Example DApp

Smart contract

```
//SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.8.7;

contract Simple {

   string public myCity = "Cizre";

}
```

Front end

```
<!DOCTYPE html>
<html>
<head>
    <title>CONNECTION TO METAMASK</title>
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/web3/1.2.7-
rc.0/web3.min.js"></script>
    <style>
        body {
            background-color: palevioletred;
            font-size: 33px;
            text-align: center;
        }
        button {
            background-color: aquamarine;
            font-size: 33px;
            font-family: Impact, Haettenschweiler, 'Arial Narrow Bold', sans-
serif;
        }
        button:hover {
            background-color: yellow;
        }
    </style>
</head>
<body>
    <button onclick="connectMetamask()">CONNECT TO METAMASK</button> <br>
    <p id="accountArea"></p>
    <button onclick="connectContract()">CONNECT TO CONTRACT</button> <br>
    <p id="contractArea"></p>
    <button onclick="readContract()">GET DATA FROM CONTRACT</button> <br>
    <p id="dataArea"></p>


    <script>
```

```
        //1- connect metamask
        let account;
        const connectMetamask = async () => {
            if(window.ethereum !== "undefined") { //standard code to connect
to connect metamask
                const accounts = await ethereum.request({method:
"eth_requestAccounts"});
                account = accounts[0];//get all the accounts connected with
the metamask
                document.getElementById("accountArea").innerHTML = account;
            }
        }

        //2- connect to smart contract
        const connectContract = async () => {
            const ABI = [
                   {
                    "inputs": [],
                    "name": "myCity",
                    "outputs": [
                        {
                            "internalType": "string",
                            "name": "",
                            "type": "string"
                        }
                    ],
                    "stateMutability": "view",
                    "type": "function"
                }

            ];
            const Address = "0x5515BE0eF82bA81EFB7910fb4E3eAC469f2C0309";
            window.web3 = await new Web3(window.ethereum);// interacting with
smart contract
            window.contract = await new window.web3.eth.Contract( ABI,
Address);  // creating instances of the contract
            document.getElementById("contractArea").innerHTML = "connected to
smart contract";
        }

        //3-read data from smart contract
        const readContract = async () => {
            const data = await window.contract.methods.myCity().call();
            document.getElementById("dataArea").innerHTML = data;
        }
    </script>
</body>
</html>
```

Connection establishment file

```
const express = require("express"); // Express is a node js web application
framework that provides broad features for building web and mobile
applications.
const path = require("path"); //  used for handling and transforming file
paths.
const app = express();

app.get("/", (req, res) => {
    res.sendFile(path.join(__dirname + "/index.html")); // res.sendFile from
with server.js
})

const server = app.listen(5000); // backend server running on port 5000
const portNumber = server.address().port;
console.log(`port is open on ${portNumber}`);
```