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 FrontendWeb3

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

- Setup **MetaMask wallet** in the browser.
- Open **Remix IDE** and write the smart contract (SimpleStorage.sol).
- Compile the smart contract to generate the **ABI**.
- Deploy the contract on **Sepolia Testnet** using MetaMask (Injected Provider).
- Copy the **deployed contract address**.
- Create a **React frontend project** using create-react-app.
- Inside src, create ABI.js to store the smart contract ABI.
- Create a .env file in the project root to store **contract address** and **network details**.
- Install **ethers.js** (primary library for blockchain interaction).
- In App.js, implement wallet connection and contract interaction logic using **ethers.js**.
- Build a **UI** to store and retrieve values from the contract.
- Run the project with npm start and verify blockchain interactions through MetaMask prompts.

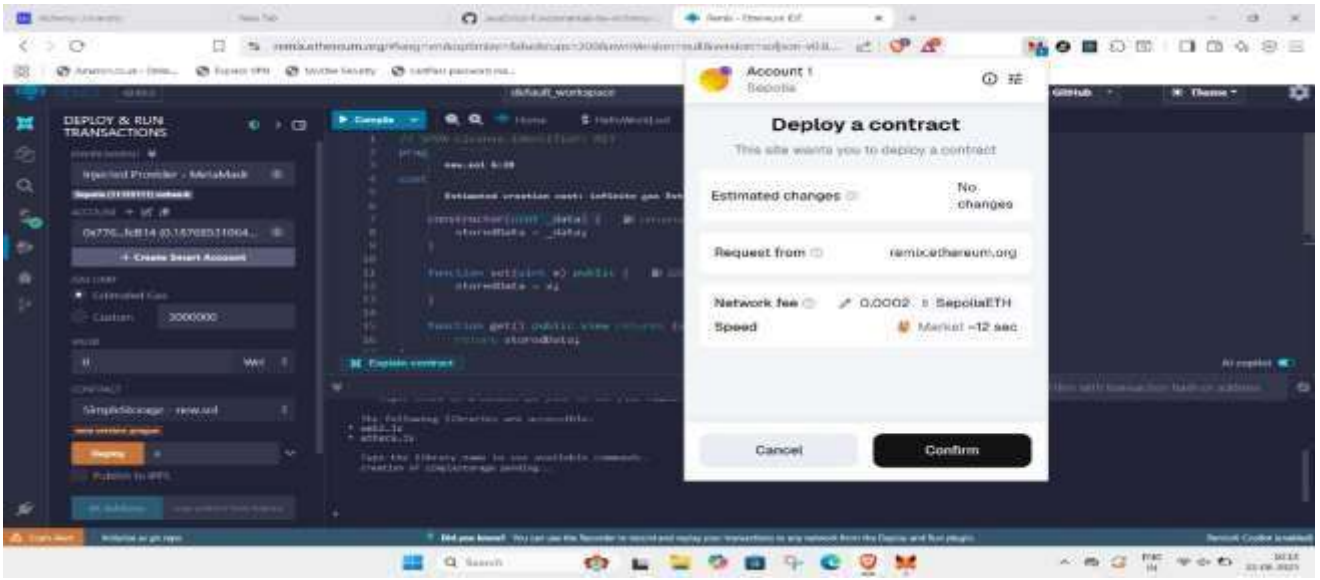
Software used

1. MetaMask Wallet
2. Remix IDE.
3. MS Word.
4. Brave for researching.

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

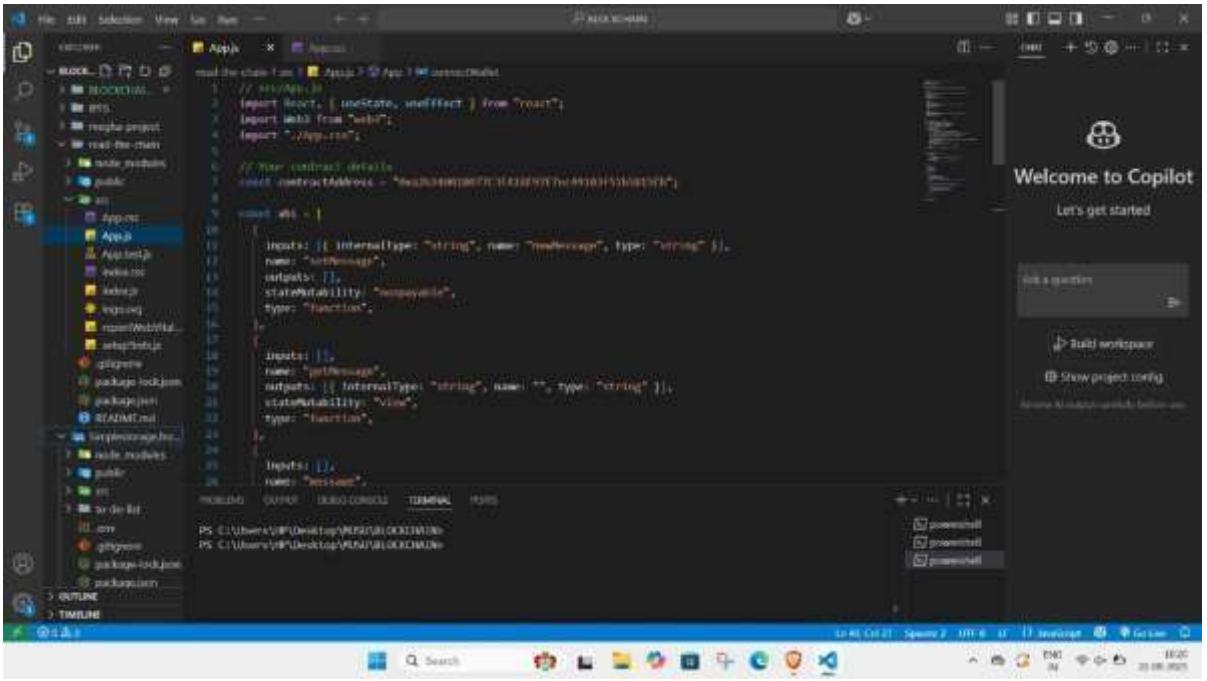
* Implementation Phase: Final Output (no error)

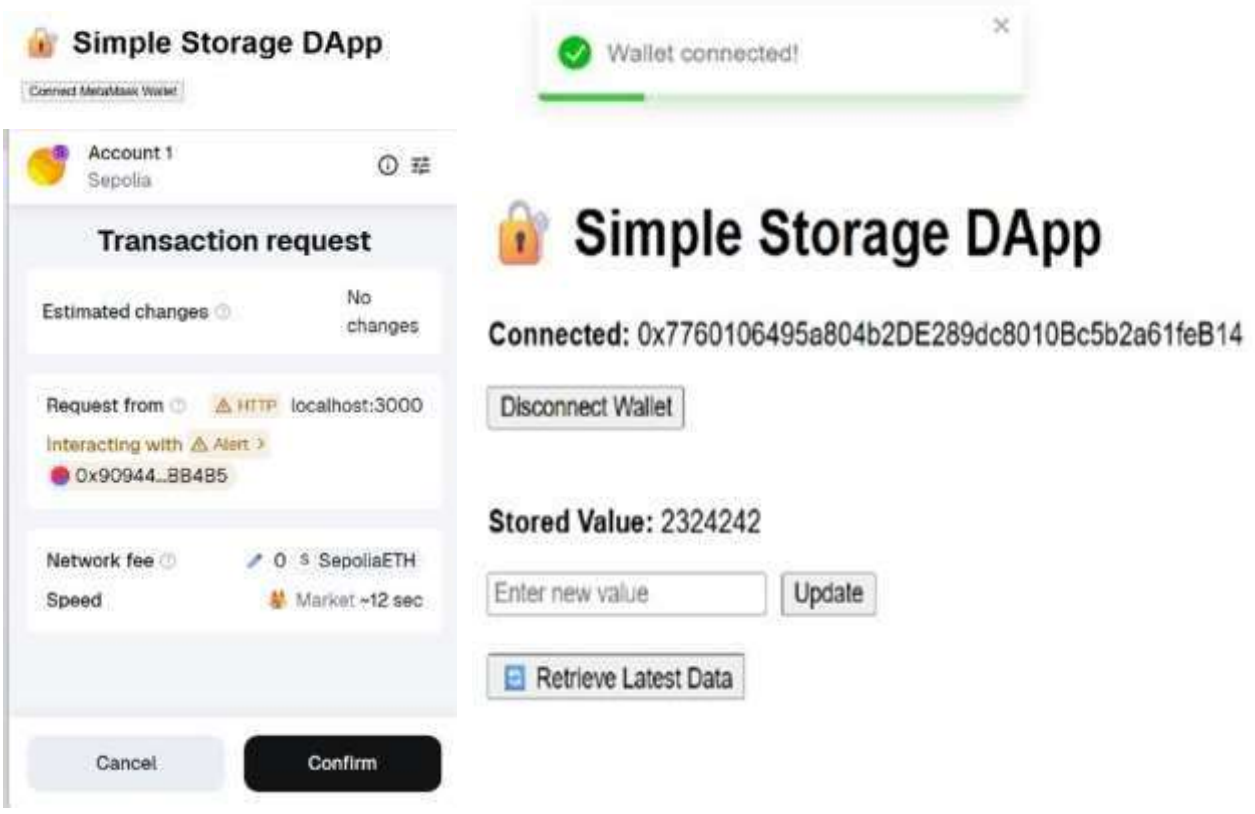
- Smart contract successfully compiled in Remix IDE without errors.
- Contract deployed on **Sepolia Testnet** using MetaMask.
- ABI and contract address correctly imported into React frontend.
- `.env` file securely holds contract address and network details.
- Wallet connection established via MetaMask in the frontend.
- User can **store** values in the contract and **retrieve** them through the UI.
- Frontend interacts smoothly with blockchain using **ethers.js** (preferred over web3.js for simplicity & security).
- Final project runs successfully with `npm start` showing no errors.



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

* Implementation Phase: Final Output (no error)





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

* Observations:

- Contract successfully deployed on **Sepolia Testnet**.
- **MetaMask wallet connection** worked smoothly with the frontend.
- Frontend UI allowed **real-time interaction** with the smart contract (store/retrieve values).
- **ethers.js** ensured secure and simplified interaction compared to web3.js.
- .env file securely handled contract details and network configuration.
- Data stored in the contract remained **persistent on the blockchain**.
- The project acted as a **scaffold for future DApp 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:

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