

```

#app.py
from flask import Flask,request, url_for, redirect, render_template, jsonify,session
import sqlite3 as sql
from flask_cors import CORS, cross_origin
import pickle
import numpy as np
import os
import pandas as pd
import joblib

```

```

app = Flask(__name__)
app.secret_key = "Secret Key"
# load the saved model file and use for prediction
model = pickle.load(open("CarPricePredictionModel.pkl", "rb"))

```

```

@app.after_request # blueprint can also be app~~
def after_request(response):
    header = response.headers
    header['Access-Control-Allow-Origin'] = '*'
    return response

```

```

# =====
# Insert data in database (SIGNUP)
# =====
def insertUser(username, email, password, contact):
    con = sql.connect("SignUP.db")
    cur = con.cursor()
    phone = int(contact)
    query = ("""INSERT INTO SignUP
                (username,email,password,contact)
                VALUES ('%s','%s','%s',%s)""" %
                (username, email, password, contact))
    cur.execute(query)
    con.commit()
    con.close()

```

```

# =====
# Validating data in database (LOGIN)
# =====
def validUser(email, password):
    con = sql.connect("SignUP.db")
    cur = con.cursor()
    query = ("""SELECT * FROM SignUP
                where email = '%s' and password = '%s'
                """) %

```

```

        (email, password))
    cur.execute(query)
    data = cur.fetchall()
    con.close()
    return data

# =====
#     Flask Routing
# =====

@app.route('/')
def home111():
    return render_template('login_1.html')

# Login page
@app.route('/login_1', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        rd = validUser(request.form['email'], request.form['password'])
        if rd:
            session['user']=rd[0]
            return render_template('homepage_1.html')
        else:
            msg="Wrong username or password"
            return render_template('login_1.html',msg=msg)
    else:
        return render_template('login_1.html')

@app.route('/logout')
def logout1():
    session.pop('user', None)
    return render_template('login_1.html')

@app.route('/s')
def student():
    if 'user' in session:
        s = session['user']
        all_data = Student.query.all()
        return render_template("homepage_1.html", all_data = all_data,user=s)
    else:
        return render_template('login_1.html')

# Signup page
@app.route('/signup/', methods=['GET', 'POST'])
def signup():

```

```

    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        contact = request.form['contact']
        insertUser(username, email, password, contact)
        msg= "account created successfully"
        return redirect(url_for('login'))
    else:
        return render_template('login_1.html')

# api json
@app.route('/sum', methods=['GET', 'POST'])
def sum():
    sum = 0
    a = int(request.args.get('a'))
    b = int(request.args.get('b'))
    sum = a+b
    return jsonify(sum)

@app.route('/mainpage')
def mainhome():
    return render_template("homepage_1.html")

@app.route('/contact')
def contact():
    return render_template("contact.html")

@app.route('/about')
def about():
    return render_template("about.html")

@app.route("/predict", methods=['GET', 'POST'])
def predict():
    if request.method == 'POST':
        year = int(request.form['year'])
        km_driven=float(request.form['km_driven'])
        owner=request.form['owner']
        if(owner=='test'):
            owner=0
        elif(owner=='first'):
            owner=1
        elif(owner=='second'):
            owner=2
        elif(owner=='third'):
            owner=3

```

```

elif(owner=='fourth'):
    owner=4

fuel=request.form['fuel']
if(fuel=='Diesel'):
    fuel=0
elif(fuel=='Petrol'):
    fuel=1
elif(fuel=='LPG'):
    fuel=2
elif(fuel=='CNG'):
    fuel=3
Current_year = 2021
years_driven = Current_year - year
seller_type=request.form['seller_type']
if(seller_type=='Individual'):
    seller_type=0
elif(seller_type=='Dealer'):
    seller_type=1
transmission=request.form['transmission']
if(transmission == 'Mannual'):
    transmission=1
elif(transmission == 'Automatic'):
    transmission=0
mileage = float(request.form['mileage'])
engine = float(request.form['engine'])
max_power = float(request.form['max_power'])
max_power = max_power - 30
torque = float(request.form['torque'])
torque = torque - 40
seats = int(request.form['seats'])
prediction=model.predict(np.array([[year, km_driven, fuel, seller_type,
transmission, owner, mileage, engine, max_power, torque, seats, Current_year,
years_driven]]))
#output=round(prediction[0],2)
output1 = str(prediction)
output = output1.strip("[].")
#if output<0:
#    return render_template('index.html',prediction_texts="Sorry you cannot
sell this car")
#else:
    return render_template('predict.html',prediction_text="You can sell the Car
at ₹{}".format(output))
else:
    return render_template('predict.html')

```

```

if __name__ == '__main__':

```

```
app.run(debug=True)
```