WEEK-5

Using Web3.js to Interact with Smart Contracts.

Code:

```javascript
const Web3 = require('web3');
const web3 = new Web3('http://localhost:7545'); // Connect to your local Ganache instance

const SimpleStorage = artifacts.require('SimpleStorage'); // Replace with your contract's
name

const interactWithContract = async () => {
    const accounts = await web3.eth.getAccounts();
    const ownerAccount = accounts[0];

    const simpleStorageInstance = await SimpleStorage.deployed();

    // Get the current data
    const currentData = await simpleStorageInstance.data.call();
    console.log('Current Data:', currentData.toNumber());

    // Set new data
    await simpleStorageInstance.setData(42, { from: ownerAccount });

    // Get the updated data
    const updatedData = await simpleStorageInstance.data.call();
    console.log('Updated Data:', updatedData.toNumber());
};

interactWithContract();
```

Output:

```
{
  "innerError": {
    "code": -32000,
    "message": "err: insufficient funds for gas * price + value: address
0x7c5EB88A3698c534B03Eb60db9846DEa0bb9E4F7 have 0 want 100000000000000000 (supplied gas
15010499)"
  },
  "code": 101,
  "data": undefined,
  "request": {
    "jsonrpc": "2.0",
        "id": "7d99d80d-28fb-4609-8854-057ca330e5d1",
        "method": "eth_estimateGas",
        "params": [
            {
                "from": "0x7c5EB88A3698c534B03Eb60db9846DEa0bb9E4F7",
                "to": "0x7c5EB88A3698c534B03Eb60db9846DEa0bb9E4F7",
                "value": "0x16345785d8a0000"
            },
            "latest"
        ]
    }
}
```