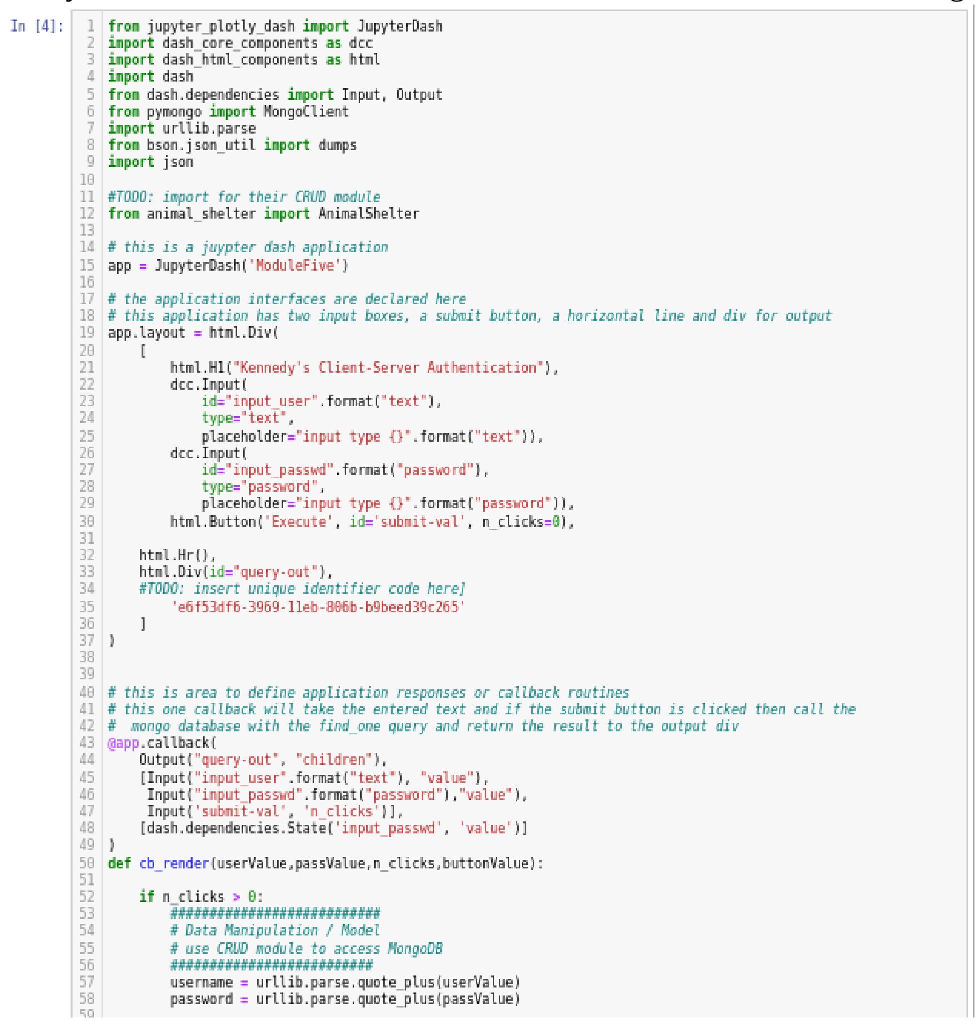
Kennedy Uzoho
5-2 Assignment
CS-340 Client/Server Development

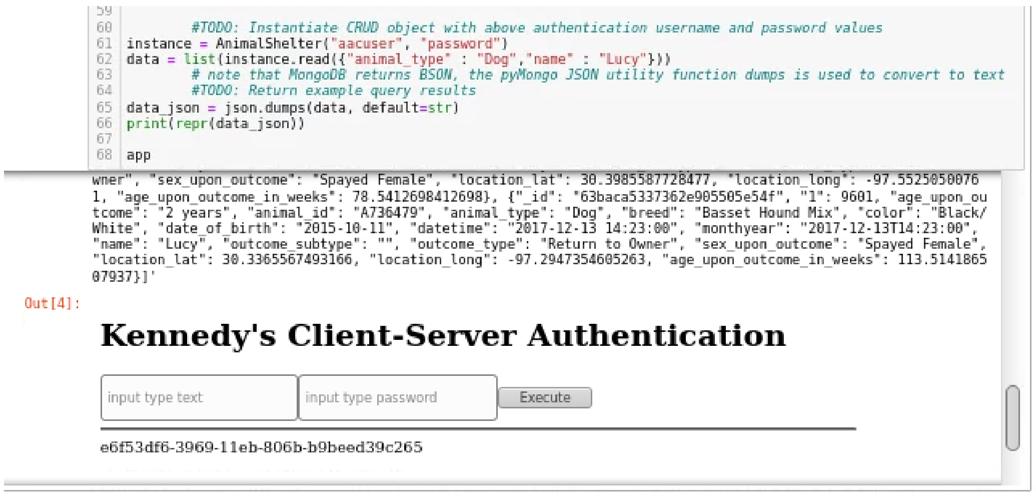
Client-Side Authentication

1. Imported the CRUD Python module that was created for Project One.

2. Add the functionality in the callback routine for the instantiation of your CRUD object. User authentication was applied in the CRUD object.

3. Lastly, added client side function to test the dashboard connection to MongoDB.


In [4]:
```
from jupyter_plotly_dash import JupyterDash
import dash_core_components as dcc
import dash_html_components as html
import dash
from dash.dependencies import Input, Output
from pymongo import MongoClient
import urllib.parse
from bson.json_util import dumps
import json

#TODO: import for their CRUD module
from animal_shelter import AnimalShelter

# this is a juypter dash application
app = JupyterDash('ModuleFive')

# the application interfaces are declared here
# this application has two input boxes, a submit button, a horizontal line and div for output
app.layout = html.Div(
    [
        html.H1("Kennedy's Client-Server Authentication"),
        dcc.Input(
            id="input_user".format("text"),
            type="text",
            placeholder="input type {}".format("text")),
        dcc.Input(
            id="input_passwd".format("password"),
            type="password",
            placeholder="input type {}".format("password")),
        html.Button('Execute', id='submit-val', n_clicks=0),

    html.Hr(),
    html.Div(id="query-out"),
    #TODO: insert unique identifier code here]
        'e6f53df6-3969-11eb-806b-b9beed39c265'
    ]
)


# this is area to define application responses or callback routines
# this one callback will take the entered text and if the submit button is clicked then call the
#  mongo database with the find_one query and return the result to the output div
@app.callback(
    Output("query-out", "children"),
    [Input("input_user".format("text"), "value"),
     Input("input_passwd".format("password"),"value"),
     Input('submit-val', 'n_clicks')],
    [dash.dependencies.State('input_passwd', 'value')]
)
def cb_render(userValue,passValue,n_clicks,buttonValue):

    if n_clicks > 0:
        ###########################
        # Data Manipulation / Model
        # use CRUD module to access MongoDB
        ##########################
        username = urllib.parse.quote_plus(userValue)
        password = urllib.parse.quote_plus(passValue)
```



```python

        #TODO: Instantiate CRUD object with above authentication username and password values
instance = AnimalShelter("aacuser", "password")
data = list(instance.read({"animal_type" : "Dog","name" : "Lucy"}))
        # note that MongoDB returns BSON, the pyMongo JSON utility function dumps is used to convert to text
        #TODO: Return example query results
data_json = json.dumps(data, default=str)
print(repr(data_json))

app
```

```
wner", "sex_upon_outcome": "Spayed Female", "location_lat": 30.3985587728477, "location_long": -97.5525050076
1, "age_upon_outcome_in_weeks": 78.5412698412698}, {"_id": "63baca5337362e905505e54f", "1": 9601, "age_upon_ou
tcome": "2 years", "animal_id": "A736479", "animal_type": "Dog", "breed": "Basset Hound Mix", "color": "Black/
White", "date_of_birth": "2015-10-11", "datetime": "2017-12-13 14:23:00", "monthyear": "2017-12-13T14:23:00",
"name": "Lucy", "outcome_subtype": "", "outcome_type": "Return to Owner", "sex_upon_outcome": "Spayed Female",
"location_lat": 30.3365567493166, "location_long": -97.2947354605263, "age_upon_outcome_in_weeks": 113.5141865
07937}]'
```

Out[4]:

Kennedy's Client-Server Authentication

e6f53df6-3969-11eb-806b-b9beed39c265