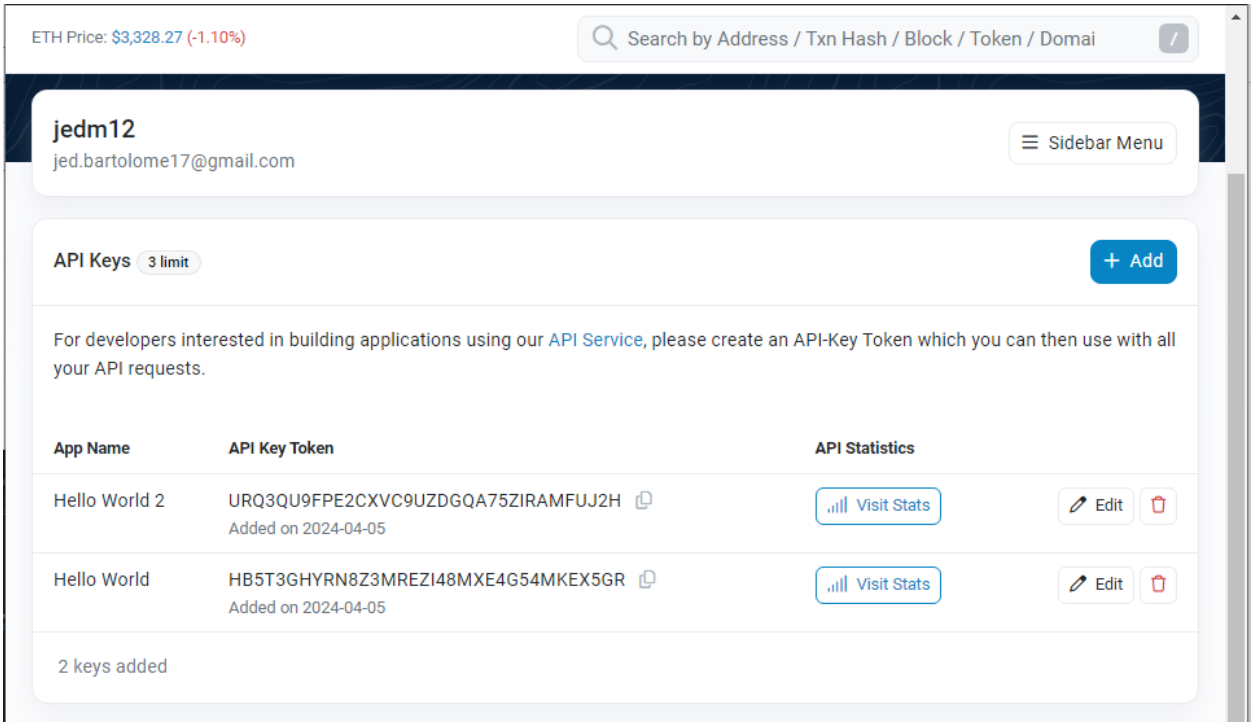
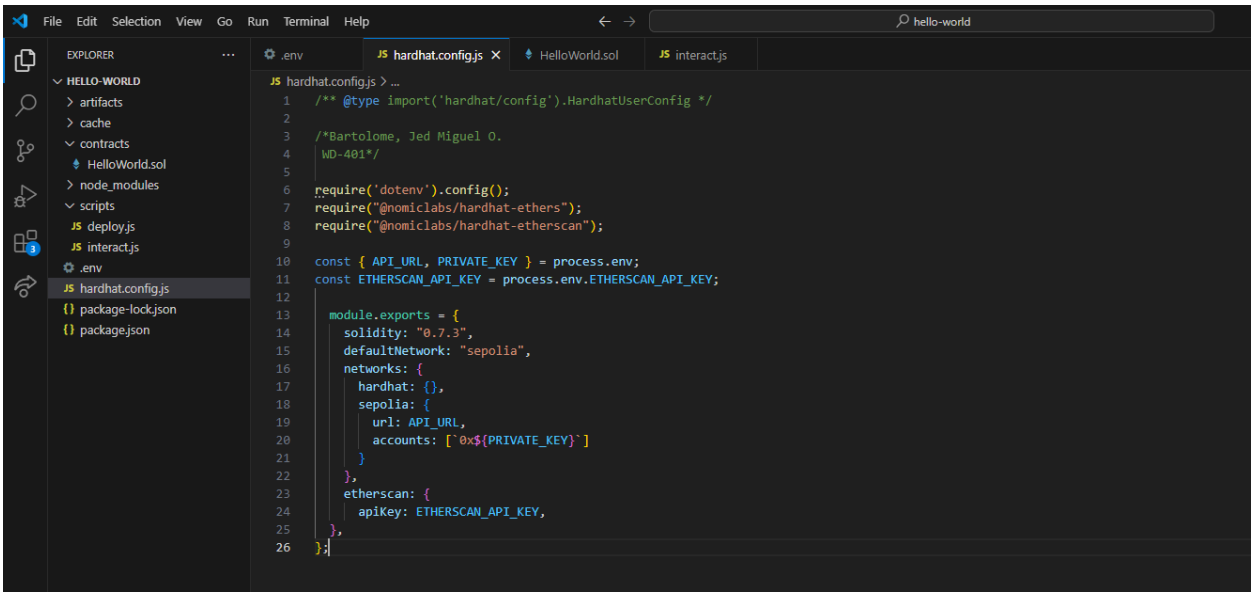
Bartolome, Jed Miguel O.

WD-401

Code:





Output:

```
E:\Stuffs\Jed\4th Year\2nd Sem\BCHAIN\hello-world>npx hardhat verify --network sepolia 0x1cFD4666eb3484a3ec9De87535Cf
1beAeA8Ed02c "Hello World!"
Nothing to compile
Successfully submitted source code for contract
contracts/HelloWorld.sol:HelloWorld at 0x1cFD4666eb3484a3ec9De87535Cf1beAeA8Ed02c
for verification on the block explorer. Waiting for verification result...

Successfully verified contract HelloWorld on Etherscan.
https://sepolia.etherscan.io/address/0x1cFD4666eb3484a3ec9De87535Cf1beAeA8Ed02c#code
```

Reflection:

I've learned some helpful techniques for developing smart contracts using Hardhat. By using environment variables and plugins like Hardhat-Ethers and Hardhat-Etherscan, I can keep sensitive information separate and secure. Setting the Solidity version and default network to "sepolia" ensures easy deployment and verification. Verifying deployments enhances security and promotes trust in the blockchain community. I used Hardhat's verification tool to confirm the deployment of the HelloWorld contract on the Sepolia network, which makes the contract's source code publicly available on Etherscan. These practices are important for decentralized applications, as they improve security and reliability for everyone involved.