**ERC-3643 Digital Tokenization DApp (Frontend Template)**

**Project Overview**

**Name:** erc3643-digital-tokenization-dapp

**Purpose:** Build a professional frontend (DApp) to interact with your ERC-3643 Smart Contracts for tokenized assets like Real Estate, Bonds, or Stocks.

**Technologies Used:**

* React 18 + Vite
* TailwindCSS
* Ethers.js (Blockchain interaction)
* React Router (Navigation)
* dotenv (.env support for secure keys)
* MetaMask wallet integration

**Folder Structure**

erc3643-digital-tokenization-dapp/

├── public/

│ └── index.html

├── src/

│ ├── assets/

│ ├── components/

│ │ ├── Header.jsx

│ │ ├── Footer.jsx

│ ├── pages/

│ │ ├── Home.jsx

│ │ ├── Dashboard.jsx

│ │ ├── MintToken.jsx

│ │ └── TransferToken.jsx

│ ├── services/

│ │ └── blockchain.js (Ethers connection)

│ ├── App.jsx

│ ├── main.jsx

└── .env

└── README.md

**Step-by-Step Setup Guide**

**1. Prerequisites**

* Install **Node.js** (>= 16.0)
* Install **VS Code**
* Install **MetaMask Extension** on your browser
* Sepolia ETH Testnet tokens for testing (You can get from a Faucet)

**2. Environment Variables (.env)**

VITE\_PRIVATE\_KEY=your\_private\_key (for backend transactions if needed)

VITE\_INFURA\_API\_KEY=your\_infura\_or\_alchemy\_project\_key

VITE\_CONTRACT\_ADDRESS=your\_deployed\_smart\_contract\_address

Never commit .env to GitHub!

**3. Commands to Run Locally**

# Step 1: Install dependencies

npm install

# Step 2: Start the development server

npm run dev

Your DApp will run on:

http://localhost:5173

**4. Basic Pages Included**

* **Home:** Project Introduction + Connect Wallet
* **Dashboard:** See your Tokenized Assets + Balance
* **MintToken:** Admin can mint new Tokens to investors
* **TransferToken:** Investors can securely transfer tokens

**Additional Details**

**Blockchain Connection (services/blockchain.js)**

import { ethers } from 'ethers';

const provider = new ethers.providers.Web3Provider(window.ethereum);

const signer = provider.getSigner();

export { provider, signer };

**Wallet Connection (Home Page)**

const connectWallet = async () => {

if (window.ethereum) {

await window.ethereum.request({ method: 'eth\_requestAccounts' });

const accounts = await provider.listAccounts();

console.log('Connected Account:', accounts[0]);

}

};

**Next Steps After Setup**

* Update Smart Contract ABI and address in services/blockchain.js
* Deploy Smart Contract to Sepolia/Mainnet
* Integrate real Mint and Transfer functions
* Improve UI further if needed (optional)
* Finally, push to GitHub

git init

git add .

git commit -m "Initial commit: ERC-3643 DApp"

git branch -M main

git remote add origin https://github.com/yourname/yourrepo.git

git push -u origin main

**Conclusion**

This template gives you a very strong starting point to build a fully functional, professional Digital Asset Tokenization DApp based on ERC-3643.

Once you launch it, you can tokenize:

* Real Estate
* Private Equity
* Bonds
* Stocks
* Anything else allowed by regulations!

**Important**

* Use testnet for first testing.
* Ensure proper KYC/AML compliance for production.
* Always secure your PRIVATE KEYS.