NFT Project:

-npm init

-npm init

-npm install --save-dev hardhat

Install openzeppeline:

-npm install @openzeppelin/contracts

Documentation for openzeppeline: https://docs.openzeppelin.com/contracts/4.x/access-control

Set up .env file:

-npm install dotenv –save

-then create env file

Installing ethersJS by hardhat:

-npm install --save-dev @nomiclabs/hardhat-ethers 'ethers@^5.0.0' –f

##solidity version had to set:

Solidity >=5.0.0 <9.0.0;

##needed to set up the hardhat-config file

##compile-- npx hardhat compile

##deploy:

* In ./scripts/deploy.js file edit it properly
* Then deploy on sepolia test net-- npx hardhat run scripts/deploy.js --network sepolia

## now have to connect the deployed contract with the image, create an NFT and transfer it in the metamask

🡪 first upload the image in the IPFS

🡪 then link it with the NFT smart-contract

* First created a file - In ./scripts/mint-nft.js file edit it properly

-node run scripts/mint-nft.js

🡪 Go to any IPFS upload the picture you want to add as nft

🡪 create a file in the root folder called “nft-metadata.json”

🡪 copy the IPFS ID and links of both of these. We will use this in the “mintNFT.js” file

🡪 then properly writing the code in the mintNFT.js file as written in this project

🡪 then

* Node run scripts/mintNFT.js

🡪 Due to some issues the hash number do not get printed. But we can check it in the sepolia etherscan by the help of the contract address. We can see all the transaction

🡪 in the metamask it may not added automatically. We need to add manually. Just give the contract address and the ID((e.g 1,2,3….)) we see in the etherscan.

🡪 can also send the NFT to another account.

🡪 the account owner may also not see it automatically. But when he adds it manually following the same process 🡪give the contract and the ID(e.g 1,2,3….)