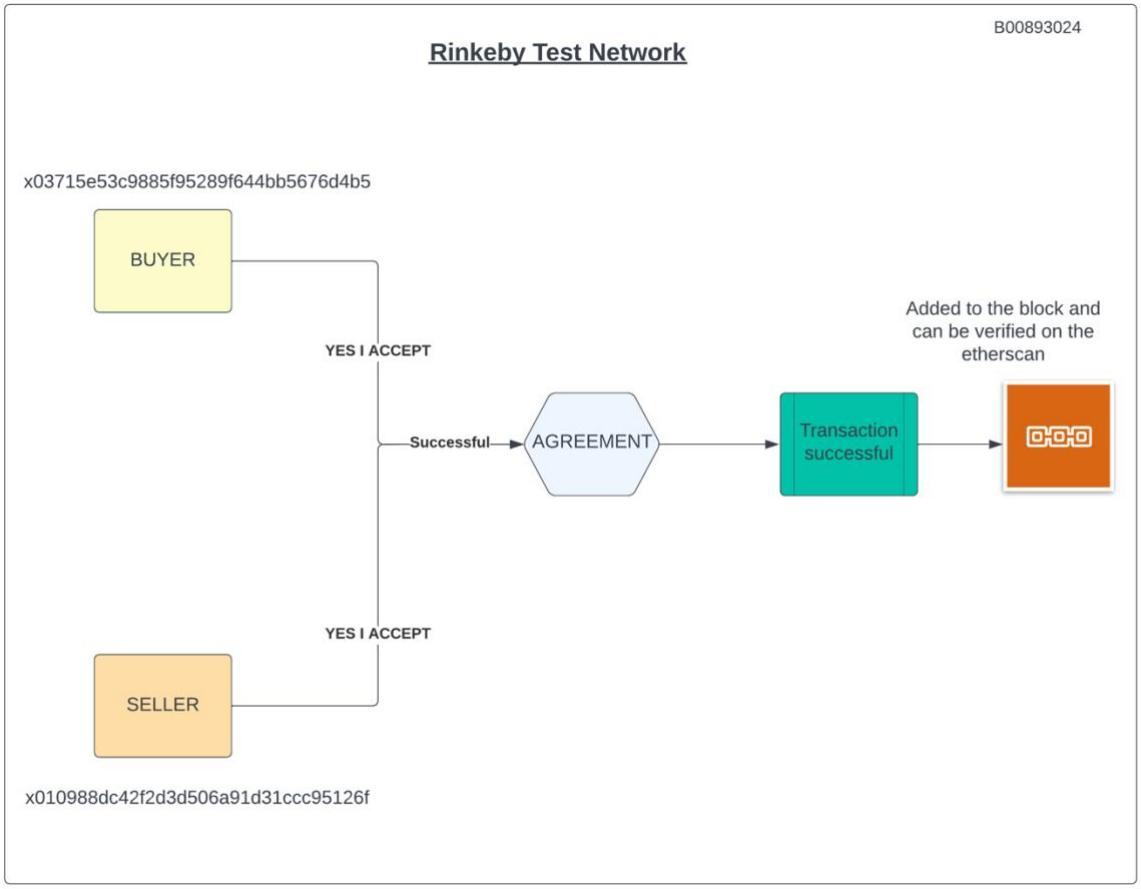
Assignment 1 - part B

Table of Contents :

Introduction

The approach followed the procedural algorithm where the smart contracts act as a notary and helps in a successful transaction. Below is the flow diagram which specifies the process followed :



Environment and Assumptions

Development Environment

1. Visual Studio Code
2. Truffle CLI
3. Ganache GUI
4. MetaMask Chrome plugin
5. Rinkeby Test Ethers from TestNet
6. Etherscan
7. Infura Platform
8. Gitlab

Gitlab Link to Submission :

https://git.cs.dal.ca/ayushv/csci_6313_verma_ayush_b00893024

Screenshots :

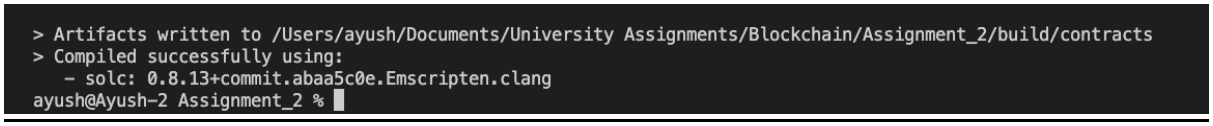


Figure 1 – Successful Compilation using truffle compile

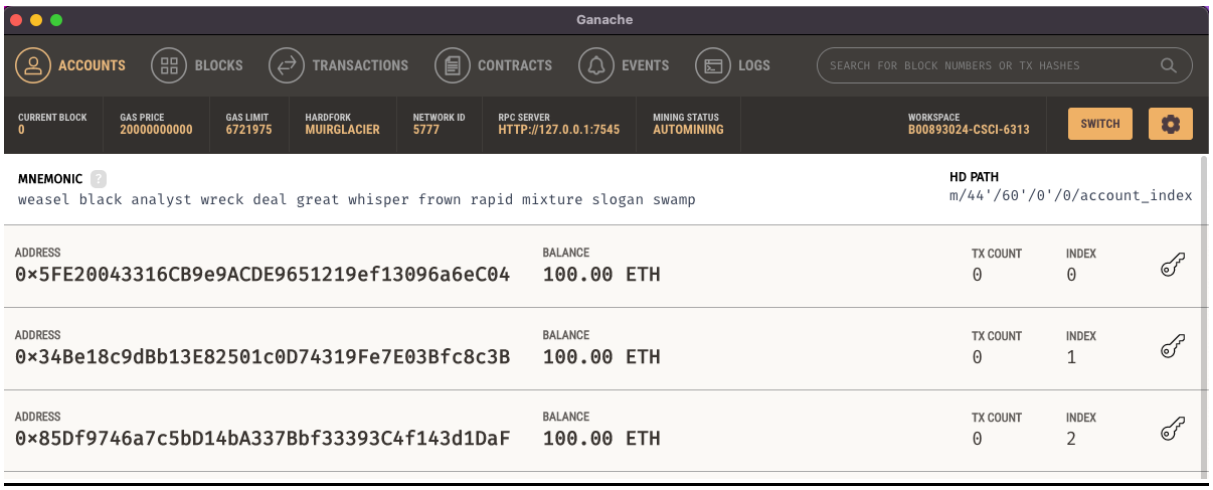


Figure 2 – Ganache GUI

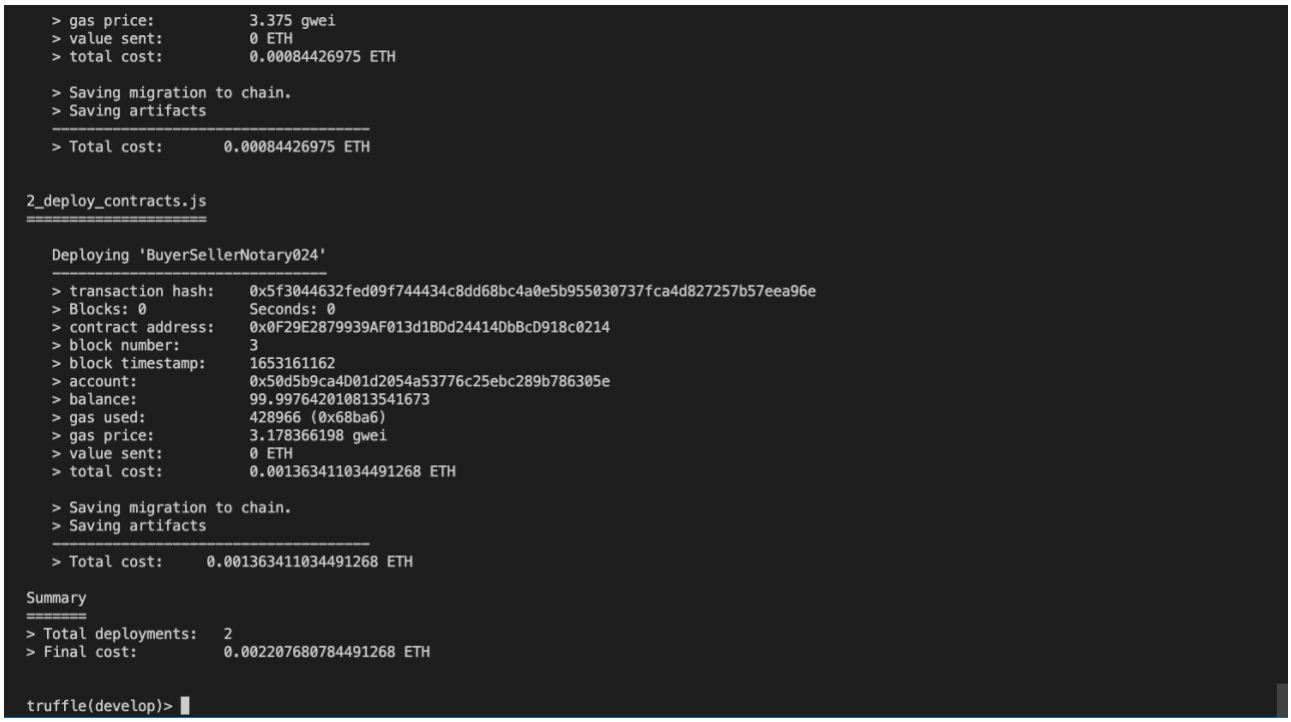


Figure 3 – Local Deployment of the Smart contract-1

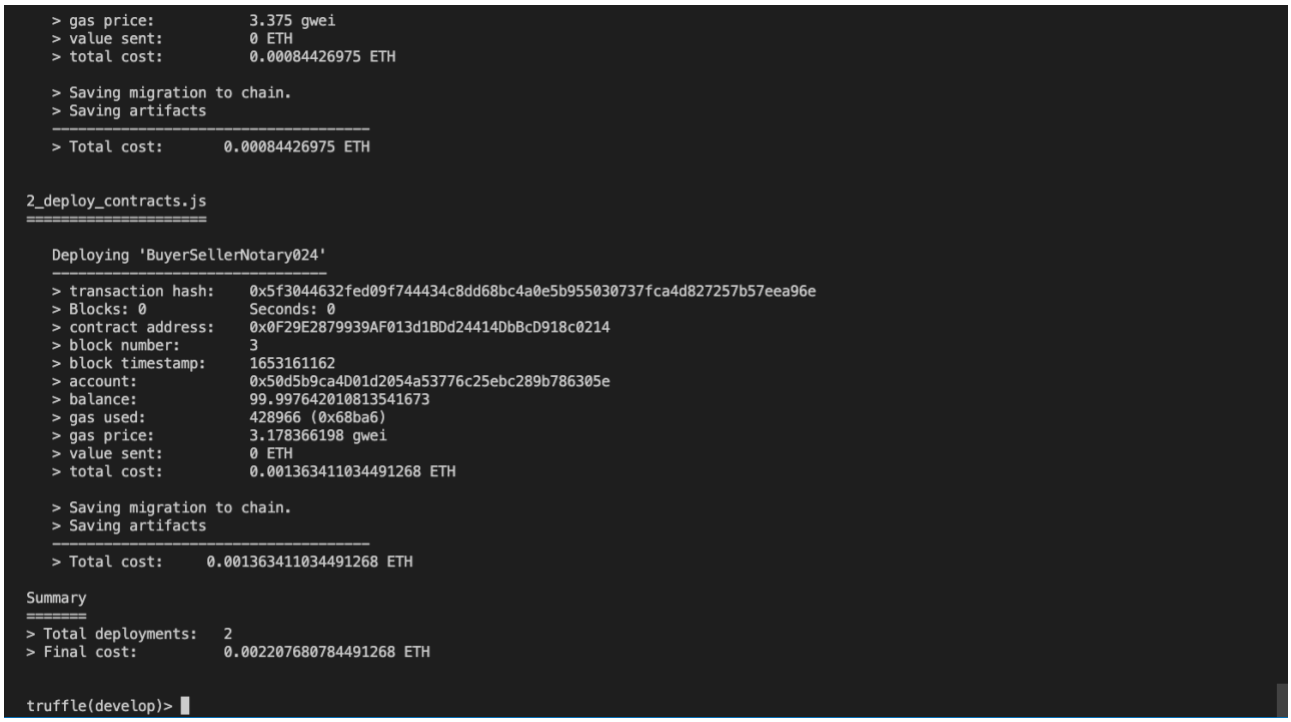


Figure 4 – Local Deployment of the Smart contract-2

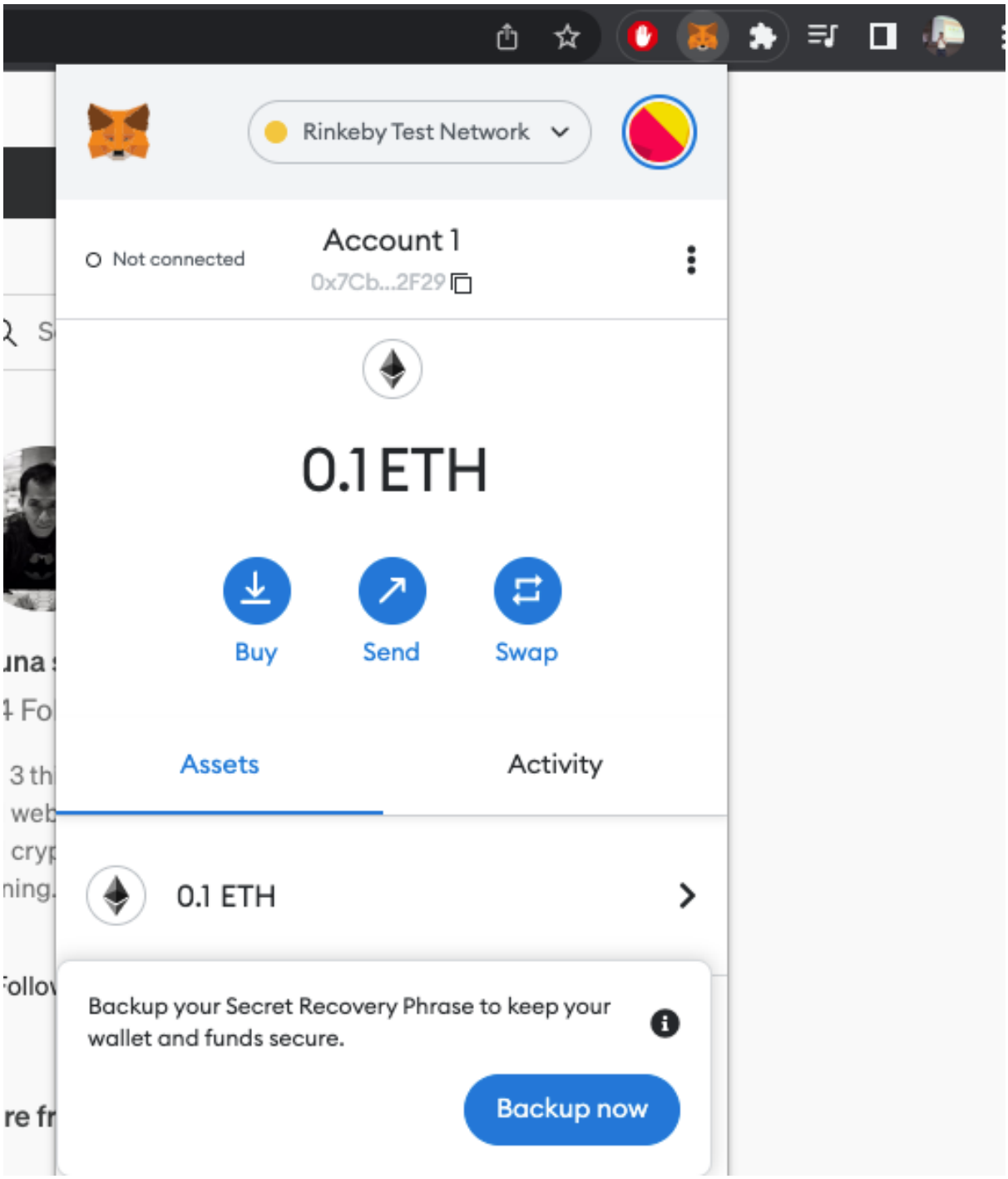


Figure 5 – Metamask wallet

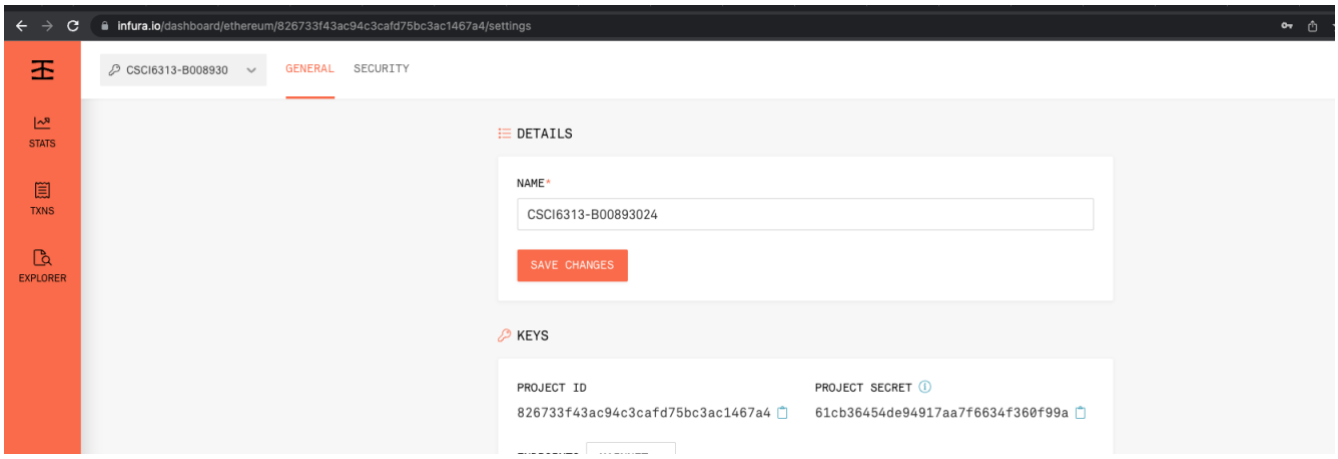


Figure 6 – Infura blockchain client workspace

```
> Compiled successfully using:
   - solc: 0.8.13+commit.abaa5c0e.Emscripten.clang


Migrations dry-run (simulation)
===============================
> Network name:    'rinkeby-fork'
> Network id:      4
> Block gas limit: 29999972 (0x1c9c364)


2_deploy_contracts.js
=====================

   Deploying 'BuyerSellerNotary024'
   --------------------------------
   > block number:        10716405
   > block timestamp:     1653162237
   > account:             0x7Cb51C6B92B7d0b5A55CbeDEcACE00f2f6B02F29
   > balance:             0.098927584996568272
   > gas used:            428966 (0x68ba6)
   > gas price:           2.500000008 gwei
   > value sent:          0 ETH
   > total cost:          0.001072415003431728 ETH

   ------------------------------------
   > Total cost:     0.001072415003431728 ETH

Summary
=======
> Total deployments:   1
> Final cost:          0.001072415003431728 ETH
```

Figure 7 – Deploying on the Rinkeby network

```
> Final cost:          0.001072415003431728 ETH



Starting migrations...
======================
> Network name:    'rinkeby'
> Network id:      4
> Block gas limit: 29999972 (0x1c9c364)


2_deploy_contracts.js
=====================

   Deploying 'BuyerSellerNotary024'
   --------------------------------
   > transaction hash:    0x8257358907638e4bea2b53b1ae572b0bee2fbe920a59c84b4097e5f31ddf2314
   > Blocks: 0            Seconds: 12
   > contract address:    0xc55FCcC0cD54DFB6ae1a3766e5bC18759E8cE175
   > block number:        10716410
   > block timestamp:     1653162253
   > account:             0x7Cb51C6B92B7d0b5A55CbeDEcACE00f2f6B02F29
   > balance:             0.09952813739613930 6
   > gas used:            428966 (0x68ba6)
   > gas price:           1.100000009 gwei
   > value sent:          0 ETH
   > total cost:          0.000471862603860694 ETH

   > Saving artifacts
   -------------------------------------
   > Total cost:     0.000471862603860694 ETH
```

Figure 8 – Deploying on the Rinkeby network

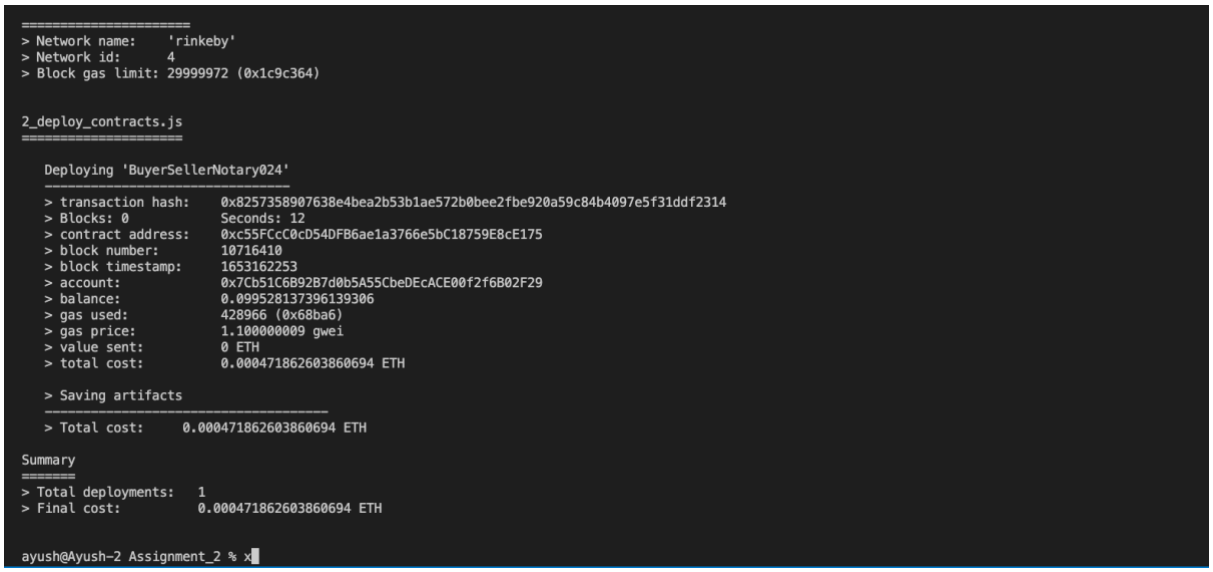


Figure 9 – Deploying on the Rinkeby network

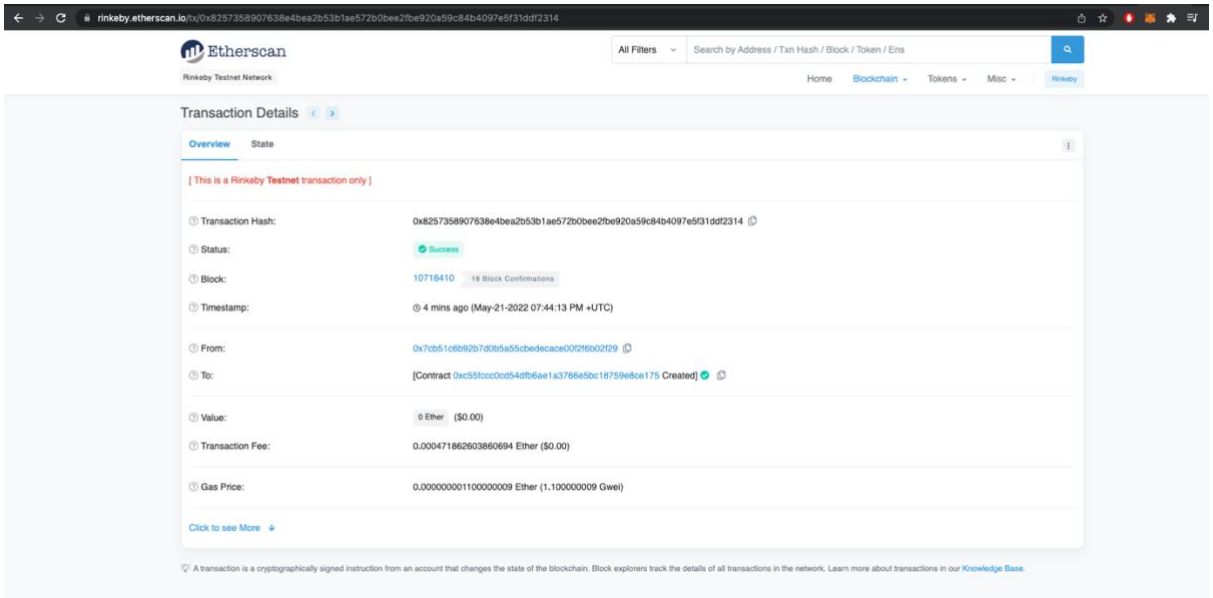


Figure 10 – Etherscan transaction

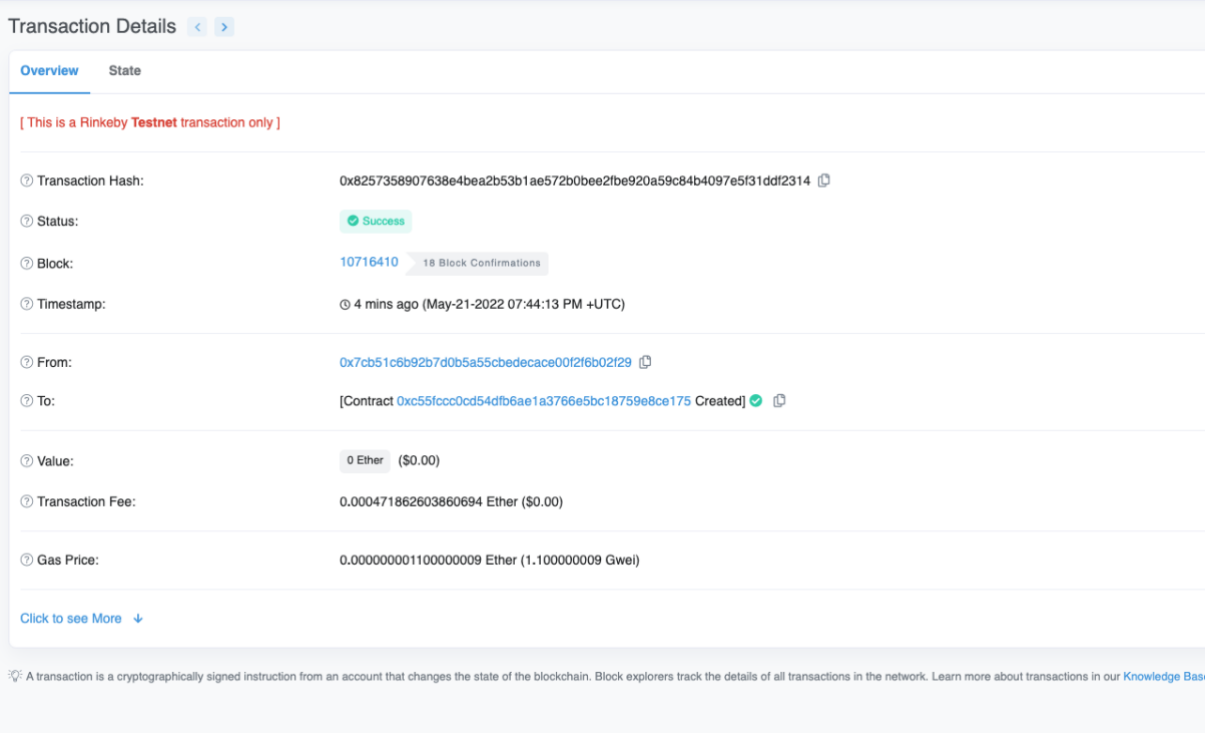


Figure 11 – Etherscan transaction of the acceptance

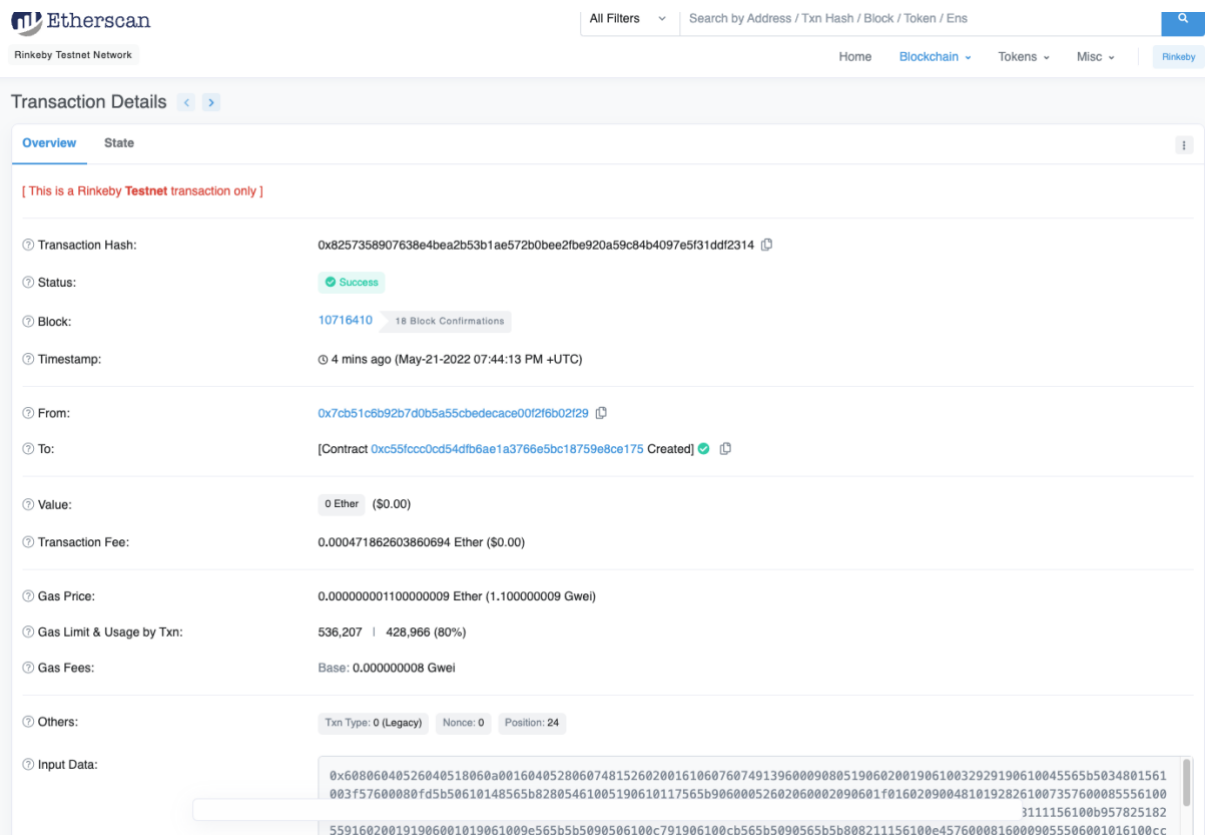


Figure 11 – Etherscan transaction for viewing the notary

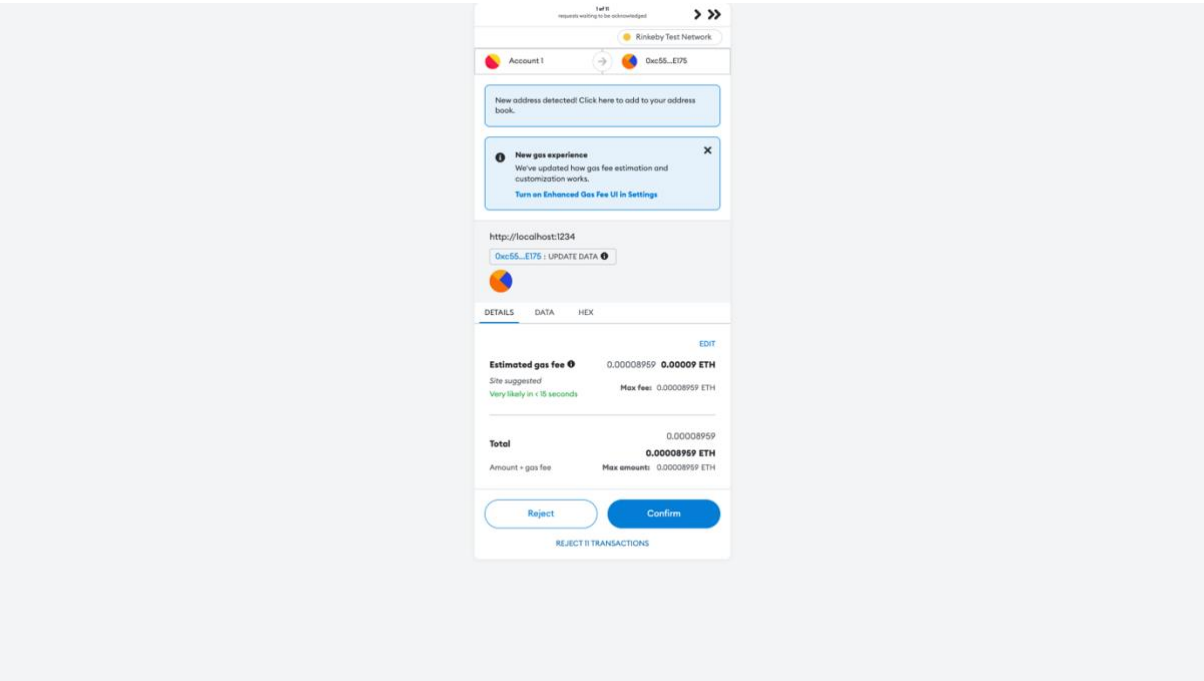


Figure 12 – Integration of the Metamask with localhost

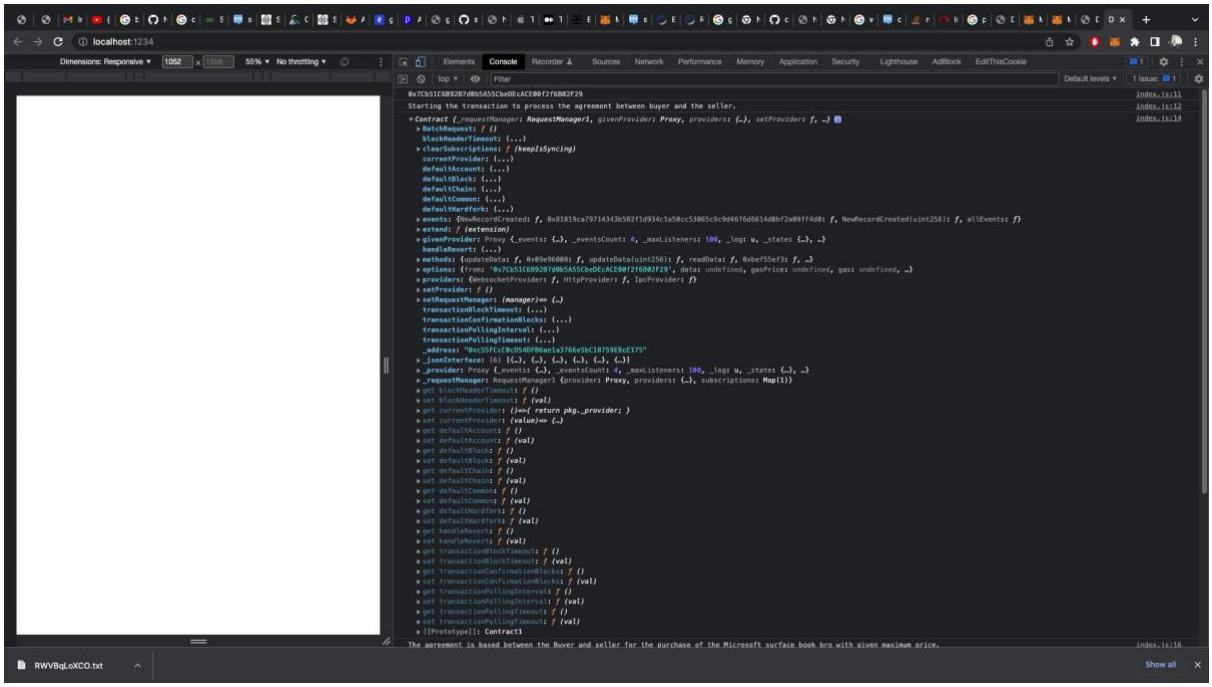


Figure 13 – DAPP running in the browser using Parcel

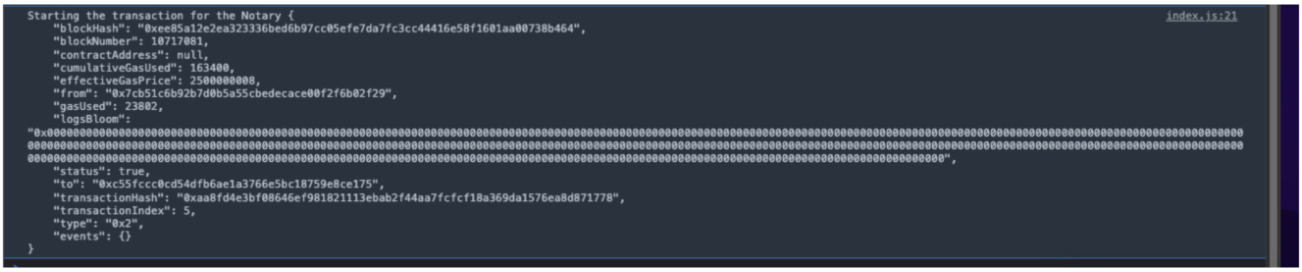


Figure 14 – DAPP running in the browser using Parcel

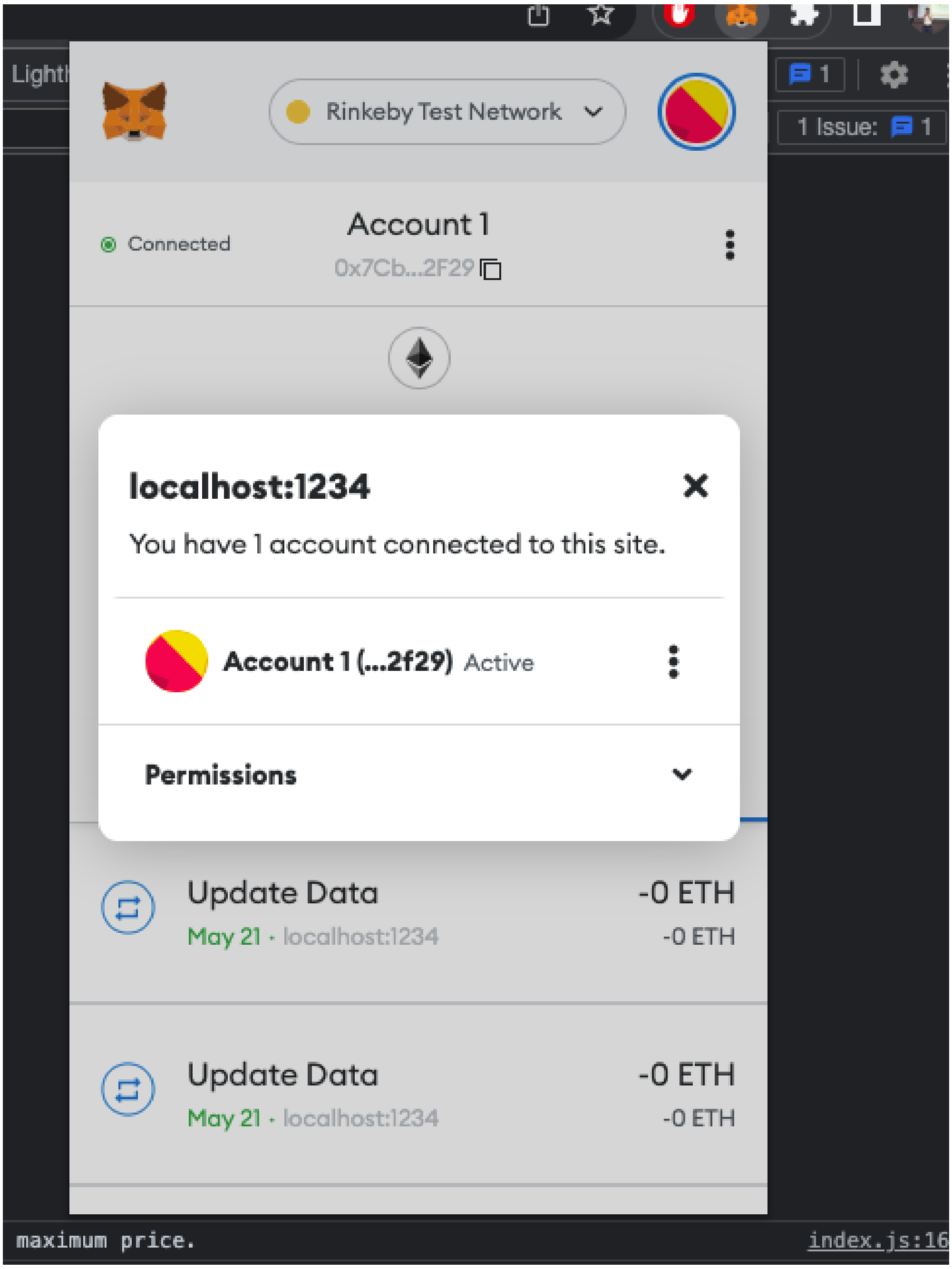


Figure 15 – Connecting Metamask with Transaction to test

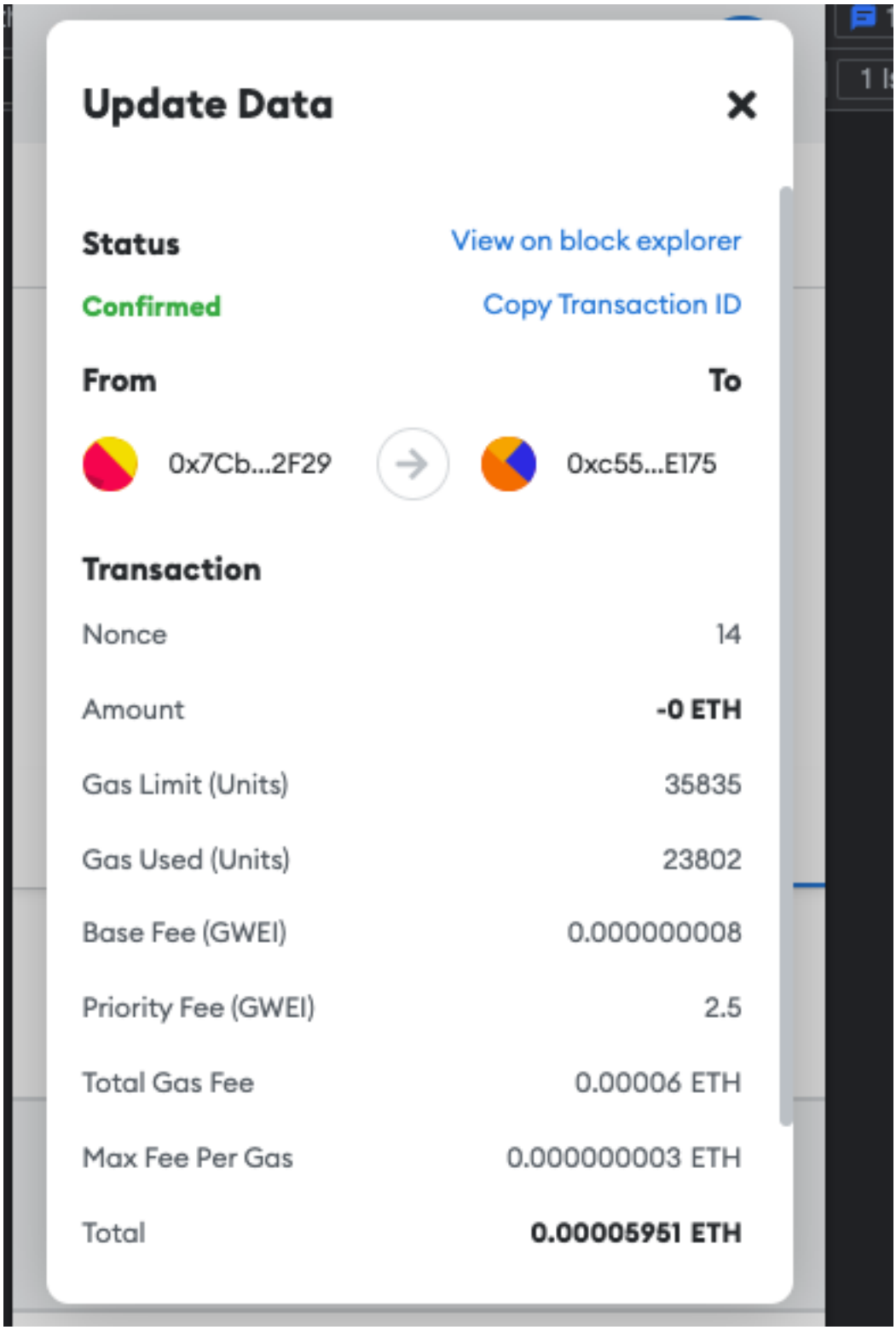


Figure 16 – Transaction running along with the processed gas

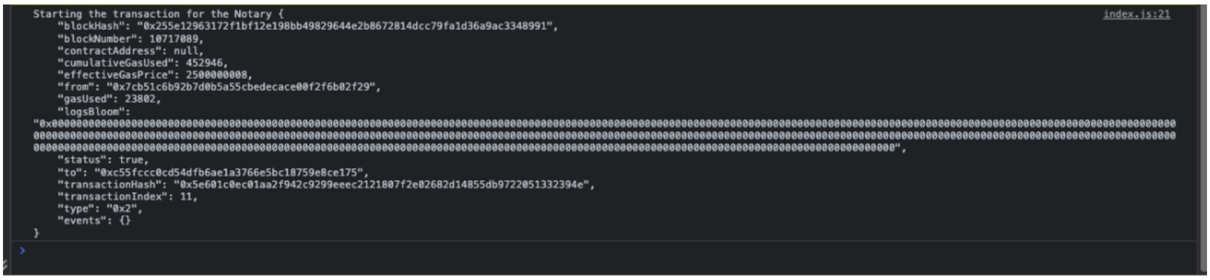


Figure 17 – Transaction running in the browser along with the status

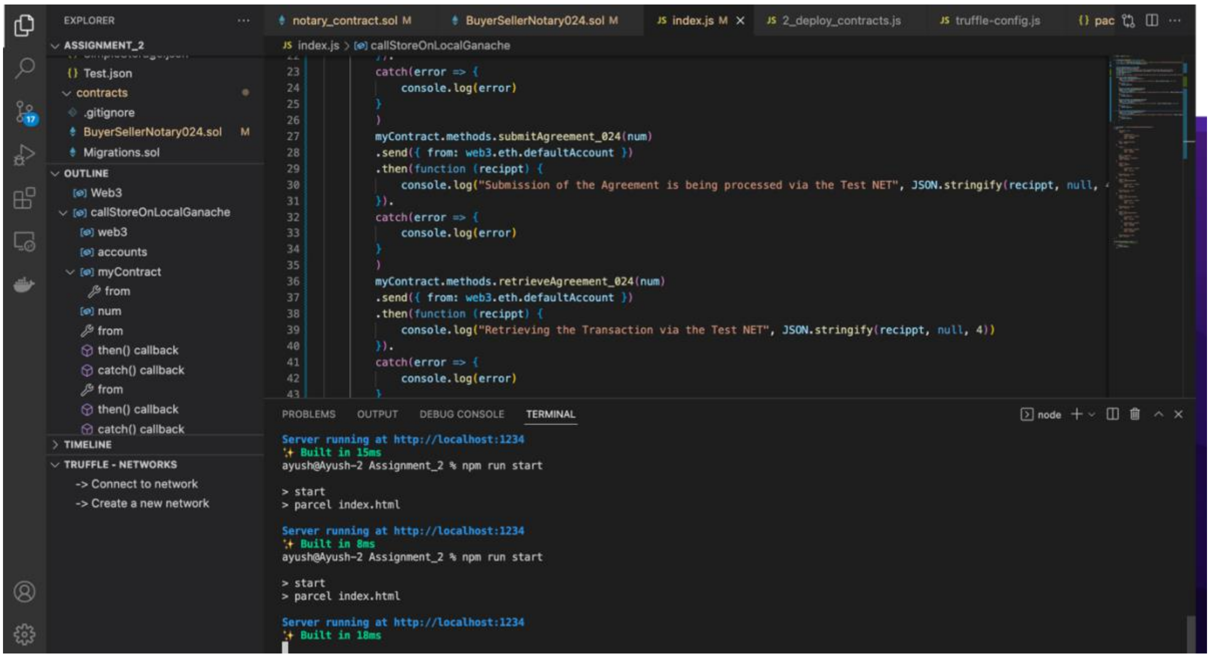


Figure 18 – Parcel running the server using the Node backend

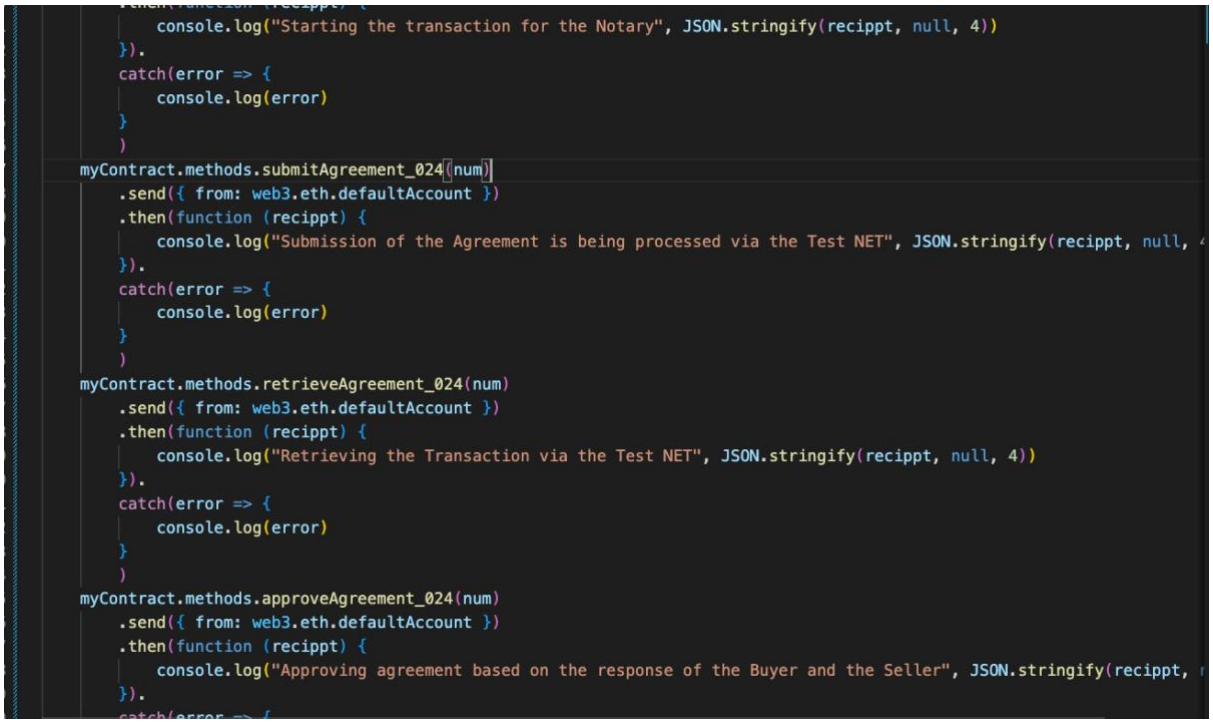


Figure 19 – Using Web3 to use the smart contract

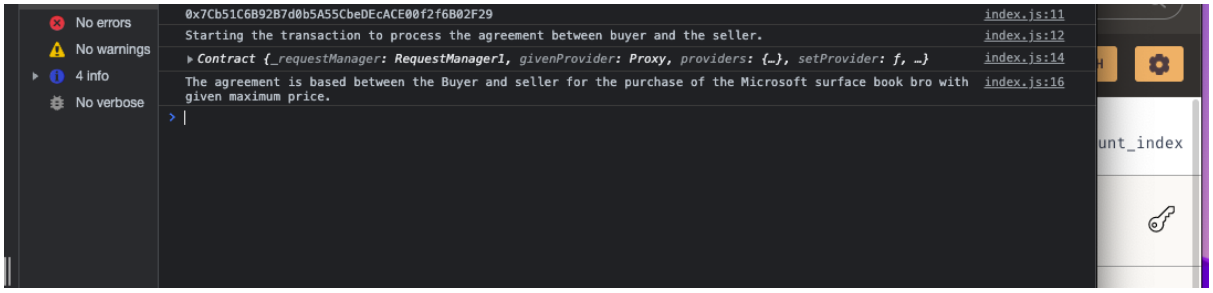


Figure 20 – Running transaction based on the input

References

"The crypto wallet for defi, Web3 Dapps and nfts," *MetaMask*. [Online]. Available: https://metamask.io/. [Accessed: 21-May-2022].

"Ethereum API: Ipfs API & gateway: ETH Nodes as a service," *Infura*. [Online]. Available: https://infura.io/. [Accessed: 21-May-2022].

"Ethereum Development Documentation," *ethereum.org*. [Online]. Available: https://ethereum.org/en/developers/docs/. [Accessed: 21-May-2022].

"Log in to access the Lucid Visual Collaboration Suite," *Lucid visual collaboration suite: Log in*. [Online]. Available: https://lucid.app/users/login. [Accessed: 21-May-2022].