```
from flask import Flask, render_template, request
import pandas as pd
import numpy as np
import pickle
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "oUHg5RCJT5AVH-CqNzu1fyA067ZbL9NdbCOH3R0DdnIh"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
                                                    API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json',
          'Authorization': 'Bearer ' + mltoken}

app = Flask(_name_)
model = pickle.load(open("CarLinearRegressionModel.pkl", 'rb'))
car = pd.read_csv("Cleaned Car.csv")



@app.route('/')
def index():

    car_models = sorted(car['car_models'].unique())
    company_name = sorted(car['company_name'].unique())
    year = sorted(car['year'].unique(), reverse=True)
    km_driven = sorted(car['year'].unique())
```

```
    fuel = sorted(car['fuel'].unique())
    seller_type = sorted(car['seller_type'].unique())
    transmission = sorted(car['transmission'].unique())
    owner = sorted(car['owner'].unique())
    return render_template('index.html', car_models=car_models, company_name=company_name, year=year, km_driven=km_driven, fuel=fuel, seller_type=seller_type, transmission=transmission, owner=owner)


@app.route('/predict', methods=['POST'])
def predict():
    company_name = request.form.get('company_name')
    car_models = request.form.get('car_models')
    year = int(request.form.get('year'))
    km_driven = int(request.form.get('kilo_driven'))
    fuel = request.form.get('fuel_type')
    seller_type = request.form.get('seller_type')
    transmission = request.form.get('transmission')
    owner = request.form.get('owner')
    # t=[[car_models,company_name,year,km_driven,fuel,seller_type,transmission,owner]]
    # NOTE: manually define and pass the array(s) of values to be scored in the next line
    payload_scoring = {"input_data": [{"fields": ['car_models', 'company_name', 'year', 'km_driven', 'fuel', 'seller_type',
                                        'transmission', 'owner'], "values":[[car_models, company_name, year, km_driven, fuel, seller_type, transmission, owner]]}]}
    response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/fb824ca5-dfcf-41e1-979f-4a731cd910b5/predictions?version=2022-11-18', json=payload_scoring,
                                     headers={'Authorization': 'Bearer ' + mltoken})
    prediction = response_scoring['predictions'][0]['values']
    return str(np.round(prediction[00], 2))
```

```
if _name == "main_":
    app.run()
```