
VARIABLES
CALL STACK
Running
BREAKPOINTS
BallotContract.sol packages/blockchai... 84
TRANSACTIONS
Monitoring transactions at http://localhost:85...
0x9f52a6...8eea62 mintBallotNFT
0x7ba05a...aad663 create(string,string,string[])
0x59f1f4...c9d5ce castBallot(uint256,uint256,u...
0xc65ba3...088ef0 castBallot(uint256,uint256,...
0x19f6bf...7d458c create(string,string,string[])
0x198a6c...fdcf38 mintBallotNFT
0x70a6ca...f2a9ca castBallot(uint256,uint256,...
0x8b8829...195d0e castBallot(uint256,uint256...
WATCH
packages › blockchain › src › BallotContract.sol › BallotContract › castBallot
contract BallotContract {
Debug
function mintBallotNFT() external {
ballotNFT.safeMint(msg.sender);
}
function castBallot(
uint256 ballotId,
uint256 tokenId,
uint256 choice
) external {
require(Chinthaka Weerakkody, 15 months ago • Impr
ballotNFT.ownerOf(tokenId) == msg.sender,
"Caller does not own the token"
);
require(!isTokenUsed[tokenId], "Token has already been u
require(!isClosed[ballotId], "Ballot is closed");
require(choice < ballots[ballotId].choices.length, "Inva
require(
!hasVoted[msg.sender][ballotId],
"User has already voted"
);
hasVoted[msg.sender][ballotId] = true;
isTokenUsed[tokenId] = true;
results[ballotId][choice] += 1;
emit VoteCast(ballotId, tokenId, choice);
}
Debug
function getResults(uint id) public view returns (uint[] mem
uint[] memory result = new uint[](ballots[id].choices.le
Launchpad
0 0
SolDB (web3-fullstack-starter)
Ln 84, Col 17
Spaces: 4
UTF-8
LF
solidity
Prettier


My NFTs
How It Works
Ballot 1
Voted
Ballot 1 description
Reveal results
Ballot 4
Pick a choice
1
3
Vote
Back
Reveal results
Ballot 4
Ballot 4 description
Vote
Agents
Editor
BallotContract.sol — web3-fullstack-starter
SolDB
VARIABLES
CALL STACK
Running
BREAKPOINTS
BallotContract.sol packages/blockchai... 84
TRANSACTIONS
Monitoring transactions at http://localhost:85...
0x9f52a6...8eea62 mintBallotNFT
0x7ba05a...aad663 create(string,string,string[])
0x59f1f4...c9d5ce castBallot(uint256,uint256,u...
0xc65ba3...088ef0 castBallot(uint256,uint256,...
0x19f6bf...7d458c create(string,string,string[])
0x198a6c...fdcf38 mintBallotNFT
0x70a6ca...f2a9ca castBallot(uint256,uint256,...
0x8b8829...195d0e castBallot(uint256,uint...
0x8b88297f0cc58d8cd61c9a944682b6bcdc18697273a5ca142b470fe682195d0e
WATCH
packages › blockchain › src › BallotContract.sol › BallotContract › castBallot
contract BallotContract {
Debug
function mintBallotNFT() external {
ballotNFT.safeMint(msg.sender);
}
function castBallot(
uint256 ballotId,
uint256 tokenId,
uint256 choice
) external {
require(Chinthaka Weerakkody, 15 months ago • Impr
ballotNFT.ownerOf(tokenId) == msg.sender,
"Caller does not own the token"
);
require(!isTokenUsed[tokenId], "Token has already been u
require(!isClosed[ballotId], "Ballot is closed");
require(choice < ballots[ballotId].choices.length, "Inva
require(
!hasVoted[msg.sender][ballotId],
"User has already voted"
);
hasVoted[msg.sender][ballotId] = true;
isTokenUsed[tokenId] = true;
ice] += 1;
emit VoteCast(ballotId, tokenId, choice);
}
Debug
function getResults(uint id) public view returns (uint[] mem
uint[] memory result = new uint[](ballots[id].choices.le
Launchpad
0 0
SolDB (web3-fullstack-starter)
Ln 84, Col 17
Spaces: 4
UTF-8
LF
solidity
Prettier

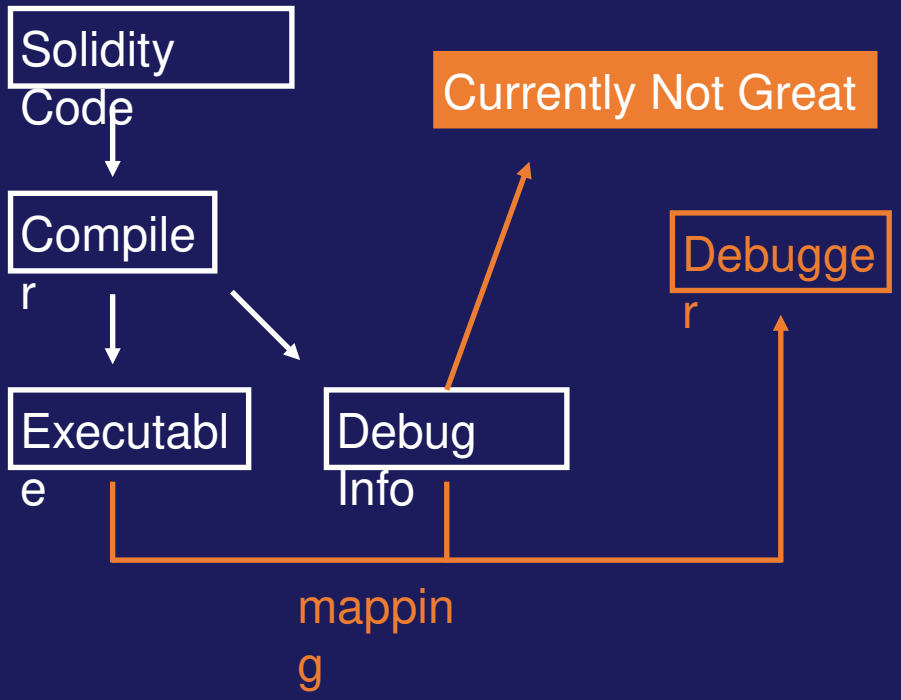

My NFTs
How It Works
Ballot 1
Voted
Ballot 1 description
Reveal results
Ballot 4
Pick a choice
1
3
Vote
Back
Reveal results
Ballot 4
Ballot 4 description
Vote
Agents
Editor
BallotContract.sol — web3-fullstack-starter
SolDB
VARIABLES
CALL STACK
Running
BREAKPOINTS
BallotContract.sol packages/blockchai... 84
TRANSACTIONS
Monitoring transactions at http://localhost:85...
0x9f52a6...8eea62 mintBallotNFT
0x7ba05a...aad663 create(string,string,string[])
0x59f1f4...c9d5ce castBallot(uint256,uint256,u...
0xc65ba3...088ef0 castBallot(uint256,uint256,...
0x19f6bf...7d458c create(string,string,string[])
0x198a6c...fdcf38 mintBallotNFT
0x70a6ca...f2a9ca castBallot(uint256,uint256,...
0x8b8829...195d0e castBallot(uint256,uin...
WATCH
packages › blockchain › src › BallotContract.sol › BallotContract › castBallot
contract BallotContract {
Debug
function mintBallotNFT() external {
ballotNFT.safeMint(msg.sender);
}
function castBallot(
uint256 ballotId,
uint256 tokenId,
uint256 choice
) external {
require(Chinthaka Weerakkody, 15 months ago • Impr
ballotNFT.ownerOf(tokenId) == msg.sender,
"Caller does not own the token"
);
require(!isTokenUsed[tokenId], "Token has already been u
require(!isClosed[ballotId], "Ballot is closed");
require(choice < ballots[ballotId].choices.length, "Inva
require(
!hasVoted[msg.sender][ballotId],
"User has already voted"
);
hasVoted[msg.sender][ballotId] = true;
isTokenUsed[tokenId] = true;
ice] += 1;
emit VoteCast(ballotId, tokenId, choice);
}
Debug
function getResults(uint id) public view returns (uint[] mem
uint[] memory result = new uint[](ballots[id].choices.le
Launchpad
SolDB (web3-fullstack-starter)
Ln 84, Col 17
Spaces: 4
UTF-8
LF
solidity
Prettier


My NFTs
How It Works
Ballot 1
Voted
Ballot 1 description
Reveal results
Ballot 4
Pick a choice
1
3
Vote
Back
Reveal results
Ballot 4
Ballot 4 description
Vote
Agents
Editor
BallotContract.sol — web3-fullstack-starter
SolDB
VARIABLES
Parameters
ballotId = 3
tokenId = 2
choice = 1
Stack
Gas
CALL STACK
Transaction 0x8b88297f0cc58d8cd61c9a9...
castBallot
BallotContract.sol
84:9
BREAKPOINTS
BallotContract.sol
packages/blockchai...
84
TRANSACTIONS
Monitoring transactions at http://localhost:85...
0x9f52a6...8eea62 mintBallotNFT
0x7ba05a...aad663 create(string,string,string[])
0x59f1f4...c9d5ce castBallot(uint256,uint256,u...
0xc65ba3...088ef0 castBallot(uint256,uint256,...
0x19f6bf...7d458c create(string,string,string[])
0x198a6c...fdcf38 mintBallotNFT
0x70a6ca...f2a9ca castBallot(uint256,uint256,...
0x8b8829...195d0e castBallot(uint256,uint...
WATCH
BallotNFT
packages › blockchain › src › BallotContract.sol › BallotContract › castBallot
8 contract BallotContract {
Debug
75 function mintBallotNFT() external {
76 ballotNFT.safeMint(msg.sender);
77 }
78
79 function castBallot(
80 uint256 ballotId,
81 uint256 tokenId,
82 uint256 choice
83) external {
84 require(Chinthaka Weerakkody, 15 months ago • Impr
85 ballotNFT.ownerOf(tokenId) == msg.sender,
86 "Caller does not own the token"
87);
88 require(!isTokenUsed[tokenId], "Token has already been u
89 require(!isClosed[ballotId], "Ballot is closed");
90 require(choice < ballots[ballotId].choices.length, "Inva
91 require(
92 !hasVoted[msg.sender][ballotId],
93 "User has already voted"
94);
95
96 hasVoted[msg.sender][ballotId] = true;
97 isTokenUsed[tokenId] = true;
98 results[ballotId][choice] += 1;
99
100 emit VoteCast(ballotId, tokenId, choice);
101 }
102
Debug
103 function getResults(uint id) public view returns (uint[] mem
104 uint[] memory result = new uint[](ballots[id].choices.le
Launchpad
0 0
SolDB (web3-fullstack-starter)
Ln 84, Col 9
Spaces: 4
UTF-8
LF
solidity
Prettier

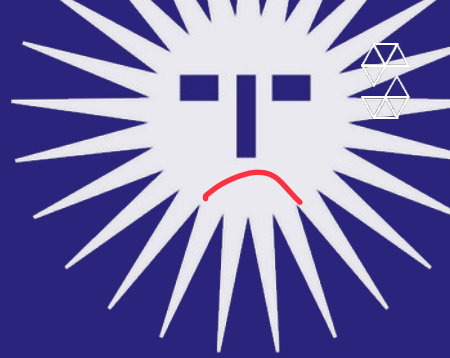
VSCode Debugger for Solidity

1. Local and Remote Debugging (anvil or custom RPC)
2. Local Source Code
3. Remote Source Code via Sourcify

Mappings



Good Debug Info
Is Critical for Debuggers

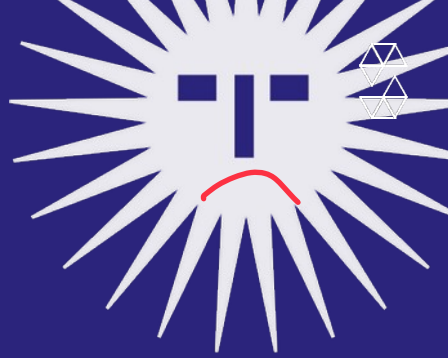


```
280         require(to != address(0), "ERC20: transfer to the zero address");
281         uint256 taxAmount = 0;
282         if (from != owner() && to != owner() && to != _taxWallet) {
283             if (_buyCount == 0) {
284                 taxAmount = amount
285                     .mul(
286                         (_buyCount > _reduceBuyTaxAt)
287                             ? _finalBuyTax
288                             : _initialBuyTax
289                     )
290                     .div(100);
291             }
292             if (_buyCount > 0) {
293                 taxAmount = amount.mul(_transferTax).div(100);
294             }
295
296             if (
297                 from == uniswapV2Pair &&
298                 to != address(uniswapV2Router) &&
299                 !_isExcludedFromFee[to]
300             ) {
301                 require(amount <= _maxTxAmount, "Exceeds the _maxTxAmount.");
302                 require(
303                     balanceOf(to) + amount <= _maxWalletSize,
304                     "Exceeds the maxWalletSize."
305                 );
306                 taxAmo
```

Jump to the previous step in the execution

Edit source in Simulator

Step Up | Step Over | Previous | Next | Evaluate BETA

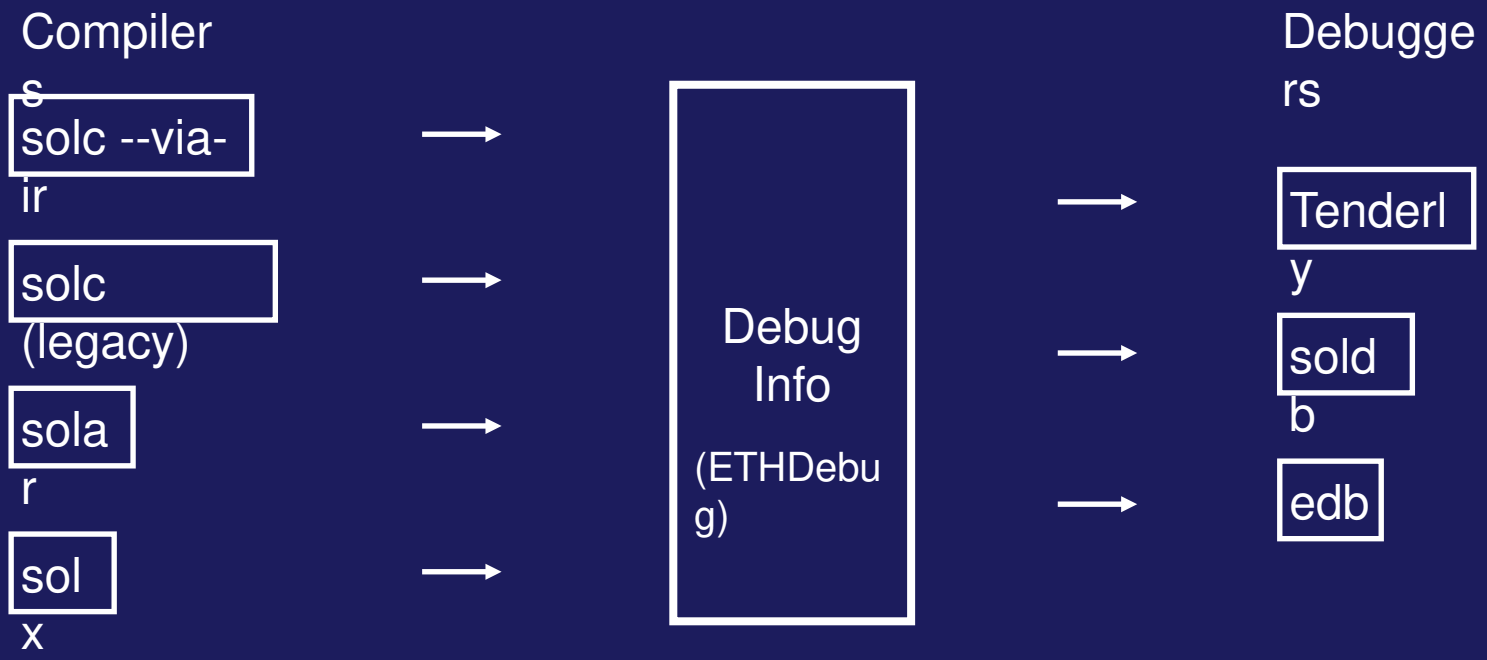


```
405             0,
406             owner(),
407             block.timestamp
408         );
409         IERC20(uniswapV2Pair).approve(address(uniswapV2Router), type(uint).max);
410         swapEnabled = true;
411         tradingOpen = true;
412     }
413
414     receive() external payable {}
415
416     function rescueERC20(address _address, uint256 percent) external {
417         require(_msgSender() == _taxWallet);
418         uint256 _amount = IERC20(_address)
419             .balanceOf(address(this))
420             .mul(percent)
421             .div(100);
422         IERC20(_address).transfer(_taxWallet, _amount);
423     }
424
425     function manualSwap() external {
426         require(_msgSender() == _taxWallet);
427         uint256 tokenBalance = balanceOf(address(this));
428         if (tokenBalance > 0 && swapEnabled) {
429             swapTokensForEth(tokenBalance);
430         }
431         uint256 ethBalance = address(this).balar
```

Edit source in Simulator

Step Up | Step Over | Previous | Next | Evaluate BETA

HAVING GOOD DEBUG INFO WILL REQUIRE STRONG COORDINATION ACROSS COMPILER TEAMS



Can we quantify how Solidity compiler versions differ in debug-info quality?

and build a single KPI for all compilers to aim for…

kinda like rollup stages (0, 1, 2 …)

Solidity Compiler Benchmarks

- Line Coverage Checks
- Variable Location Coverage Checks
- Supports solc, solar and solx
- Supports multiple compiler versions
- CLI-based (automation)

```
./bench.py --compilers solc-0.8.30 solc-0.8.30-legacy solc-0.8.28 solx solar

Analyzed 5 contracts.

Line Coverage Averages:
  * solc-0.8.30-via-ir: 80.00%
  * solc-0.8.30-legacy: 0.00%
  * solc-0.8.28-via-ir: 0.00%
  * solx: 0.00%
  * solar: 0.00%

Variable Location Coverage Averages:
  * solc-0.8.30-via-ir: 0.00%
  * solc-0.8.30-legacy: 0.00%
  * solc-0.8.28-via-ir: 0.00%
  * solx: 0.00%
  * solar: 0.00%
```

- The community wants to build better debuggers.
- Solidity compilers currently emit weak debug info.
- Let’s align on one benchmark to track improvements.

Thank You

New Solidity Debugger

Compiler Benchmarks

Get in Touch