Experiment No.8
To create a private ethereum blockchain using Ganache and Truffle
Date of Performance:5—10—23
Date of Submission:10—10—23

AIM: To create a private ethereum blockchain using Ganache and Truffle

Objective: To create private ethereum blockchain and deploy smart contract on it

Theory:

Smart contracts are programs stored on a blockchain. The term 'smart contract' was first coined by Nick Szabo in 1994. It's a collection of code (its functions) and data (its state) that resides at a specific address on the Ethereum blockchain. One of the main features is that they are immutable once deployed on the blockchain. Solidity and Vyper are the two most active programming languages used to write smart contracts on the Ethereum blockchain with Solidity being the top choice for most developers as it is an object-oriented, statically-typed language and is strongly influenced by more popular OOP languages like JavaScript and C++.

Ganache

Ganache is a personalized blockchain for Ethereum development. It can be used to run tests, execute commands, and inspect states while controlling how the chain operates. Ganache is an Ethereum simulator that makes developing Ethereum applications faster, easier, and safer. It is provided by Truffle Suite and can be downloaded from https://www.trufflesuite.com/ganache. The below image shows the view of Ganache

Fig.8.1 Ganache view

Truffle

Truffle is the most popular development framework for Ethereum. Truffle takes care of managing your contract artifacts so you don't have to. Includes support for custom deployments, library linking and complex Ethereum applications. Truffle is used to test contracts in both javascript and solidity.

Process:

Step 1. Install Ganache , Download Ganache from the website https://truffleframework.com/ganache

Step 2. Install Truffle by executing the following command at the command prompt:

```
npm install –g truffle
```

To install Truffle you need to have Node and NPM along with Python setup on your machine.

Step 3. To verify if Truffle is installed successfully, execute the following command at the command prompt.

```
truffle version
```

Step 4. To start a project in Truffle, go into a directory and type the init command:

```
truffle init
```

This will create a new project with the required directory: contracts, migrations, test

Step 5. Create the required contract file in solidity and save it in the contracts directory

Step 6. Create a migration file in javascript and save it in migrations directory

Step 7. Run the following command to deploy the contract on Ganache

truffle Test

Ganache should be running while executing this command. The contract deployed will be visible in the contracts tab of the Ganache.

Code:

Election.sol

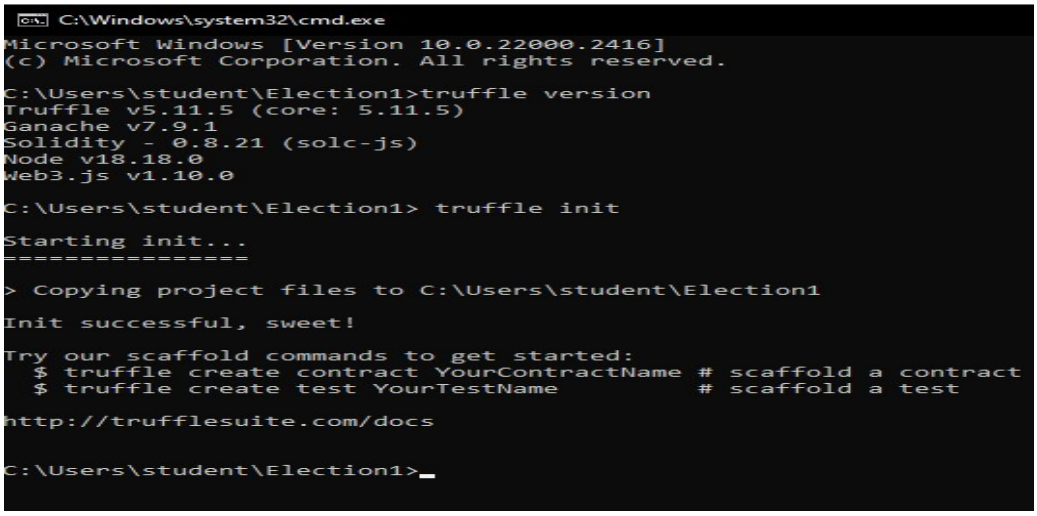
```
pragma solidity ^0.5.16;

contract Election {

    // Model a Candidate

    struct Candidate {

        uint id;

        string name;

        uint voteCount;

    }

    // Store accounts that have voted

    mapping(address => bool) public voters;

    // Read/write candidates

    mapping(uint => Candidate) public candidates;

    // Store Candidates Count

    uint public candidatesCount;
```

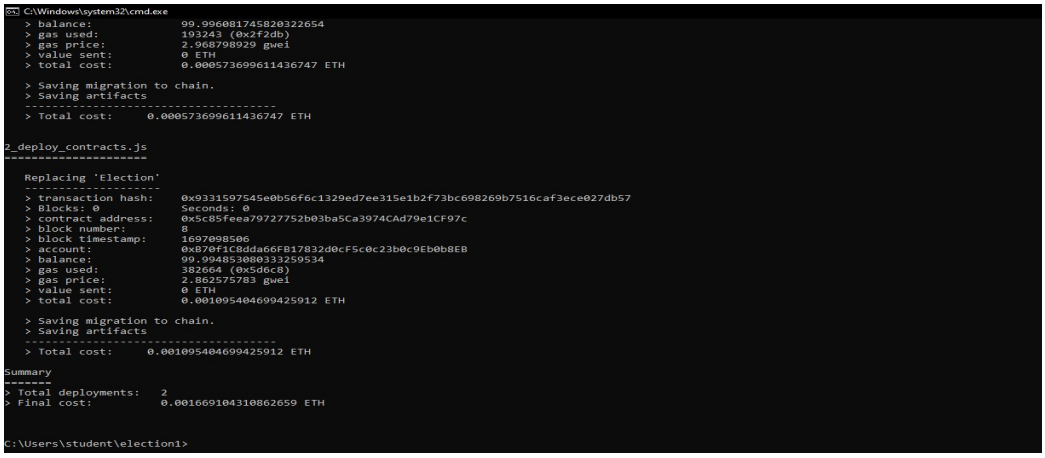
```
    constructor () public {
        addCandidate("Candidate 1");
        addCandidate("Candidate 2");
    }
    function addCandidate (string memory _name) private {
        candidatesCount ++;
        candidates[candidatesCount] = Candidate(candidatesCount, _name, 0);
    }
    function vote (uint _candidateId) public {
        // require that they haven't voted before
        require(!voters[msg.sender]);
        // require a valid candidate
        require(_candidateId > 0 && _candidateId <= candidatesCount);
        // record that voter has voted
        voters[msg.sender] = true;
        // update candidate vote Count
        candidates[_candidateId].voteCount ++;}
}
```

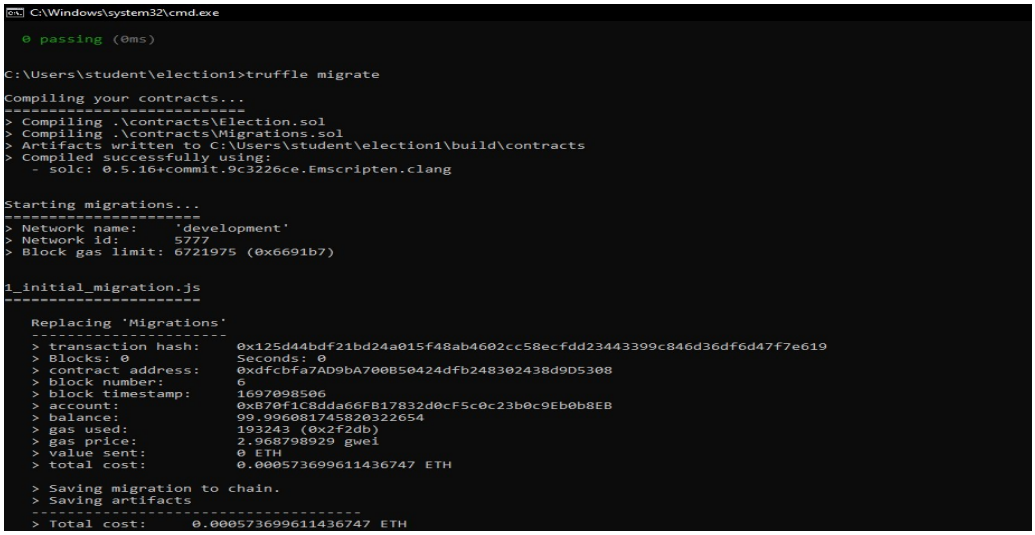
2_deploy_contracts

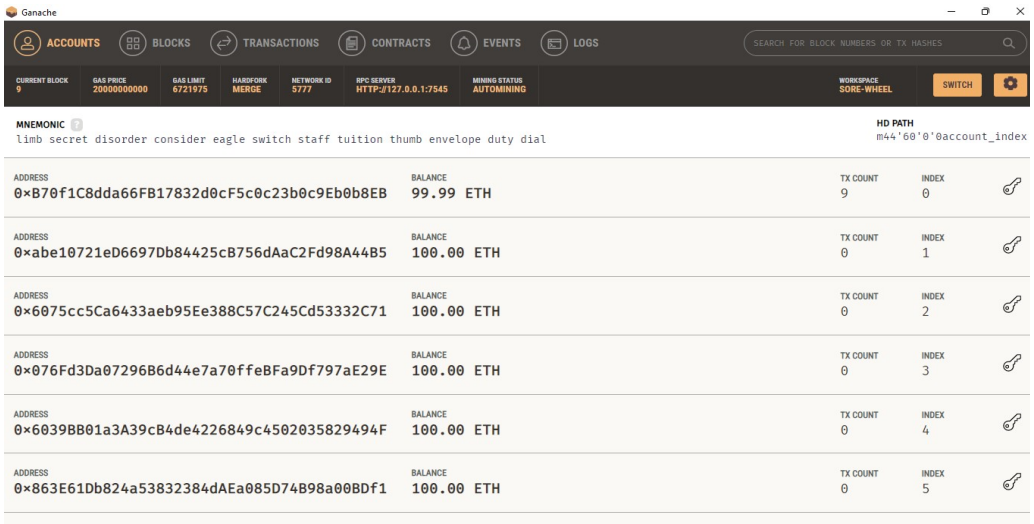
```
var Election = artifacts.require("./Election.sol");

module.exports = function(deployer) {

  deployer.deploy(Election);

};
```

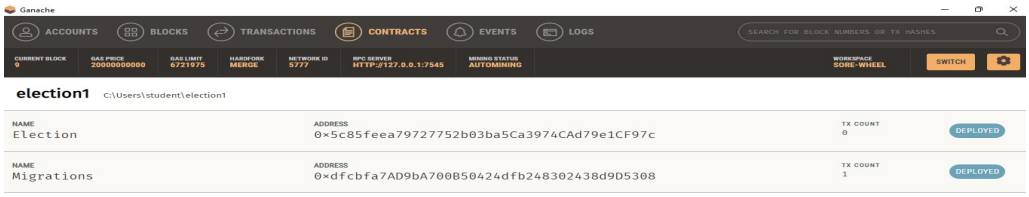
Output:






C:\Windows\system32\cmd.exe
0 passing (0ms)
C:\Users\student\election1>truffle migrate
Compiling your contracts...
> Compiling .\contracts\Election.sol
> Compiling .\contracts\Migrations.sol
> Artifacts written to C:\Users\student\election1\build\contracts
> Compiled successfully using:
- solc: 0.5.16+commit.9c3226ce.Emscripten.clang
Starting migrations...
> Network name: 'development'
> Network id: 5777
> Block gas limit: 6721975 (0x6691b7)
1_initial_migration.js
Replacing 'Migrations'
> transaction hash: 0x125d44bdf21bd24a015f48ab4602cc58ecfdd23443399c846d36df6d47f7e619
> Blocks: 0 Seconds: 0
> contract address: 0xdfcbfa7AD9bA700B50424dfb248302438d9D5308
> block number: 6
> block timestamp: 1697098506
> account: 0xB70f1C8dda66FB17832d0cF5c0c23b0c9Eb0b8EB
> balance: 99.996081745820322654
> gas used: 193243 (0x2f2db)
> gas price: 2.968798929 gwei
> value sent: 0 ETH
> total cost: 0.000573699611436747 ETH
> Saving migration to chain.
> Saving artifacts
> Total cost: 0.000573699611436747 ETH


Ganache
ACCOUNTS
BLOCKS
TRANSACTIONS
CONTRACTS
EVENTS
LOGS
SEARCH FOR BLOCK NUMBERS OR TX HASHES
CURRENT BLOCK 9
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK MERGE
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:7545
MINING STATUS AUTOMINING
WORKSPACE SORE-WHEEL
SWITCH
MNEMONIC
limb secret disorder consider eagle switch staff tuition thumb envelope duty dial
HD PATH
m44'60'0'0account_index
ADDRESS 0x B70f1C8dda66FB17832d0cF5c0c23b0c9Eb0b8EB BALANCE 99.99 ETH TX COUNT 9 INDEX 0
ADDRESS 0xabe10721eD6697Db84425cB756dAaC2Fd98A44B5 BALANCE 100.00 ETH TX COUNT 0 INDEX 1
ADDRESS 0x6075cc5Ca6433aeb95Ee388C57C245Cd53332C71 BALANCE 100.00 ETH TX COUNT 0 INDEX 2
ADDRESS 0x076Fd3Da07296B6d44e7a70ffeBFa9Df797aE29E BALANCE 100.00 ETH TX COUNT 0 INDEX 3
ADDRESS 0x6039BB01a3A39cB4de4226849c4502035829494F BALANCE 100.00 ETH TX COUNT 0 INDEX 4
ADDRESS 0x863E61Db824a53832384dAEa085D74B98a00BDf1 BALANCE 100.00 ETH TX COUNT 0 INDEX 5


Ganache
ACCOUNTS
BLOCKS
TRANSACTIONS
CONTRACTS
EVENTS
LOGS
SEARCH FOR BLOCK NUMBERS OR TX HASHES
CURRENT BLOCK 9
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK MERGE
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:7545
MINING STATUS AUTOMINING
WORKSPACE SORE-WHEEL
SWITCH
election1
C:\Users\student\election1
NAME Election ADDRESS 0x5c85feea79727752b03ba5Ca3974CAd79e1CF97c TX COUNT 0 DEPLOYED
NAME Migrations ADDRESS 0xdfcbfa7AD9bA700B50424dfb248302438d9D5308 TX COUNT 1 DEPLOYED

Conclusion:

Ganache and Truffle are crucial tools in creating a personal Ethereum blockchain. Ganache provides a local, lightweight Ethereum network for development and testing, making it easy to experiment with smart contracts and DApps. Truffle, on the other hand, offers a robust suite of development and deployment tools, simplifying the process of creating and managing smart contracts on Ethereum. Together, they streamline the development cycle, enhance efficiency, and reduce the complexity of building and testing Ethereum-based solutions. In conclusion, Ganache and Truffle play indispensable roles in facilitating the creation and deployment of personal Ethereum blockchains for development and experimentation.