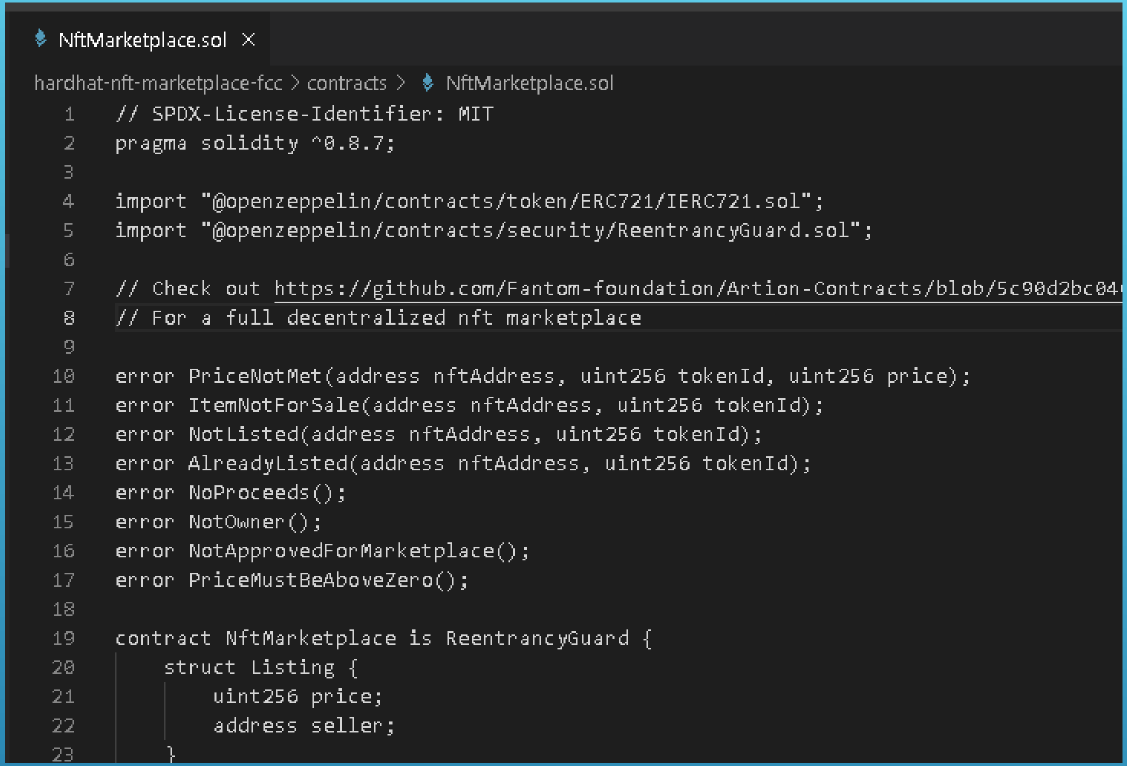
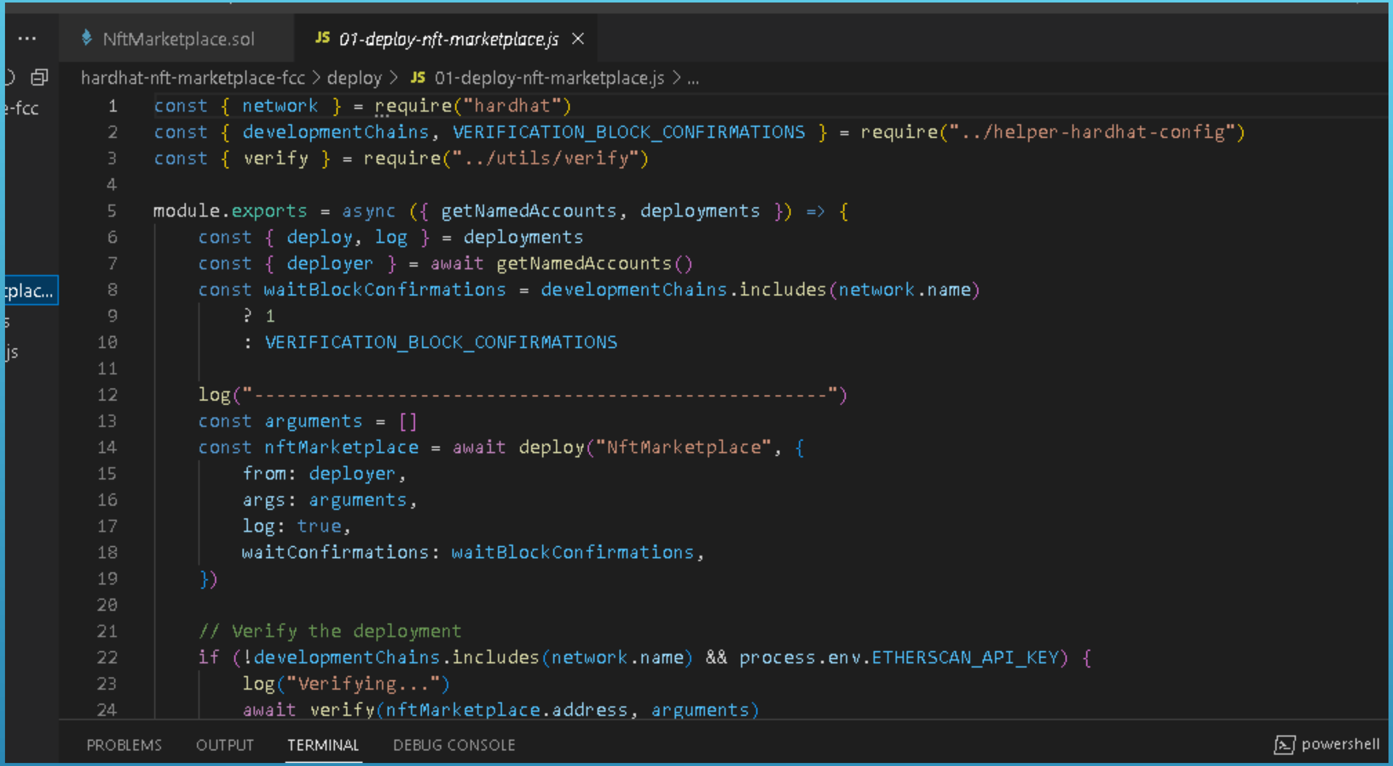
LESSON 15 FULL STACK NFT MARKETPLACE


NftMarketplace.sol
hardhat-nft-marketplace-fcc > contracts > NftMarketplace.sol

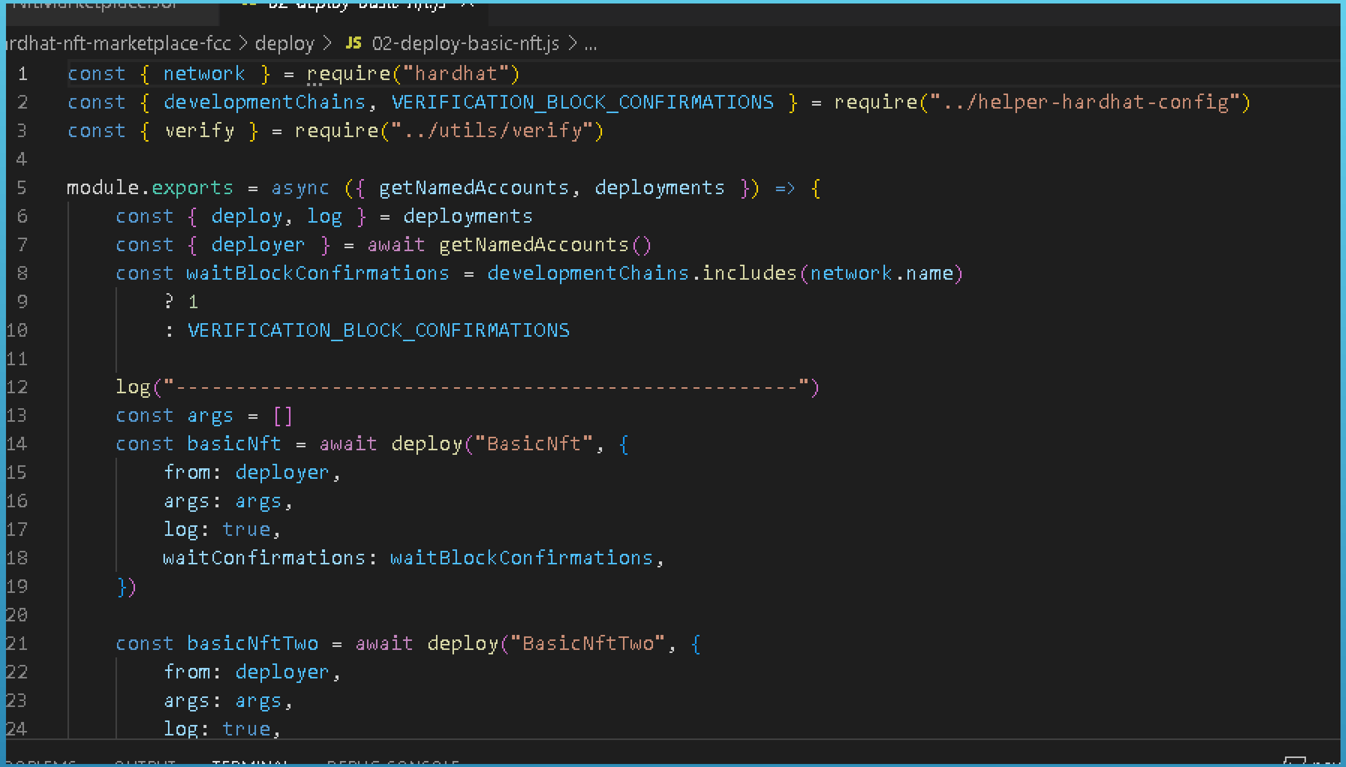
```solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.7;

import "@openzeppelin/contracts/token/ERC721/IERC721.sol";
import "@openzeppelin/contracts/security/ReentrancyGuard.sol";

// Check out https://github.com/Fantom-foundation/Artion-Contracts/blob/5c90d2bc04
// For a full decentralized nft marketplace

error PriceNotMet(address nftAddress, uint256 tokenId, uint256 price);
error ItemNotForSale(address nftAddress, uint256 tokenId);
error NotListed(address nftAddress, uint256 tokenId);
error AlreadyListed(address nftAddress, uint256 tokenId);
error NoProceeds();
error NotOwner();
error NotApprovedForMarketplace();
error PriceMustBeAboveZero();

contract NftMarketplace is ReentrancyGuard {
    struct Listing {
        uint256 price;
        address seller;
    }
```

CONTRACTS


NftMarketplace.sol
01-deploy-nft-marketplace.js
hardhat-nft-marketplace-fcc > deploy > 01-deploy-nft-marketplace.js > ...
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
powershell

```javascript
const { network } = require("hardhat")
const { developmentChains, VERIFICATION_BLOCK_CONFIRMATIONS } = require("../helper-hardhat-config")
const { verify } = require("../utils/verify")

module.exports = async ({ getNamedAccounts, deployments }) => {
    const { deploy, log } = deployments
    const { deployer } = await getNamedAccounts()
    const waitBlockConfirmations = developmentChains.includes(network.name)
        ? 1
        : VERIFICATION_BLOCK_CONFIRMATIONS

    log("----------------------------------------------------")
    const arguments = []
    const nftMarketplace = await deploy("NftMarketplace", {
        from: deployer,
        args: arguments,
        log: true,
        waitConfirmations: waitBlockConfirmations,
    })

    // Verify the deployment
    if (!developmentChains.includes(network.name) && process.env.ETHERSCAN_API_KEY) {
        log("Verifying...")
        await verify(nftMarketplace.address, arguments)
```

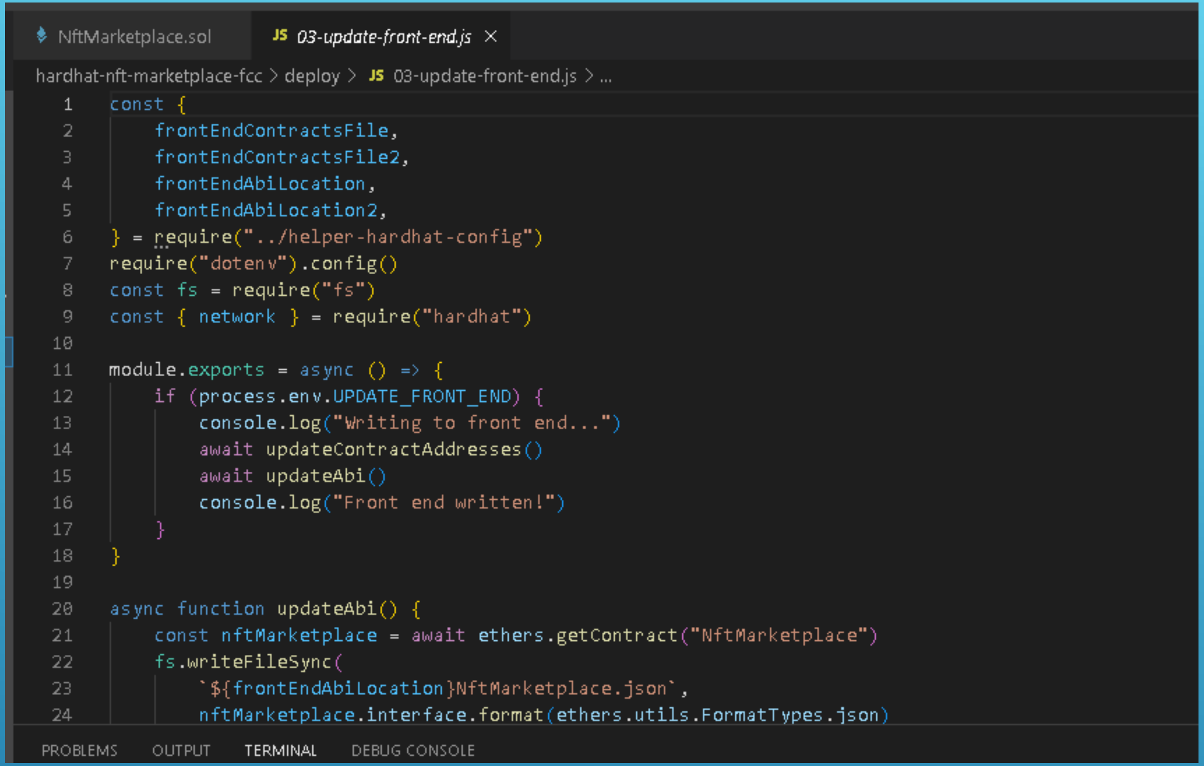
NFT MARKETPLACE DEPLOY



rdhat-nft-marketplace-fcc > deploy > JS 02-deploy-basic-nft.js > ...

```javascript
const { network } = require("hardhat")
const { developmentChains, VERIFICATION_BLOCK_CONFIRMATIONS } = require("../helper-hardhat-config")
const { verify } = require("../utils/verify")

module.exports = async ({ getNamedAccounts, deployments }) => {
    const { deploy, log } = deployments
    const { deployer } = await getNamedAccounts()
    const waitBlockConfirmations = developmentChains.includes(network.name)
        ? 1
        : VERIFICATION_BLOCK_CONFIRMATIONS

    log("----------------------------------------------------")
    const args = []
    const basicNft = await deploy("BasicNft", {
        from: deployer,
        args: args,
        log: true,
        waitConfirmations: waitBlockConfirmations,
    })

    const basicNftTwo = await deploy("BasicNftTwo", {
        from: deployer,
        args: args,
        log: true,
```

NFT BASIC DEPLOY

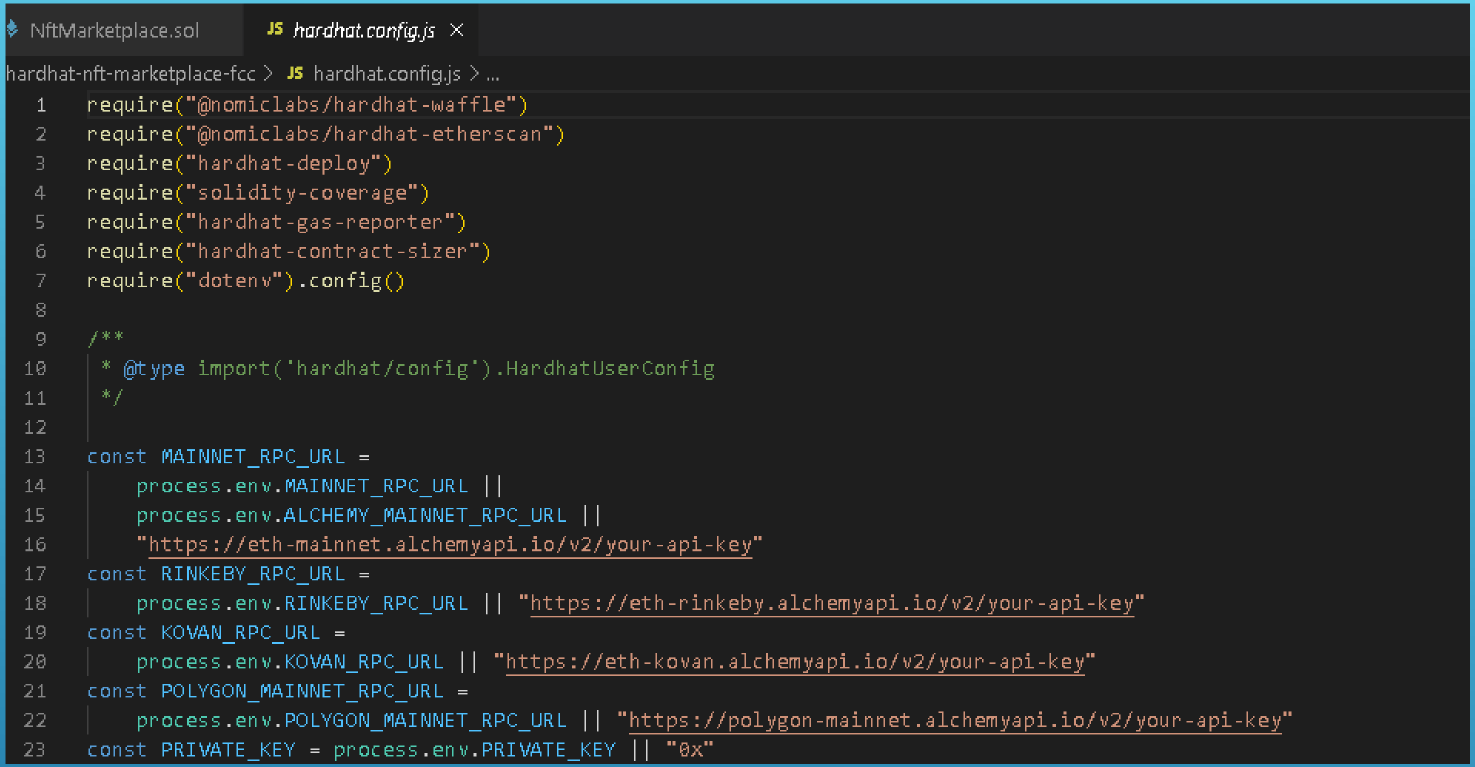


NftMarketplace.sol | 03-update-front-end.js

hardhat-nft-marketplace-fcc > deploy > 03-update-front-end.js > ...

```
const {
    frontEndContractsFile,
    frontEndContractsFile2,
    frontEndAbiLocation,
    frontEndAbiLocation2,
} = require("../helper-hardhat-config")
require("dotenv").config()
const fs = require("fs")
const { network } = require("hardhat")

module.exports = async () => {
    if (process.env.UPDATE_FRONT_END) {
        console.log("Writing to front end...")
        await updateContractAddresses()
        await updateAbi()
        console.log("Front end written!")
    }
}

async function updateAbi() {
    const nftMarketplace = await ethers.getContract("NftMarketplace")
    fs.writeFileSync(
        `${frontEndAbiLocation}NftMarketplace.json`,
        nftMarketplace.interface.format(ethers.utils.FormatTypes.json)
```

UPDATE FRONT END



NftMarketplace.sol | hardhat.config.js

hardhat-nft-marketplace-fcc > hardhat.config.js > ...

```js
require("@nomiclabs/hardhat-waffle")
require("@nomiclabs/hardhat-etherscan")
require("hardhat-deploy")
require("solidity-coverage")
require("hardhat-gas-reporter")
require("hardhat-contract-sizer")
require("dotenv").config()

/**
 * @type import('hardhat/config').HardhatUserConfig
 */

const MAINNET_RPC_URL =
    process.env.MAINNET_RPC_URL ||
    process.env.ALCHEMY_MAINNET_RPC_URL ||
    "https://eth-mainnet.alchemyapi.io/v2/your-api-key"
const RINKEBY_RPC_URL =
    process.env.RINKEBY_RPC_URL || "https://eth-rinkeby.alchemyapi.io/v2/your-api-key"
const KOVAN_RPC_URL =
    process.env.KOVAN_RPC_URL || "https://eth-kovan.alchemyapi.io/v2/your-api-key"
const POLYGON_MAINNET_RPC_URL =
    process.env.POLYGON_MAINNET_RPC_URL || "https://polygon-mainnet.alchemyapi.io/v2/your-api-key"
const PRIVATE_KEY = process.env.PRIVATE_KEY || "0x"
```

HARDHAT CONFIG