|  |  |
|---|---|
| CENTURION UNIVERSITY *Shaping Lives... Empowering Communities!* | 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: Web3 Connect – Contract Calls via Frontend**

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

```
➢ Start
➢ Import Ethers.js / Web3.js library.
➢ Detect MetaMask availability in the browser.
➢ Request connection to the user's wallet.
➢ Retrieve connected account address.
➢ Load the smart contract using contract ABI and contract address.
➢ Create buttons on UI to:
      Read data (non-transactional call).
      Write data (transactional call).
➢ When a user clicks "Read", fetch data from contract and display.
➢ When a user clicks "Write", send transaction via MetaMask.
➢ Confirm transaction and update data.
➢ End
```

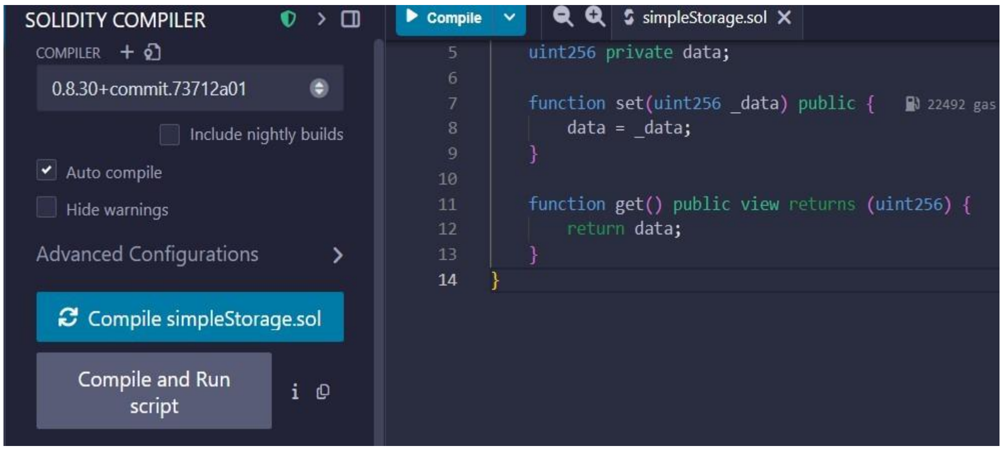
## * Software used:

- Laptop
- Visual Studio Code (code editor)
- MetaMask Wallet (browser extension)
- Remix IDE (web-based smart contract IDE)
- Node.js
- React (via create-react-app)
- Web3.js (Ethereum JavaScript library)
- dotenv (for environment variables)

***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.



SOLIDITY COMPILER

COMPILER: 0.8.30+commit.73712a01

☐ Include nightly builds

☑ Auto compile

☐ Hide warnings

Advanced Configurations

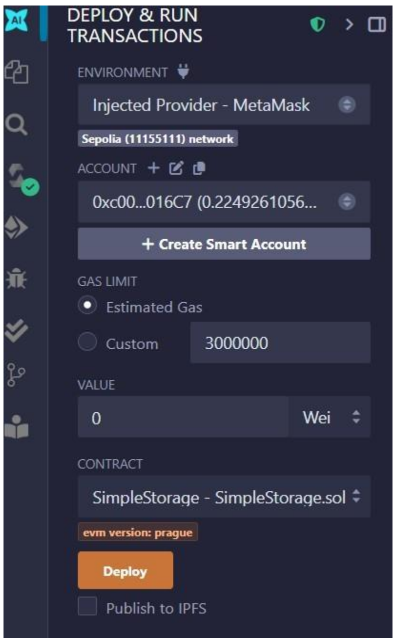
Compile simpleStorage.sol

Compile and Run script

```
    uint256 private data;

    function set(uint256 _data) public {    22492 gas
        data = _data;
    }

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



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Injected Provider - MetaMask

Sepolia (11155111) network

ACCOUNT: 0xc00...016C7 (0.2249261056...

+ Create Smart Account

GAS LIMIT: ● Estimated Gas ○ Custom 3000000

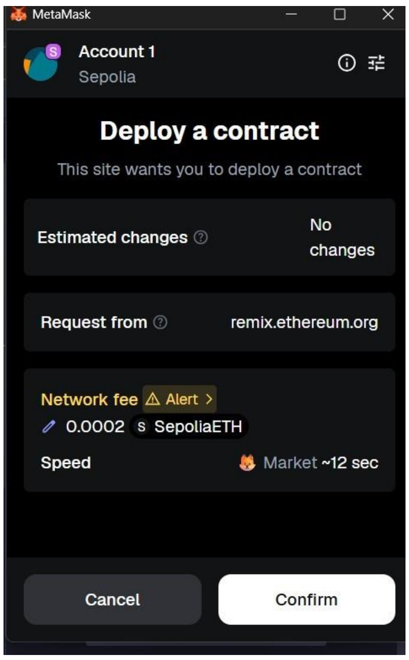
VALUE: 0 Wei

CONTRACT: SimpleStorage - SimpleStorage.sol

evm version: prague

Deploy

☐ Publish to IPFS



MetaMask

Account 1

Sepolia

**Deploy a contract**

This site wants you to deploy a contract

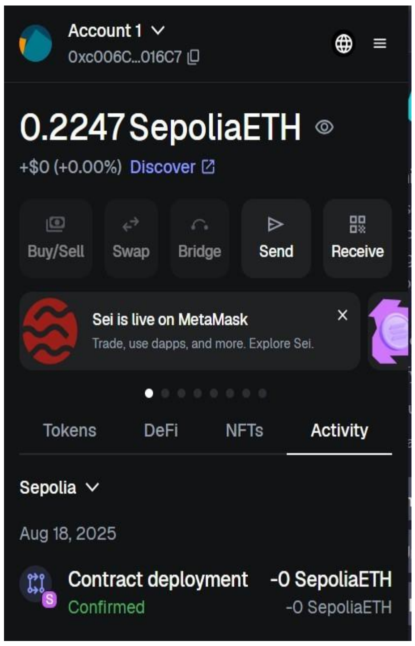
Estimated changes: No changes

Request from: remix.ethereum.org

Network fee ⚠ Alert: 0.0002 SepoliaETH

Speed: Market ~12 sec

Cancel | Confirm



Account 1

0xc006C...016C7

0.2247 SepoliaETH

+$0 (+0.00%) Discover

Buy/Sell | Swap | Bridge | Send | Receive

Sei is live on MetaMask

Trade, use dapps, and more. Explore Sei.

Tokens | DeFi | NFTs | Activity

Sepolia

Aug 18, 2025

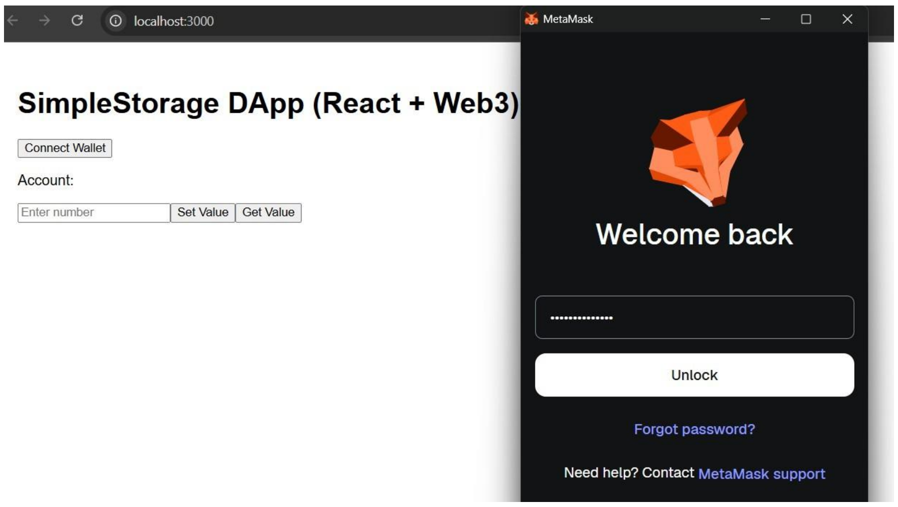
Contract deployment: -0 SepoliaETH

Confirmed: -0 SepoliaETH

## * 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.
- Then we can interact with the UI such as connecting to wallet and set and get functions.

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


localhost:3000

SimpleStorage DApp (React + Web3)

Connect Wallet

Account:

Enter number | Set Value | Get Value

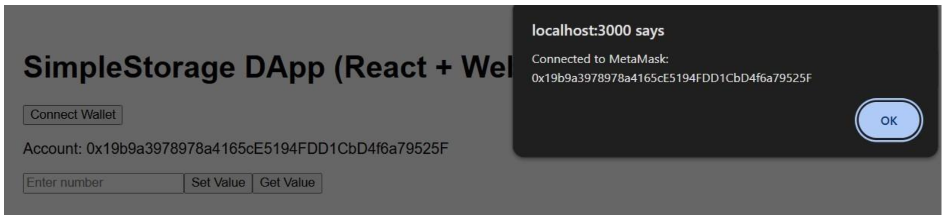
MetaMask

Welcome back

Unlock

Forgot password?

Need help? Contact MetaMask support


SimpleStorage DApp (React + Wel

Connect Wallet

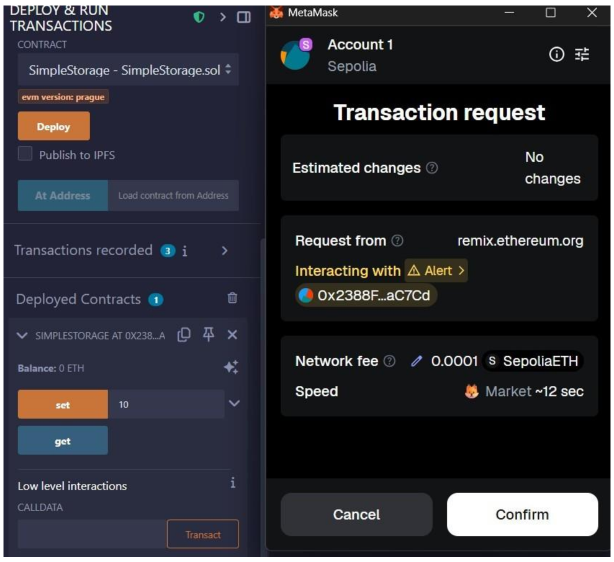
Account: 0x19b9a3978978a4165cE5194FDD1CbD4f6a79525F

Enter number | Set Value | Get Value

localhost:3000 says

Connected to MetaMask:
0x19b9a3978978a4165cE5194FDD1CbD4f6a79525F

OK


DEPLOY & RUN TRANSACTIONS

CONTRACT

SimpleStorage - SimpleStorage.sol

evm version: prague

Deploy

Publish to IPFS

At Address | Load contract from Address

Transactions recorded 3

Deployed Contracts 1

SIMPLESTORAGE AT 0X238...A

Balance: 0 ETH

set | 10

get

Low level interactions

CALLDATA

Transact

MetaMask

Account 1
Sepolia

Transaction request

Estimated changes: No changes

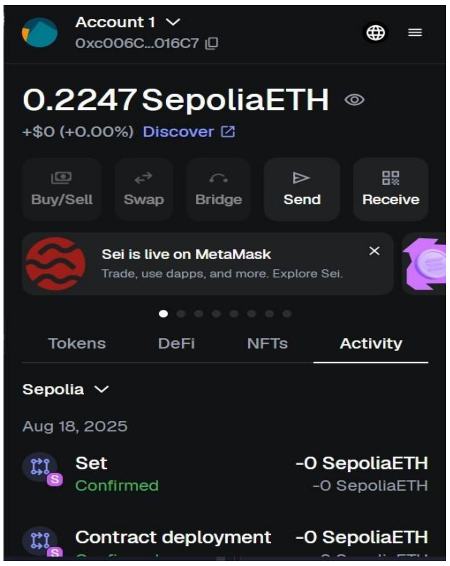
Request from: remix.ethereum.org

Interacting with ⚠ Alert: 0x2388F...aC7Cd

Network fee: 0.0001 SepoliaETH

Speed: Market ~12 sec

Cancel | Confirm


Account 1
0xc006C...016C7

0.2247 SepoliaETH

+$0 (+0.00%) Discover

Buy/Sell | Swap | Bridge | Send | Receive

Sei is live on MetaMask
Trade, use dapps, and more. Explore Sei.

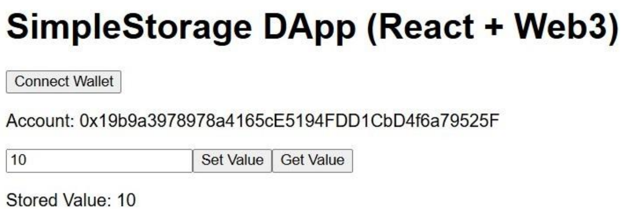
Tokens | DeFi | NFTs | Activity

Sepolia

Aug 18, 2025

Set — Confirmed — -0 SepoliaETH — -0 SepoliaETH

Contract deployment — -0 SepoliaETH


SimpleStorage DApp (React + Web3)

Connect Wallet

Account: 0x19b9a3978978a4165cE5194FDD1CbD4f6a79525F

10 | Set Value | Get Value

Stored Value: 10

** 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. :***

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