BÁO CÁO LAB

Họ và tên: Trương Chí Chọn

MSHV: 24C12025

LAB: Xây dựng DAPP trên một mạng ETHEREUM

Nội dung:

- Thiết lập một local Ethereum blockchain
- Triển khai một smart contract trên Ethereum
- Tiển khai một Dapp tương tác với smart contract

Thực hiện

1. **Cài đặt**

   Private Ethereum blockchain – sử dụng Ganache

- Mạng ảo này sẽ không có node nào và không có mining time.
- Tạo ra 10 địa chỉ Ethereum ảo

  Bước 1: Run → cmd

  Bước 2: Gõ lệnh

  >npm install -g ganache-cli

  >ganache-cli

```
C:\Users\chont>ganache-cli
Ganache CLI v6.12.2 (ganache-core: 2.13.2)

Available Accounts
==================
(0) 0x49537042FE641d12768C6921714D1470F1523527 (100 ETH)
(1) 0x4e1F38B9bE1095624bAC9379DF25451caa5052bf (100 ETH)
(2) 0xC9BD6fC2373Aad8D62916d89878514638 99A0D24 (100 ETH)
(3) 0x6464d522f98734003DEA250b1b7dDCfE77F3EfB6 (100 ETH)
(4) 0x2bd2a8008F40a3C1863248b6EE0439f5b85de861 (100 ETH)
(5) 0x92A2d1F3A7A97943c4628bB77CAF68488F9049ca (100 ETH)
(6) 0xdb820EAEe05Db6d042d33BE0C6e54B8F88E99A53 (100 ETH)
(7) 0x180e06A88cdc2122EedC67dE045ae620CA29A167 (100 ETH)
(8) 0x214eAC27b4D0246b3AD336BdfD674660Fe3be5fF (100 ETH)
(9) 0x01dA1040363679F8b272570AF72015aCf37F0D46 (100 ETH)

Private Keys
==================
(0) 0xa8a172ec9cc7168a802c4a4904357b1eca7f6cb1f9267e4788b7cd67bfd35517
(1) 0xf32b811aeb41022b249991c13b783054ff44d7bbb1c51d0a83e9885910e5091c
(2) 0x84d6df9031e604d25ea737d45f10822dfefe8774a26207d78c55612e6ee062e8
(3) 0x9fd265fdb77b87f87c16dfb4b20ce8aa711098320699cc60c053bab0bf605a34
(4) 0x0d856b2291862a41855a39bce244238c8c6181c65e96c473a5afc088bb3e1d5c
(5) 0xa0586ccff50fefd50fd9855954954bc417267c3944bcda8824ae9e928e6d71e4
(6) 0xc0171a7a3f8a5fee80b41c026433866db3cfaa09e36fa0644eb73493599b0885
(7) 0x6b9200e9e96a603845eb3bd227d03b3ca4e6b6fc49c50590284867e278636402
(8) 0xd5ac7dba919ee63cef9a670b75a9ebc2a296b3b0fac5085994677a9043d2e195
(9) 0xba36b985bf6680fabc0177447a8a10fba5ba70fc00ea586c522cee65d82373cb

HD Wallet
==================
Mnemonic:      twist broccoli raccoon imitate stomach fruit neck tree faint clean prize october
Base HD Path:  m/44'/60'/0'/0/{account_index}
```

Bước 3: Cài MetaMask (là một ví Ethereum (Wallet Ethereum))

https://chrome.google.com/webstore/detail/metamask/nkbihfbeogaeaoehlefnkodbefgpgknn

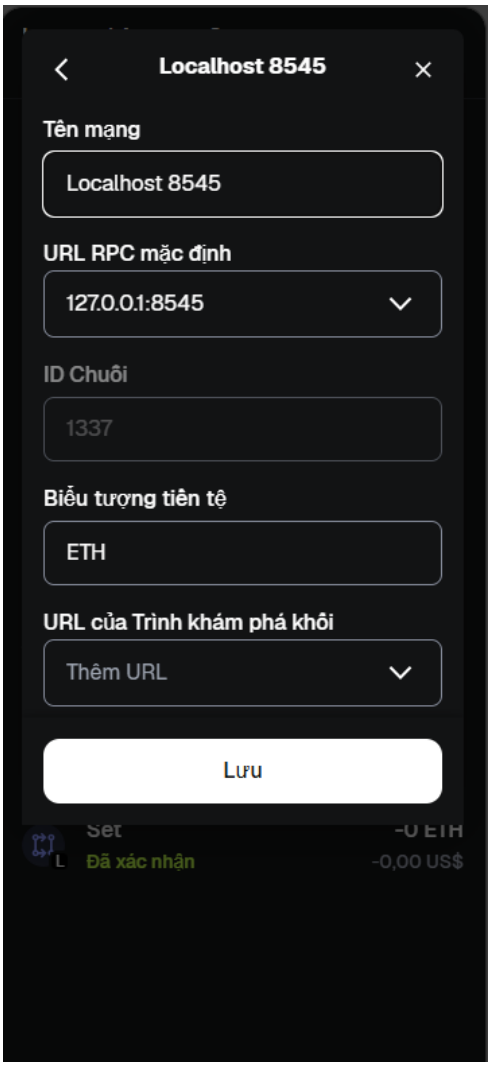
Bước 4: Import 1 private key từ một trong 10 account của Ethereum vào MetaMask

Bước 5: Khai báo mạng để tương tác ganache-cli

URL RPC: http://127.0.0.1:8545

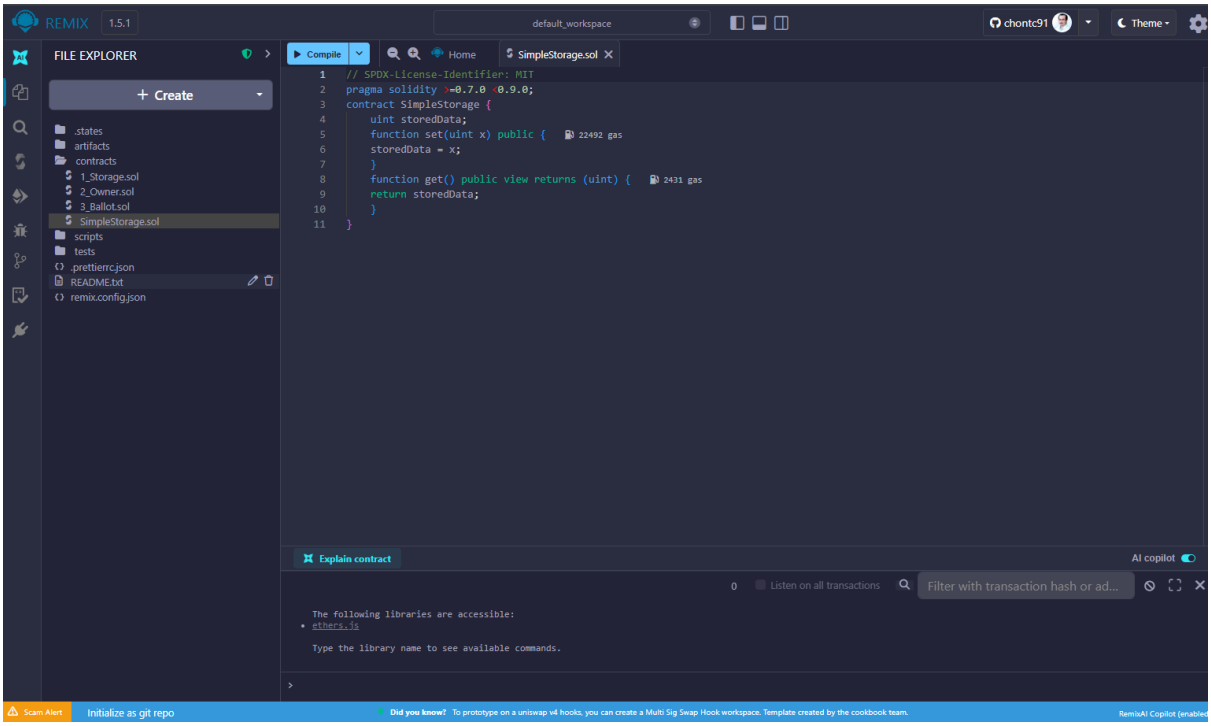
Chain ID: 1337

Currency: ETH



2. **Triển khai một smart contract trên mạng ETH**

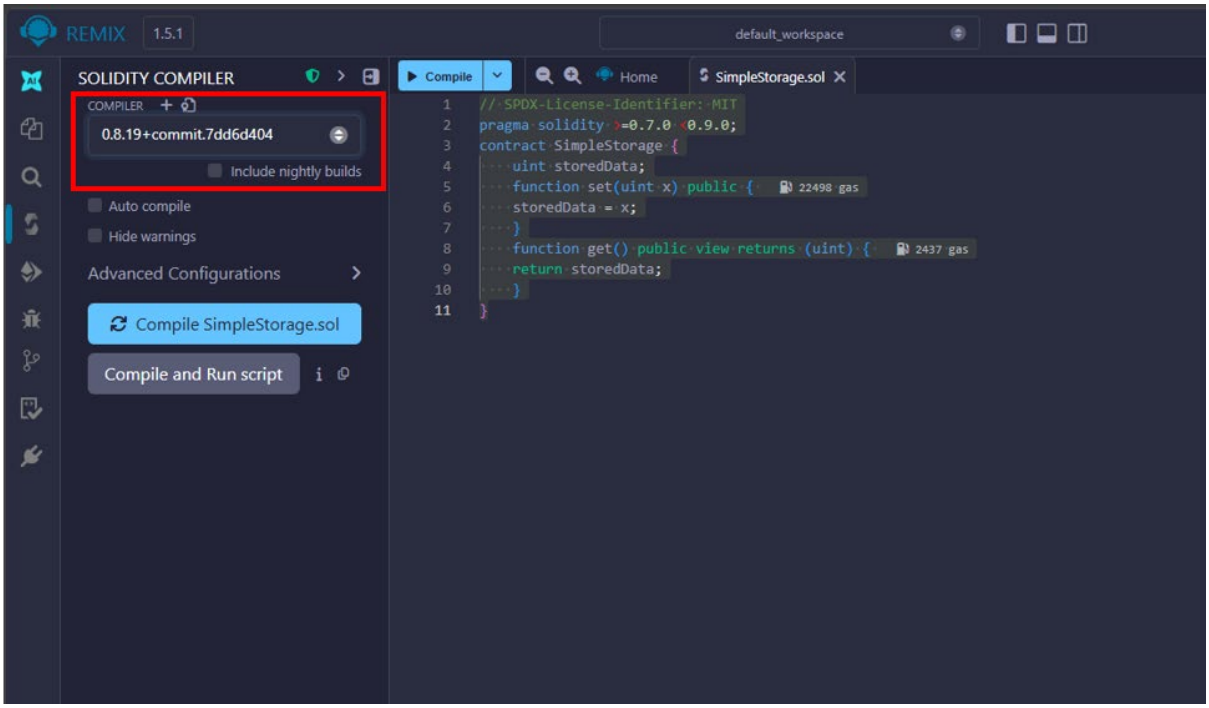
   Viết smart contract trên remix: https://remix.ethereum.org



- **Tạo trong thư mục contracts, tạo mới file tên SimpleStorage.sol**

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.7.0 <0.9.0;
contract SimpleStorage {
    uint storedData;
    function set(uint x) public {
    storedData = x;
    }
    function get() public view returns (uint) {
    return storedData;
    }
}
```

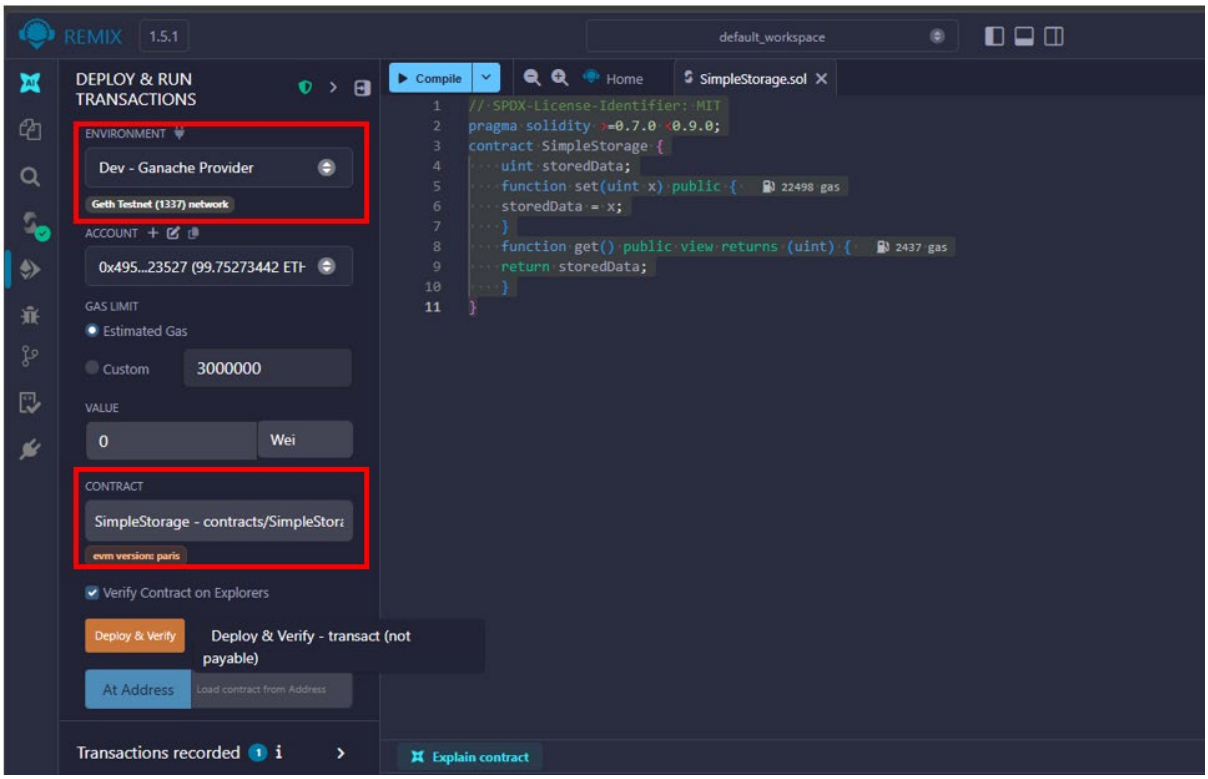
- **Biên dịch:**



Chọn Compiler: 0.8.19….

- **Liên kết ứng dụng với private Ethereum**

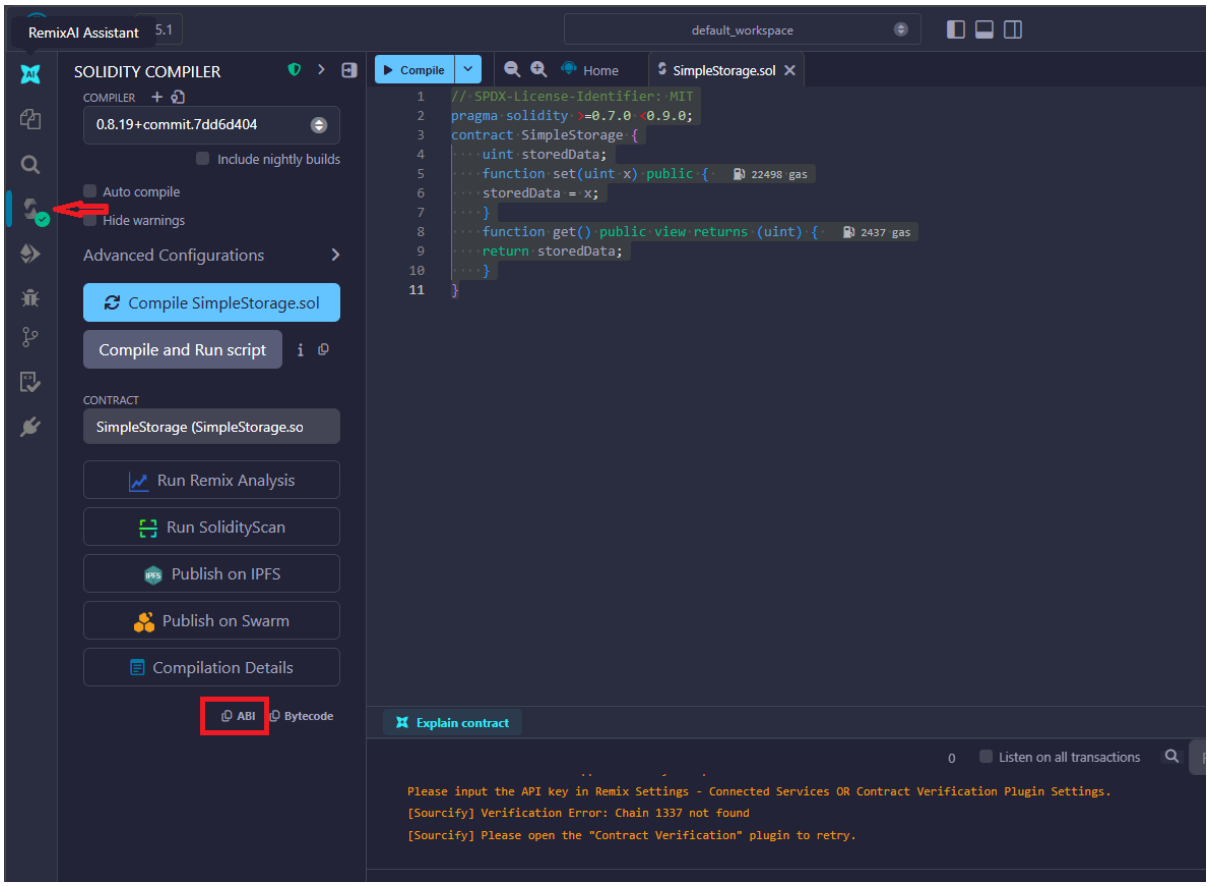
Sử dụng thư viện Web3 trên Web Browser và Deploy smart contract trên Ethereum.



- **Quan sát trên ganache-cli**

```
Transaction: 0x96fac1b1c7ffee344e4c41f71b939e8527028825e92f9b7e20f25be753c38446
Contract created: 0x1f2468ad5075de38b1ffaab46f4ffb7c5113c926
Gas usage: 125641
Block Number: 8
Block Time: Wed Jan 07 2026 16:30:44 GMT+0700 (Indochina Time)
```

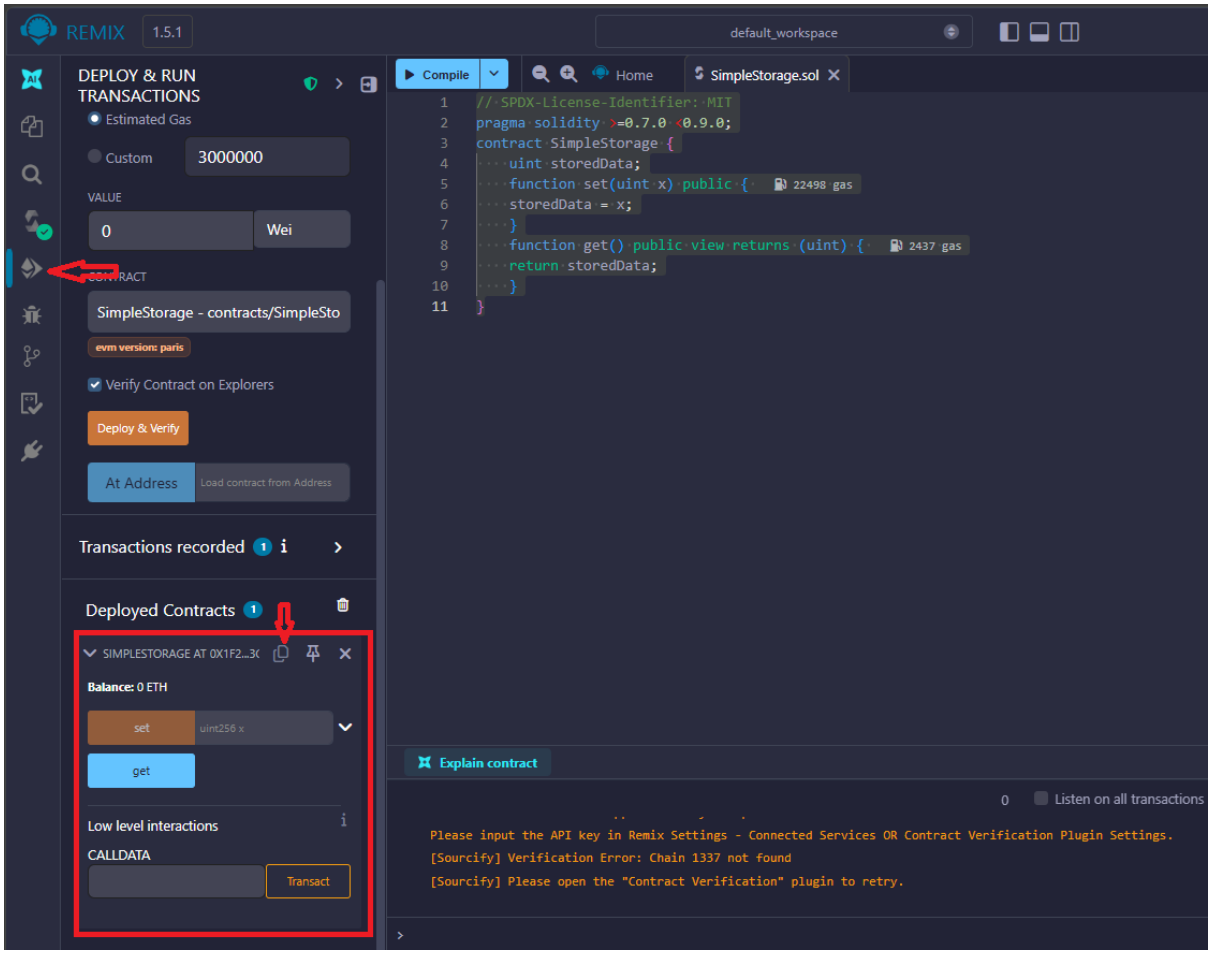
- **Xem ABI của smart contract**




```
[
{
	"inputs": [
		{
			"internalType": "uint256",
			"name": "x",
			"type": "uint256"
		}
	],
	"name": "set",
	"outputs": [],
	"stateMutability": "nonpayable",
	"type": "function"
},
{
	"inputs": [],
```

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


- **Địa chỉ của smart contract trên ETH**



0x1F2468aD5075dE38B1fFAAb46f4Ffb7c5113C926

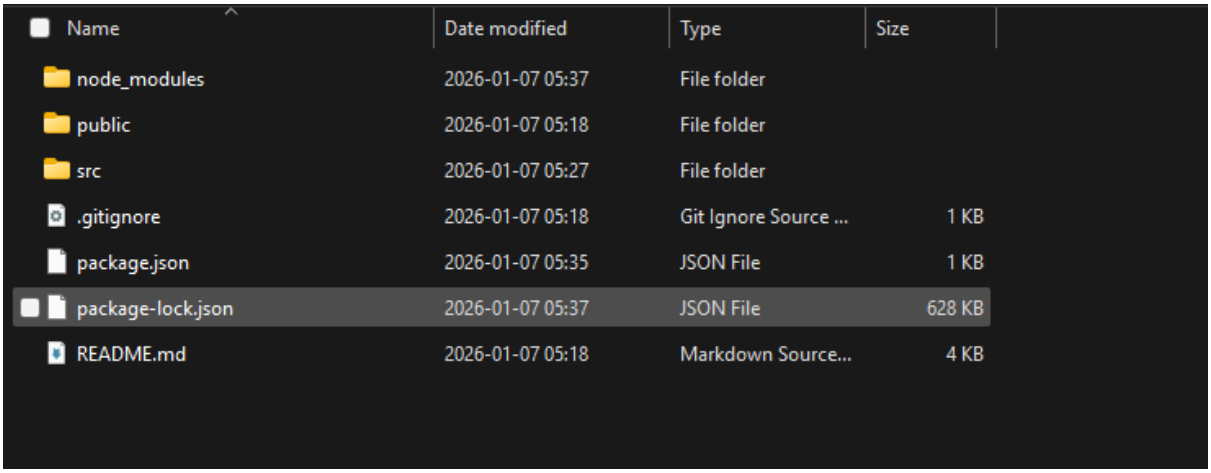
3. Viết Dapp

- **Tạo Frontend**

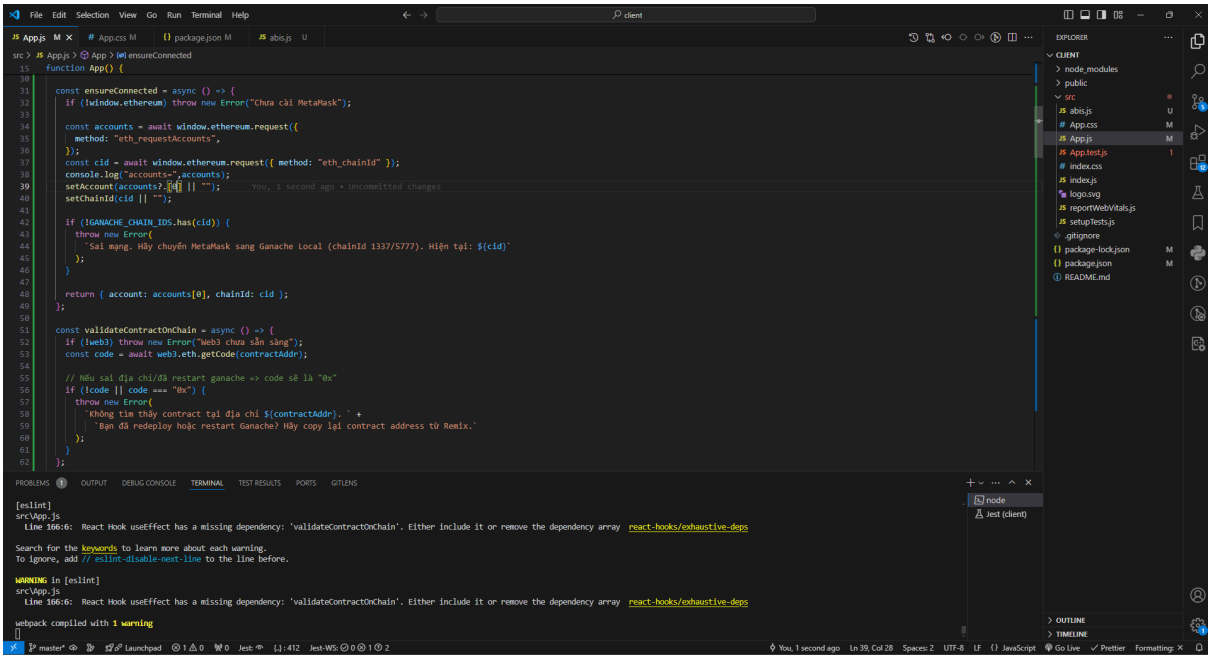
Sử dụng React để tạo web

Điều kiện: máy đã cài đặt node, nmp

- run → cmd
- npm create-react-app <project name>



- cd <project name>
- npm install web3
- Sử dụng Visual studio code để mở project



- **Lập trình Dapp**

  Chỉnh sửa file App.js

```
import "./App.css";
import React, { useEffect, useMemo, useState } from "react";
import Web3 from "web3";
import { simpleStorageAbi } from "./abis";

// QUAN TRỌNG: cập nhật đúng địa chỉ contract vừa deploy trên Ganache hiện tại
const contractAddr = "0x97FE3FBccFe27A36f127507B68Fcb273984a8C50";
```

```
const GANACHE_CHAIN_IDS = new Set(["0x539", "0x1691"]); // 1337, 5777

function shortAddr(a = "") {
  return a ? `${a.slice(0, 6)}...${a.slice(-4)}` : "";
}

function App() {
  const [number, setNumber] = useState("13");
  const [getNumber, setGetNumber] = useState("0");
  const [status, setStatus] = useState("—");

  const [account, setAccount] = useState("");
  const [chainId, setChainId] = useState("");
  const [isReady, setIsReady] = useState(false);

  const { web3, contract } = useMemo(() => {
    if (!window.ethereum) return { web3: null, contract: null };
    const w3 = new Web3(window.ethereum);
    const c = new w3.eth.Contract(simpleStorageAbi, contractAddr);
    return { web3: w3, contract: c };
  }, []);

  const ensureConnected = async () => {
    if (!window.ethereum) throw new Error("Chưa cài MetaMask");

    const accounts = await window.ethereum.request({
      method: "eth_requestAccounts",
    });
    const cid = await window.ethereum.request({ method: "eth_chainId" });
    console.log("accounts=",accounts);
    setAccount(accounts?.[0] || "");
    setChainId(cid || "");

    if (!GANACHE_CHAIN_IDS.has(cid)) {
      throw new Error(
        `Sai mạng. Hãy chuyển MetaMask sang Ganache Local (chainId 1337/5777).
Hiện tại: ${cid}`
      );
    }

    return { account: accounts[0], chainId: cid };
  };

  const validateContractOnChain = async () => {
    if (!web3) throw new Error("Web3 chưa sẵn sàng");
    const code = await web3.eth.getCode(contractAddr);

    // Nếu sai địa chỉ/đã restart ganache => code sẽ là "0x"
```

```
    if (!code || code === "0x") {
      throw new Error(
        `Không tìm thấy contract tại địa chỉ ${contractAddr}. ` +
          `Bạn đã redeploy hoặc restart Ganache? Hãy copy lại contract address
từ Remix.`
      );
    }
  };

  const handleGet = async () => {
    try {
      if (!contract) throw new Error("Contract chưa sẵn sàng");

      setStatus("Đang đọc get()...");
      await ensureConnected();
      await validateContractOnChain();

      const result = await contract.methods.get().call();
      setGetNumber(String(result));
      setStatus("Đọc thành công");
    } catch (err) {
      console.error("GET ERROR FULL:", err);
      setStatus(`Lỗi get(): ${err?.data?.message || err?.message ||
"Unknown"}`);
    }
  };

  const handleSet = async (e) => {
    e.preventDefault();
    try {
      if (!contract || !web3) throw new Error("Web3/Contract chưa sẵn sàng");
      if (number === "" || number === null || number === undefined) {
        throw new Error("Vui lòng nhập số trước khi Set");
      }

      setStatus("Đang kiểm tra ví/chain/contract...");
      const { account } = await ensureConnected();
      await validateContractOnChain();

      setStatus("Đang ước lượng gas...");
      const gas = await contract.methods.set(number).estimateGas({ from:
account });

      setStatus("Đang lấy gasPrice (legacy)...");
      const gasPrice = await web3.eth.getGasPrice();

      setStatus("Đang gửi transaction (MetaMask sẽ hiện popup)...");
      const receipt = await contract.methods.set(number).send({
```

```
        from: account,
        gas,
        gasPrice,
        type: "0x0", // ép legacy để tránh EIP-1559 trên Ganache
      });

      setStatus(`Ghi thành công. Tx: ${receipt.transactionHash}. Đang đọc
lại...`);

      // Đọc lại sau khi ghi để UI luôn ra đúng (vd: 13)
      const latest = await contract.methods.get().call();
      setGetNumber(String(latest));
      setStatus(`Hoàn tất. Giá trị hiện tại: ${latest}`);
    } catch (err) {
      console.error("SET ERROR FULL:", err);

      // Hiển thị message "thật" hơn thay vì generic JSON-RPC
      const msg =
        err?.data?.message ||
        err?.cause?.message ||
        err?.message ||
        "Unknown error";

      setStatus(`Lỗi set(): ${msg}`);
    }
  };

  // Auto-init: lấy account/chainId, kiểm tra contract, và đọc giá trị hiện
tại
  useEffect(() => {
    (async () => {
      try {
        if (!window.ethereum || !web3 || !contract) return;

        setStatus("Đang khởi tạo kết nối...");
        await ensureConnected();
        await validateContractOnChain();
        setIsReady(true);

        const current = await contract.methods.get().call();
        setGetNumber(String(current));
        setStatus("Sẵn sàng");
      } catch (err) {
        console.error("INIT ERROR FULL:", err);
        setIsReady(false);
        setStatus(err?.message || "Chưa sẵn sàng");
      }
    })();
```

```
    if (!window.ethereum) return;

    const onAccountsChanged = (accs) => {
      setAccount(accs?.[0] || "");
    };
    const onChainChanged = (cid) => {
      setChainId(cid || "");
      // Reload nhẹ để tránh provider cache sai network
      window.location.reload();
    };

    window.ethereum.on("accountsChanged", onAccountsChanged);
    window.ethereum.on("chainChanged", onChainChanged);

    return () => {
      window.ethereum.removeListener("accountsChanged", onAccountsChanged);
      window.ethereum.removeListener("chainChanged", onChainChanged);
    };
  }, [web3, contract]);

  return (
    <div className="App">
      <main className="container">
        <section className="card">
          <h1 className="title">SimpleStorage DApp</h1>
          <p className="subtitle">Ganache + MetaMask + Web3.js</p>
          <p className="subtitle">Họ và tên: Trương Chí Chọn - MSHV:
24C12025</p>

          <div className="statusWrap" style={{ marginTop: 12 }}>
            <div className="statusLabel">Kết nối</div>
            <div className="statusText">
              Account: {account ? shortAddr(account) : "—"} | ChainId:{" "}
              {chainId || "—"} | Contract: {shortAddr(contractAddr)} |
Ready:{" "}
              {isReady ? "Yes" : "No"}
            </div>
          </div>

          <div className="grid">
            <form className="row" onSubmit={handleSet}>
              <label className="label" htmlFor="setNumber">
                Set number
              </label>

              <div className="inputGroup">
                <input
```

```
                id="setNumber"
                className="input"
                type="number"
                value={number}
                onChange={(e) => setNumber(e.target.value)}
                placeholder="Ví dụ: 13"
              />
              <button className="btn btnPrimary" type="submit"
disabled={!isReady}>
                Set
              </button>
            </div>
          </form>

          <div className="row">
            <label className="label">Get number</label>

            <div className="inputGroup">
              <div className="valueBox" aria-label="Current value">
                {getNumber}
              </div>
              <button
                className="btn btnSecondary"
                onClick={handleGet}
                type="button"
                disabled={!isReady}
              >
                Get
              </button>
            </div>
          </div>
        </div>

        <div className="statusWrap">
          <div className="statusLabel">Trạng thái</div>
          <div className="statusText">{status}</div>
        </div>
      </section>
    </main>
  </div>
 );
}

export default App;
```

- **Chỉnh sửa giao diện**

```
-  * { box-sizing: border-box; }
```

```
-   html, body { height: 100%; }
-   body { margin: 0; }
-
-   /* Nền và căn giữa */
-   .App {
-     min-height: 100vh;
-     background: #0f172a; /* slate-900 */
-     color: #e5e7eb;      /* gray-200 */
-     font-family: system-ui, -apple-system, Segoe UI, Roboto, Arial, sans-
    serif;
-   }
-
-   .container {
-     min-height: 100vh;
-     display: grid;
-     place-items: center;
-     padding: 24px;
-   }
-
-   /* Card trung tâm */
-   .card {
-     width: min(720px, 100%);
-     background: rgba(255, 255, 255, 0.06);
-     border: 1px solid rgba(255, 255, 255, 0.12);
-     border-radius: 16px;
-     padding: 24px;
-     backdrop-filter: blur(10px);
-   }
-
-   .title {
-     margin: 0 0 6px 0;
-     font-size: 22px;
-     font-weight: 700;
-     letter-spacing: 0.2px;
-     text-align: center;
-   }
-
-   .subtitle {
-     margin: 0 0 18px 0;
-     font-size: 13px;
-     opacity: 0.85;
-     text-align: center;
-   }
-
-   /* Lưới nội dung */
-   .grid {
-     display: grid;
-     gap: 16px;
```

```
-     margin-top: 8px;
-   }
-
-   .row {
-     display: grid;
-     gap: 8px;
-     padding: 14px;
-     border-radius: 12px;
-     background: rgba(255, 255, 255, 0.04);
-     border: 1px solid rgba(255, 255, 255, 0.08);
-   }
-
-   .label {
-     font-size: 13px;
-     opacity: 0.9;
-   }
-
-   /* Input + button thẳng hàng */
-   .inputGroup {
-     display: grid;
-     grid-template-columns: 1fr auto;
-     gap: 10px;
-     align-items: center;
-   }
-
-   .input {
-     width: 100%;
-     height: 40px;
-     padding: 0 12px;
-     border-radius: 10px;
-     border: 1px solid rgba(255, 255, 255, 0.18);
-     background: rgba(15, 23, 42, 0.6);
-     color: #e5e7eb;
-     outline: none;
-   }
-
-   .input:focus {
-     border-color: rgba(99, 102, 241, 0.8); /* indigo */
-   }
-
-   /* Box hiển thị value cân đối như input */
-   .valueBox {
-     height: 40px;
-     padding: 0 12px;
-     border-radius: 10px;
-     display: flex;
-     align-items: center;
-     justify-content: center;
```

```
-     font-variant-numeric: tabular-nums;
-     border: 1px dashed rgba(255, 255, 255, 0.25);
-     background: rgba(15, 23, 42, 0.35);
-     color: #e5e7eb;
-   }
-
-   /* Buttons */
-   .btn {
-     height: 40px;
-     padding: 0 14px;
-     border-radius: 10px;
-     border: 1px solid rgba(255, 255, 255, 0.14);
-     background: rgba(255, 255, 255, 0.06);
-     color: #e5e7eb;
-     cursor: pointer;
-     font-weight: 600;
-   }
-
-   .btn:hover {
-     background: rgba(255, 255, 255, 0.10);
-   }
-
-   .btnPrimary {
-     background: rgba(99, 102, 241, 0.35);
-     border-color: rgba(99, 102, 241, 0.55);
-   }
-   .btnPrimary:hover {
-     background: rgba(99, 102, 241, 0.45);
-   }
-
-   .btnSecondary {
-     background: rgba(16, 185, 129, 0.25);
-     border-color: rgba(16, 185, 129, 0.45);
-   }
-   .btnSecondary:hover {
-     background: rgba(16, 185, 129, 0.33);
-   }
-
-   /* Status */
-   .statusWrap {
-     margin-top: 16px;
-     padding: 12px 14px;
-     border-radius: 12px;
-     background: rgba(255, 255, 255, 0.04);
-     border: 1px solid rgba(255, 255, 255, 0.08);
-   }
-
-   .statusLabel {
```

```
-     font-size: 12px;
-     opacity: 0.8;
-     margin-bottom: 6px;
-   }
-
-   .statusText {
-     font-size: 13px;
-     line-height: 1.4;
-     word-break: break-word;
-     font-variant-numeric: tabular-nums;
-   }
-
-   /* Responsive: mobile */
-   @media (max-width: 520px) {
-     .card { padding: 18px; }
-     .inputGroup { grid-template-columns: 1fr; }
-     .btn { width: 100%; }
-     .valueBox { justify-content: center; }
-   }
-
```

- **Tạo file abis.js**

  Nội dung lấy từ ABI từ bước trên sau đó tạo 1 function để gọi

```
export const simpleStorageAbi = [
    {
        "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"
    }
]
```

- **Chạy React**

  Tại Visual studio code, nhấn Ctrl + ` để mở terminal

  Gõ lênh: npm run start