Centurion
UNIVERSITY
Shaping Lives...
Empowering Communities...

School: .... Campus: ....

Academic Year: .... Subject Name: .... Subject Code: ....

Semester: .... Program: .... Branch: .... Specialization: ....

Date: ....

Applied and Action Learning

(Learning by Doing and Discovery)

Name of the Experiement : Blockchain Dev Tools – Setting Up Environment

Objective/Aim:

To understand and set up the essential blockchain development tools and frameworks required for building, compiling, deploying, and testing smart contracts on local or test blockchain networks.

Apparatus/Software Used:

1. **Node.js & npm** – for JavaScript package management and blockchain tool installation
2. **Truffle / Hardhat** – smart contract development frameworks
3. **Ganache** – local blockchain simulator for testing contracts
4. **MetaMask Wallet** – browser wallet to interact with blockchain networks
5. **Remix IDE** – online smart contract editor and compiler
6. **VS Code** – code editor for development and integration
7. **Ethereum Testnets** – Sepolia networks for deployment

Theory/Concept:

Blockchain developers use specialized tools to manage contracts, compile Solidity code, deploy to testnets, and interact with nodes.
Setting up this environment allows you to create decentralized applications (DApps) and test them safely before mainnet deployment.

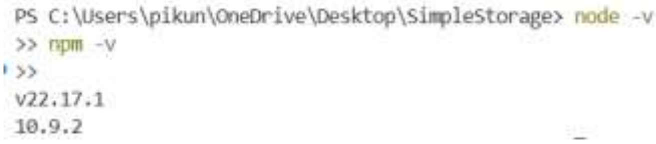
Key Components:

1. **Node.js & npm:**
   - Provides runtime environment for JavaScript-based blockchain tools.
   - Installed globally and used to manage packages.
     - ➢ **node -v**
     - ➢ **npm -v**
2. **Truffle Suite:**
   - ➢ Framework for smart contract compilation, deployment, and testing.
     - ➢ **npm install -g truffle**
     - ➢ **truffle init**
     - ➢ **truffle compile**
3. **Hardhat:**
   - Advanced Ethereum development environment with built-in testing and scripting.
     - ➢ **npm install --save-dev hardhat**
     - ➢ **npx hardhat**
4. **Ganache:**
   - ➢ Local blockchain simulator that provides fake accounts with ETH for testing.
5. **MetaMask Wallet:**
   - ➢ Browser extension wallet for connecting to local or test networks.
6. **Remix IDE:**
   - ➢ Web-based IDE to write and deploy contracts quickly without local setup.

Coding Phase: Algorithm / procedure

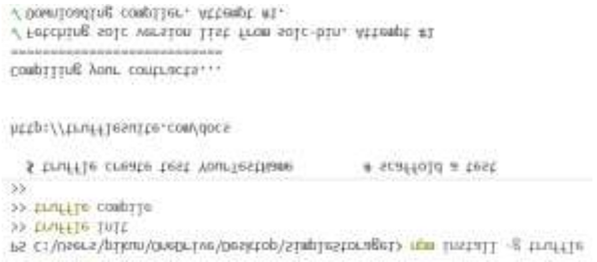
Algorithm :

- Start Node.js installation → verify using **node -v**.



```
PS C:\Users\pikun\OneDrive\Desktop\SimpleStorage> node -v
>> npm -v
>>
v22.17.1
10.9.2
```

- Install globally using npm.

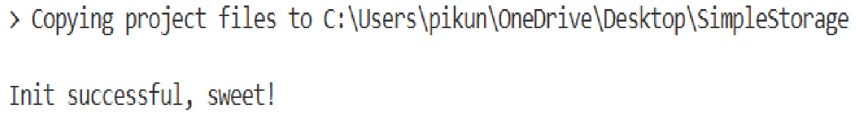


```
PS C:/Users/pikun/OneDrive/Desktop/SimpleStorage> npm install -g truffle
>> truffle init
>> truffle compile
>>
  $ truffle create test YourTestName         # scaffold a test

http://trufflesuite.com/docs

Compiling your contracts...
===========================
√ Fetching solc version list from solc-bin. Attempt #1
√ Downloading compiler. Attempt #1.
```

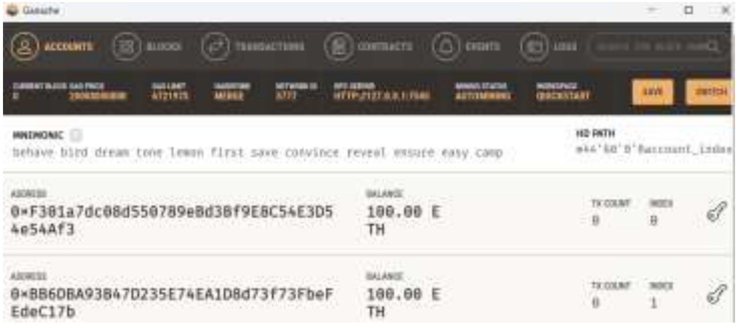
- Create a new project folder → initialize with truffle **init** .



```
> Copying project files to C:\Users\pikun\OneDrive\Desktop\SimpleStorage

Init successful, sweet!
```

- Set up a local blockchain.(Ganache)

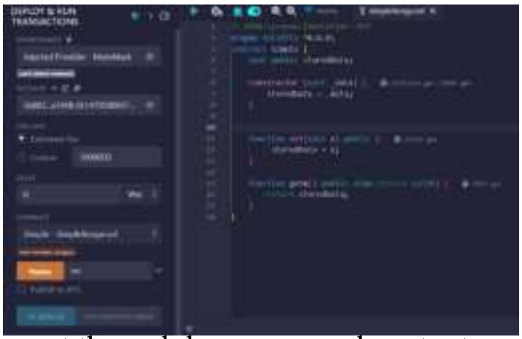


```
MNEMONIC
behave bird dream tone lemon first save convince reveal ensure easy camp
HD PATH
m44'60'0'0account_index

ADDRESS                                        BALANCE     TX COUNT  INDEX
0×F301a7dc08d550789eBd3Bf9E8C54E3D5            100.00 E    0         0
4e54Af3                                        TH

ADDRESS                                        BALANCE     TX COUNT  INDEX
0×BB6DBA93847D235E74EA1D8d73f73FbeF            100.00 E    0         1
EdeC17b                                        TH
```

- Connect MetaMask to the local network
- Write a sample Solidity contract (e.g., **SimpleStorage.sol).**
- Compile and deploy contract using Truffle.



- Verify successful deployment through logs or console output.
- Interact with the contract via console or MetaMask.

Observation Table:

Step	Task	Result
1	Ganache local blockchain setup	10 accounts created, 100 ETH each
2	MetaMask connection	Successfully connected (localhost:7545)
3	Contract creation	SimpleStorage.sol compiled with no errors
4	Deployment	Contract deployed successfully with transaction hash
5	Interaction	Value stored and retrieved correctly from blockchain

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Planning and Execution/ Practical Simulation/ Programming	10		
Result and Interpretation	10		
Record of Applied and Action Learning	10		
Viva	10		
Total	**50**		

Signature of the Student:

Name :

Regn. No. :

Signature of the Faculty: