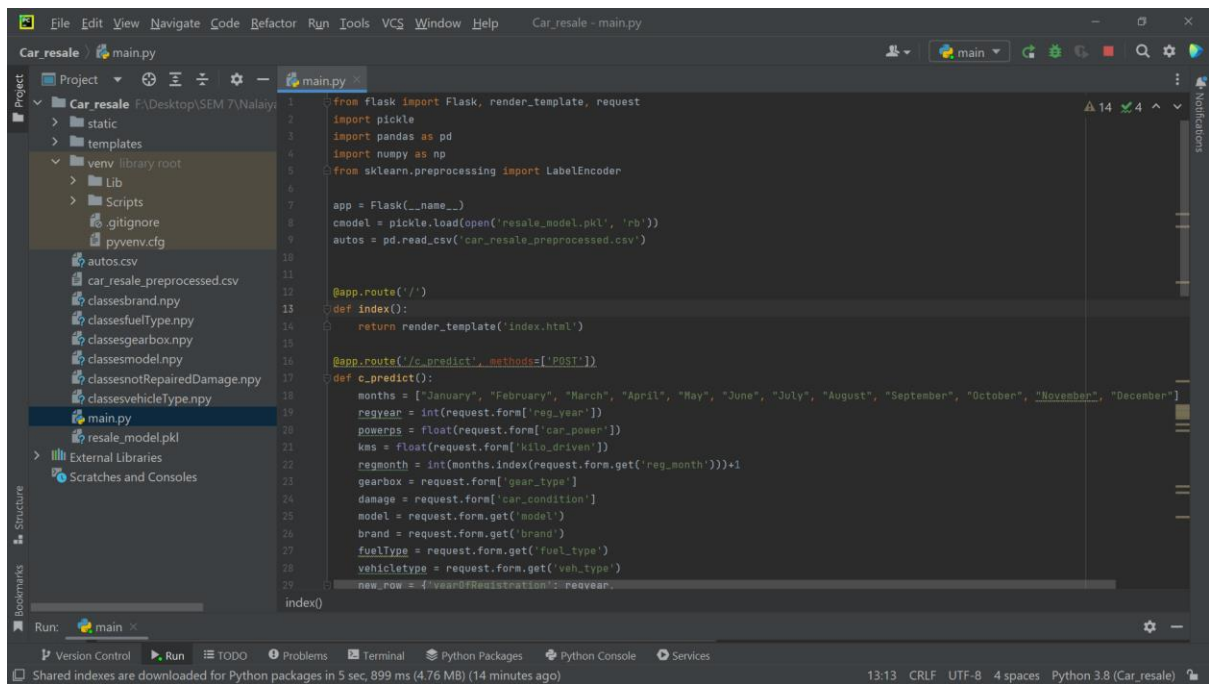


## Development Phase Application Building

Date	17 November 2022
Team ID	PNT2022TMID21553
Project Name	Project – Car Resale Value Prediction

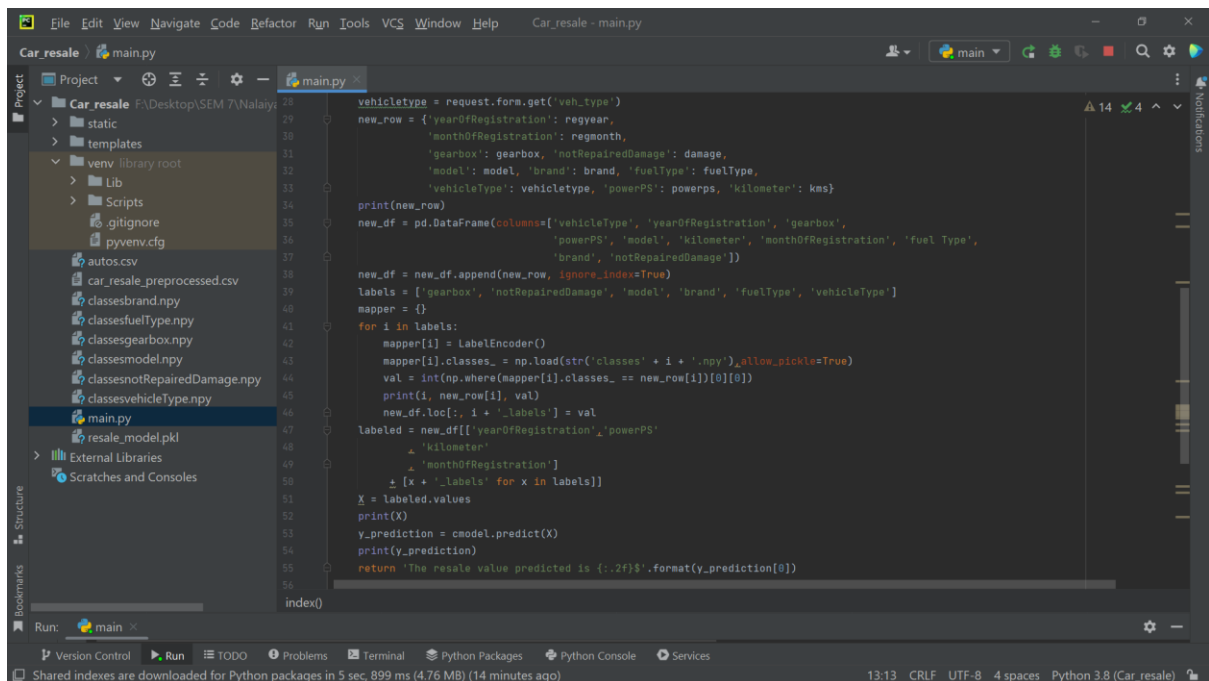
### Build the python flask:



```

1 from flask import Flask, render_template, request
2 import pickle
3 import pandas as pd
4 import numpy as np
5 from sklearn.preprocessing import LabelEncoder
6
7 app = Flask(__name__)
8 cmodel = pickle.load(open('resale_model.pkl', 'rb'))
9 autos = pd.read_csv('car_resale_preprocessed.csv')
10
11 @app.route('/')
12 def index():
13     return render_template('index.html')
14
15 @app.route('/c_predict', methods=['POST'])
16 def c_predict():
17     months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December']
18     regyear = int(request.form['reg_year'])
19     powerps = float(request.form['car_power'])
20     kms = float(request.form['kilo_driven'])
21     regmonth = int(months.index(request.form.get('reg_month')))+1
22     gearbox = request.form['gear_type']
23     damage = request.form['car_condition']
24     model = request.form.get('model')
25     brand = request.form.get('brand')
26     fuelType = request.form.get('fuel_type')
27     vehicleType = request.form.get('veh_type')
28     new_row = {'yearOfRegistration': regyear,
29

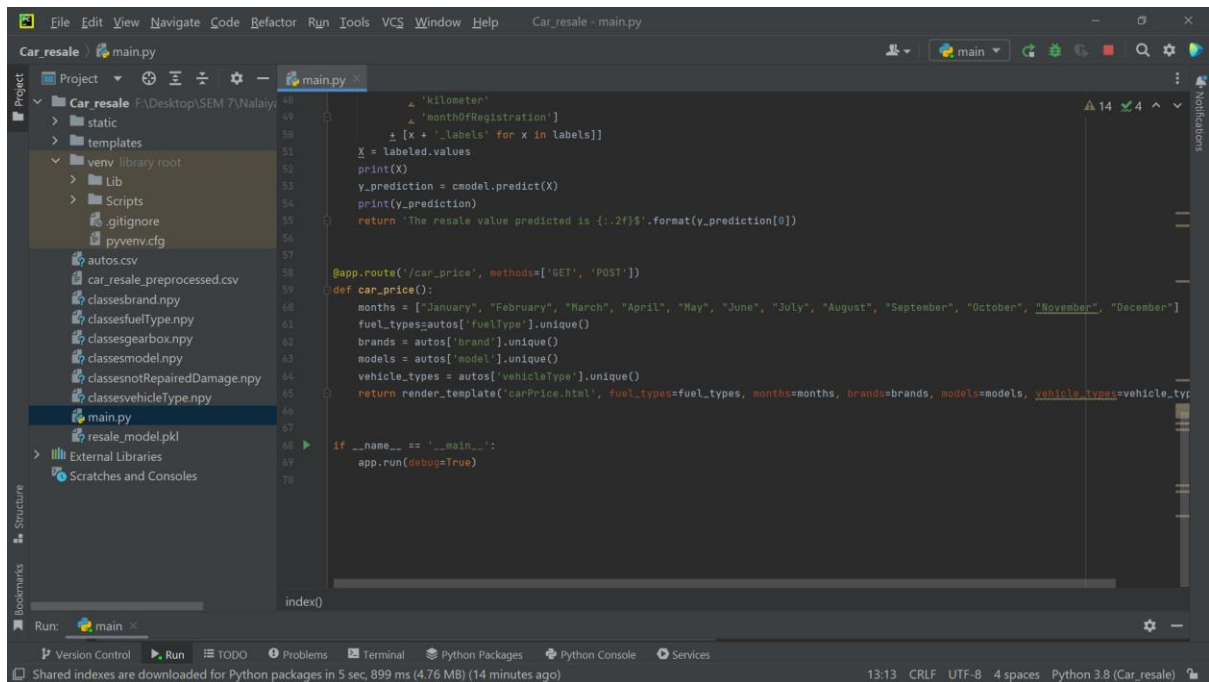
```



```

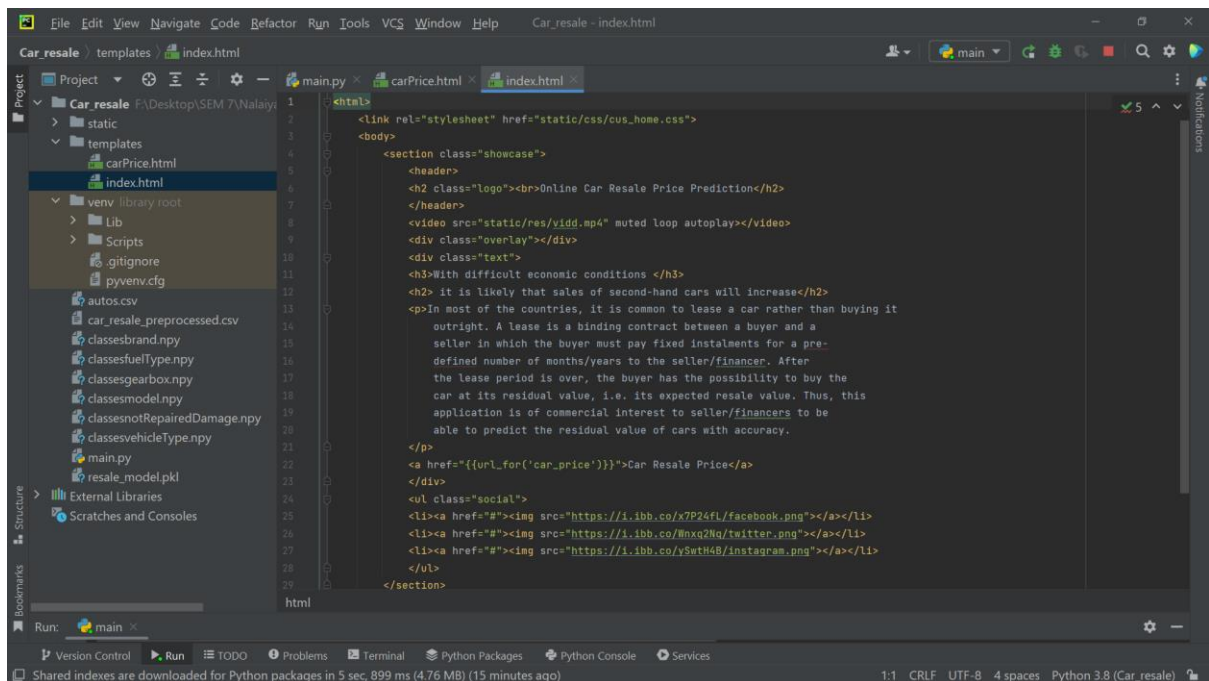
30     'yearOfRegistration': regyear,
31     'monthOfRegistration': regmonth,
32     'gearbox': gearbox, 'notRepairedDamage': damage,
33     'model': model, 'brand': brand, 'fuelType': fuelType,
34     'vehicleType': vehicleType, 'powerPS': powerps, 'kilometer': kms}
35     print(new_row)
36     new_df = pd.DataFrame(columns=['vehicleType', 'yearOfRegistration', 'gearbox',
37     'powerPS', 'model', 'kilometer', 'monthOfRegistration', 'fuel Type',
38     'brand', 'notRepairedDamage'])
39     new_df.append(new_row, ignore_index=True)
40     labels = ['gearbox', 'notRepairedDamage', 'model', 'brand', 'fuelType', 'vehicleType']
41     mapper = {}
42     for i in labels:
43         mapper[i] = LabelEncoder()
44         mapper[i].classes_ = np.load(str('classes' + i + '.npy'), allow_pickle=True)
45         val = int(np.where(mapper[i].classes_ == new_row[i])[0][0])
46         print(i, new_row[i], val)
47         new_df.loc[:, i + '_labels'] = val
48     labeled = new_df[['yearOfRegistration', 'powerPS',
49     'kilometer',
50     'monthOfRegistration']]
51     X = labeled.values
52     print(X)
53     y_prediction = cmodel.predict(X)
54     print(y_prediction)
55     return 'The resale value predicted is {:.2f}$'.format(y_prediction[0])
56

```

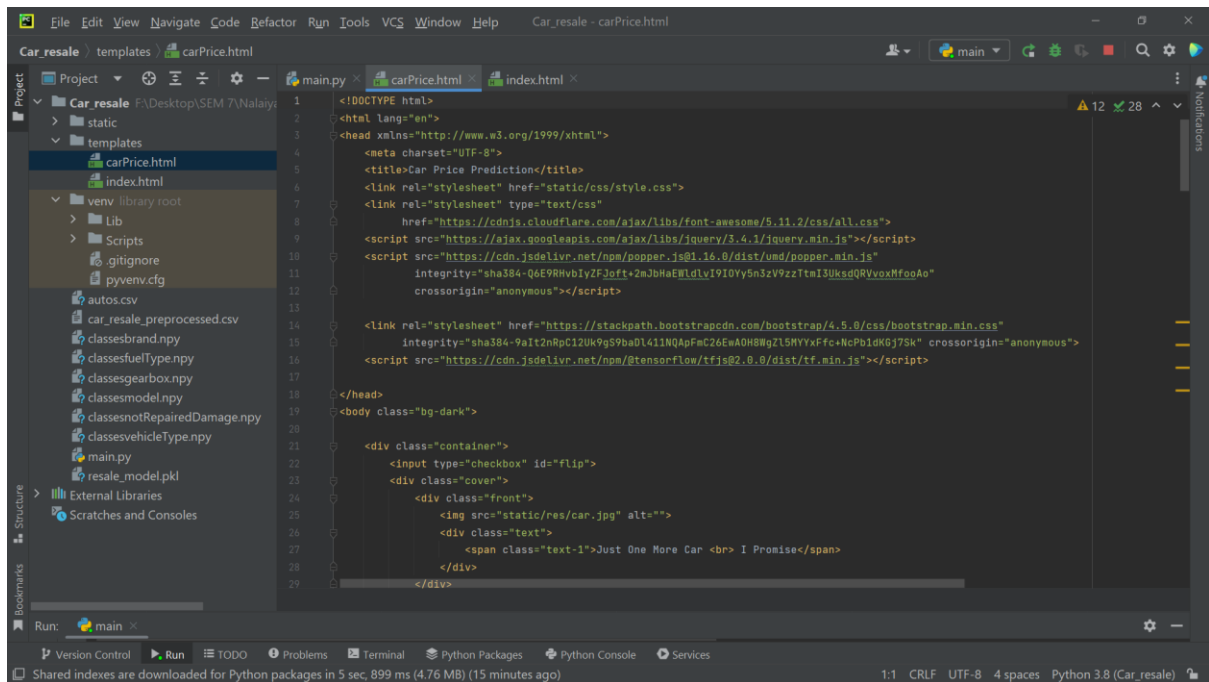


```
48     'kilometer'
49     'monthOfRegistration']
50     + [x + 'Labels' for x in labels]]
51     X = labeled.values
52     print(X)
53     y_prediction = cmodel.predict(X)
54     print(y_prediction)
55     return 'The resale value predicted is {:.2f}$'.format(y_prediction[0])
56
57
58 @app.route('/car_price', methods=['GET', 'POST'])
59 def car_price():
60     months = ["January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"]
61     fuel_types = autos['fuelType'].unique()
62     brands = autos['brand'].unique()
63     models = autos['model'].unique()
64     vehicle_types = autos['vehicleType'].unique()
65     return render_template('carPrice.html', fuel_types=fuel_types, months=months, brands=brands, models=models, vehicle_types=vehicle_types)
66
67
68 if __name__ == '__main__':
69     app.run(debug=True)
70
71
72 index()
```

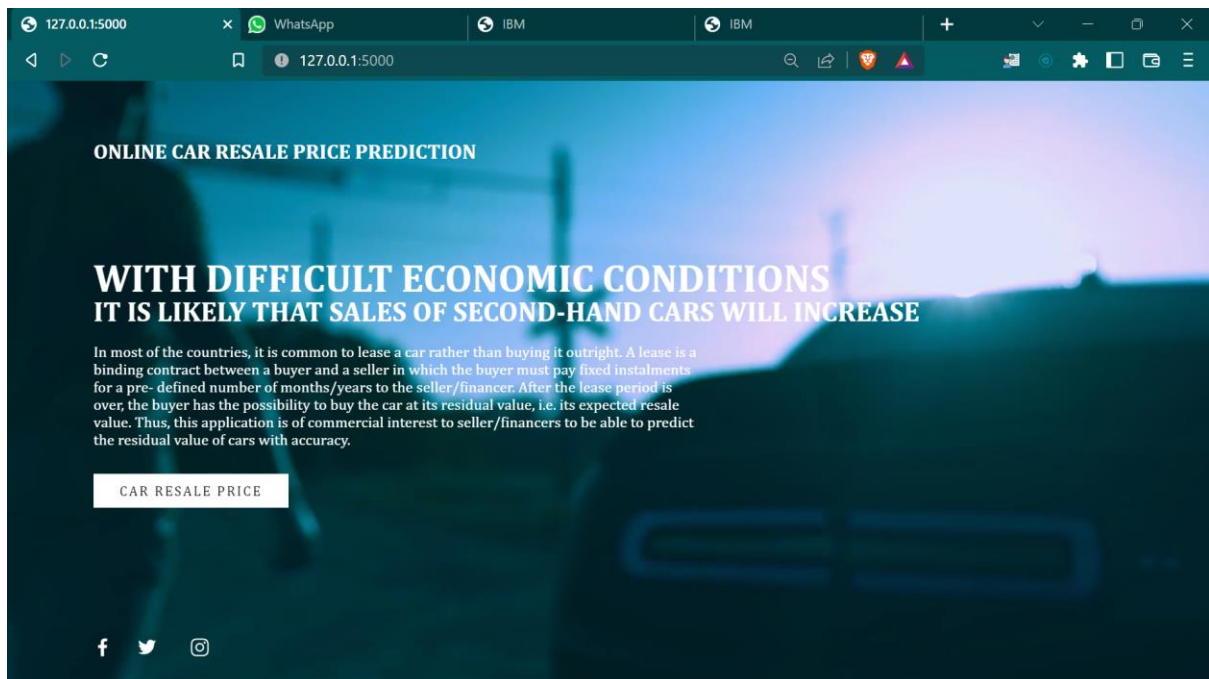
## Build HTML page:



```
1 <html>
2 <link rel="stylesheet" href="static/css/cus_home.css">
3
4 <body>
5     <section class="showcase">
6         <header>
7             <h2 class="logo"><br>Online Car Resale Price Prediction</h2>
8         </header>
9         <video src="static/res/vidd.mp4" muted loop autoplay></video>
10        <div class="overlay"></div>
11        <div class="text">
12            <h3>With difficult economic conditions </h3>
13            <h2> it is likely that sales of second-hand cars will increase</h2>
14        </div>
15        <p>In most of the countries, it is common to lease a car rather than buying it outright. A lease is a binding contract between a buyer and a seller in which the buyer must pay fixed instalments for a pre-defined number of months/years to the seller/financer. After the lease period is over, the buyer has the possibility to buy the car at its residual value, i.e. its expected resale value. Thus, this application is of commercial interest to seller/financers to be able to predict the residual value of cars with accuracy.</p>
16        <a href="{{url_for('car_price')}}">Car Resale Price</a>
17    </div>
18    <ul class="social">
19        <li><a href="#"></a></li>
20        <li><a href="#"></a></li>
21        <li><a href="#"></a></li>
22    </ul>
23    </section>
24
25 html
```



Execute and test the model:



Car Price Prediction

WhatsApp

IBM

IBM

127.0.0.1:5000/car\_price

### Car Price Prediction

Enter the Registration Year:

Select the Registration Month:

January

Enter the Power of Car in PS:

Enter the Number of Kilometres that the car has travelled:


Select the Gear Box Type:

☒ Manual ☐ Automatic ☐ Not Declared

Your Car is damaged or repaired:

☐ Yes ☒ No ☐ Not Declared

Select the Model Type:



Just One More Car  
I Promise

Car Price Prediction

WhatsApp

IBM

IBM

127.0.0.1:5000/car\_price

Select the Gear Box Type:

☒ Manual ☐ Automatic ☐ Not Declared

Your Car is damaged or repaired:

☐ Yes ☒ No ☐ Not Declared

Select the Model Type:

not-declared

Select the Brand of the Car:

audi

Select the Fuel Type of Car:

diesel


Select the Vehicle Type of Car:

coupe

Predict Price

[Go to HomePage : Click Here..](#)

The resale value predicted is 5973.54\$



Just One More Car  
I Promise