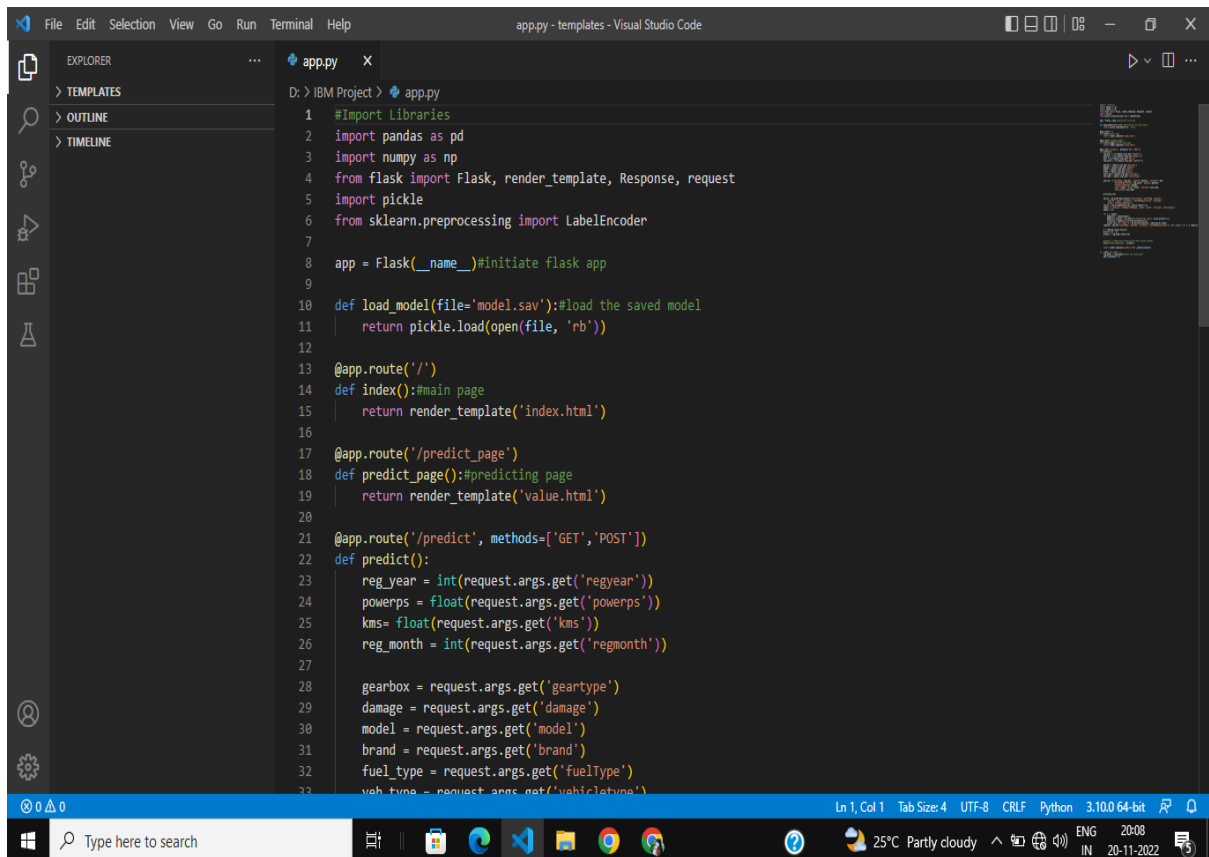
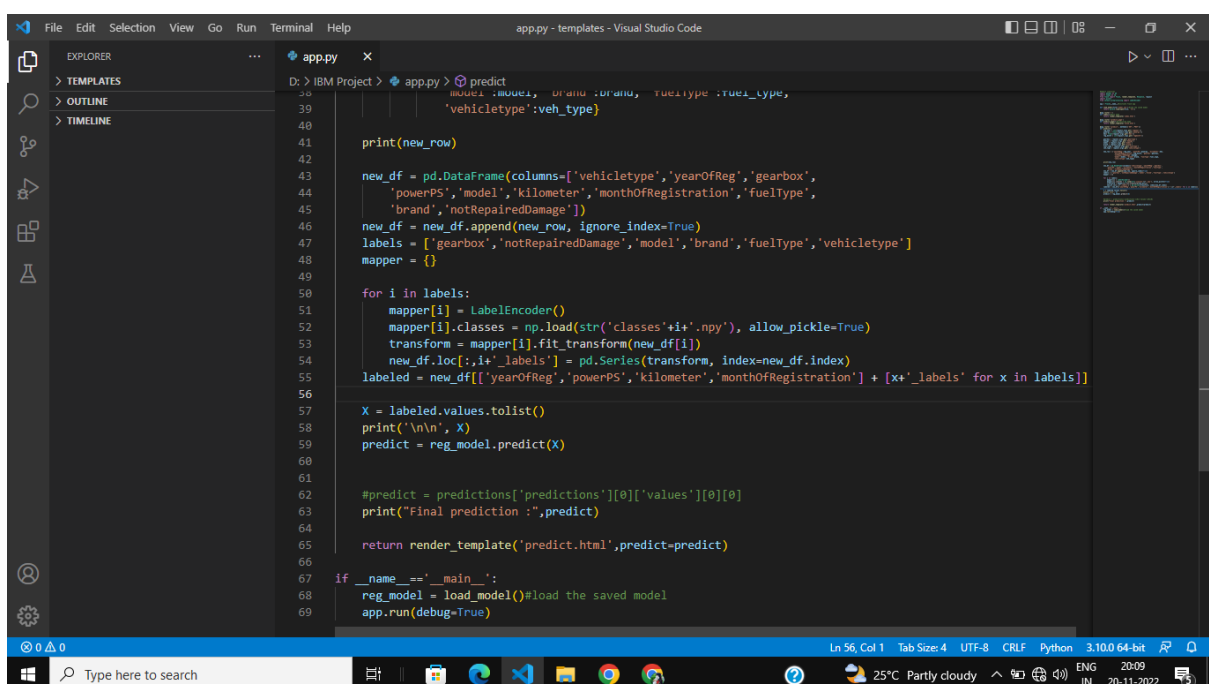


INTEGRATE FLASK WITH SCORING END POINT

Execute the python code and after the module is running, open the localhost and click on the button to predict the output.



```
1 #Import Libraries
2 import pandas as pd
3 import numpy as np
4 from flask import Flask, render_template, Response, request
5 import pickle
6 from sklearn.preprocessing import LabelEncoder
7
8 app = Flask(__name__)#initiate flask app
9
10 def load_model(file='model.sav'):#load the saved model
11     return pickle.load(open(file, 'rb'))
12
13 @app.route('/')
14 def index():#main page
15     return render_template('index.html')
16
17 @app.route('/predict_page')
18 def predict_page():#predicting page
19     return render_template('value.html')
20
21 @app.route('/predict', methods=['GET','POST'])
22 def predict():
23     reg_year = int(request.args.get('regyear'))
24     powerps = float(request.args.get('powerps'))
25     kms = float(request.args.get('kms'))
26     reg_month = int(request.args.get('regmonth'))
27
28     gearbox = request.args.get('geartype')
29     damage = request.args.get('damage')
30     model = request.args.get('model')
31     brand = request.args.get('brand')
32     fuel_type = request.args.get('fuelType')
33     veh_type = request.args.get('vehicletype')
```



```
38     model=model, brand=brand, fuel_type=fuel_type,
39     vehicletype=veh_type}
40
41 print(new_row)
42
43 new_df = pd.DataFrame(columns=['vehicletype','yearOfReg','gearbox',
44     'powerPS','model','kilometer','monthOfRegistration','fuelType',
45     'brand','notRepairedDamage'])
46 new_df = new_df.append(new_row, ignore_index=True)
47 labels = ['gearbox','notRepairedDamage','model','brand','fuelType','vehicletype']
48 mapper = {}
49
50 for i in labels:
51     mapper[i] = LabelEncoder()
52     mapper[i].classes = np.load(str('classes'+i+'.npy'), allow_pickle=True)
53     transform = mapper[i].fit_transform(new_df[i])
54     new_df.loc[:,i+'_labels'] = pd.Series(transform, index=new_df.index)
55 labeled = new_df[['yearOfReg','powerPS','kilometer','monthOfRegistration'] + [x+'_labels' for x in labels]]
56
57 X = labeled.values.tolist()
58 print('\n\n', X)
59 predict = reg_model.predict(X)
60
61 #predict = predictions['predictions'][0]['values'][0][0]
62 print("Final prediction :",predict)
63
64 return render_template('predict.html',predict=predict)
65
66
67 if __name__ == '__main__':
68     reg_model = load_model()#load the saved model
69     app.run(debug=True)
```

