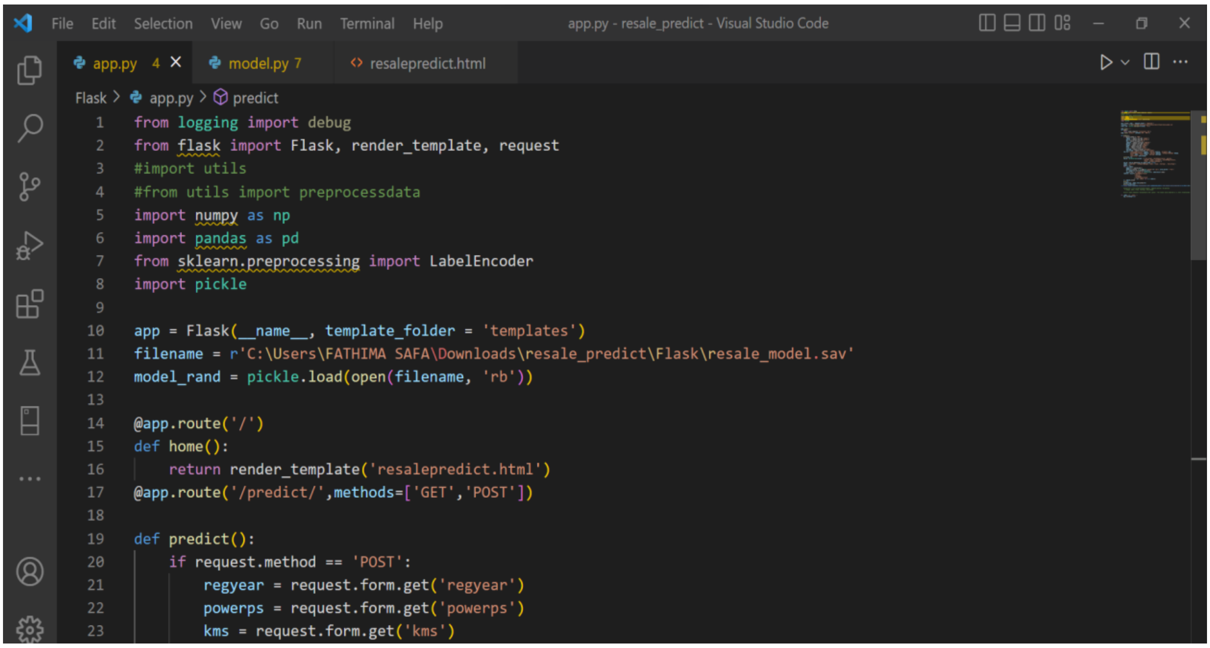
APPLICATION BUILDING

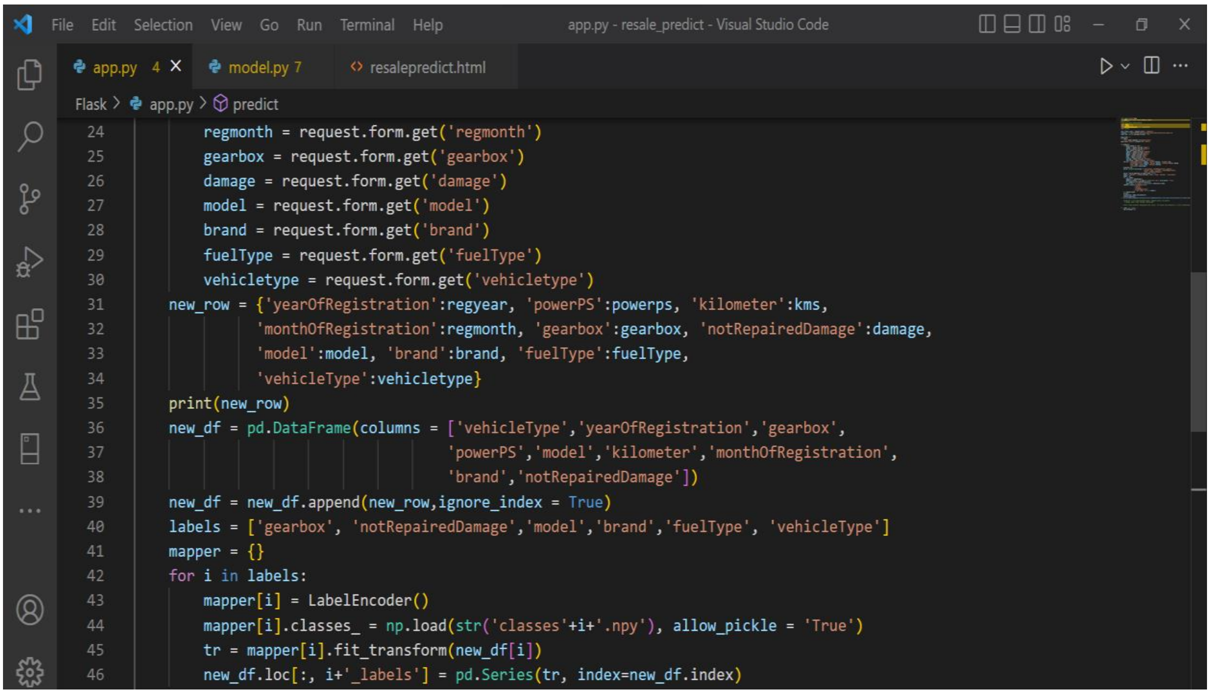
BUILD THE PYTHON FLASK APP

Date	17 November 2022
Team Id	PNT2022TMID21221
Project Name	Project – Car Resale Value Prediction

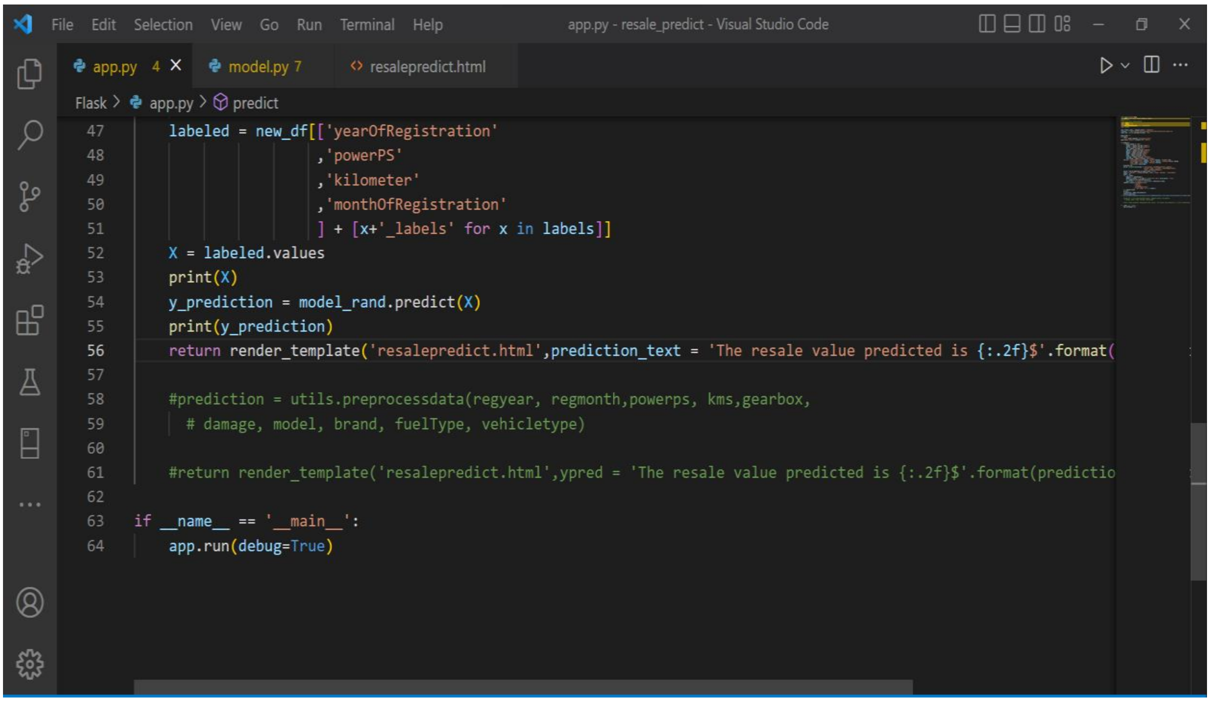
The file for flask application building is saved as app.py. All the necessary libraries are imported first and the flask application is started. Load the resale_model.sav file and start defining the function predict. Encode the inputted data and use them for prediction.


File Edit Selection View Go Run Terminal Help
app.py - resale_predict - Visual Studio Code
app.py 4
model.py 7
resalepredict.html
Flask > app.py > predict

```python
from logging import debug
from flask import Flask, render_template, request
#import utils
#from utils import preprocessdata
import numpy as np
import pandas as pd
from sklearn.preprocessing import LabelEncoder
import pickle

app = Flask(__name__, template_folder = 'templates')
filename = r'C:\Users\FATHIMA SAFA\Downloads\resale_predict\Flask\resale_model.sav'
model_rand = pickle.load(open(filename, 'rb'))

@app.route('/')
def home():
    return render_template('resalepredict.html')
@app.route('/predict/',methods=['GET','POST'])

def predict():
    if request.method == 'POST':
        regyear = request.form.get('regyear')
        powerps = request.form.get('powerps')
        kms = request.form.get('kms')
```


File Edit Selection View Go Run Terminal Help
app.py - resale_predict - Visual Studio Code
app.py 4
model.py 7
resalepredict.html
Flask > app.py > predict

```python
        regmonth = request.form.get('regmonth')
        gearbox = request.form.get('gearbox')
        damage = request.form.get('damage')
        model = request.form.get('model')
        brand = request.form.get('brand')
        fuelType = request.form.get('fuelType')
        vehicletype = request.form.get('vehicletype')
    new_row = {'yearOfRegistration':regyear, 'powerPS':powerps, 'kilometer':kms,
               'monthOfRegistration':regmonth, 'gearbox':gearbox, 'notRepairedDamage':damage,
               'model':model, 'brand':brand, 'fuelType':fuelType,
               'vehicleType':vehicletype}
    print(new_row)
    new_df = pd.DataFrame(columns = ['vehicleType','yearOfRegistration','gearbox',
                                     'powerPS','model','kilometer','monthOfRegistration',
                                     'brand','notRepairedDamage'])
    new_df = new_df.append(new_row,ignore_index = True)
    labels = ['gearbox', 'notRepairedDamage','model','brand','fuelType', 'vehicleType']
    mapper = {}
    for i in labels:
        mapper[i] = LabelEncoder()
        mapper[i].classes_ = np.load(str('classes'+i+'.npy'), allow_pickle = 'True')
        tr = mapper[i].fit_transform(new_df[i])
        new_df.loc[:, i+'_labels'] = pd.Series(tr, index=new_df.index)
```



```
    labeled = new_df[['yearOfRegistration'
                     ,'powerPS'
                     ,'kilometer'
                     ,'monthOfRegistration'
                     ] + [x+'_labels' for x in labels]]
    X = labeled.values
    print(X)
    y_prediction = model_rand.predict(X)
    print(y_prediction)
    return render_template('resalepredict.html',prediction_text = 'The resale value predicted is {:.2f}$'.format(

    #prediction = utils.preprocessdata(regyear, regmonth,powerps, kms,gearbox,
      # damage, model, brand, fuelType, vehicletype)

    #return render_template('resalepredict.html',ypred = 'The resale value predicted is {:.2f}$'.format(predictio

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