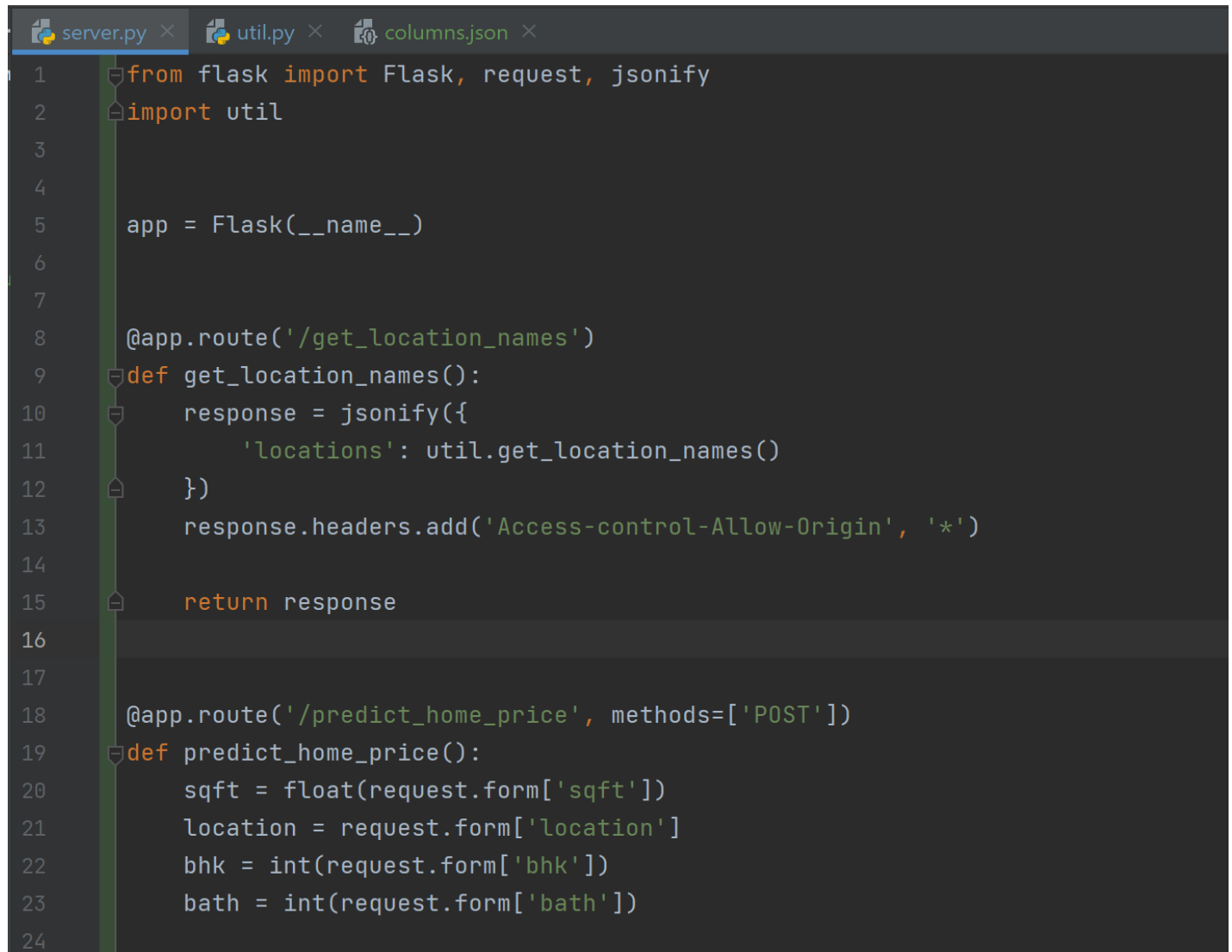


OVUOWO RUKEVWE

LISUM11: 30

JULY 28

GITHUB (OVUOWO- RUKEVWE)



The image shows a code editor with three tabs: 'server.py', 'util.py', and 'columns.json'. The 'server.py' tab is active, displaying Python code for a Flask web application. The code includes imports for Flask, request, jsonify, and a custom util module. It defines a Flask app and two routes: one for GET requests to '/get_location_names' and another for POST requests to '/predict_home_price'. The GET route uses the util module to fetch location names and sets an 'Access-Control-Allow-Origin' header. The POST route processes form data for sqft, location, bhk, and bath.

```
1 from flask import Flask, request, jsonify
2 import util
3
4
5 app = Flask(__name__)
6
7
8 @app.route('/get_location_names')
9 def get_location_names():
10     response = jsonify({
11         'locations': util.get_location_names()
12     })
13     response.headers.add('Access-control-Allow-Origin', '*')
14
15     return response
16
17
18 @app.route('/predict_home_price', methods=['POST'])
19 def predict_home_price():
20     sqft = float(request.form['sqft'])
21     location = request.form['location']
22     bhk = int(request.form['bhk'])
23     bath = int(request.form['bath'])
24
```

```
server.py x util.py x columns.json x
C:\Users\PC\Desktop\learning\House Price Prediction
18 @app.route('/predict_home_price', methods=['POST'])
19 def predict_home_price():
20     sqft = float(request.form['sqft'])
21     location = request.form['location']
22     bhk = int(request.form['bhk'])
23     bath = int(request.form['bath'])
24
25     response = jsonify({
26         'estimated_price': util.get_estimated_price(location, sqft, bhk, bath)
27     })
28
29     response.headers.add('Access-control-Allow-Origin', '*')
30
31     return response
32
33
34 if __name__ == '__main__':
35     print('starting a python project')
36     util.load_saved_artifacts()
37     app.run()
38
```

```
server.py x util.py x columns.json x
C:\Users\PC\Desktop\learning\House Price Prediction
2 import pickle
3 import numpy as np
4
5 __locations = None
6 __data_columns = None
7 __model = None
8
9
10 def get_estimated_price(location, sqft, bhk, bath):
11     try:
12         loc_index = __data_columns.index(location.lower())
13     except:
14         loc_index = -1
15
16     x = np.zeros(len(__data_columns))
17     x[0] = sqft
18     x[1] = bath
19     x[2] = bhk
20     if loc_index >= 1:
21         x[loc_index] = 1
22
23     return round(__model.predict([x])[0])
24
```

```

server.py x util.py x columns.json x
C:\Users\PC\Desktop\learning\House Price Prediction
27     return __locations
28
29
30 def load_saved_artifacts():
31     print('loading saved artifacts...starting')
32     global __data_columns
33     global __locations
34
35     with open('./artifacts/columns.json', 'r') as f:
36         __data_columns = json.load(f)['data_columns']
37         __locations = __data_columns[3:]
38
39     global __model
40     with open('./artifacts/predicted_house_price_model', 'rb') as f:
41         __model = pickle.load(f)
42     print('loading saved artifacts...done')
43
44
45 if __name__ == '__main__':
46     load_saved_artifacts()
47     print(get_location_names())
48     print(round(get_estimated_price('1st Phase JP Nagar', 1000, 3, 3), 2))
49     print(round(get_estimated_price('Vishveshwarya Layout', 2500, 2, 3), 2))

```

```

server.py x util.py x columns.json x
1 {"data_columns": ["total_sqft", "bath", "bhk", "1st block jayanagar", "1st phase jp nagar",
2 "2nd phase judicial layout", "2nd stage nagarbhavi", "5th block hbr layout", "5th phase jp nagar",
3 "6th phase jp nagar", "7th phase jp nagar", "8th phase jp nagar", "9th phase jp nagar", "aecs layout",
4 "abbigere", "akshaya nagar", "ambalipura", "ambedkar nagar", "amruthahalli", "anandapura", "ananth nagar",
5 "anekal", "anjanapura", "ardendale", "arekere", "attibele", "beml layout", "btm 2nd stage", "btm layout",
6 "babusapalaya", "badavala nagar", "balagere", "banashankari", "banashankari stage ii",
7 "banashankari stage iii", "banashankari stage v", "banashankari stage vi", "banaswadi", "banjara layout",
8 "bannerghatta", "bannerghatta road", "basavangudi", "basaveshwara nagar", "battarahalli", "begur", "begur road",
9 "bellandur", "benson town", "bharathi nagar", "bhoganhalli", "billekahalli", "binny pete", "bisuvanahalli",
10 "bommanahalli", "bommasandra", "bommasandra industrial area", "bommenahalli", "brookefield", "budigere",
11 "cv raman nagar", "chamrajpet", "chandapura", "channasandra", "chikka tirupathi", "chikkabanavar",
12 "chikkalasandra", "choodasandra", "cooke town", "cox town", "cunningham road", "dasanapura", "dasarahalli",
13 "devanahalli", "devarachikkanahalli", "dodda nekkundi", "doddaballapur", "doddakallasandra", "doddathoguru",
14 "domlur", "dommasandra", "epip zone", "electronic city", "electronic city phase ii", "electronics city phase 1",
15 "frazer town", "gm palaya", "garudachar palya", "giri nagar", "gollarapalya hosahalli", "gottigere",
16 "green glen layout", "gubbalala", "gunjur", "hal 2nd stage", "hbr layout", "hrbr layout", "hsr layout",
17 "haralur road", "harlur", "hebbal", "hebbal kempapura", "hegde nagar", "hennur", "hennur road", "hoodi",
18 "horamavu agara", "horamavu banaswadi", "hormavu", "hosa road", "hosakerehalli", "hoskote", "hosur road",
19 "hulimavu", "isro layout", "itpl", "iblr village", "indira nagar", "jp nagar", "jakkur", "jalahalli",
20 "jalahalli east", "jigani", "judicial layout", "kr puram", "kadubeesanahalli", "kadugodi", "kaggadasapura",
21 "kaggalipura", "kaikondrahalli", "kalena agrahara", "kalyan nagar", "kambipura", "kammanahalli", "kammasandra",
22 "kanakapura", "kanakapura road", "kannamangala", "karuna nagar", "kasavanahalli", "kasturi nagar", "kathriguppe",
23 "kaval byrasandra", "kenchenahalli", "kengeri", "kengeri satellite town", "keneguddadahalli", "kodichikkanahalli",
24 "kodigehalli", "kodigehalli", "kodi halli", "kogilu", "konanakunte", "koramangala", "kothannur", "kothanur", "kudlu", "kudlu ga

```