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

AcademicYear:.....SubjectName:.....SubjectCode:.....

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

Date: .....

Applied and Action Learning

(Learning by Doing and Discovery)

Name of the Experiement : Truffle vs Hardhat – Dev Environment Showdown

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

ALGORITHM / STEPS:

Install Dependencies

- Install Node.js and npm.
- Install Truffle globally: npm install -g truffle
- Install Hardhat locally: npm install --save-dev hardhat

Initialize Project

- Truffle: truffle init
- Hardhat: npx hardhat → select project type interactively

Write Smart Contracts

- Create .sol files in contracts/ folder.

Compile Contracts

- Truffle: truffle compile
- Hardhat: npx hardhat compile

Deploy Contracts to Local Blockchain

- Truffle: truffle migrate (using Ganache)
- Hardhat: npx hardhat run scripts/deploy.js --network localhost

Test Smart Contracts

- Truffle: truffle test
- Hardhat: npx hardhat test

Debug & Verify

- Use Truffle's console or Hardhat's stack traces and console logs.

* Softwares used

- Node.js & npm
- Truffle
- Hardhat
- Ganache (for Truffle local blockchain)
- VS Code (IDE)

* Testing Phase: Compilation of Code (error detection)

For installing truffle
npm install -g truffle

```
C:\Users\HP>npm install -g truffle
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache i
f you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerfu
l.
npm warn deprecated rimraf@2.7.1: Rimraf versions prior to v4 are no longer supported
npm warn deprecated mkdirp-promise@5.0.1: This package is broken and no longer maintained. 'mkdirp' itself supports prom
ises now, please switch to that.
npm warn deprecated har-validator@5.1.5: this library is no longer supported
npm warn deprecated yaeti@0.0.6: Package no longer supported. Contact Support at https://www.npmjs.com/support for more
info.
npm warn deprecated memdown@1.4.1: Superseded by memory-level (https://github.com/Level/community#faq)
npm warn deprecated glob@7.2.0: Glob versions prior to v9 are no longer supported
npm warn deprecated level-errors@2.0.1: Superseded by abstract-level (https://github.com/Level/community#faq)
npm warn deprecated encoding-down@6.3.0: Superseded by abstract-level (https://github.com/Level/community#faq)
npm warn deprecated deferred-leveldown@5.3.0: Superseded by abstract-level (https://github.com/Level/community#faq)
npm warn deprecated levelup@4.4.0: Superseded by abstract-level (https://github.com/Level/community#faq)
npm warn deprecated level-js@5.0.2: Superseded by browser-level (https://github.com/Level/community#faq)
npm warn deprecated level-packager@5.1.1: Superseded by abstract-level (https://github.com/Level/community#faq)
npm warn deprecated level-codec@9.0.2: Superseded by level-transcoder (https://github.com/Level/community#faq)
npm warn deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm warn deprecated multibase@0.6.1: This module has been superseded by the multiformats module
npm warn deprecated apollo-server-errors@3.3.1: The `apollo-server-errors` package is part of Apollo Server v2 and v3, w
hich are now end-of-life (as of October 22nd 2023 and October 22nd 2024, respectively). This package's functionality is
now found in the `@apollo/server` package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for m
ore details.
```

```
C:\Users\HP>npm install -g ganache-cli
npm warn deprecated ganache-cli@6.12.2: ganache-cli is now ganache; visit https://trfl.io/g7 for details

added 1 package in 6s

2 packages are looking for funding
  run `npm fund` for details

C:\Users\HP>
```

Install Hardhat:

```
C:\Users\HP>cd hardhat-project

C:\Users\HP\hardhat-project>npm init -y
Wrote to C:\Users\HP\hardhat-project\package.json:

{
  "name": "hardhat-project",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}



C:\Users\HP\hardhat-project>npm install --save-dev hardhat

added 57 packages, and audited 58 packages in 22s

14 packages are looking for funding
  run `npm fund` for details
```

```
C:\Users\HP\hardhat-project>npm install --save-dev hardhat

added 57 packages, and audited 58 packages in 22s

14 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\Users\HP\hardhat-project>
```

2. Initialize Hardhat Project

```
npx hardhat
```

- Choose “Create a basic sample project”.
- This sets up contracts/, scripts/, and test/ folders.

3. Write Smart Contract

- Create or edit a contract in contracts/, e.g., MyContract.sol.

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

contract MyContract {
    string public message = "Hello Hardhat!";
}
```

4. Compile Contract

```
npx hardhat compile
```

5. Write Deployment Script

- Create a script in scripts/deploy.js:

```
async function main() {
    const MyContract = await ethers.getContractFactory("MyContract");
    const contract = await MyContract.deploy();
    await contract.deployed();
    console.log("Contract deployed to:", contract.address);
}

main().catch((error) => {
    console.error(error);
    process.exitCode = 1;
});
```

6. Run Local Blockchain (Hardhat Network)

- Hardhat automatically runs a local blockchain when deploying, or you can start manually:

npx hardhat node

7. Deploy Contract

npx hardhat run scripts/deploy.js --network localhost

- Outputs the deployed contract address.

8. Verify Deployment

Implementation Phase: Final Output (no error

- **Compilation Success**
- `Compiling 1 Solidity file with 0 warnings`
- `Compilation finished successfully`
- Shows all contracts compiled without errors.
- **Deployment Success**
- **Truffle (Ganache):**
- `Using network 'development'.`
- `Running migration: 1_initial_migration.js`
- `Replacing Migrations...`
- `Migrations deployed at <contract-address>`
-
- **Hardhat (Local Network):**
- `Contract deployed to address:0xa0a68a18938AFD0099b75fF0245AfF8568Cf3608`

* Observations

- Compilation, deployment, and testing complete without warnings or errors.
- Smart contracts work as expected on the local blockchain.
- Both Truffle and Hardhat are ready for integration with frontend or further development.

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:

**Asapplicableaccordingtotheexperiment. Twosheetsperexperiment(10-20)tobeused.*