

```
1 #-----
2 # Name:          This script is for interacting front end with Ethereum chain
3 # Author:        Rejith Reghunathan
4 # Created:       18/04/2022
5 # Licence:       MIT
6 #-----
7
8
9 from flask import Flask, render_template, request, flash, redirect
10 import config
11 from web3 import Web3
12 from web3 import Web3, HTTPProvider
13 import json
14 from web3.middleware import geth_poa_middleware
15 import os
16 from dotenv import load_dotenv
17
18 load_dotenv()
19
20 PRIVATE_KEY = os.getenv('PRIVATE_KEY')
21 OWNER_ADDRESS = os.getenv('OWNER_ADDRESS')
22
23 # reading contract ABI data
24 with open("brownie/build/contracts/IOT_Web3.json") as f:
25     info_json = json.load(f)
26 abi = info_json["abi"]
27
28
29 # HTTP Provider
30 rinkeby_url =
31 "https://rinkeby.infura.io/v3/6044b6684296420ca3436a6055ed50c3"
32 w3 = Web3(Web3.HTTPProvider(rinkeby_url))
33 w3.middleware_onion.inject(geth_poa_middleware, layer=0)
34
35 # Smart contract address - convert to checksum address
36 address =
37 w3.toChecksumAddress('0xCE2E449b25268F59bc0B2D3661df1b33ABe6c0e6')
38
39 # Initialize Smart contract
40 contract = w3.eth.contract(address=address, abi=abi)
41
42 # Set a default account to sign transactions - this account is unlocked
43 # with Ganache
44 #w3.eth.defaultAccount = w3.eth.accounts[0]
45
46
47
48 ##### FLASK #####
49 app = Flask(__name__)
50
51 @app.route("/", methods = ['POST', 'GET'])
52 def index():
53
```

```

54 value_from_ether_bc=contract.functions.getiot_datas().call(block_identifie
r='latest')
55
56     ## for displaying latest 4 datas
57     latest_data = []
58     temp_data =[]
59     vib_data =[]
60     index = 0
61
62     if(len(value_from_ether_bc)>5):
63         for data in range(len(value_from_ether_bc)-1,
len(value_from_ether_bc)-5, -1):
64             sensor_data_list = value_from_ether_bc[data]
65             latest_data.append(sensor_data_list)
66             if(latest_data[index][0] == "VIBR_SENSOR_001"):
67                 vib_data.append(sensor_data_list)
68             if(latest_data[index][0] == "TEMP_SENSOR_001"):
69                 temp_data.append(sensor_data_list)
70             index+=1
71
72     latest_vib_data = vib_data[0][2]
73     latest_temp_data = temp_data[0][2]
74
75     labels_vib = [row[1] for row in vib_data]
76     data_vib = [row[2] for row in vib_data]
77     labels_temp = [row[1] for row in temp_data]
78     data_temp = [row[2] for row in temp_data]
79
80     print(data_vib)
81     print(data_temp)
82
83     form_data =[]
84     service_data =[]
85
86     data_from_contract=contract.functions.getstatus().call(block_identifier='l
atest')
87
88     if(len(data_from_contract)>2):
89         for data in range(len(data_from_contract)-1,
len(data_from_contract)-2, -1):
90             data_temp = data_from_contract[data]
91             service_data.append(data_temp)
92             print(service_data)
93
94     if request.method == 'POST':
95         form_data = request.form
96         print(str(form_data["service_date"]))
97         nonce = w3.eth.getTransactionCount(OWNER_ADDRESS)
98         txn =
contract.functions.setstatus(str(form_data["service_date"]),str(form_data["
next_service"]),str(form_data["healthy"]),str(form_data["company"]),str(for
m_data["engineer"])).buildTransaction(
99             {"gasPrice": w3.eth.gas_price.toNumber() * 1.40 , "from":
OWNER_ADDRESS, "nonce": nonce, "gasPrice": 35000000000 })
100
101         signed_txn = w3.eth.account.sign_transaction(txn,
private_key=PRIVATE_KEY)
102         txn_hash = w3.eth.sendRawTransaction(signed_txn.rawTransaction)

```

```
103
104     print(txn_hash)
105
106
107
108
109     return render_template('index-
eth.html', latest_data=latest_data, latest_vib_data=latest_vib_data, latest_te
mp_data=latest_temp_data, labels_vib=labels_vib,
labels_temp=labels_temp, data_temp=data_temp, data_vib=data_vib,
service_data= service_data)
110
111
112
113 #####
114 if __name__ == '__main__':
115     app.run(port=8002, host='0.0.0.0', debug=True)
116
117 #####
118
```