**ERC Tokens lab**

**Successfully logged into Metamask can see we still have some BNB**

**A screenshot of a computer

Description automatically generated**

**Visited** [**https://faucets.chain.link/**](https://faucets.chain.link/) **and created a github account and sent some Ethereum Sepolia 0.1ETH**

**A screenshot of a computer

Description automatically generated**

**We pasted some code into Remix-Ethereum IDE** [**https://remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.26+commit.8a97fa7a.js**](https://remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.26+commit.8a97fa7a.js)

ERC20 Token is a token standard for creating Ethereum Token contracts

ERC stands for Ethereum Request for Comments

Any token contract that adheres to the specification follows ERC (Can use with any wallets that understand ERC. Most ICOs (Initial Coin Offerings) adhered to ERC20.

**We Create and Deploy an ERC20 Token**

**A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generatedPasted code from Interfacesol.**

**andCoinInterface.sol**

**A screenshot of a computer program

Description automatically generated**

**andCoin.sol**

A screen shot of a computer

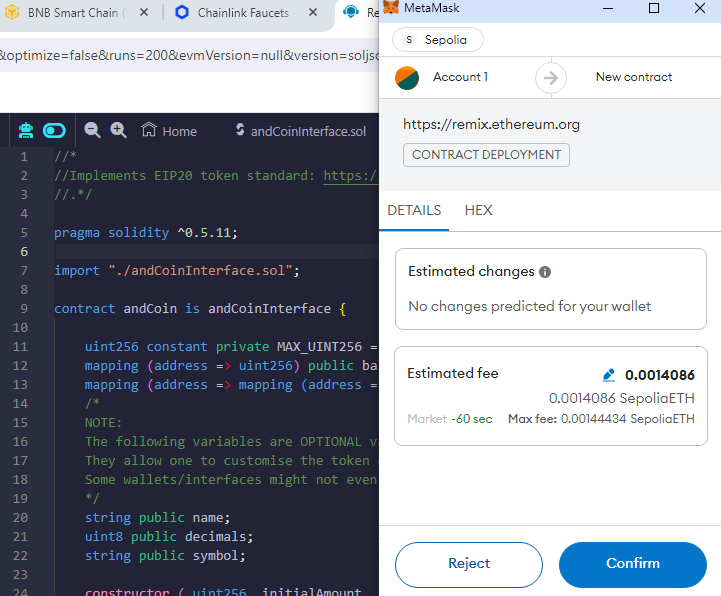
Description automatically generated

**Both contracts have compiled without errors**

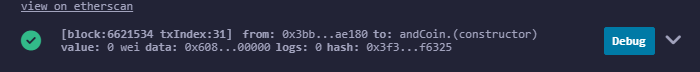
**A screenshot of a computer

Description automatically generated**

**These are the settings for the customized andCoin**

****

**Below message confirms the creation of the andCoin (after paying the fee to deploy the contract**



**Successfully copied the address**

0x5e04bDA72Edbed2e26ea38b44ecff3a6702d1A27

**A screenshot of a computer

Description automatically generatedChecking the address of the new contract andCoin**

**Using metamask address (**0x3bB59688853227ff584f2D25Fd0bD75a9f3Ae180**)**

**This was entered into balanceOf**

A screenshot of a computer program

Description automatically generated