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**

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

Algorithm: Start the process.

1. **Initialize network parameters** such as number of nodes (peers) and assign each node a unique ID and empty ledger.
2. **Establish peer connections** by linking each node randomly to other nodes to form a mesh network.
3. **Broadcast message** from one initiating node to its connected peers.
4. **Forward message** from each peer to their respective connections, excluding the sender.
5. **Verify each message** received by checking if it is new and valid; ignore duplicates or invalid entries.
6. **Update ledger** of each node with the valid transaction or block.
7. **Apply consensus rule** — accept the first valid message across all nodes and reject conflicting ones.
8. **Display final results** showing how many nodes accepted the message and confirm synchronization.
9. **End** the simulation.

* Softwares used

1. **Node.js**
2. **Truffle Suite**
3. **Ganache**
4. **Visual Studio Code (VS Code)**
5. **Solidity Extension (VS Code Plugin)**

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

* Implementation Phase: Final Output (no error)

Algorithm: 6 Blockchain Dev Tools – Setting Up Environment

1. **Install Node.js:**
   Download and install Node.js for your OS (Windows/macOS/Linux), verify using `node -v` and `npm -v`. It includes npm for managing blockchain packages.
2. **Install Truffle Suite:**
   Use the command `npm install -g truffle` to set up Truffle, a framework for compiling, testing, and deploying Ethereum smart contracts.
3. **Install Ganache:**
   Download and run Ganache to create a local Ethereum blockchain for testing. Customize ports and RPC endpoint (`http://127.0.0.1:7545`) as needed.
4. **Set Up Visual Studio Code & Solidity Extension:**
   Install VS Code and add the Solidity extension to enable smart contract development with syntax highlighting and compilation support.
5. **Create and Deploy Smart Contract:**
   Initialize a new Truffle project using `truffle init`, create a contract (`truffle create contract HelloWorld`), compile it with `truffle compile`, and deploy using `truffle migrate`.

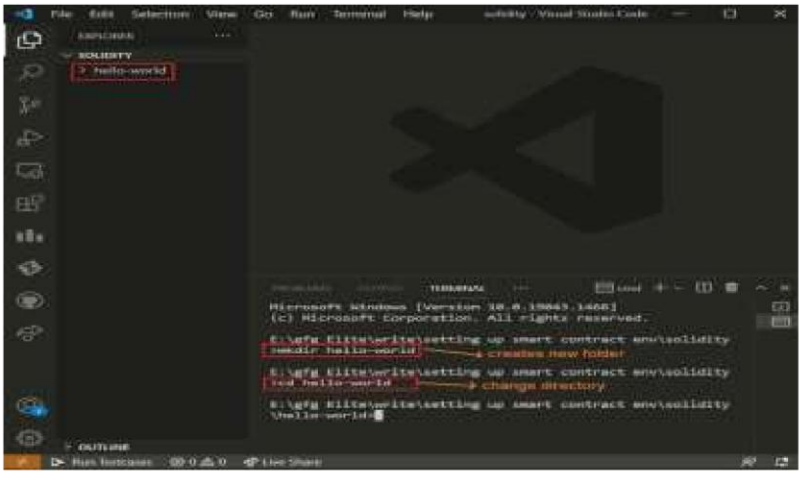
```
PS C:\Users\kamal> node -v
v16.13.2
PS C:\Users\kamal> npm -v
8.1.2
```

```
PS C:\Users\kamal> npm install truffle -g
npm WARN           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           source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
npm WARN           urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN           har-validator@5.1.5: this library is no longer supported
npm WARN           source-map-resolve@0.5.3: See https://github.com/lydell/source-map-resolve#deprecated
npm WARN           circular-json@0.5.9: CircularJSON is in maintenance only, flatted is its successor.
npm WARN           ipld-raw@6.0.0: This module has been superseded by the multiformats module
npm WARN           resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN           apollo-tracing@0.15.0: The `apollo-tracing` package is no longer part of Apollo Server 3. See https:
//www.apollographql.com/docs/apollo-server/migration/#tracing for details
[                  ] | reify:lazystream: timing                                                    Completed in 20
```

Step 1: Open command prompt terminal in VS code and type command-

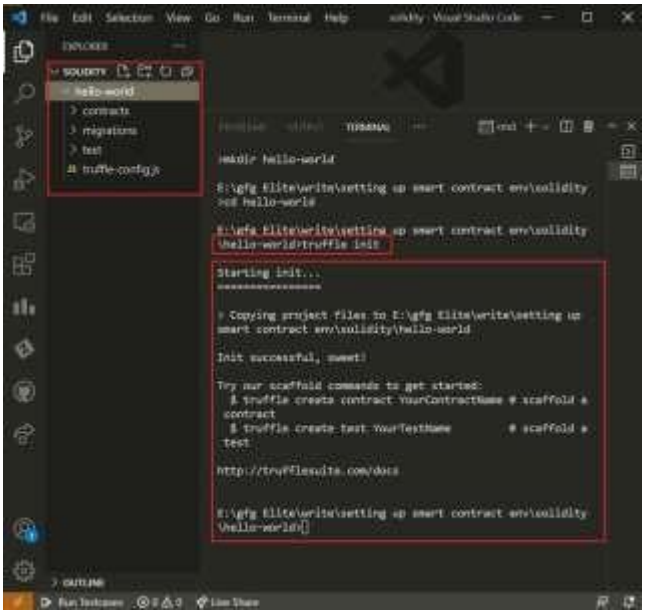
mkdir hello-world

cd hello-world

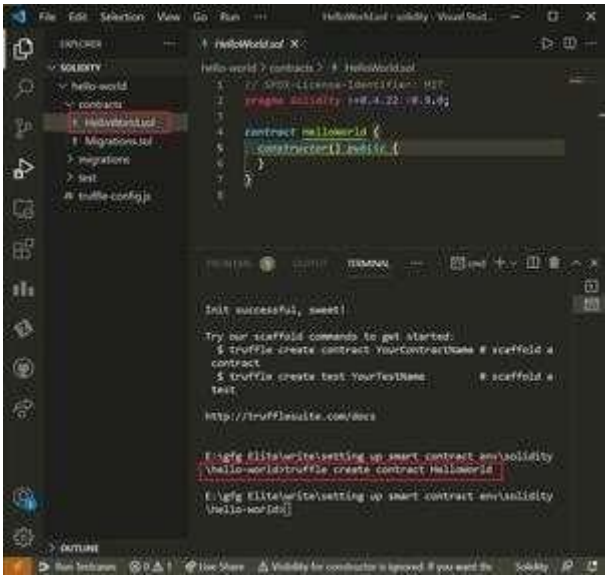


*As applicable according to the experiment.
Two sheets per experiment (10-20) to be used.

Step 2: Generates the template of the smart contract by executing the command- truffle init



Step 3: Now, Create HelloWorld.sol file by executing the command- truffle create contract HelloWorld



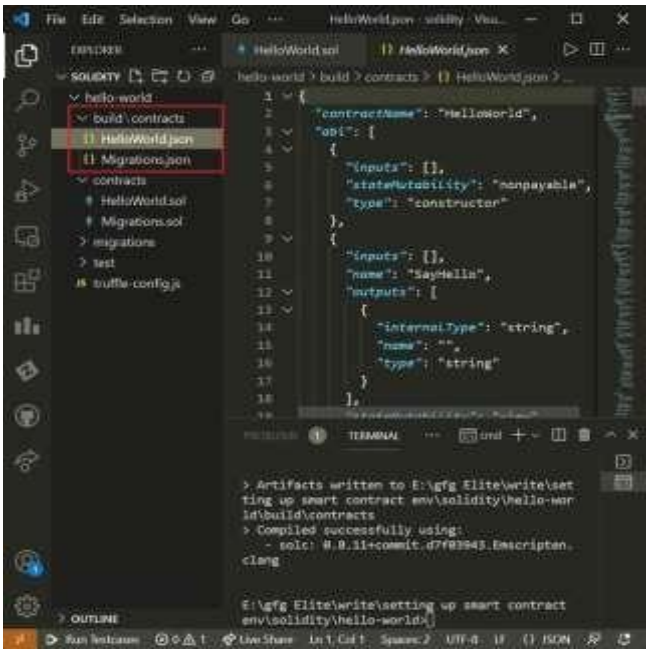
Step 4: Now copy or type the below code to your code editor.

```
pragma solidity >=0.4.22 <0.9.0;

contract HelloWorld
{
  string message;
  constructor() public
  {
    message = "Hello World!";
  }

  function SayHello() view public returns (string memory)
  {
    return message;
  }
}
```


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

Step 5: After typing the above code execute the command-

truffle compile



*As the experiment. Two (10-20) to be

applicable according to sheets per experiment used.

* Implementation Phase: Final Output (no error)

• **Development Environment Configuration:**
Established a structured blockchain development setup using essential tools like Node.js, Truffle, Ganache, and VS Code, ensuring a seamless workflow for compiling, deploying, and testing smart contracts.

• **Efficiency & Integration Optimization:**
Enhanced the development and testing process by integrating Truffle Suite with Ganache for a local Ethereum network, improving debugging speed, compilation accuracy, and deployment efficiency.

• **Compliance & Standardization:**
Adopted globally recognized Ethereum development standards (EVM compatibility) to ensure reliability, transparency, and consistency across smart contract execution environments.

• **Flexibility & Future Scalability:**
Designed a modular, adaptable setup that supports integration with new blockchain frameworks, updates, and development tools, promoting long-term usability and learning adaptability.

* Observations

1. The setup process established a complete local blockchain environment for efficient smart contract development and testing.
2. Integration of Truffle, Ganache, and VS Code streamlined compilation, deployment, and debugging workflows.
3. The environment proved flexible and scalable, supporting future upgrades and diverse blockchain development needs.

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:

applicable according to
sheets per experiment *used.*