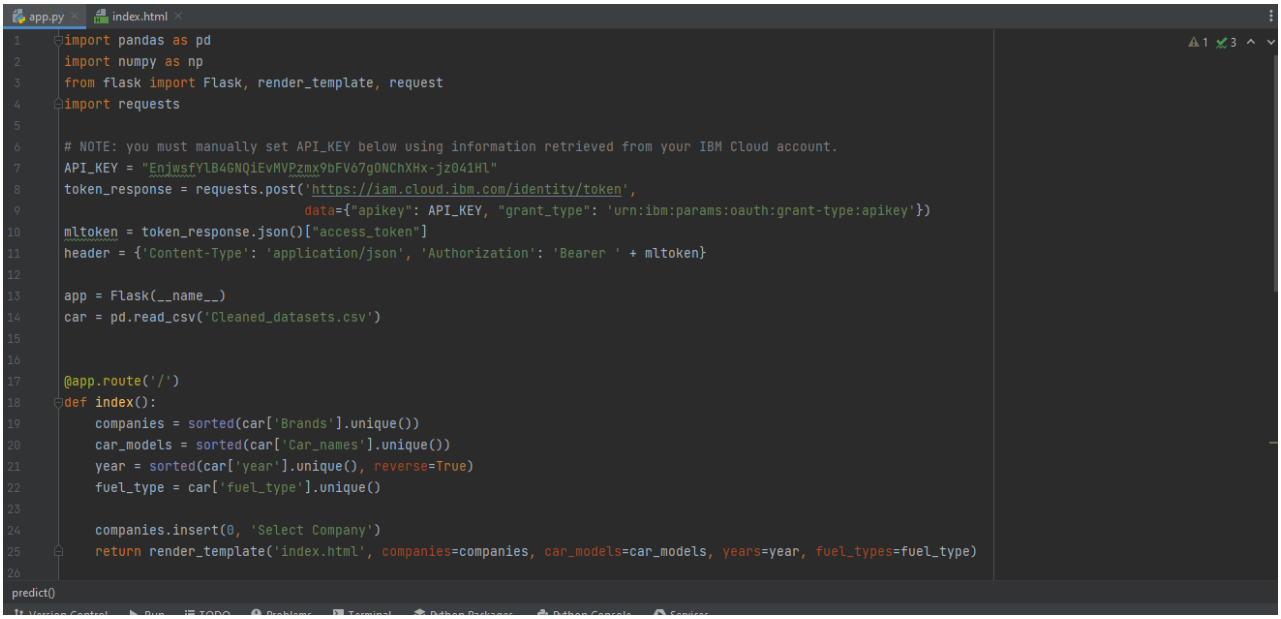
EXECUTE AND TEST MODEL

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



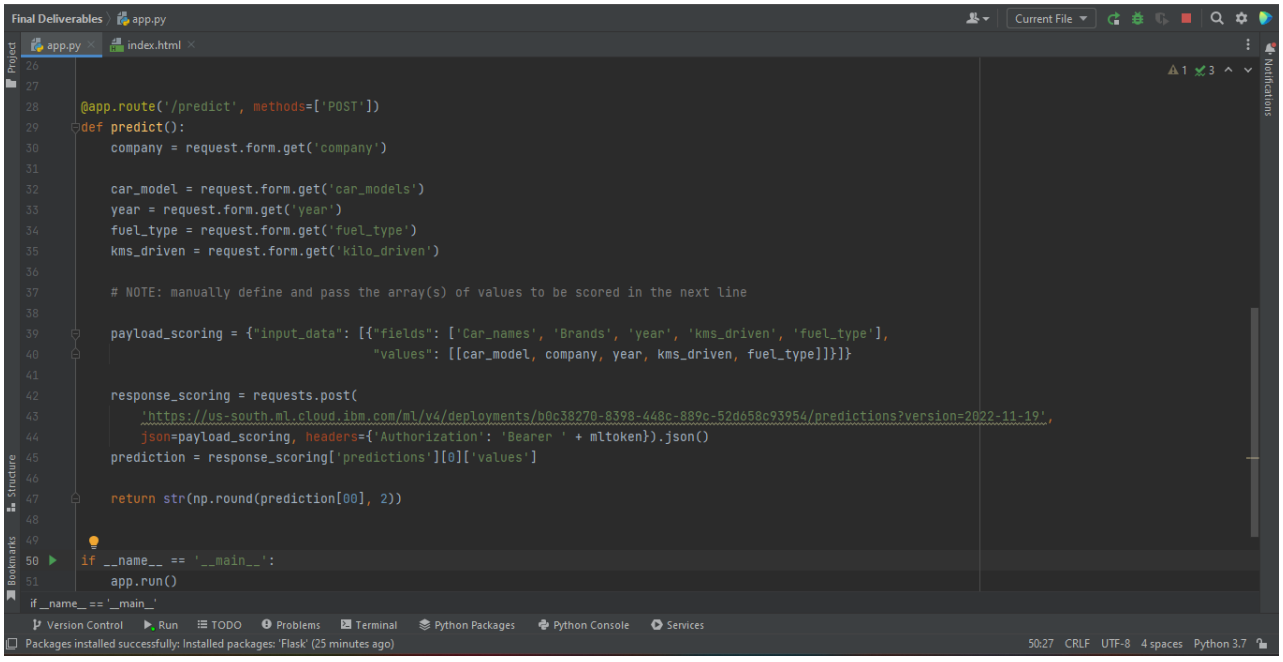
```python
import pandas as pd
import numpy as np
from flask import Flask, render_template, request
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "EnjwsfYlB4GNQiEvMVPzmx9bFV67gONChXHx-jzO41Hl"
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__)
car = pd.read_csv('Cleaned_datasets.csv')


@app.route('/')
def index():
    companies = sorted(car['Brands'].unique())
    car_models = sorted(car['Car_names'].unique())
    year = sorted(car['year'].unique(), reverse=True)
    fuel_type = car['fuel_type'].unique()

    companies.insert(0, 'Select Company')
    return render_template('index.html', companies=companies, car_models=car_models, years=year, fuel_types=fuel_type)
```



```python
@app.route('/predict', methods=['POST'])
def predict():
    company = request.form.get('company')

    car_model = request.form.get('car_models')
    year = request.form.get('year')
    fuel_type = request.form.get('fuel_type')
    kms_driven = request.form.get('kilo_driven')

    # NOTE: manually define and pass the array(s) of values to be scored in the next line

    payload_scoring = {"input_data": [{"fields": ['Car_names', 'Brands', 'year', 'kms_driven', 'fuel_type'],
                                       "values": [[car_model, company, year, kms_driven, fuel_type]]}]}

    response_scoring = requests.post(
        'https://us-south.ml.cloud.ibm.com/ml/v4/deployments/b0c38270-8398-448c-889c-52d658c93954/predictions?version=2022-11-19',
        json=payload_scoring, headers={'Authorization': 'Bearer ' + mltoken}).json()
    prediction = response_scoring['predictions'][0]['values']

    return str(np.round(prediction[00], 2))


if __name__ == '__main__':
    app.run()
```

This is the home page of the application