



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: React Start – DApp Frontend Scaffolding

*Coding Phase: Pseudo Code / Flow Chart / Algorithm



Start

- Initialize the frontend framework (React app).
- Import blockchain libraries (ethers.js).
- Detect if MetaMask is installed:
- If not, prompt the user to install it.
- Request wallet connection from the user.
- Retrieve and display user's Ethereum address.
- Load contract ABI and address.
- Create a contract instance using Ethers.js.
- Read contract data and display on the UI.
- When user performs action (e.g., submit transaction):
- Capture input values
- Call smart contract function
- Await transaction confirmation
- Update UI with transaction result.



End

* Software used:

- Laptop
- Visual Studio Code (code editor)
- MetaMask Wallet (browser extension)
- Remix IDE (web-based smart contract IDE)
- Node.js
- React (via create-react-app)
- Ether.js (Ethereum JavaScript library)
- dotenv (for environment variables)

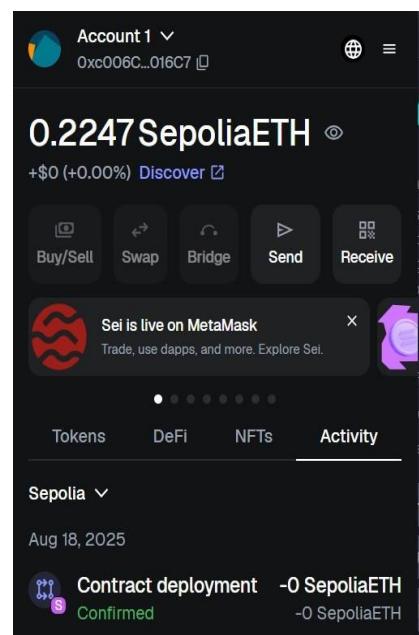
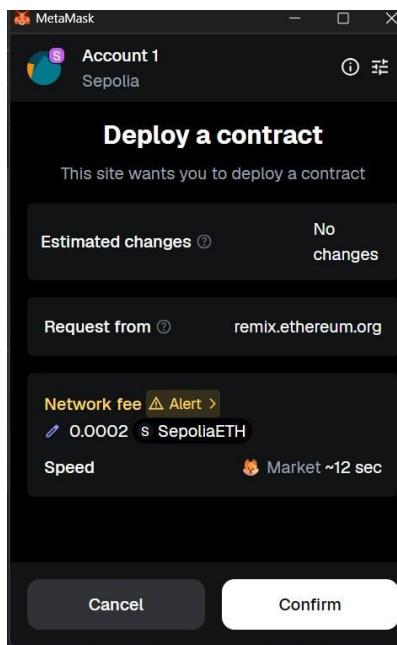
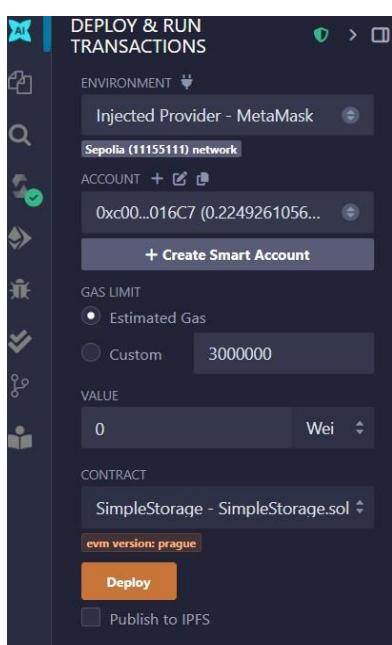
Page No.....

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

* Testing Phase: Compilation of Code (error detection)

- First we have to go Remix IDE and create a .sol file named as simpleStorage.sol and write our smart contract.
- Then we need to compile our smart contract and copy the generated ABI
- After successful compilation deploy the smart contract and choose the environment to Injected Provider - MetaMask
- After deployment under Deployed Contracts section copy the contract address for future use.
- Then using web3.js library we create frontend and interact with our wallet.

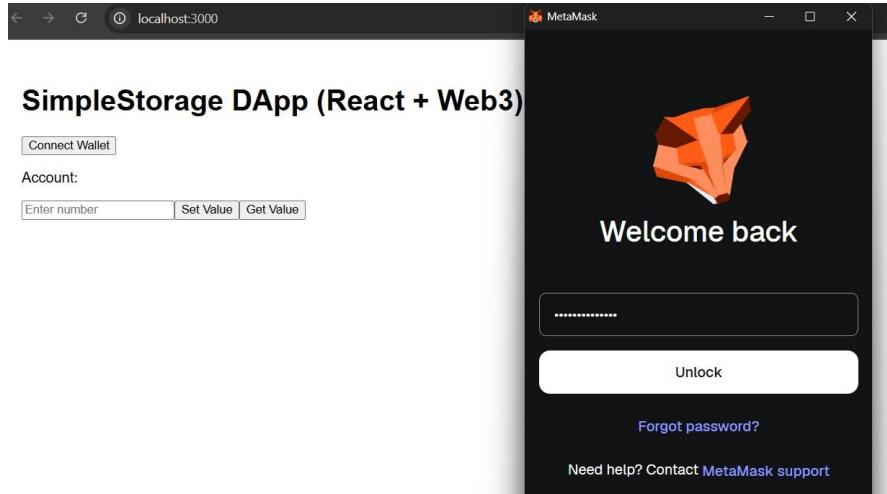
```
5     uint256 private data;
6
7     function set(uint256 _data) public {
8         data = _data;
9     }
10
11    function get() public view returns (uint256) {
12        return data;
13    }
14 }
```



* Implementation Phase: Final Output (no error)

- Now we have to create a folder named as “frontend” and open the terminal and move to the current frontend directory.
- Inside frontend we have to create a ‘.env’ file where we will store our contract address.
- In the frontend/src/ folder we have to create a ABI.json file to store our contract ABI.
- Now in the App.js file we have to write our frontend code and wallet connection function.

* Implementation Phase: Final Output (no error)



SimpleStorage DApp (React + Web3)

Connect Wallet

Account: 0x19b9a3978978a4165cE5194FDD1CbD4f6a79525F

Set Value

Stored Value: 10

Page No.....

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

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. :

Page No.....

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

