# Task 1: ERC20 Token Development & Deployment (Beginner Level) – Blockchain Domain

Brief Task Description:

* In this task, we were asked to build and deploy a basic smart contract following the ERC20 token contract.
* Before approaching this task, I knew a bit of solidity and blockchain and Ethereum theory and had zero knowledge of the procedure through which smart contracts are developed and deployed.
* I learned during this task that smart contracts require a special program just to verify them as they hold key transaction details.
* I started out the task by first learning the fundamentals of blockchain and Ethereum protocol.
* I preferred using Foundry over Hardhat for developing smart contracts because of my illiteracy in javascript (it would’ve been a lot of work learning both solidity and javascript if I had used Hardhat). I thus used Foundry – where smart contracts are written and tested in solidity. I also got to know the better utility of Foundry than Hardhat.
* The smart contract submitted is basic and performs simple operations of transferring and approving tokens of type ERC20 (fungible assets).
* Resources:

1. <https://www.youtube.com/watch?v=sas02qSFZ74>  
   For understanding the basics of ERC20 contracts
2. <https://youtube.com/playlist?list=PLgPmWS2dQHW-BRQCQCNYgmHUfCN115pn0&si=MdGPlrhCn5vfp2Bs>

For learning the fundaments of blockchain and Ethereum

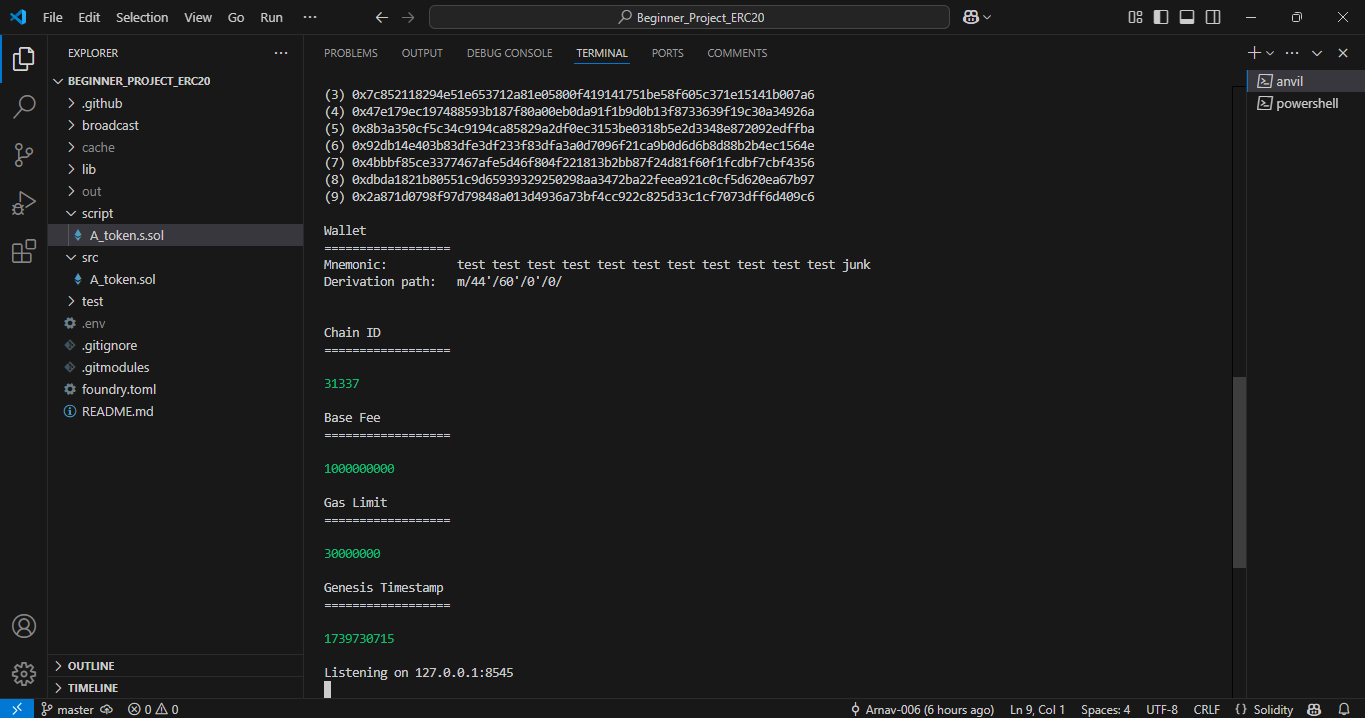
1. <https://cryptozombies.io/en/solidity?lessonId=1>  
   For learning solidity
2. <https://docs.alchemy.com/docs/how-to-add-sepolia-to-metamask#3-add-sepolia-to-metamask>

For obtaining the Sepolia RPC URL

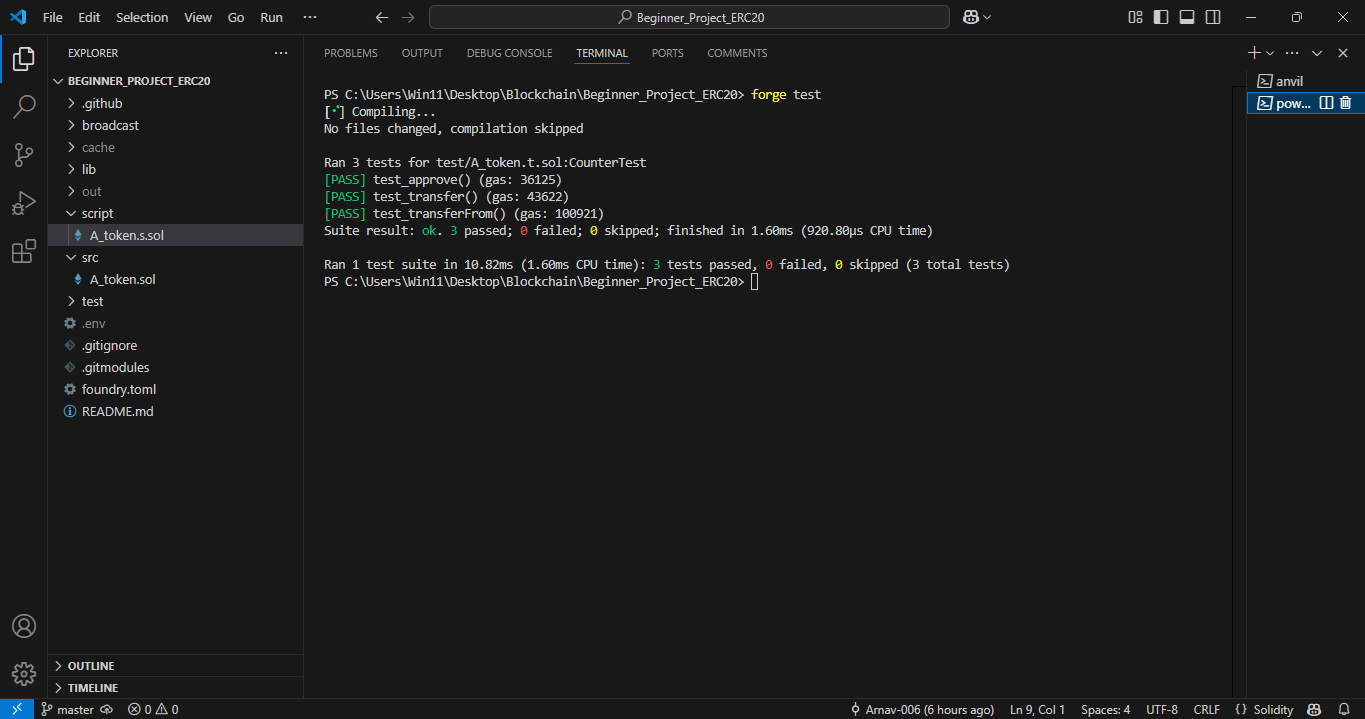
1. MetaMask (crypto wallet)
2. Google Cloud Web3 (for obtaining free ethers)
3. Chat GPT (Foundry installation guide)
4. Copilot (helped me with the syntax) 😊

Related Screenshots:

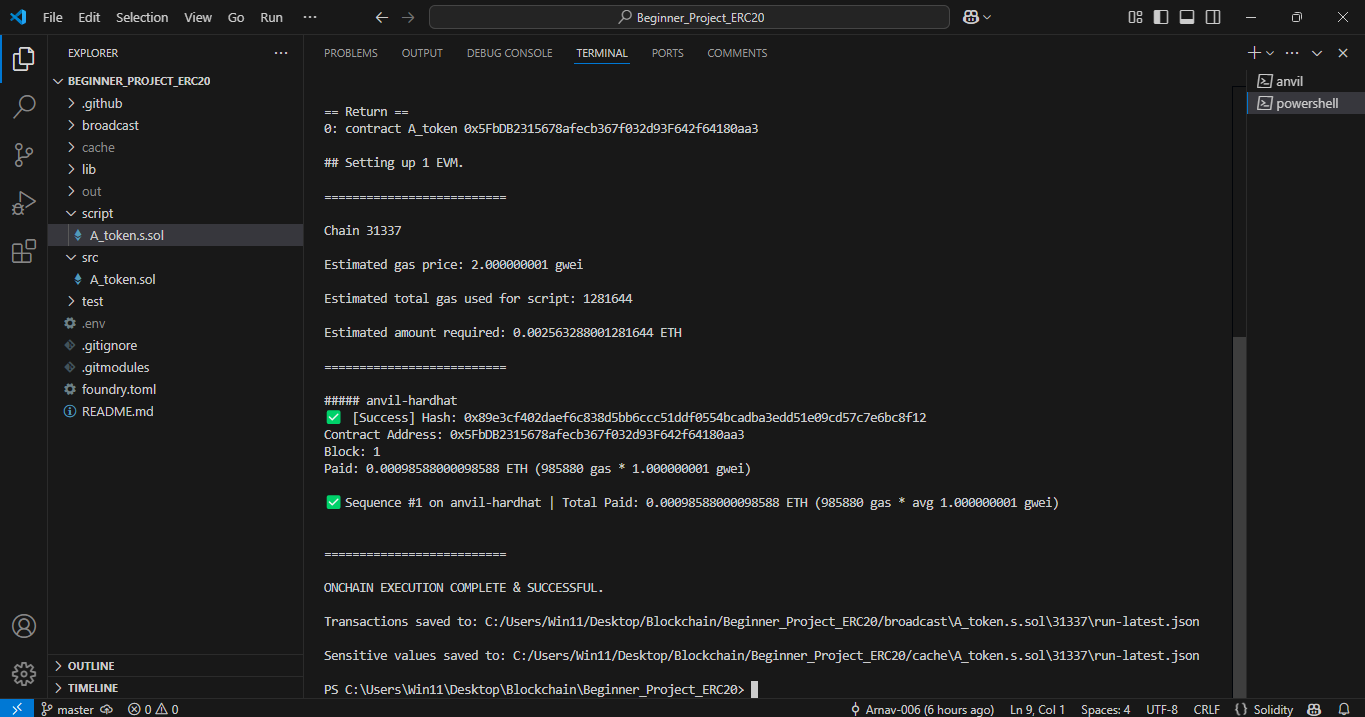
* Local test net anvil setup:



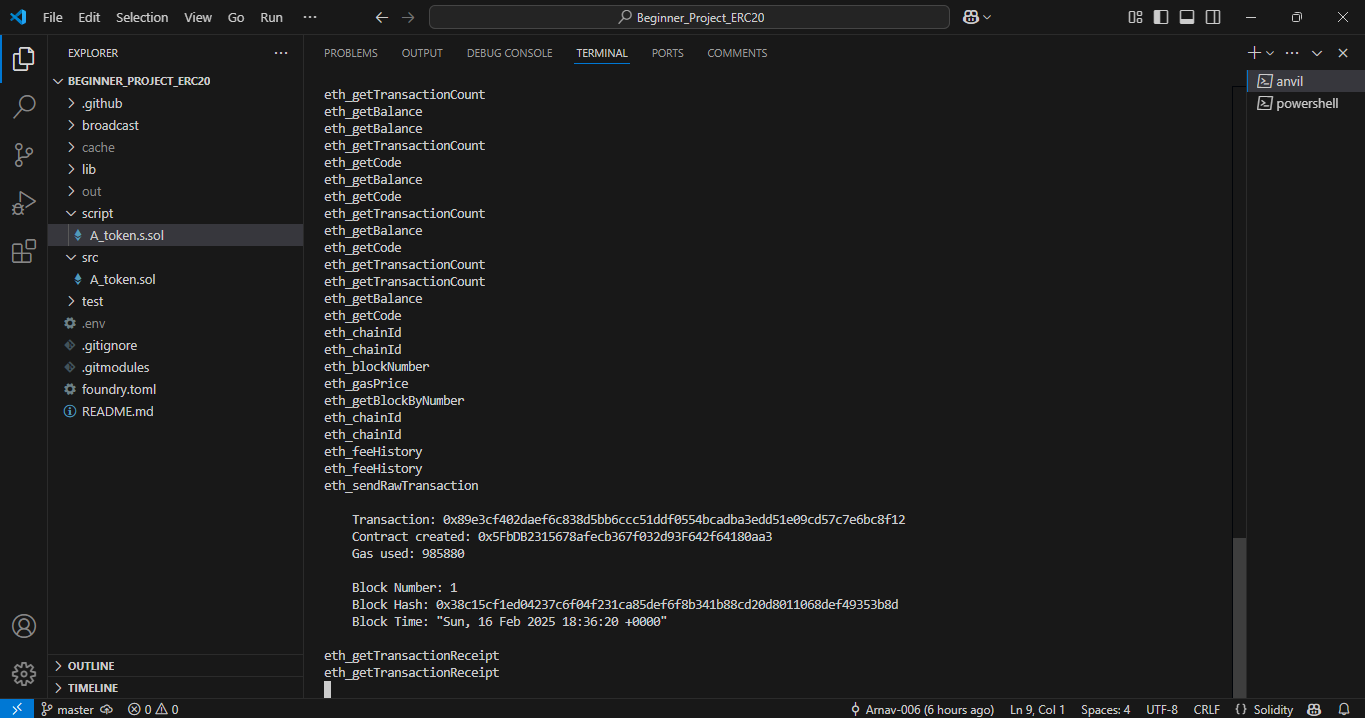
* 3 tests conducted successfully:



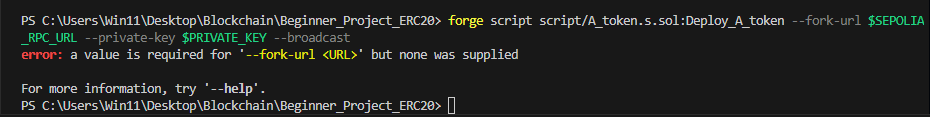
* Contract successfully deployed on local test net (using first private key of the 10 private keys provided on anvil)



* Transaction receipt



* Deployment on public test net failed due to lack of time to resolve the errors.



Instructions to Run the Project:

* forge build

This compiles the smart contract, making it ready for testing.

* anvil

Sets up anvil – local test net

* forge test

Runs all the tests

* forge script script/A\_token.s.sol:Deploy\_A\_token –rpc url (continued)

Deploys the smart contract using the specified private key.

# Task-2: ERC721 NFT DEVELOPMENT & DEPLOYMENT (Intermediate Level)

Brief Task Description:

* In this task, we were asked to build and deploy a basic smart contract following ERC721 token contract.
* Task included developing functions to interact with ERC721, which are essentially non-fungible assets.
* Task completion pending due to dearth of time.
* Resources:

1. <https://docs.openzeppelin.com/contracts/2.x/erc721>

(for obtaining source code)