

Assingment-3: Flask, sqlalchemy, sqlite

Name: Adil Ergazin CS-2115

Assignment-3

**Connect to the site's documentation by API and Flask SQLAlchemy
and connect to the database by sqlite**

1. **Step:** Connect to the site's information documentation by API(Rescountries, weather) and create the database connection.

```
1 import requests
2 from flask import Flask, render_template, request
3 from flask_sqlalchemy import SQLAlchemy
4
5 db = SQLAlchemy()
6 # create the app
7 app = Flask(__name__)
8 app.app_context().push()
9 # configure the SQLite database, relative to the app instance folder
10 app.config["SQLALCHEMY_DATABASE_URI"] = "sqlite:///assignment3.db"
11 # initialize the app with the extension
12 db.init_app(app)
13
14 class user(db.Model):
15     id = db.Column(db.Integer, primary_key=True)
16     countryName = db.Column(db.String, nullable=False)
17     offName = db.Column(db.String, nullable=False)
18     nativeName = db.Column(db.String, nullable=False)
19     currenciesName = db.Column(db.String, nullable=False)
20     curSymbol = db.Column(db.String, nullable=False)
21     capital = db.Column(db.String, nullable=False)
22     region = db.Column(db.String, nullable=False)
23     subregion = db.Column(db.String, nullable=False)
24     languages = db.Column(db.String, nullable=False)
25     population = db.Column(db.String, nullable=False)
26     area = db.Column(db.String, nullable=False)
27     flags = db.Column(db.String, nullable=False)
28
```

```
12 db.init_app(app)
13
14 class user(db.Model):
15     id = db.Column(db.Integer, primary_key=True)
16     countryName = db.Column(db.String, nullable=False)
17     offName = db.Column(db.String, nullable=False)
18     nativeName = db.Column(db.String, nullable=False)
19     currenciesName = db.Column(db.String, nullable=False)
20     curSymbol = db.Column(db.String, nullable=False)
21     capital = db.Column(db.String, nullable=False)
22     region = db.Column(db.String, nullable=False)
23     subregion = db.Column(db.String, nullable=False)
24     languages = db.Column(db.String, nullable=False)
25     population = db.Column(db.String, nullable=False)
26     area = db.Column(db.String, nullable=False)
27     flags = db.Column(db.String, nullable=False)
28
29
30 with app.app_context():
31     db.create_all()
32
33 @app.route('/')
34 def index():
35     return render_template('index.html')
```

```

@app.route('/result', methods=['GET', 'POST'])
def result():
    try:
        print(request.values['country'])
        country = request.values['country']
        url = f'https://restcountries.com/v3.1/name/{country}'
        r = requests.get(url)
        data = r.json()[0]
        officialhr = data['name']['official']
        official = data['name']['nativeName']
        offcc = ''
        offc = []
        for c, d in official.items():
            for j, b in d.items():
                offc.append(b)
        for i in range(len(offc)):
            offc[i] = offc[i].replace(' ', '')

        flags = f'{data["flags"]}',
        flags_ = f'{data["flags"]}'
        key = '6ae37b61af66ad1033a5747eb164a39b'
        weurl = f'https://api.openweathermap.org/data/2.5/weather?q={capital}&appid={key}&units=metric'
        getw = requests.get(weurl)
        dataw = getw.json()
        apiweather = dataw['main']['temp']
        iconn = dataw['weather'][0]['icon']
        if (any(c['data']['name'] == 'common' in dataw for dataw in quer())):

```

2. **Step2:** By “*route*” mapping of the URL to the Python function and by “*GET, POST*” methods workin with codes which are going by flask to the sql as a query of sql commands.

```

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/result', methods=['GET', 'POST'])
def result():
    try:
        print(request.values['country'])
        country = request.values['country']
        url = f'https://restcountries.com/v3.1/name/{country}'
        r = requests.get(url)
        data = r.json()[0]
        officialhr = data['name']['official']
        official = data['name']['nativeName']
        offcc = ''
        offc = []
        for c, d in official.items():
            for j, b in d.items():
                offc.append(b)
        for i in range(len(offc)):
            offc[i] = offc[i].replace(' ', '')
            break
        flagg = data['flags']['png']
        currencies = data['currencies'][0][data['currencies'][0]['name']]
        symb = data['currencies'][0][data['currencies'][0]['symbol']]
        capital = data['capital'][0]
        region = data['region']
        subreg = data['subregion']

```

```

49     offc = []
50     for c, d in official.items():
51         for j, b in d.items():
52             offc.append(b)
53         for i in range(len(offc)):
54             offcc = offcc + offc[i]
55         break
56     flagg = data['flags']['png']
57     currencies = data['currencies'][list(data['currencies'].keys())[0]]['name']
58     symb = data['currencies'][list(data['currencies'].keys())[0]]['symbol']
59     capital = data['capital'][0]
60     region = data['region']
61     subreg = data['subregion']
62     langd = data['languages']
63     langls = []
64     langst = ''
65     for k, v in langd.items():
66         langls.append(v)
67     for i in range(len(langls)):
68         if (len(langls) == 1):
69             langst = langst + langls[i]
70         elif (i == 0):
71             langst = langst + langls[i] + ', ' + langls[i + 1]
72         elif (i > 1):
73             langst = langst + ', ' + langls[i]
74     population = data['population']
75     s = data['area']
76     def quer():

```

```

76     def quer():
77         f = []
78         for i in user.query.all():
79             f.append(i.countryName)
80         return f
81     if r.status_code == 200:
82         user1 = user(countryName=f"{data['name']}['common']",
83                     offName=f"{data['name']}['officialName']",
84                     nativeName=f"{data['name']}['nativeName']",
85                     currenciesName=f"{data['currencies']}['name']",
86                     curSymbol=f"{data['currencies']}['symbol']",
87                     capital=f"{data['capital']}[0]",
88                     region=f"{data['region']}",
89                     subregion=f"{data['subregion']}",
90                     languages=f"{data['languages']}",
91                     population=f"{data['population']}",
92                     area=f"{data['area']}",
93                     flags=f"{data['flags']}")
94         key = '6ae37b61af06ad1833a5747eb164a39b'
95         weurl = f"https://api.openweathermap.org/data/2.5/weather?q={capital}&appid={key}&units=metric"
96         getw = requests.get(weurl)
97         dataw = getw.json()
98         apiweather = dataw['main']['temp']
99         iconn = dataw['weather'][0]['icon']
100         if (any(f"{data['name']}['common']" in datas for datas in quer())):
101             return render_template('country.html',
102                                   weather=apiweather,
103                                   weathericon=iconn,
104                                   offici=user1.offName,
105                                   png=user1.flags,
106                                   ofic_name=user1.nativeName,
107                                   curr=user1.currenciesName,
108                                   symb=user1.curSymbol,
109                                   capCity=user1.capital,
110                                   reg=user1.region,
111                                   subr=user1.subregion,
112                                   languages=user1.languages,
113                                   population=user1.population,
114                                   area=user1.area)

```

```

100         iconn = dataw['weather'][0]['icon']
101         if (any(f"{data['name']}['common']" in datas for datas in quer())):
102             return render_template('country.html',
103                                   weather=apiweather,
104                                   weathericon=iconn,
105                                   offici=user1.offName,
106                                   png=user1.flags,
107                                   ofic_name=user1.nativeName,
108                                   curr=user1.currenciesName,
109                                   symb=user1.curSymbol,
110                                   capCity=user1.capital,
111                                   reg=user1.region,
112                                   subr=user1.subregion,
113                                   languages=user1.languages,
114                                   population=user1.population,
115                                   area=user1.area)
116         else:
117             db.session.add(user1)
118             db.session.commit()
119         except KeyError:
120             return render_template('error.html')
121         return render_template('country.html',
122                               weather=apiweather,
123                               weathericon=iconn,
124                               offici=user1.offName,
125                               png=user1.flags,
126                               ofic_name=user1.nativeName,
127                               curr=user1.currenciesName,
128                               symb=user1.curSymbol,
129                               capCity=user1.capital,
130                               reg=user1.region,
131                               subr=user1.subregion,
132                               languages=user1.languages,
133                               population=user1.population,
134                               area=user1.area)

```

```

114         population=user1.population,
115         area=user1.area)
116     else:
117         db.session.add(user1)
118         db.session.commit()
119     except KeyError:
120         return render_template('error.html')
121     return render_template('country.html',
122                             weathe=apiweather,
123                             weathericon=iconn,
124                             offi=user1.offName,
125                             png=user1.flags,
126                             ofic_name=user1.nativeName,
127                             curr=user1.currenciesName,
128                             symbol=user1.curSymbol,
129                             capCity=user1.capital,
130                             reg=user1.region,
131                             subr=user1.subregion,
132                             languages=user1.languages,
133                             population=user1.population,
134                             area=user1.area)
135
136 if __name__ == '__main__':
137     app.run(debug=True)

```

3. **Step3:** Getting information by result function from “*route result*” and by “*index.html*” and “*country.html*” we can see our results joining to the host “*http://127.0.0.1:5000*” from site’s documentation(Restcountries, weather). In order to get info fast we are using our database and it’s(from API docs) automatically inserting and by data from database it will search faster.

```

37
38 @app.route('/result', methods=['GET', 'POST'])
39 def result():
40     try:
41         print(request.values['country'])
42         country = request.values['country']
43         url = f'https://restcountries.com/v3.1/name/{country}'
44         r = requests.get(url)

```

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6 </head>
7 <body>
8     <h1>Hello user. Please enter your country:</h1>
9     <form action="/result" method="post">
10         <input type="text" name="country"><br><br>
11         <button type="submit">Search</button>
12     </form>
13 </body>
14 </html>

```

```
Adil_Ergazin_CS2115.py x country.html x index.html x
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Title</title>
6 </head>
7 <body>
8 <h1>{{ offici }}</h1>
9 <p>{{ weathe }} &degC, {{ capCity }}</p>
10 
11 <hr>
12 
13 <hr>
14 <p>Native Naming: {{ ofic_name }}</p>
15 <p>Currencies: {{ curr }} '{{ symbol }}'</p>
16 <p>Capital city: {{ capCity }}</p>
17 <p>Region: {{ reg }} ({{ subr }})</p>
18 <p>Languages: {{ languages }}</p>
19 <p>Population: {{ population }}</p>
20 <p>Area: {{ area }}</p>
21 <hr size="4" color="gray" style="opacity:0.2">
22 <form>
23 <input type="button" value="Go back" onclick="history.go(-1)">
24 </form>
25 </body>
26 </html>
```

Hello user. Please enter your country:

Republic of Kazakhstan

-0.93 °C, Nur-Sultan



Native Naming: Қазақстан Республикасы

Currencies: Kazakhstani tenge '₸'

Capital city: Nur-Sultan

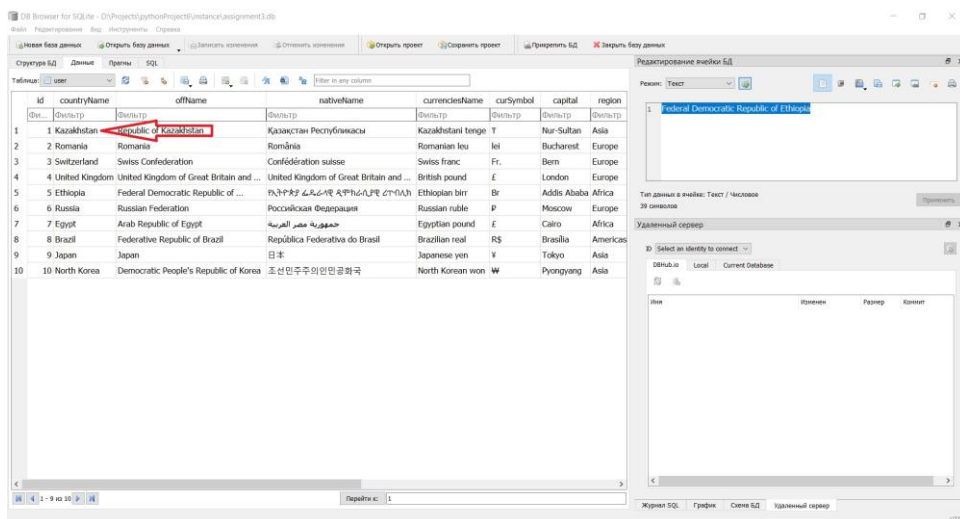
Region: Asia (Central Asia)

Languages: Kazakh, Russian

Population: 18754440

Area: 2724900.0

[Go back](#)



The screenshot shows a web browser window displaying a table of countries. The table has columns: id, countryname, offname, nativeName, currenciesName, curSymbol, capital, and region. The first row is highlighted, showing Kazakhstan. A red box highlights the 'countryname' column header. To the right of the table, there is a sidebar with a search bar and a list of countries. The search bar contains the text 'Federal Democratic Republic of Ethiopia'. Below the search bar, there is a section titled 'Удаленный сервер' (Remote server) with a dropdown menu for 'ID' and a button 'Подключить' (Connect).

id	countryname	offname	nativeName	currenciesName	curSymbol	capital	region
1	Kazakhstan	Republic of Kazakhstan	Қазақстан Республикасы	Kazakhstani tenge	₸	Nur-Sultan	Asia
2	Romania	Romania	România	Romanian leu	lei	Bucharest	Europe
3	Switzerland	Swiss Confederation	Confédération suisse	Swiss franc	Fr.	Bern	Europe
4	United Kingdom	United Kingdom of Great Britain and ...	United Kingdom of Great Britain and ...	British pound	£	London	Europe
5	Ethiopia	Federal Democratic Republic of ...	የኢትዮጵያ ፌዴራል ዲሞክራሲያዊ ሪፐብሊክ	Ethiopian birr	Br	Addis Ababa	Africa
6	Russia	Russian Federation	Российская Федерация	Russian ruble	₽	Moscow	Europe
7	Egypt	Arab Republic of Egypt	جمهورية مصر العربية	Egyptian pound	£	Cairo	Africa
8	Brazil	Federative Republic of Brazil	República Federativa do Brasil	Brazilian real	R\$	Brasilia	Americas
9	Japan	Japan	日本	Japanese yen	¥	Tokyo	Asia
10	North Korea	Democratic People's Republic of Korea	조선민주주의인민공화국	North Korean won	₩	Pyongyang	Asia

```
100     _iconn = data[weather][0]['icon']
101     if(any(["data[name]" + _iconn"]" in datas for datas in quer())):
102         return render_template('country.html',
103                                weather=apiweather,
104                                weathericon = _iconn,
105                                offit=user1.offName,
106                                png=user1.flags,
107                                ofic_name=user1.nativeName,
108                                curr=user1.currenciesName,
109                                symbol=user1.curSymbol,
110                                capCity=user1.capital,
111                                reg=user1.region,
112                                subre=user1.subregion,
113                                languages=user1.languages,
114                                population=user1.population,
115                                area=user1.area)
116     else:
117         db.session.add(user1)
118         db.session.commit()
```