🚀 **Internship Title:**
Blockchain & Crypto Internship

📘 **Task:**
Task 2 – Build a Free Crypto Token on Polygon or BNB Testnet

📌 **Track Code:**
BC (Blockchain & Crypto)

📜 **Task Objective:**
To write a custom ERC-20 token using Solidity and deploy it on the **BNB Smart Chain Testnet** using **Remix IDE**, **MetaMask**, and **OpenZeppelin**. This task builds understanding of smart contract development, testnet deployment, and Web3 tooling.

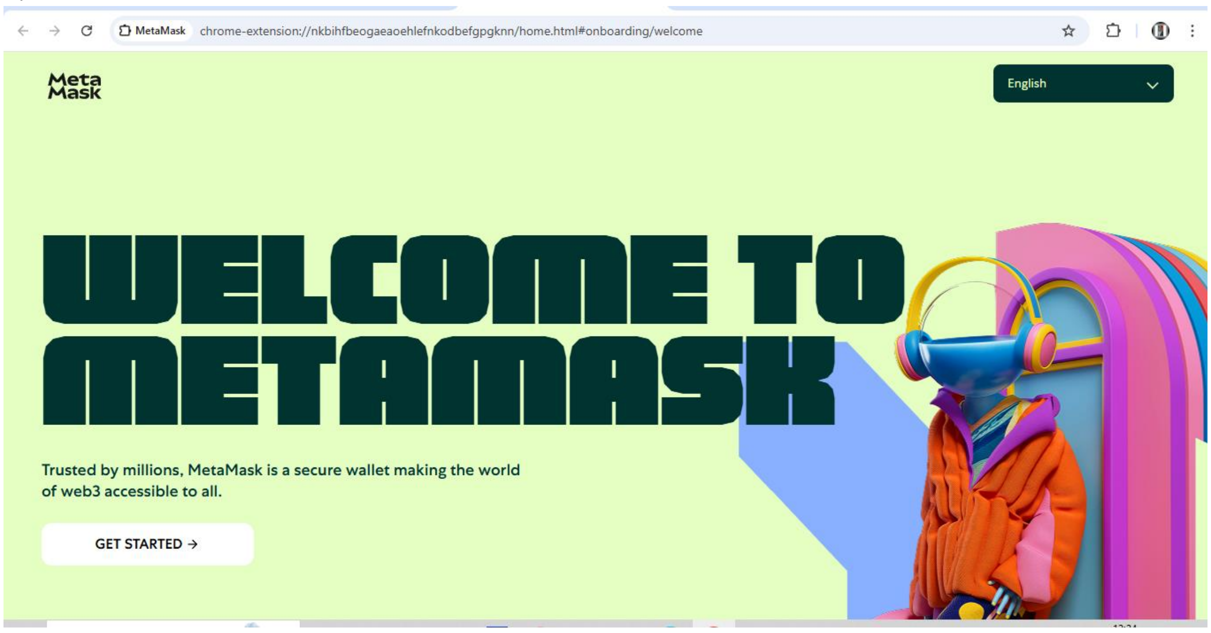
👦💻 **Intern Name:**
Avichal Avneel Nath

🆔 **CIN ID:**
FIT/JUL25/BC726

📍 **Token Details:**

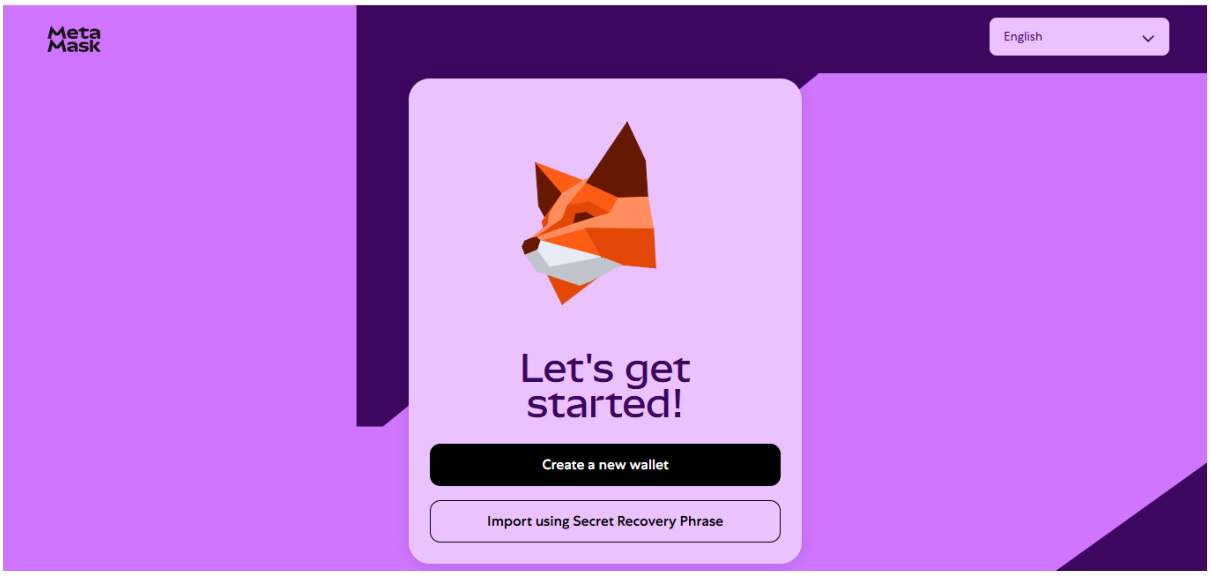
- **Token Name:** Avichal_Token
- **Symbol:** MTK
- **Total Supply:** 1,000,000 MTK
- **Deployed Network:** BNB Smart Chain Testnet

1) Metamask:

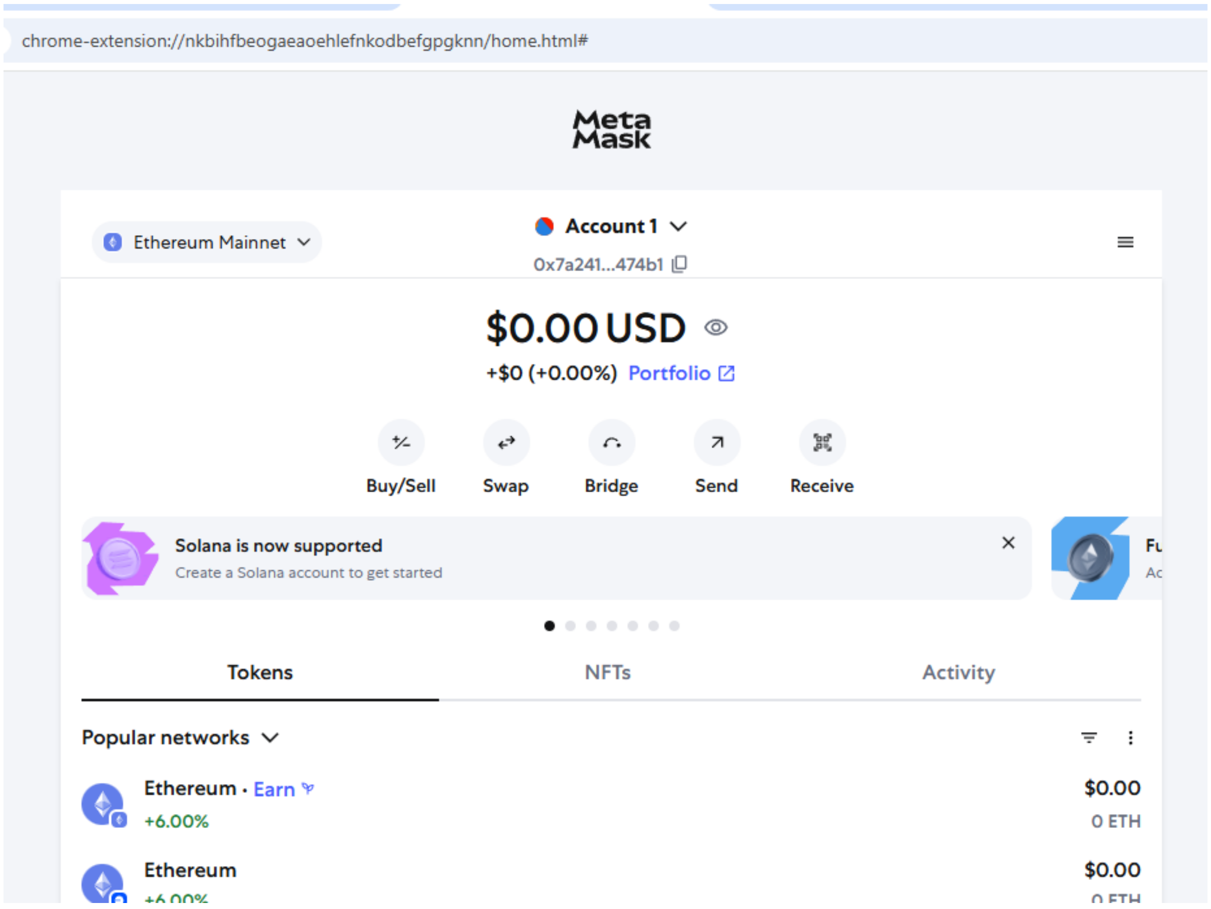

MetaMask
chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/home.html#onboarding/welcome
Meta Mask
English
WELCOME TO METAMASK
GET STARTED →

Trusted by millions, MetaMask is a secure wallet making the world of web3 accessible to all.

2) create a wallet:



3) wallet created with advanced security measures:



chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/home.html#

$0.00 USD

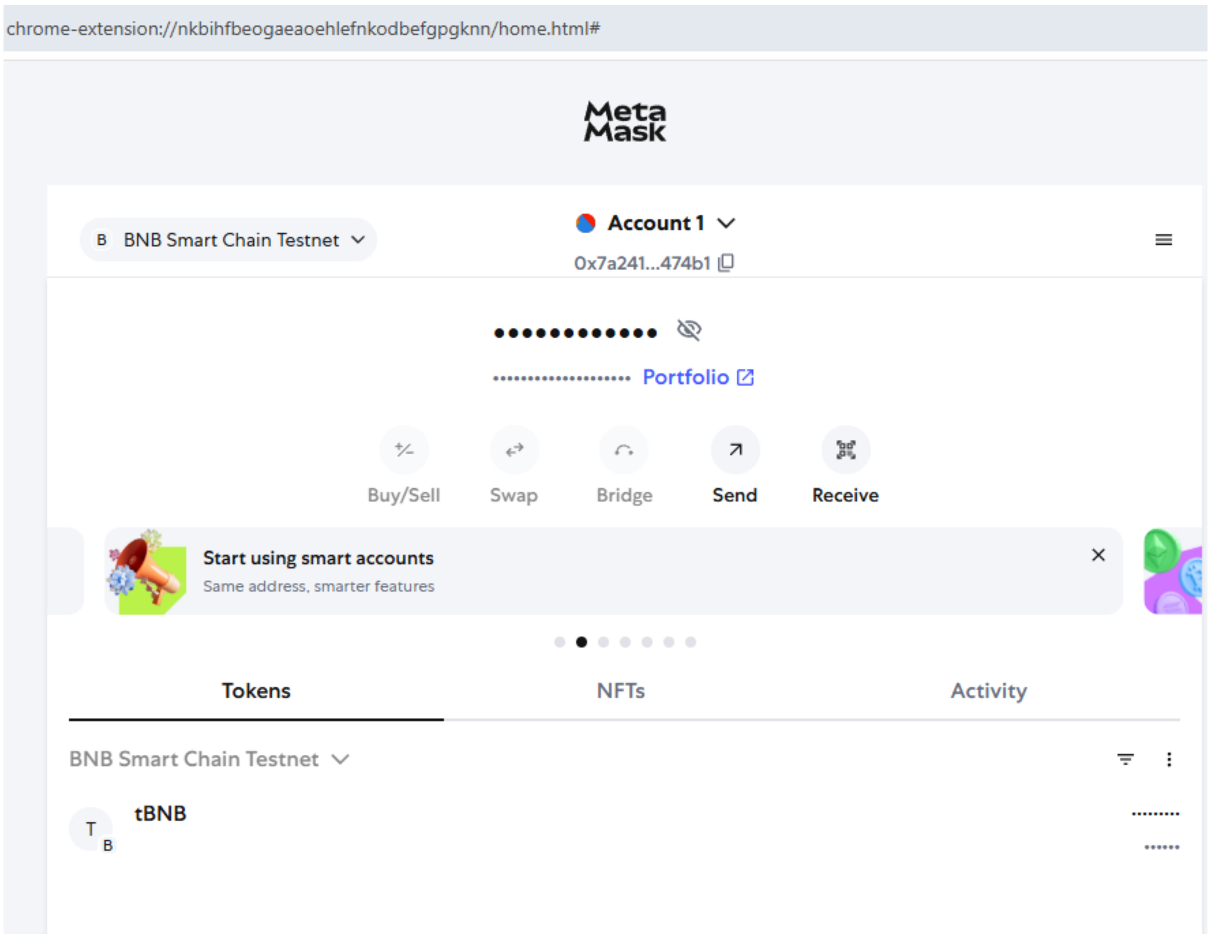
+$0 (+0.00%) Portfolio

Solana is now supported

Create a Solana account to get started

Ethereum · Earn +6.00%	$0.00 0 ETH
Ethereum +6.00%	$0.00 0 ETH

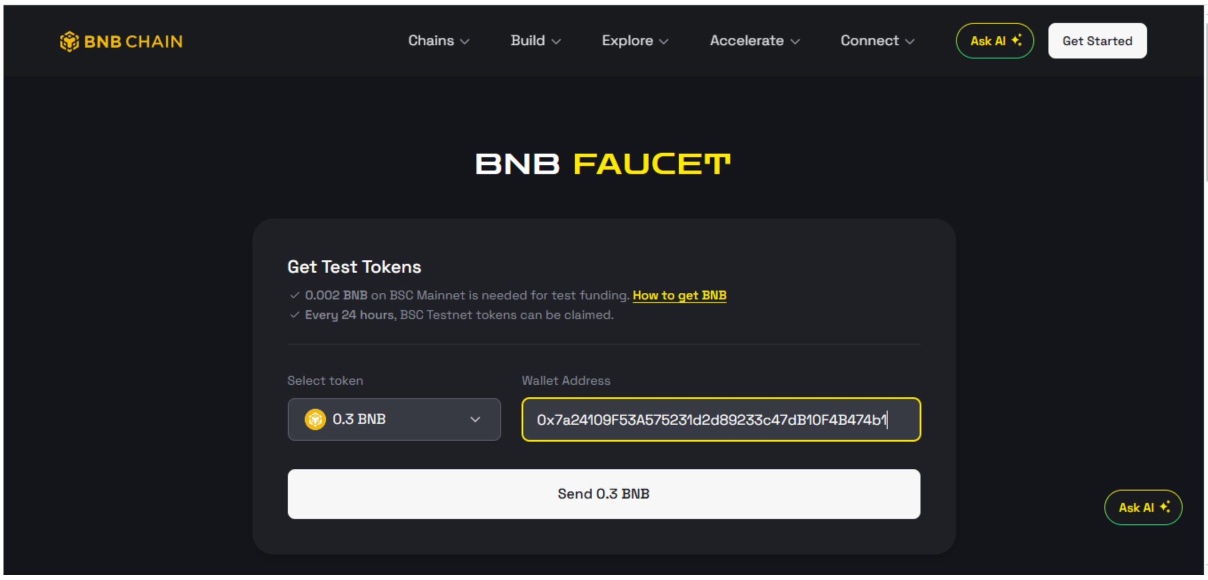
4) BNB smart chain testnet created using custom credentials


chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/home.html#

Start using smart accounts
Same address, smarter features

tBNB ••••••••• ••••••

5) Testing BNB to pay for contract deployment gas



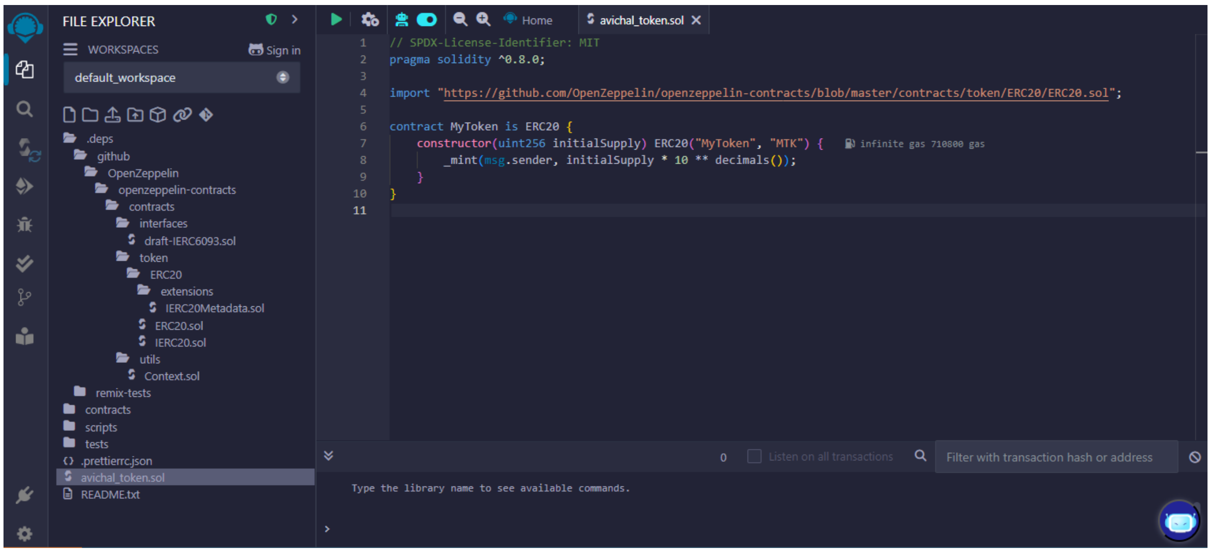
Get Test Tokens

✓ 0.002 BNB on BSC Mainnet is needed for test funding. How to get BNB
✓ Every 24 hours, BSC Testnet tokens can be claimed.

Select token: 0.3 BNB

Wallet Address: 0x7a24109F53A575231d2d89233c47dB10F4B474b1

6) writing, compiling, and deploying smart contracts for custom token: avichal_token



```
.deps
  github
    OpenZeppelin
      openzeppelin-contracts
        contracts
          interfaces
            draft-IERC6093.sol
          token
            ERC20
              extensions
                IERC20Metadata.sol
              ERC20.sol
              IERC20.sol
          utils
            Context.sol
  remix-tests
contracts
scripts
tests
.prettierrc.json
avichal_token.sol
README.txt
```

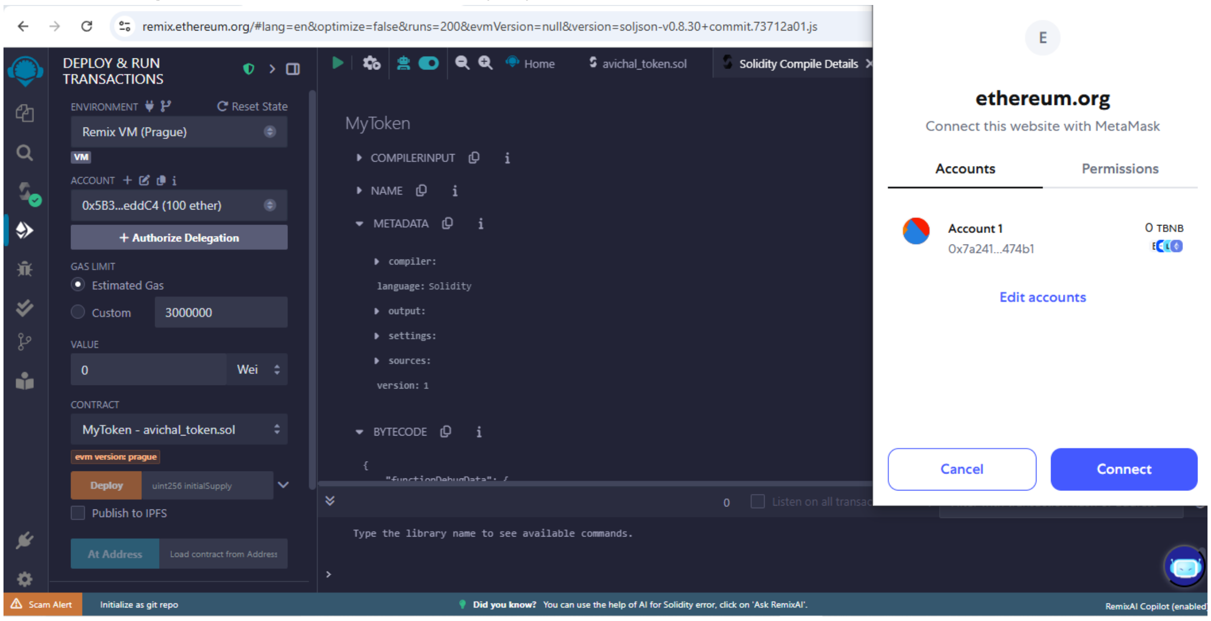
```solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC20/ERC20.sol";

contract MyToken is ERC20 {
    constructor(uint256 initialSupply) ERC20("MyToken", "MTK") {    infinite gas 710800 gas
        _mint(msg.sender, initialSupply * 10 ** decimals());
    }
}
```

```
0   Listen on all transactions   Filter with transaction hash or address
Type the library name to see available commands.
>
```

7) connecting metmask and remix for deployment of contract:


remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.30+commit.73712a01.js
DEPLOY & RUN TRANSACTIONS
Home
avichal_token.sol
Solidity Compile Details
Reset State
+ Authorize Delegation
Deploy
Publish to IPFS
At Address
Load contract from Address
Scam Alert
Initialize as git repo
RemixAI Copilot (enabled)

ENVIRONMENT: Remix VM (Prague)

VM

ACCOUNT: 0x5B3...eddC4 (100 ether)

GAS LIMIT
- (●) Estimated Gas
- ( ) Custom 3000000

VALUE: 0 Wei

CONTRACT: MyToken - avichal_token.sol

evm version: prague

Deploy: uint256 initialSupply

MyToken

```
COMPILERINPUT
NAME
METADATA
  compiler:
  language: Solidity
  output:
  settings:
  sources:
  version: 1
BYTECODE
  {
    "functionDebugData": {
```

```
0   Listen on all transa
Type the library name to see available commands.
>
```

Did you know? You can use the help of AI for Solidity error, click on 'Ask RemixAI'.

E

ethereum.org

Connect this website with MetaMask

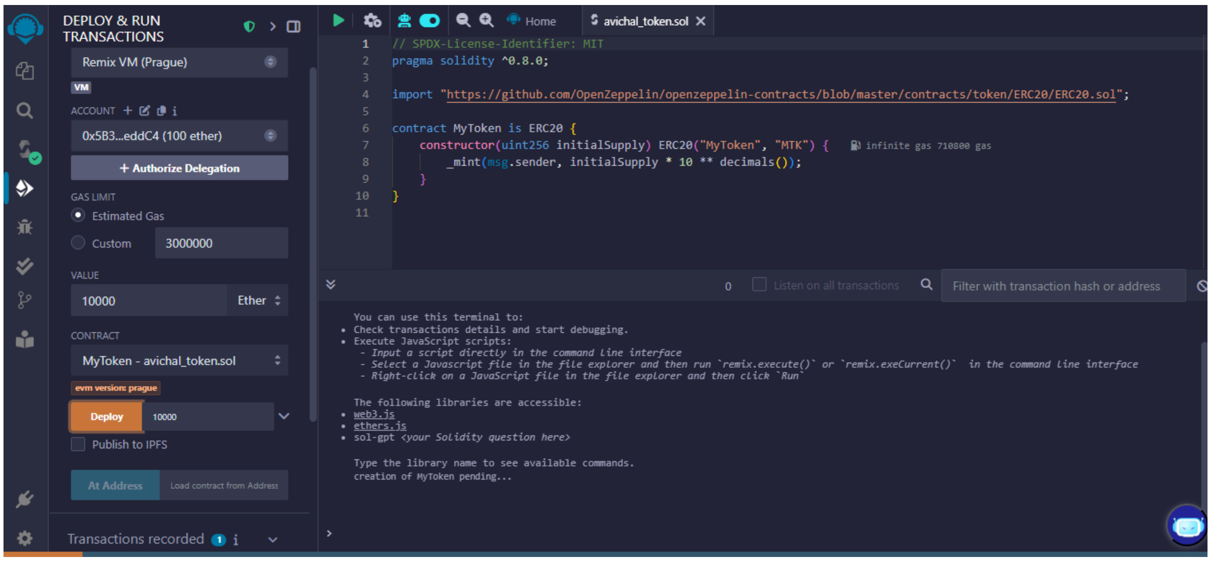
Accounts | Permissions

Account 1 — 0x7a241...474b1 — 0 TBNB

Edit accounts

Cancel | Connect

8)Deploying contract:



```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC20/ERC20.sol";

contract MyToken is ERC20 {
    constructor(uint256 initialSupply) ERC20("MyToken", "MTK") {    infinite gas 710800 gas
        _mint(msg.sender, initialSupply * 10 ** decimals());
    }
}
```

```
 You can use this terminal to:
• Check transactions details and start debugging.
• Execute JavaScript scripts:
  - Input a script directly in the command line interface
  - Select a Javascript file in the file explorer and then run `remix.execute()` or `remix.exeCurrent()`  in the command line interface
  - Right-click on a JavaScript file in the file explorer and then click `Run`

 The following libraries are accessible:
• web3.js
• ethers.js
• sol-gpt <your Solidity question here>

 Type the library name to see available commands.
 creation of MyToken pending...
```