



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 :

### \* **Coding Phase: Pseudo Code / Flow Chart / Algorithm**

#### **ALGORITHM:**

1. Install Node.js (which also installs npm) for running JavaScript-based blockchain frameworks.
2. Install MetaMask browser extension for wallet and blockchain interactions.
3. Open Remix IDE in the browser for smart contract development and testing.
4. Install Truffle or Hardhat (using npm) for advanced smart contract compilation and deployment.
5. Compile, deploy, and interact with a smart contract locally.
6. Verify the contract execution by checking transactions in Ganache GUI or MetaMask activity.

### \* **Software used**

1. Node.js
2. MetaMask Wallet (Browser Extension)
3. Remix IDE (web-based)
4. Truffle / Hardhat, Ganache.

## \* Testing Phase: Compilation of Code (error detection)

- 1.Verified Node.js installation by running node -v and npm -v in the terminal.
- 2.Successfully installed MetaMask extension and created a test wallet.
- 3.Successfully accessed Remix IDE and compiled a sample Solidity smart contract.
- 4.Installed Truffle using npm install -g truffle and confirmed installation with truffle version.
- 5.Installed and launched Ganache UI, and verified it created 10 test accounts with balances.
- 6.Connected MetaMask to Ganache by importing private keys and switching to custom RPC.
- 7.Deployed a simple smart contract using Remix and verified the transaction in Ganache logs.

## \*Implementation Phase: Final Output (no error)

- 1.Successfully set up complete blockchain development environment.
- 2.Node.js, npm, Git, VS Code and MetaMask, all working correctly.
- 3.Web3.js able to connect to local blockchain and fetch accounts.

The screenshot displays three windows side-by-side:

- MetaMask Wallet Interface:** Shows account balance (\$0.00), network addresses (5), and activity history for Nov 1, 2025, including a successful contract deployment and an approved spending cap.
- Remix IDE Interface:** Shows the XPLORER sidebar with recent folders like 'sc' and 'SMARTCONTRACTS'. The REMIX workspace shows the v1.1.0 Release page with documentation and website links. Recent workspaces include 'electron' and 'Remix Desktop Release'. A sidebar for 'RemixAI Assistant' is also visible.
- Terminal Window:** Shows the command 'npm install --save-dev hardhat' being run in a Windows command prompt. The output indicates 57 packages added, 33 packages looking for funding, and 0 vulnerabilities found.

## \* Implementation Phase: Final Output (no error)

Applied and Action Learning

C:\Users\HP>node -v  
v22.20.0  
  
C:\Users\HP>npm -v  
10.9.3  
  
C:\Users\HP>

C:\Users\HP>npm install -g ganache  
added 336 packages in 31s  
6 packages are looking for funding  
run `npm fund` for details  
C:\Users\HP>

The screenshot shows a terminal window with the command prompt C:\Users\HP>. It displays two commands: 'node -v' showing version v22.20.0 and 'npm -v' showing version 10.9.3. Below the terminal is a code editor interface for a blockchain application named 'FRESHERS'. The code editor shows several files: LoadingScreen.js, InvitationSection.js, ParticleBackground.js, and HeroSection.js. The HeroSection.js file is open and contains the following code:

```

const HeroSection = () => {
  return (
    <section style={heroStyles.section}>
      > ...
      > Thank You for Making UDBHAV 2.0 Unforgettable! 🎉
    </motion.div>
    /* University name */
    <motion.div
      style={heroStyles.universityInfo}
      initial={{ y: 40, opacity: 0 }}
      animate={{ y: 0, opacity: 1 }}
      transition={{ duration: 0.8, delay: 0.9 }}>
      <p style={heroStyles.universityName}>
        Centurion University of Technology and Management
    </motion.div>
  </section>
}

export default HeroSection;

```

The code editor has tabs for PROBLEMS, POSTMAN CONSOLE, OUTPUT, TERMINAL, and PORTS. The TERMINAL tab is selected, showing the command 'npm install -g ganache' and its output. The right side of the interface shows a sidebar with various icons and a search bar.

## \* Observations

- 1.Tools like Ganache and MetaMask allow for complete blockchain interaction without requiring real networks.
- 2.Node.js and npm serve as the foundation for installing blockchain frameworks like Truffle and Hardhat.
- 3.Remix IDE is a powerful online tool for smart contract development with zero configuration.
- 4.After environment setup, blockchain apps can be built, tested, and debugged easily using this local setup.

## 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		
<b>Total</b>	<b>50</b>		

*Signature of the Student:*

Name :

Regn. No. :

Page No.....

*Signature of the Faculty:*

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