**Deploy and Upgrade contracts using HARDHAT and OPENZEPPELIN**

* Install nodeJS
* npm install –save-dev hardhat
* mkdir “myproject” && cd myproject
* npx create-next-app myAppName (you can add css like “npm I bulma”)
* npx hardhat (select : Create a sample project)
* If you rerun this command , it will show you all the possible things you can do with hardhat.
* Npm add @openzeppelin/hardhat-upgrades
* Npm add @nomiclabs/hardhat-ethers ethers
* (add the packages in hardhat.config.js if not present “require(“@nomiclabs/hardhat-ethers”)” and “require(“@openzeppelin/hardhat-upgrades”)”)
* Add your contract.sol to the “contracts” folder.
* Add your contract2.sol to the “contracts” folder. (the new contract that the original will upgrade to, should we need it to)
* You can compile your contract using the “npx hardhat compile” command (this will create some JSON files in artifacts folder – this is where the contracts ABI is located)
* Create 2 new scripts inside /scripts “deploy.js” and “upgrade.js” (provided in GIT repo)
* Deploy your smart contract @localhost using the following commands:
* “npx hardhat node” and “npx hardhat run scripts/deploy.js --network localhost” (on separate command line windows). This will deploy 3 contracts – contract.sol implementation , proxyContract, transparentUpgradeableProject.
* Please note that IF you want to deploy it on another network you should add the “defaultNetwork” under “module.exports” inside hardhat.config.js , add the NETWORK\_RPC\_URL and PRIVATE\_KEY (you should change your command too, to “npx hardhat run scripts/deploy.js --network rinkeby”.
* If you want to UPGRADE your contract.sol to contract2.sol you want to run the following command “npx hardhat run scripts/upgrade.js --network localhost”