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 : Web3 Connect – Contract Calls via Frontend

Objective/Aim:

To demonstrate how a **React frontend** can connect to a deployed **smart contract** using **Web3.js** or **Ethers.js** and perform both **read** and **write** operations (contract calls).

Apparatus/Software Used:

- Visual Studio Code (VS Code)
- MetaMask Wallet
- Hardhat / Truffle Framework
- Web3.js or Ethers.js Library
- React.js Framework
- Internet Connection

Theory/Concept:

In a **Decentralized Application (DApp)**, the frontend interacts with blockchain smart contracts using Web3 libraries.

The process involves connecting a wallet, loading the contract ABI, and calling contract functions from the user interface.

Key Concepts:

Contract ABI (Application Binary Interface):

Defines how to interact with the smart contract — functions, events, and parameters.

Web3.js / Ethers.js:

JavaScript libraries that allow web applications to communicate with Ethereum-compatible blockchains.

Read Calls (Call):

Used to fetch blockchain data (no gas fee).

Write Calls (Send/Transaction):

Used to modify blockchain data (requires gas fee and wallet confirmation).

PROCEDURE

Step 1: Deploy Smart Contract (Backend Setup)

contract: SimpleStorage.sol

```
pragma solidity ^0.8.0;

contract SimpleStorage {
    uint256 public storedData;

    function set(uint256 x) public {
        storedData = x;
    }

    function get() public view returns (uint256) {
        return storedData;
    }
}
```

Deploy the above contract using **Hardhat** or **Remix** on a testnet (e.g., Sepolia).

Copy the **contract address** and **ABI** after deployment.

Step 2: React Project Setup

1. Create a new React app:

2.Create a file contractABI.json and paste your contract ABI there.

Step 3: Connect Frontend to Smart Contract

In App.js:

```
import React, { useState, useEffect } from "react";
import Web3 from "web3";
import contractABI from "./contractABI.json";

function App() {
  const [account, setAccount] = useState("");
  const [contract, setContract] = useState(null);
  const [inputValue, setInputValue] = useState("");
  const [storedValue, setStoredValue] = useState("");

  const contractAddress = "0xYourDeployedContractAddress";

  useEffect(() => {
    connectWallet();
  }, []);
```

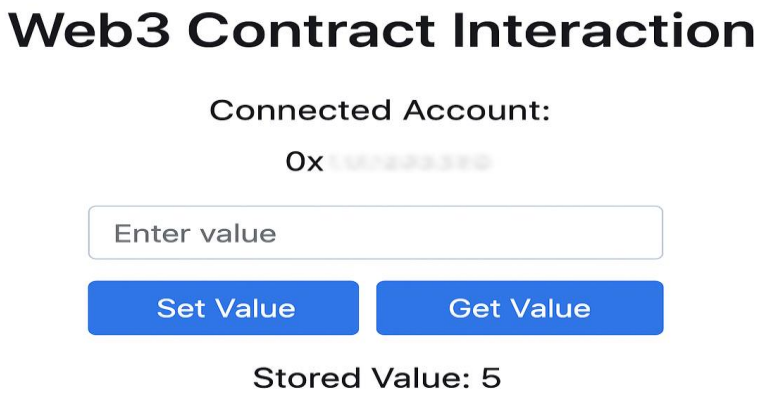
```
const connectWallet = async () => {
  if (window.ethereum) {
    const web3 = new Web3(window.ethereum);
    await window.ethereum.request({ method: "eth_requestAccounts" });
    const accounts = await web3.eth.getAccounts();
    setAccount(accounts[0]);
    const instance = new web3.eth.Contract(contractABI, contractAddress);
    setContract(instance);
  } else {
    alert("MetaMask not detected!");
  }
};
```

```
  return (
    <div style={{ textAlign: "center", marginTop: "50px" }}>
      <h1>Web3 Contract Interaction</h1>
      <p>Connected Account: {account}</p>
      <input
        type="number"
        placeholder="Enter value"
        value={inputValue}
        onChange={(e) => setInputValue(e.target.value)}
      />
      <button onClick={setValue}>Set Value</button>
      <button onClick={getValue}>Get Value</button>
      <p>Stored Value: {storedValue}</p>
    </div>
  );
}

export default App;
```

Step 4: Run the Frontend

1. Start the React app:
2. Open your browser → `http://localhost:3000`
3. Connect your MetaMask wallet and interact with the contract.



Observation Table:

- MetaMask successfully connected to the React frontend.
- The user was able to **set** and **get** values from the blockchain contract.
- set() required a transaction confirmation in MetaMask (write operation).
- get() retrieved stored data instantly (read operation).

ASSESSMENT

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

Signature of the Student:

Name :

Regn. No.

Signature of the Faculty: