St. Francis Institute of Technology, Mumbai-400 103
Department Of Information Technology

A.Y. 2025-2026
Class: BE-ITA/B, Semester: VIII
Subject: BlockChain Lab

Experiment – 5

1. Aim: To implement smart contract using truffle framework
2. Objective: To …
- explore working of truffle framework.
- explore working of Ganache local ethernet network.
- explore deployment scripts.

3. Lab outcome: After performing the experiment, the students will be able to **implement** smart contracts in Ethereum using different development frameworks (PO3, PSO2, BL3)

4. Prerequisite:
- Fundamental knowledge of blockchain
- Knowledge of the Ethereum platform
- Familiarity with the Solidity programming language and JavaScript

5. Requirements: The following are the requirements –
Truffle Framework, Ganache Provider, Visual Studio Code etc.

6. Pre-Experiment Theory:
What is Truffle Framework?
We have various setup options for deploying, migrating, and accessing smart contracts. Depending on the level of control and visibility we want into the EVM (Ethereum Virtual Machine), we can choose from using an online IDE like Remix, to running a full Ethereum mining node via Geth. Truffle is a world-class development environment, testing framework and asset pipeline for blockchains using the EVM. Truffle is widely considered the most popular tool for blockchain application development.
Some of the features of Truffle suite are
- Built-in smart contract compilation, linking, deployment and binary management.
- Automated contract testing for rapid development.
- Scriptable, extensible deployment & migrations framework.
- Network management for deploying to any number of public & private networks.
- Package management with EthPM & NPM, using the ERC190 standard.
- Interactive console for direct contract communication.
- Configurable build pipeline with support for tight integration.
- External script runner that executes scripts within a Truffle environment.

7. Laboratory Exercise
A. Steps to be implemented.
To Follow the procedure given below to build smart contract in Remix IDE
1. Download and install Nodejs from command-prompt terminal, if it is not installed.
2. Create a new directory using command mkdir, with name Ethereum (or any other name).
3. Change directory to new directory using cd Ethereum
4. Set up truffle using command npm install -g truffle
5. Open folder/directory Ethereum using Visual Studio, observe empty folder.

6. From terminal type command, truffle init and observe the directory Ethereum with new subfolders
7. In Visual Studio, under contract folder create a new file HelloWorld.sol and write a smart contract (as done in exp-1).
8. Under migration folder create a new js file with name 2_helloWorld_migration.js and paste the following js code in the file

```
var HelloWorld = artifacts.require("./HelloWorld.sol");
module.exports = function(deployer) {
   deployer.deploy(HelloWorld);
  };
```

9. Open truffle-config.js, delete all its contents and paste the following code

```
module.exports = {
  networks: {
     development: {
        // from: "", // Defaults to first address from Ganache
        host: "127.0.0.1",
        port: 7545,
        network_id: "*"
     }
  }
};
```

10. Now from terminal run command truffle compile
11. Open Ganache provider and from terminal run command truffle migrate
12. Now to see the output of smart contract run command truffle console

G. Program Code

1. Write a smart contract HelloWorld.sol and deploy it using Ganache and Truffle.
2. Write a program DonateEther.sol, create uint public variable 'balance' and initialize it to 0 using constructor() which is public, write a function contribute(), make it public and payable. Inside the function body write balance+=msg.Value; then create migration.js file for this contract as done in HelloWorld program. Deploy the contract on Ganche local Ethereum network using truffle.

8. Post Experimental Exercise-

H. Questions:

1. List down the details of components of Truffle suite.
2. Deploy all smart contracts studied in expt-1 and expt-2 using Truffle and Ganache.

I. Results/Observations/Program output:

Present the program input/output results if any and comment on the same.

J. Conclusion:

1. Write what was performed in the experiment.
2. Write which tools you used to perform the experiment
3. Write what you inferred from the output obtained.

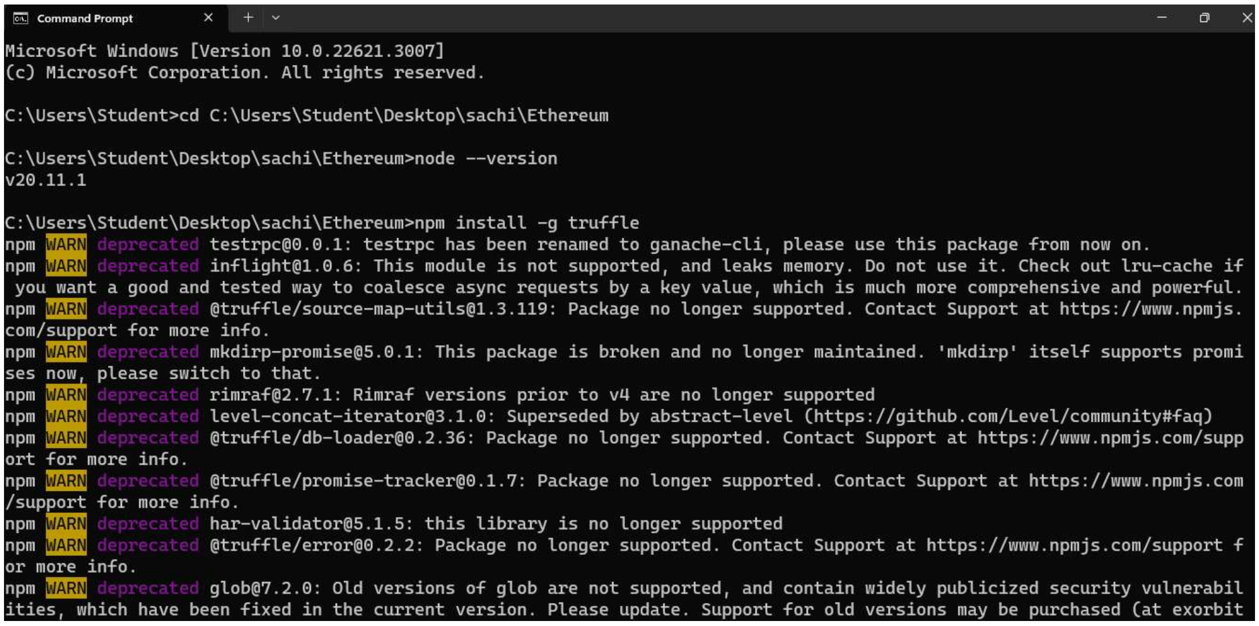
K. References:

[1] https://trufflesuite.com/docs/truffle/quickstart/

[2] Mastering Ethereum, Building Smart Contract and Dapps, Andreas M. Antonopoulos Dr. Gavin Wood, O'reilly

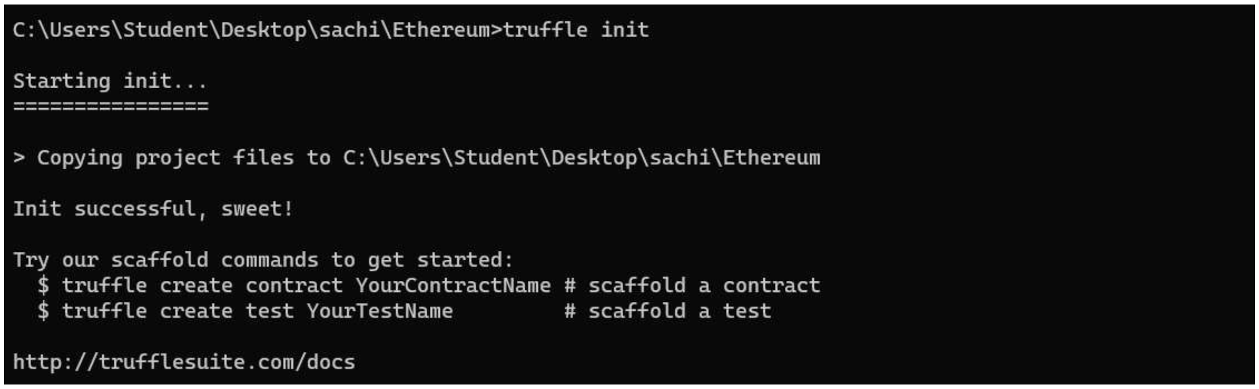
G. Program Code

1. Write a smart contract HelloWorld.sol and deploy it using Ganache and Truffle.



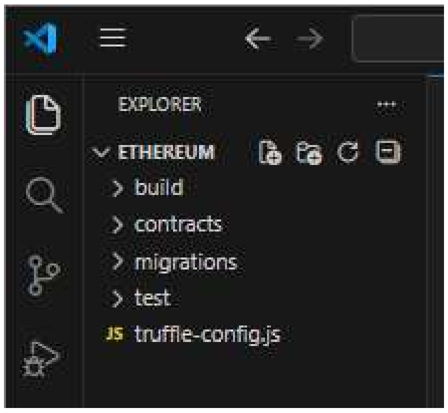
```
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Student>cd C:\Users\Student\Desktop\sachi\Ethereum

C:\Users\Student\Desktop\sachi\Ethereum>node --version
v20.11.1

C:\Users\Student\Desktop\sachi\Ethereum>npm install -g truffle
npm WARN deprecated testrpc@0.0.1: testrpc has been renamed to ganache-cli, please use this package from now on.
npm WARN deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if
 you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.
npm WARN deprecated @truffle/source-map-utils@1.3.119: Package no longer supported. Contact Support at https://www.npmjs.
com/support for more info.
npm WARN deprecated mkdirp-promise@5.0.1: This package is broken and no longer maintained. 'mkdirp' itself supports promi
ses now, please switch to that.
npm WARN deprecated rimraf@2.7.1: Rimraf versions prior to v4 are no longer supported
npm WARN deprecated level-concat-iterator@3.1.0: Superseded by abstract-level (https://github.com/Level/community#faq)
npm WARN deprecated @truffle/db-loader@0.2.36: Package no longer supported. Contact Support at https://www.npmjs.com/supp
ort for more info.
npm WARN deprecated @truffle/promise-tracker@0.1.7: Package no longer supported. Contact Support at https://www.npmjs.com
/support for more info.
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated @truffle/error@0.2.2: Package no longer supported. Contact Support at https://www.npmjs.com/support f
or more info.
npm WARN deprecated glob@7.2.0: Old versions of glob are not supported, and contain widely publicized security vulnerabil
ities, which have been fixed in the current version. Please update. Support for old versions may be purchased (at exorbit
```

Start Truffle using truffle init command:

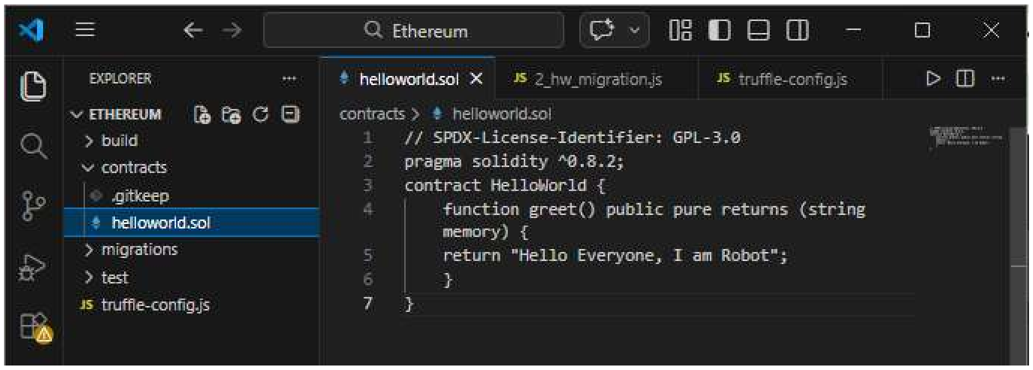


```
C:\Users\Student\Desktop\sachi\Ethereum>truffle init

Starting init...
================

> Copying project files to C:\Users\Student\Desktop\sachi\Ethereum

Init successful, sweet!

Try our scaffold commands to get started:
  $ truffle create contract YourContractName # scaffold a contract
  $ truffle create test YourTestName         # scaffold a test

http://trufflesuite.com/docs
```

Folder structure:

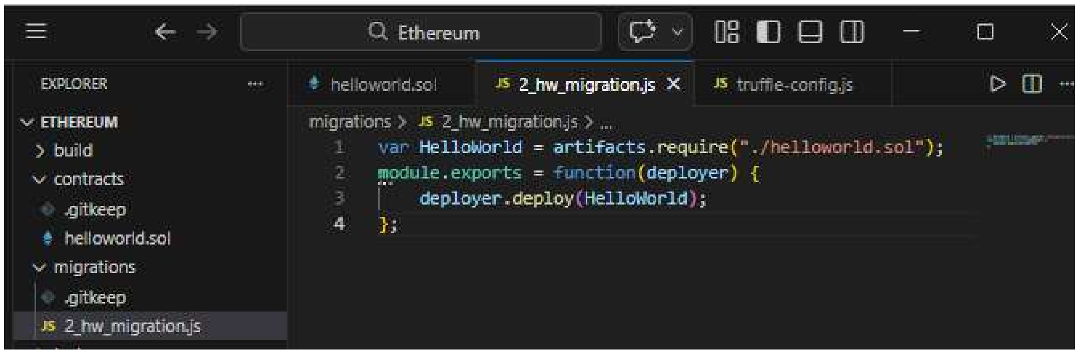


helloworld.sol



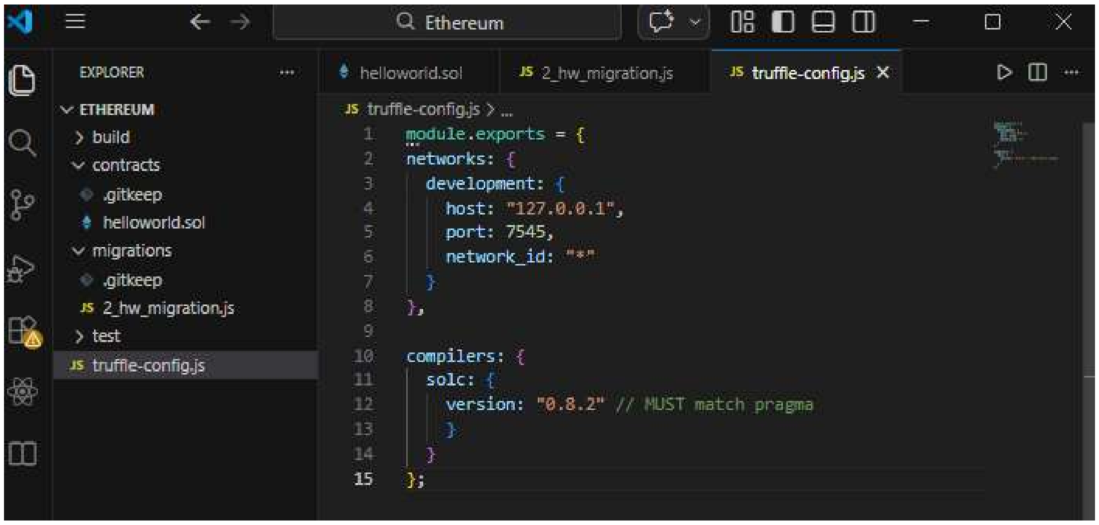
```
// SPDX-License-Identifier: GPL-3.0
pragma solidity ^0.8.2;
contract HelloWorld {
    function greet() public pure returns (string
    memory) {
    return "Hello Everyone, I am Robot";
    }
}
```

Migration code



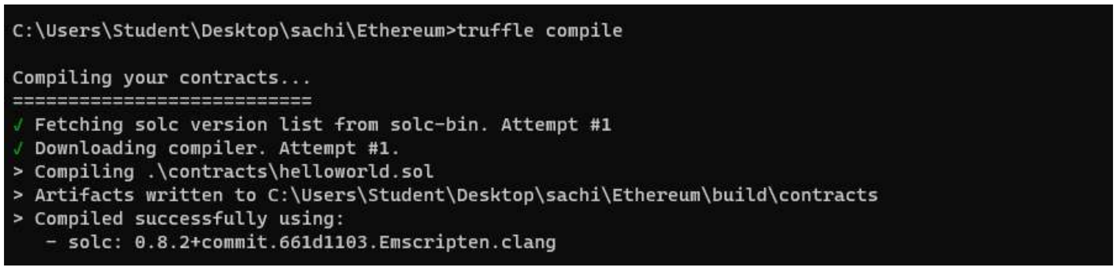
```
var HelloWorld = artifacts.require("./helloworld.sol");
module.exports = function(deployer) {
  deployer.deploy(HelloWorld);
};
```

configuration



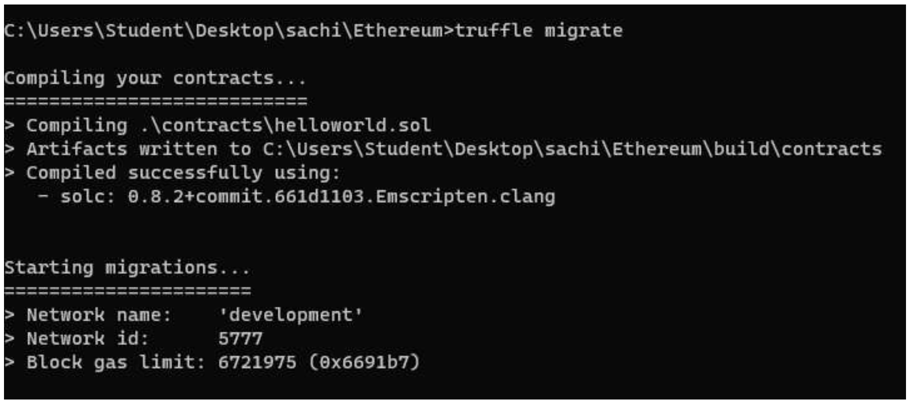
```
module.exports = {
networks: {
  development: {
    host: "127.0.0.1",
    port: 7545,
    network_id: "*"
  }
},

compilers: {
  solc: {
    version: "0.8.2" // MUST match pragma
    }
  }
};
```

Compile truffle

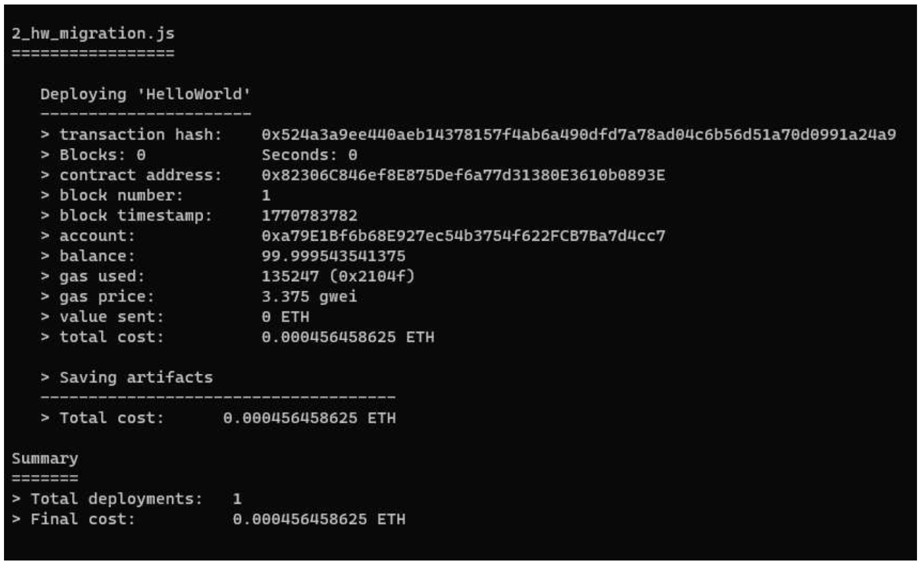


```
C:\Users\Student\Desktop\sachi\Ethereum>truffle compile

Compiling your contracts...
===========================
✓ Fetching solc version list from solc-bin. Attempt #1
✓ Downloading compiler. Attempt #1.
> Compiling .\contracts\helloworld.sol
> Artifacts written to C:\Users\Student\Desktop\sachi\Ethereum\build\contracts
> Compiled successfully using:
   - solc: 0.8.2+commit.661d1103.Emscripten.clang
```

Migrate

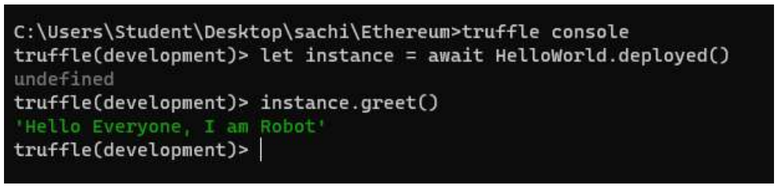


```
C:\Users\Student\Desktop\sachi\Ethereum>truffle migrate

Compiling your contracts...
===========================
> Compiling .\contracts\helloworld.sol
> Artifacts written to C:\Users\Student\Desktop\sachi\Ethereum\build\contracts
> Compiled successfully using:
   - solc: 0.8.2+commit.661d1103.Emscripten.clang


Starting migrations...
======================
> Network name:    'development'
> Network id:      5777
> Block gas limit: 6721975 (0x6691b7)
```



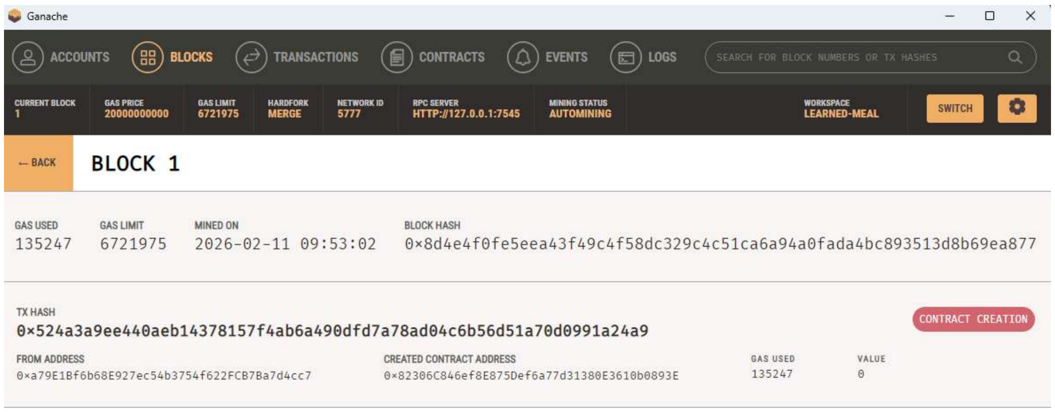
```
2_hw_migration.js
=================

   Deploying 'HelloWorld'
   ----------------------
   > transaction hash:    0x524a3a9ee440aeb14378157f4ab6a490dfd7a78ad04c6b56d51a70d0991a24a9
   > Blocks: 0            Seconds: 0
   > contract address:    0x82306C846ef8E875Def6a77d31380E3610b0893E
   > block number:        1
   > block timestamp:     1770783782
   > account:             0xa79E1Bf6b68E927ec54b3754f622FCB7Ba7d4cc7
   > balance:             99.999543541375
   > gas used:            135247 (0x2104f)
   > gas price:           3.375 gwei
   > value sent:          0 ETH
   > total cost:          0.000456458625 ETH

   > Saving artifacts
   -------------------------------------
   > Total cost:      0.000456458625 ETH

Summary
=======
> Total deployments:   1
> Final cost:          0.000456458625 ETH
```

Open truffle console for taking user input



```
C:\Users\Student\Desktop\sachi\Ethereum>truffle console
truffle(development)> let instance = await HelloWorld.deployed()
undefined
truffle(development)> instance.greet()
'Hello Everyone, I am Robot'
truffle(development)>
```

Blocks



CURRENT BLOCK 1 | GAS PRICE 20000000000 | GAS LIMIT 6721975 | HARDFORK MERGE | NETWORK ID 5777 | RPC SERVER HTTP://127.0.0.1:7545 | MINING STATUS AUTOMINING | WORKSPACE LEARNED-MEAL

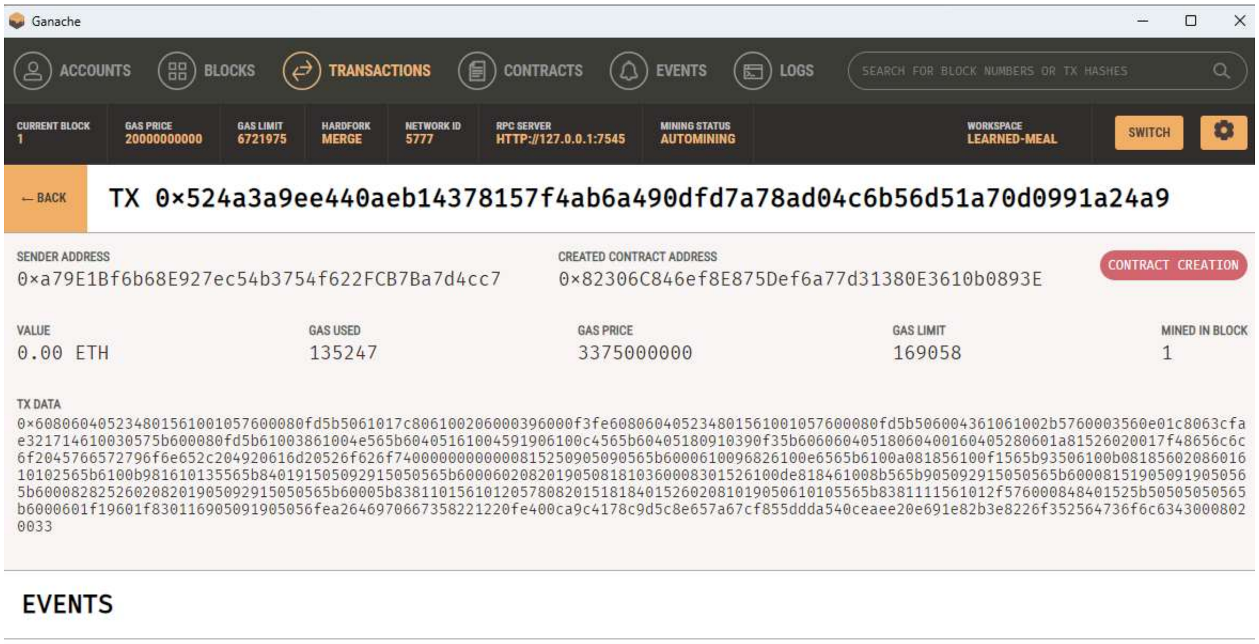
BLOCK 1

GAS USED	GAS LIMIT	MINED ON	BLOCK HASH
135247	6721975	2026-02-11 09:53:02	0×8d4e4f0fe5eea43f49c4f58dc329c4c51ca6a94a0fada4bc893513d8b69ea877

TX HASH
0×524a3a9ee440aeb14378157f4ab6a490dfd7a78ad04c6b56d51a70d0991a24a9
CONTRACT CREATION

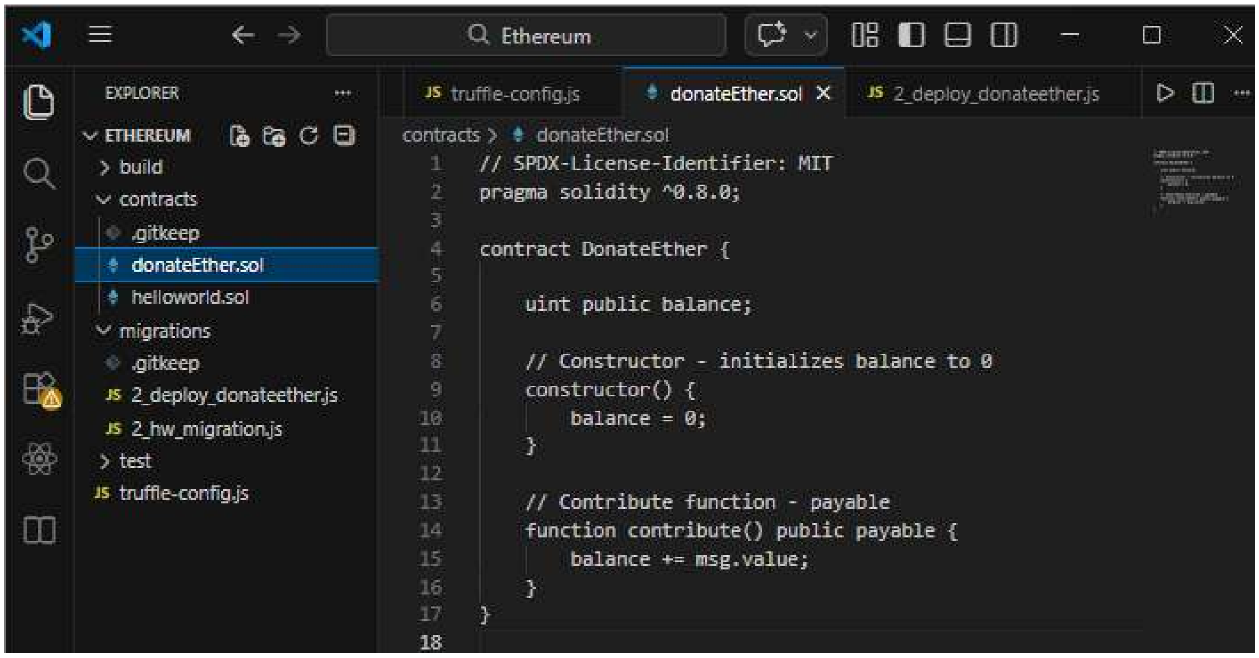
FROM ADDRESS	CREATED CONTRACT ADDRESS	GAS USED	VALUE
0×a79E1Bf6b68E927ec54b3754f622FCB7Ba7d4cc7	0×82306C846ef8E875Def6a77d31380E3610b0893E	135247	0

Transactions

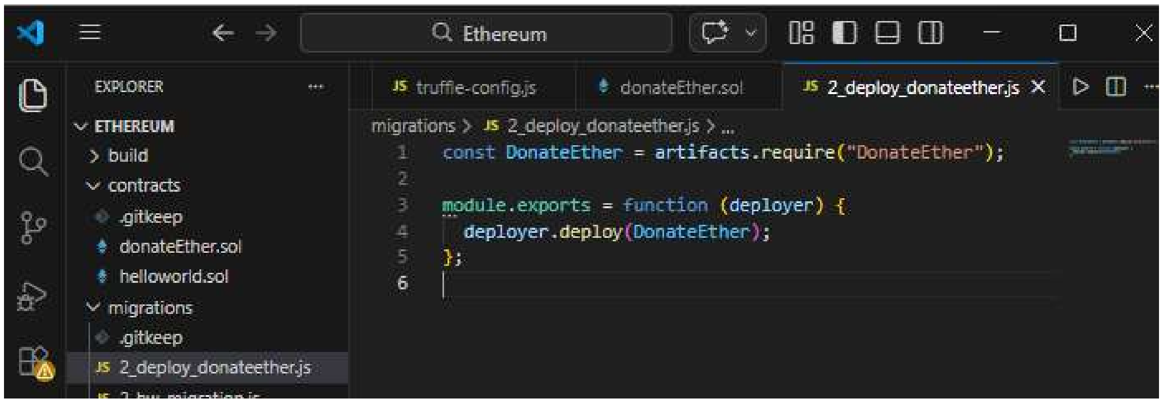


```
CURRENT BLOCK  GAS PRICE    GAS LIMIT  HARDFORK  NETWORK ID  RPC SERVER             MINING STATUS  WORKSPACE
1              20000000000  6721975    MERGE     5777        HTTP://127.0.0.1:7545  AUTOMINING     LEARNED-MEAL

TX 0x524a3a9ee440aeb14378157f4ab6a490dfd7a78ad04c6b56d51a70d0991a24a9

SENDER ADDRESS                                 CREATED CONTRACT ADDRESS
0x79E1Bf6b68E927ec54b3754f622FCB7Ba7d4cc7      0x82306C846ef8E875Def6a77d31380E3610b0893E

VALUE       GAS USED     GAS PRICE     GAS LIMIT     MINED IN BLOCK
0.00 ETH    135247       3375000000    169058        1

TX DATA
0x608060405234801561001057600080fd5b5061017c806100206000396000f3fe608060405234801561001057600080fd5b506004361061002b5760003560e01c8063cfa
e321714610030575b600080fd5b61003861004e565b60405161004591906100c4565b60405180910390f35b60606040518060400160405280601a81526020017f48656c6c
6f2045766572796f6e652c204920616d20526f626f7400000000000081525090509056 5b6000610096826100e6565b6100a081856100f1565b93506100b08185602086016
10102565b6100b981610135565b840191505092915050565b600060208201905081810360008301526100de818461008b565b905092915050565b60008151905091905056
5b600082825260208201905092915050565b60005b83811015610120578082015181840152602081019050610105565b8381111561012f576000848401525b50505050565
b6000601f19601f8301169050919050 56fea2646970667358221220fe400ca9c4178c9d5c8e657a67cf855ddda540ceaee20e691e82b3e8226f352564736f6c634300080 2
0033

EVENTS
```

2. Write a program DonateEther.sol, create uint public variable ‘balance’ and initialize it to 0 using constructor() which is public, write a function contribute(), make it public and payable. Inside the function body write balance+=msg.Value; then create migration.js file for this contract as done in the HelloWorld program. Deploy the contract on Ganche local Ethereum network using truffle.

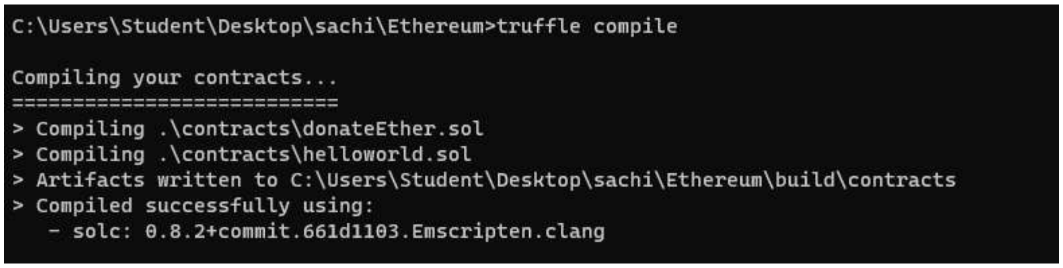


```solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract DonateEther {

    uint public balance;

    // Constructor - initializes balance to 0
    constructor() {
        balance = 0;
    }

    // Contribute function - payable
    function contribute() public payable {
        balance += msg.value;
    }
}
```



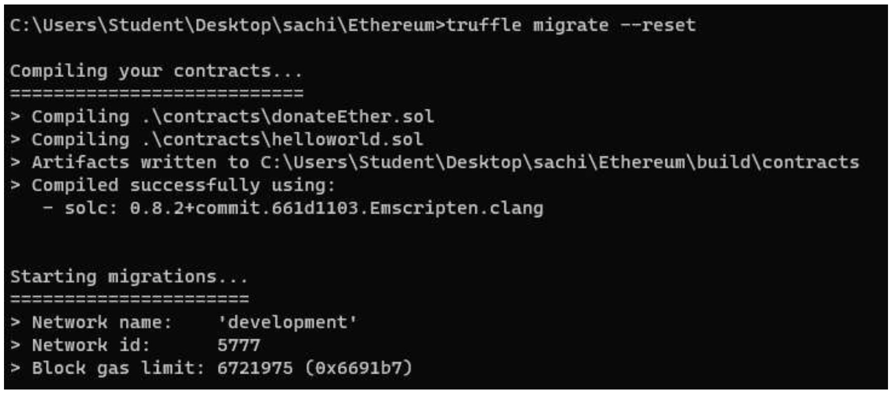
```javascript
const DonateEther = artifacts.require("DonateEther");

module.exports = function (deployer) {
  deployer.deploy(DonateEther);
};

```

Truffle compile

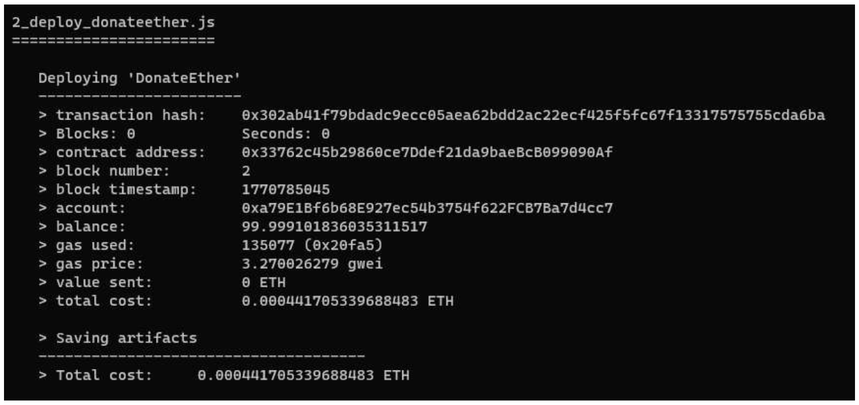


```
C:\Users\Student\Desktop\sachi\Ethereum>truffle compile

Compiling your contracts...
===========================
> Compiling .\contracts\donateEther.sol
> Compiling .\contracts\helloworld.sol
> Artifacts written to C:\Users\Student\Desktop\sachi\Ethereum\build\contracts
> Compiled successfully using:
   - solc: 0.8.2+commit.661d1103.Emscripten.clang
```

Migration

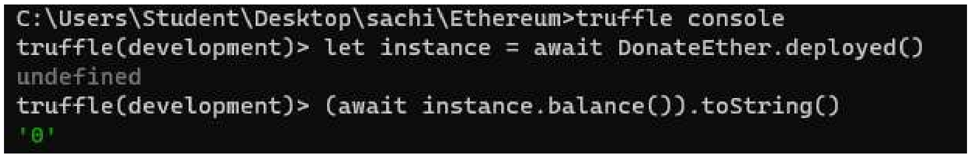


```
C:\Users\Student\Desktop\sachi\Ethereum>truffle migrate --reset

Compiling your contracts...
===========================
> Compiling .\contracts\donateEther.sol
> Compiling .\contracts\helloworld.sol
> Artifacts written to C:\Users\Student\Desktop\sachi\Ethereum\build\contracts
> Compiled successfully using:
   - solc: 0.8.2+commit.661d1103.Emscripten.clang


Starting migrations...
======================
> Network name:    'development'
> Network id:      5777
> Block gas limit: 6721975 (0x6691b7)
```



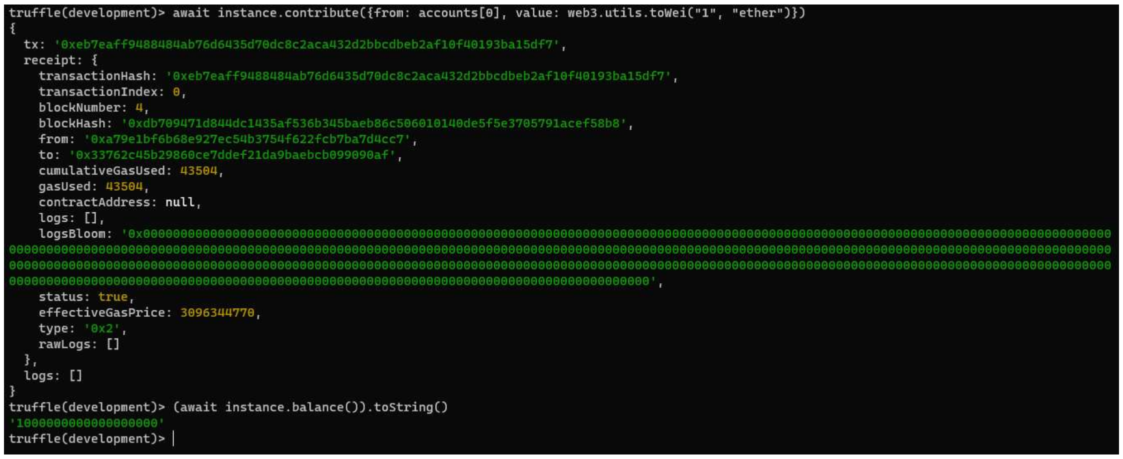
```
2_deploy_donateether.js
=======================

   Deploying 'DonateEther'
   -----------------------
   > transaction hash:    0x302ab41f79bdadc9ecc05aea62bdd2ac22ecf425f5fc67f13317575755cda6ba
   > Blocks: 0            Seconds: 0
   > contract address:    0x33762c45b29860ce7Ddef21da9baeBcB099090Af
   > block number:        2
   > block timestamp:     1770785045
   > account:             0xa79E1Bf6b68E927ec54b3754f622FCB7Ba7d4cc7
   > balance:             99.999101836035311517
   > gas used:            135077 (0x20fa5)
   > gas price:           3.270026279 gwei
   > value sent:          0 ETH
   > total cost:          0.000441705339688483 ETH

   > Saving artifacts
   -------------------------------------
   > Total cost:     0.000441705339688483 ETH
```

Check current balance



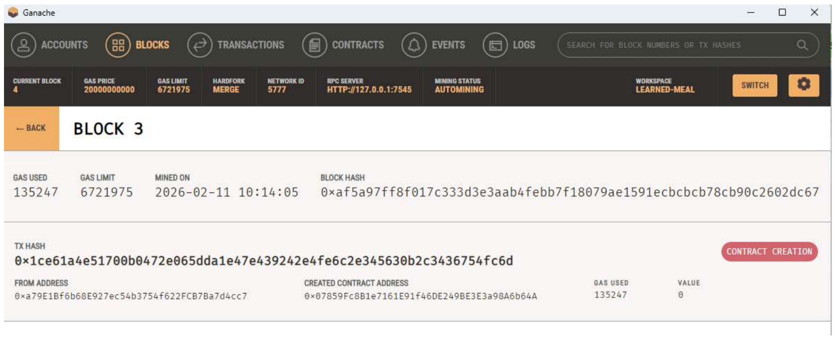
```
C:\Users\Student\Desktop\sachi\Ethereum>truffle console
truffle(development)> let instance = await DonateEther.deployed()
undefined
truffle(development)> (await instance.balance()).toString()
'0'
```

Send Ether and check balance



```
truffle(development)> await instance.contribute({from: accounts[0], value: web3.utils.toWei("1", "ether")})
{
  tx: '0xeb7eaff9488484ab76d6435d70dc8c2aca432d2bbcdbeb2af10f40193ba15df7',
  receipt: {
    transactionHash: '0xeb7eaff9488484ab76d6435d70dc8c2aca432d2bbcdbeb2af10f40193ba15df7',
    transactionIndex: 0,
    blockNumber: 4,
    blockHash: '0xdb709471d844dc1435af536b345baeb86c506010140de5f5e3705791acef58b8',
    from: '0xa79e1bf6b68e927ec54b3754f622fcb7ba7d4cc7',
    to: '0x33762c45b29860ce7ddef21da9baebcb099090af',
    cumulativeGasUsed: 43504,
    gasUsed: 43504,
    contractAddress: null,
    logs: [],
    logsBloom: '0x0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000',
    status: true,
    effectiveGasPrice: 3096344770,
    type: '0x2',
    rawLogs: []
  },
  logs: []
}
truffle(development)> (await instance.balance()).toString()
'1000000000000000000'
truffle(development)> |
```

Blocks



CURRENT BLOCK 4 | GAS PRICE 20000000000 | GAS LIMIT 6721975 | HARDFORK MERGE | NETWORK ID 5777 | RPC SERVER HTTP://127.0.0.1:7545 | MINING STATUS AUTOMINING | WORKSPACE LEARNED-MEAL

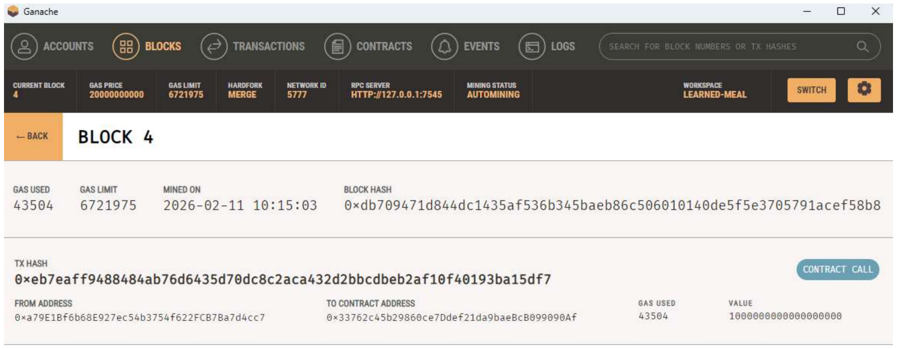
BLOCK 3

GAS USED	GAS LIMIT	MINED ON	BLOCK HASH
135247	6721975	2026-02-11 10:14:05	0×af5a97ff8f017c333d3e3aab4febb7f18079ae1591ecbcbcb78cb90c2602dc67

TX HASH
0×1ce61a4e51700b0472e065dda1e47e439242e4fe6c2e345630b2c3436754fc6d

CONTRACT CREATION

FROM ADDRESS	CREATED CONTRACT ADDRESS	GAS USED	VALUE
0×a79E1Bf6b68E927ec54b3754f622FCB7Ba7d4cc7	0×07859Fc8B1e7161E91f46DE249BE3E3a98A6b64A	135247	0



CURRENT BLOCK 4 | GAS PRICE 20000000000 | GAS LIMIT 6721975 | HARDFORK MERGE | NETWORK ID 5777 | RPC SERVER HTTP://127.0.0.1:7545 | MINING STATUS AUTOMINING | WORKSPACE LEARNED-MEAL

BLOCK 4

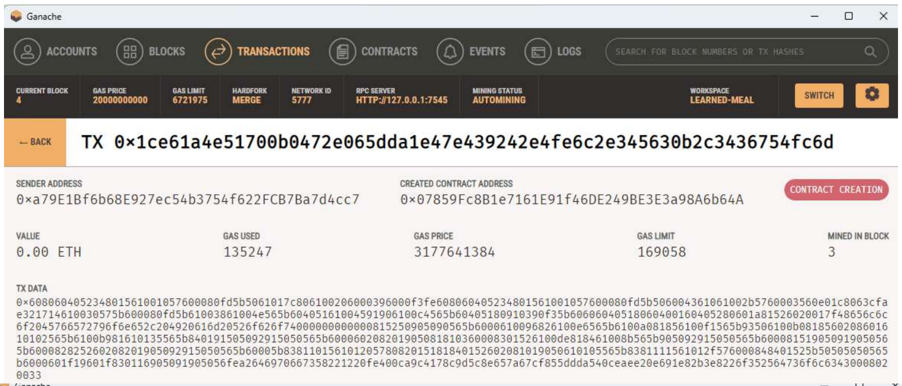
GAS USED	GAS LIMIT	MINED ON	BLOCK HASH
43504	6721975	2026-02-11 10:15:03	0×db709471d844dc1435af536b345baeb86c506010140de5f5e3705791acef58b8

TX HASH
0×eb7eaff9488484ab76d6435d70dc8c2aca432d2bbcdbeb2af10f40193ba15df7

CONTRACT CALL

FROM ADDRESS	TO CONTRACT ADDRESS	GAS USED	VALUE
0×a79E1Bf6b68E927ec54b3754f622FCB7Ba7d4cc7	0×33762c45b29860ce7Ddef21da9baeBcB099090Af	43504	1000000000000000000

Transactions



CURRENT BLOCK 4 | GAS PRICE 20000000000 | GAS LIMIT 6721975 | HARDFORK MERGE | NETWORK ID 5777 | RPC SERVER HTTP://127.0.0.1:7545 | MINING STATUS AUTOMINING | WORKSPACE LEARNED-MEAL

TX 0×1ce61a4e51700b0472e065dda1e47e439242e4fe6c2e345630b2c3436754fc6d

SENDER ADDRESS
0×a79E1Bf6b68E927ec54b3754f622FCB7Ba7d4cc7

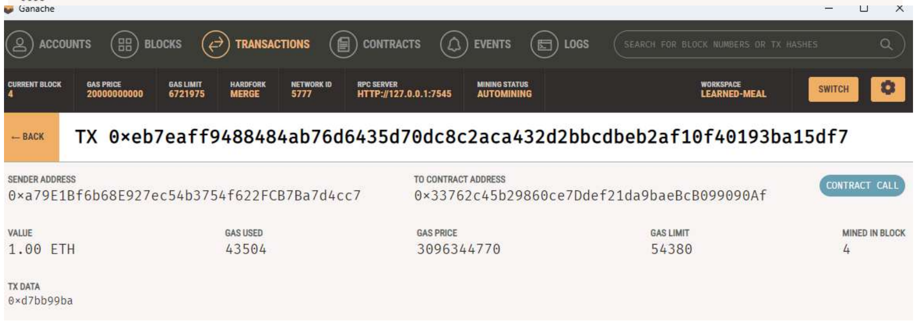
CREATED CONTRACT ADDRESS
0×07859Fc8B1e7161E91f46DE249BE3E3a98A6b64A

CONTRACT CREATION

VALUE	GAS USED	GAS PRICE	GAS LIMIT	MINED IN BLOCK
0.00 ETH	135247	3177641384	169058	3

TX DATA

```
0×6080604052348015610010576000 80fd5b5061017c806100206000396000f3fe6080604052348015610010576000 80fd5b50600436106100 2b5760003560e01c8063cfa
e321714610030575b600080fd5b61003861004e565b6040516100459190 6100c4565b60405180910390f35b60606040518060400160405280601a81526020017f48656c6c
6f2045766572796f6e652c204920616d20526f626f740000000000000081525090509056 5b60006100968261 00e6565b6100a081856100f1565b93506100b08185602086016
10102565b6100b981610135565b840191505092915050565b600060208201905081810360008301526100de818461008b565b90509291505056 5b600081519050919050 56
5b6000828252602082019050929150505 65b60005b83811015610120578082015181840152602081019050610105565b83811115610 12f576000848401525b50505050565
b6000601f19601f8301169050919050 56fea2646970667358221220fe400ca9c4178c9d5c8e657a67cf855ddda540ceaee20e691e82b3e8226f352564736f6c6343000802
0033
```



CURRENT BLOCK 4 | GAS PRICE 20000000000 | GAS LIMIT 6721975 | HARDFORK MERGE | NETWORK ID 5777 | RPC SERVER HTTP://127.0.0.1:7545 | MINING STATUS AUTOMINING | WORKSPACE LEARNED-MEAL

TX 0×eb7eaff9488484ab76d6435d70dc8c2aca432d2bbcdbeb2af10f40193ba15df7

SENDER ADDRESS
0×a79E1Bf6b68E927ec54b3754f622FCB7Ba7d4cc7

TO CONTRACT ADDRESS
0×33762c45b29860ce7Ddef21da9baeBcB099090Af

CONTRACT CALL

VALUE	GAS USED	GAS PRICE	GAS LIMIT	MINED IN BLOCK
1.00 ETH	43504	3096344770	54380	4

TX DATA
0×d7bb99ba