



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 : UI for DApps – Building a DApp Frontend

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

ALGORITHM:

- **Start** by setting up the development environment with **Node.js**, **npm**, and a code editor like VS Code.
- **Initialize a frontend project** using frameworks like **React.js** or **Vue.js** to build the DApp interface.
- **Install Web3.js or Ethers.js** libraries to enable communication between the frontend and Ethereum blockchain.
- **Connect the DApp to a blockchain** by linking the Web3/Ethers instance to MetaMask or a local blockchain (Ganache/testnet).
- **Import the smart contract ABI and address** into the frontend to interact with deployed contracts.
- **Create UI components** for user actions such as reading data, sending transactions, or displaying balances.
- **Test and debug the DApp frontend** to ensure real-time interaction with the smart contract and accurate display of blockchain data.
- **End** the process after verifying that the DApp frontend communicates correctly with the blockchain and provides a user-friendly interface.

* Software used

1. Metamask wallet
2. Remix IDE
3. Brave Browser

* Testing Phase: Compilation of Code (error detection)

Smart contract solidity code

```
//SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
contract SimpleStorage {
    uint public storedData;

    constructor(uint _data) {
        storedData = _data;
    }
    function set(uint x) public {
        storedData = x;
    }
    function get() public view returns (uint) {
        return storedData;
    }
}
```

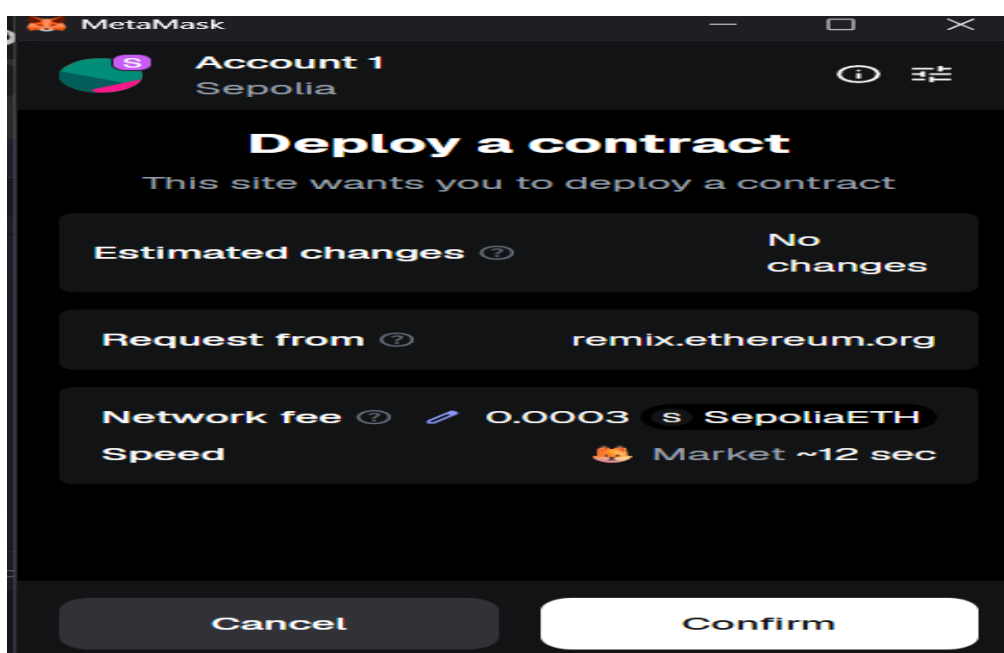
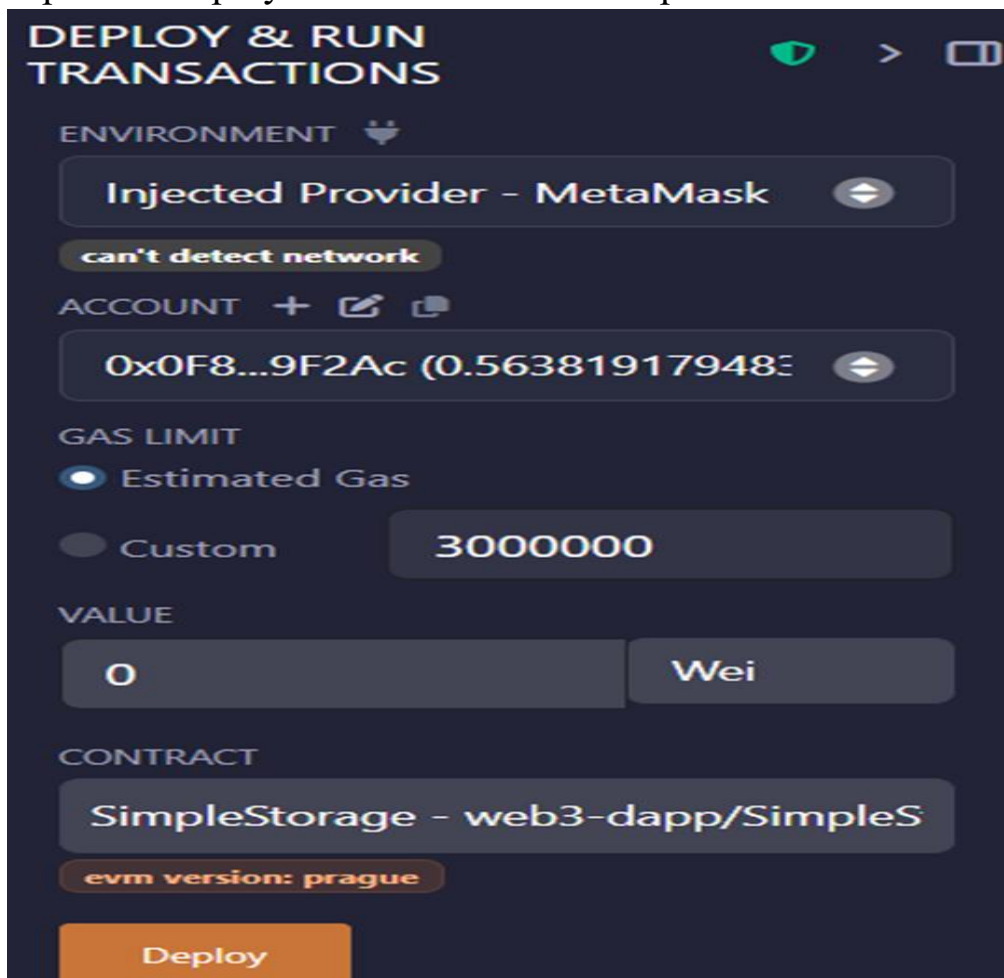
ABI key

```
export const simpleStorageABI =
[
  {
    "inputs": [
      {
        "internalType": "uint256",
        "name": "_data",
        "type": "uint256"
      }
    ],
    "stateMutability": "nonpayable",
    "type": "constructor"
  },
  {
    "inputs": [],
    "name": "get",
    "outputs": [
      {
        "internalType": "uint256",
        "name": "",
        "type": "uint256"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "uint256",
        "name": "x",
        "type": "uint256"
      }
    ],
    "name": "set",
    "outputs": [],
    "stateMutability": "nonpayable",
```

```
    "type": "function"
  },
  {
    "inputs": [],
    "name": "storedData",
    "outputs": [
      {
        "internalType": "uint256",
        "name": "",
        "type": "uint256"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  }
]
```

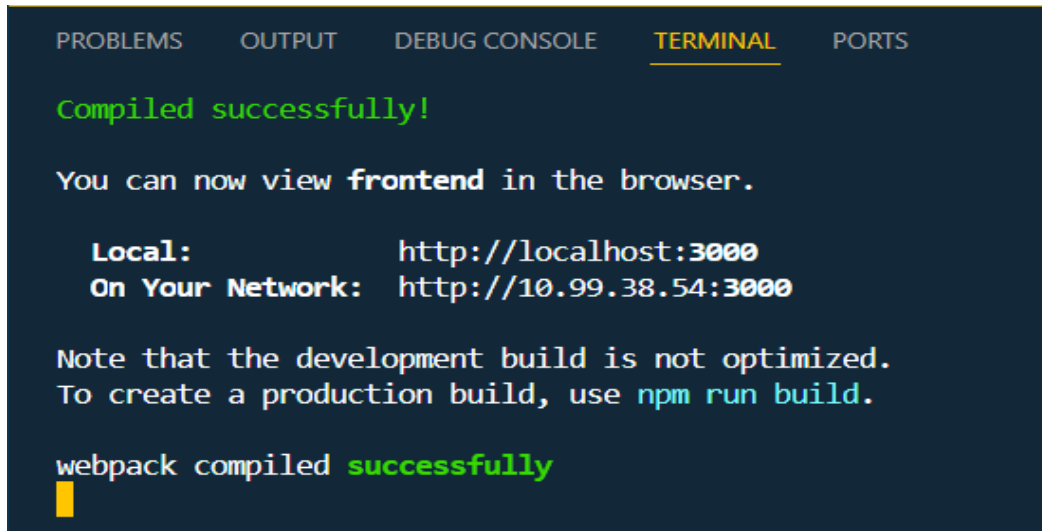
* Implementation Phase: Final Output (no error)

After compilation deploy the smart contract in sepolia test network using metamask



* Implementation Phase: Final Output (no error)

Now open terminal and run the project by writing the code `npm start`



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

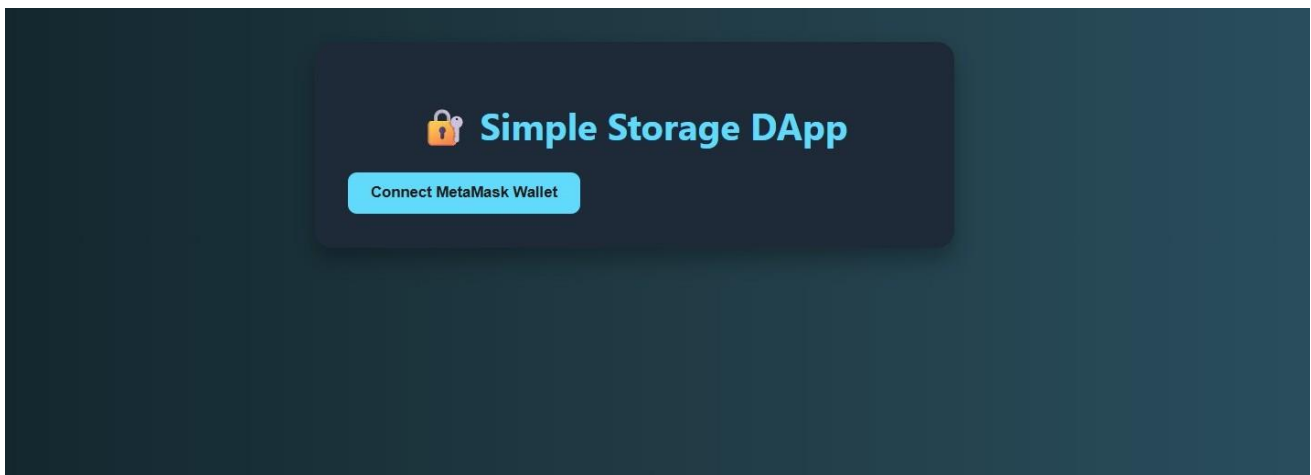
Compiled successfully!

You can now view frontend in the browser.

Local:      http://localhost:3000
On Your Network: http://10.99.38.54:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```



Then connect the wallet with the Dapp.

* Implementation Phase: Final Output (no error)

Applied and Action Learning

* Observations

- The DApp frontend was successfully set up using React.js and integrated with Web3.js/Ethers.js.
- Connection to the blockchain (local Ganache or testnet) was established via MetaMask.
- Smart contract functions were correctly called from the frontend, enabling reading and writing of blockchain data.
- UI components displayed real-time information such as account transaction status.

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:

Page No.....

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