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 Experiment: Truffle vs Hardhat – Dev Environment Showdown

*Coding Phase: Pseudo Code / Flow Chart / Algorithm

1.Start
2.Install Node.js and npm
3.Install Truffle (npm install -g truffle)
4.Create a Truffle project (truffle init)
5.Compile and deploy a sample smart contract (truffle compile, truffle migrate)
6.Install Hardhat (npm install --save-dev hardhat)
7.Create a Hardhat project (npx hardhat)
8.Compile and deploy the same smart contract (npx hardhat compile, npx hardhat run scripts/deploy.js)
9.Record outputs and screenshots
10.End

*Software used :

- npm (Node Package Manager)
- Truffle Suite – Ethereum development framework
- Hardhat – Ethereum development environment
- VS Code – Code editor
- Ganache

* As applicable according to the experiment.
Two sheets per experiment (10-20) to be used.

* Testing Phase: Compilation of Code (error detection)

- Install Node.js and npm on your system.
- Install Truffle using npm install -g truffle.
- Create a Truffle project using truffle init.
- Write a simple Solidity contract and compile using truffle compile.
- Deploy locally using Ganache or Truffle's migration scripts.
- Next, install Hardhat using npm install --save-dev hardhat.
- Initialize a Hardhat project using npx hardhat.
- Compile and deploy the same contract using Hardhat commands.
- Compare gas usage, deployment speed, and error messages in both environments.
- Record your observations

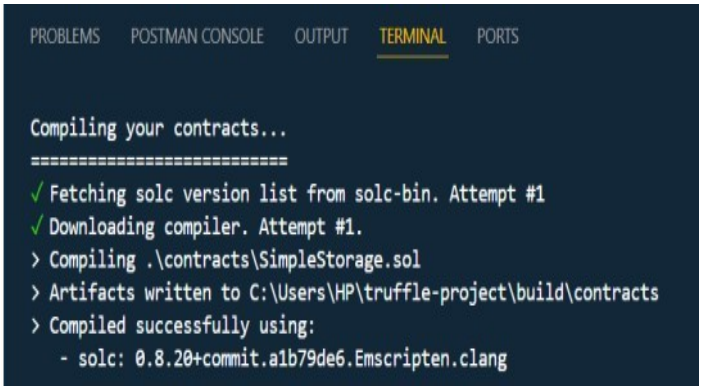
* Implementation Phase: Final Output (no error)

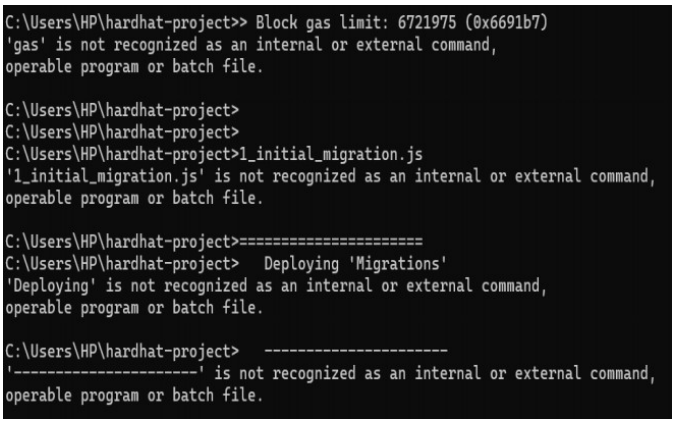
- Both environments successfully compile and deploy the HelloWorld contract.
- Gas usage and execution speed differ slightly — Hardhat executes faster.
- Deployment logs and error traces are more detailed in Hardhat.
-

* As applicable according to the experiment.
Two sheets per experiment (10-20) to be used.

* Implementation Phase: Final Output (no error)

Steps to Deploy Smart Contract in Hardhat :
1.Create a new folder for project
mkdir hardhat-project
cd hardhat-project
2.Initialize npm
npm init -y
4.Install Hardhat
npm install --save-dev hardhat
Setup Hardhat project
5.npx hardhat
Select “Create a JavaScript project”, press Enter for defaults.
6.Write smart contract (contracts/SimpleStorage.sol)
7.Add deployment script (scripts/deploy.js)
8.Compile the contract
9.npx hardhat compile
10.Start local Hardhat blockchain





* As applicable according to the experiment.
Two sheets per experiment (10-20) to be used.

* Observation :

It was observed that both Truffle and Hardhat provide efficient Ethereum development workflows. Truffle is simple and stable, best suited for beginners, while Hardhat offers better performance, advanced debugging, and flexible plugin support. Hardhat's real-time stack tracing and modern interface make it preferable for large and complex blockchain projects.

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 :

Signature of the Faculty :

Regn. No. :

* As applicable according to the experiment.
Two sheets per experiment (10-20) to be used