| Experiment No.8 |
| --- |
| To create a private ethereum blockchain using Ganache and Truffle |
| Date of Performance:7/10/23 |
| Date of Submission:7/10/23 |

**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.

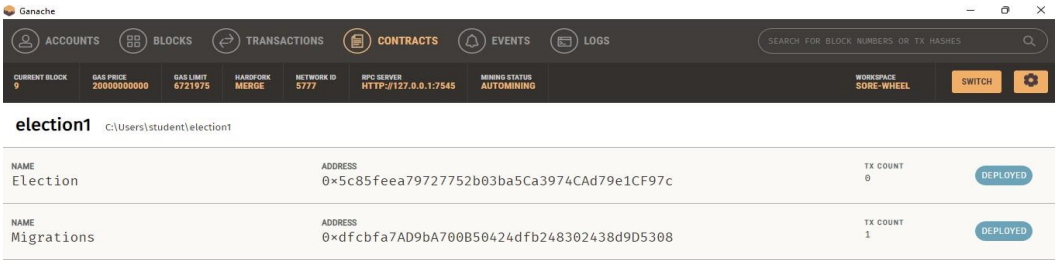
**output:**

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.22000.2416]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin\Election>truffle init

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

> Copying project files to C:\Users\admin\Election

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


C:\Users\admin\Election>truffle test

Compiling your contracts...
===========================
> Everything is up to date, there is nothing to compile.


  0 passing (0ms)


C:\Users\admin\Election>truffle migrate

Compiling your contracts...
===========================
```

```
C:\Windows\system32\cmd.exe
   > 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


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

   Replacing 'Election'
   --------------------
   > transaction hash:    0x9331597545e0b56f6c1329ed7ee315e1b2f73bc698269b7516caf3ece027db57
   > Blocks: 0            Seconds: 0
   > contract address:    0x5c85feea79727752b03ba5Ca3974CAd79e1CF97c
   > block number:        8
   > block timestamp:     1697098506
   > account:             0xB70f1C8dda66FB17832d0cF5c0c23b0c9Eb0b8EB
   > balance:             99.994853080333259534
   > gas used:            382664 (0x5d6c8)
   > gas price:           2.862575783 gwei
   > value sent:          0 ETH
   > total cost:          0.001095404699425912 ETH

   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:     0.001095404699425912 ETH

Summary
=======
> Total deployments:   2
> Final cost:          0.001669104310862659 ETH


C:\Users\student\election1>
```

S

```
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:             99.99608174582032265 4
   > 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
```



**Conclusion:**

For a number of reasons, Ganache and Truffle are necessary tools when building a personal Ethereum blockchain.

For Ethereum developers, Ganache is a useful tool that offers a lightweight, local blockchain environment that is perfect for testing and development. It provides a setup, reducing the need for complex configuration and speeding up the development process. Additionally, this platform gives developers complete control over crucial blockchain characteristics like gas caps, block durations, and network IDs, enabling them to accurately mimic various network scenarios and conditions. Additionally, Ganache works with the well-liked Truffle development framework with ease. By simplifying the process of compiling, migrating, and testing contracts during the development phase, this integration ensures a more productive and efficient workflow for smart contract development.