



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 :** UI for DApps – Building a DApp Frontend

### Objective/Aim:

To design and implement a **frontend interface** for a Decentralized Application (DApp) using **React.js** and connect it to the Ethereum blockchain through **Web3.js** or **Ethers.js**.

### Apparatus/Software Used:

1. **React.js (Frontend Framework)**
2. **Node.js & NPM (for package management)**
3. **MetaMask Wallet (browser extension)**
4. **Web3.js / Ethers.js (Blockchain interaction library)**
5. **Solidity Smart Contract (deployed on testnet)**
6. **VS Code (IDE)**

### 1. Theory/Concept:

A **Decentralized Application (DApp)** consists of:

1. **Frontend (UI)** – built using web technologies like React, HTML, CSS, JavaScript.
2. **Smart Contract (Backend)** – written in Solidity and deployed on blockchain.
3. **Wallet Connection** – using MetaMask to authenticate users and handle transactions.

The frontend interacts with the blockchain via Web3.js or Ethers.js to:

- Read smart contract data
- Execute transactions
- Display blockchain states dynamically

# procedure

## Setup React Project

```
C:\Users\pikun\OneDrive\Desktop>npx create-react-app dapp-ui

Creating a new React app in C:\Users\pikun\OneDrive\Desktop\dapp-ui.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

added 1324 packages in 2m

271 packages are looking for funding
  run 'npm fund' for details

Initialized a git repository.

Installing template dependencies using npm...

added 17 packages, and changed 1 package in 10s

271 packages are looking for funding
  run 'npm fund' for details
Removing template package using npm...

C:\Users\pikun\OneDrive\Desktop>cd dapp-ui

C:\Users\pikun\OneDrive\Desktop\dapp-ui>npm install ethers

added 9 packages, and audited 1350 packages in 7s

274 packages are looking for funding
  run 'npm fund' for details

9 vulnerabilities (3 moderate, 6 high)

To address all issues (including breaking changes), run:
  npm audit fix --force
```

## Create Smart Contract Connection

Inside src/App.js, connect MetaMask and smart contract:

The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows the project structure for 'DAPP-UI', including 'node\_modules', 'public', 'src', and 'App.js'. The code editor shows the content of 'App.js', which includes imports for 'useState' from 'react' and 'ethers' from 'ethers'. It also imports 'abi' from './SimpleStorageABI.json'. The code defines an 'App' function that uses 'useState' to manage 'account', 'value', and 'contractAddress'. It includes an async function 'connectWallet' that prompts the user to install MetaMask and connects to the MetaMask account. It also includes an async function 'readData' that uses 'ethers' to connect to the MetaMask provider and read data from the smart contract. The code is wrapped in a 'function App()' block. Below the code editor, the terminal shows the command 'npm start' and the output 'You can now view dapp-ui in the browser.' with local and network URLs. It also notes that the development build is not optimized and provides instructions for creating a production build using 'npm run build'. The terminal also shows 'webpack compiled successfully'.

```
src > App.js > App
1 import { useState } from "react";
2 import { ethers } from "ethers";
3 import abi from "./SimpleStorageABI.json";
4
5 function App() {
6   const [account, setAccount] = useState("");
7   const [value, setValue] = useState("");
8   const contractAddress = "0xDA0Bab807633f07f013f94DD0E6A4F96F8742853";
9
10  async function connectWallet() {
11    if (!window.ethereum) return alert("Install MetaMask");
12    const [address] = await window.ethereum.request({ method: "eth_requestAccounts" });
13    setAccount(address);
14  }
15
16  async function readData() {
17    const provider = new ethers.BrowserProvider(window.ethereum);
18    const contract = new ethers.Contract(contractAddress, abi, provider);
19    const storedValue = await contract.data();
20    setValue(storedValue.toString());
21  }
22
23  return (
24    <div>
25      <p>Account: {account}</p>
26      <p>Value: {value}</p>
27      <p>Contract Address: {contractAddress}</p>
28      <p>Connect Wallet: {connectWallet}</p>
29      <p>Read Data: {readData}</p>
30    </div>
31  );
32}
33
34export default App;
```

```
PS C:\Users\pikun\OneDrive\Desktop\dapp-ui> npm start

You can now view dapp-ui in the browser.

Local:      http://localhost:3000
On Your Network: http://192.168.29.107:3000

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

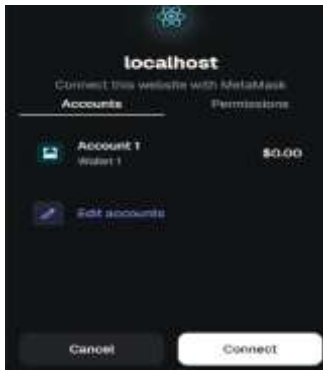
webpack compiled successfully
```

### Connect MetaMask

- Open MetaMask → Connect to **Sepolia Test Network**.
- Approve site connection when prompted.

### Run the App

- The DApp will launch in the browser at <http://localhost:3000>.



## Simple Storage DApp

Connected Account: 0xb835d4728e2811f5ced713db1d0622a101ba194b

Stored Value:

### Observation Table:

Step	Action Taken	Output/Result
React project setup	<code>npm create-react-app dapp-ui</code>	Frontend project initialized
MetaMask connection	Click "Connect Wallet"	Account address displayed
Smart contract read	Click "Read Value"	On-chain data displayed in UI
Browser console log	Shows RPC calls to blockchain	Successful connection logs

## 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		
<b>Total</b>	<b>50</b>		

**Signature of the Student:**

Name :

Regn. No. :

**Signature of the Faculty:**