

## PROJECT DEVELOPMENT PHASE

### (DELIVERY OF SPRINT-3)

Date	18 November 2022
Team ID	PNT2022TMID30426
Project Name	Car Resale Value prediction
Maximum marks	4 Marks

- **Flask Framework:**

```
# Import Libraries

import pandas as pd
import numpy as np
from flask import Flask, render_template, Response, request
import pickle
from sklearn.preprocessing import LabelEncoder

app = Flask(__name__)#initiate flask app

def load_model(file='model.sav'):#load the saved model
    return pickle.load(open(file, 'rb'))

@app.route('/')
def index():#main page
    return render_template('car.html')

@app.route('/predict_page')
def predict_page():#predicting page
    return render_template('value.html')

@app.route('/predict', methods=['GET', 'POST'])
def predict():
    reg_year = int(request.args.get('regyear'))
    powerps = float(request.args.get('powerps'))
    kms= float(request.args.get('kms'))
    reg_month = int(request.args.get('regmonth'))

    gearbox = request.args.get('geartype')
    damage = request.args.get('damage')
    model = request.args.get('model')
    brand = request.args.get('brand')
    fuel_type = request.args.get('fuelType')
    veh_type = request.args.get('vehicletype')
```

```

new_row = {'yearOfReg':reg_year, 'powerPS':powerps, 'kilometer':kms,
           'monthOfRegistration':reg_month, 'gearbox':gearbox,
           'notRepairedDamage':damage,
           'model':model, 'brand':brand, 'fuelType':fuel_type,
           'vehicletype':veh_type}

print(new_row)

new_df = pd.DataFrame(columns=['vehicletype','yearOfReg','gearbox',
                              'powerPS','model','kilometer','monthOfRegistration','fuelType',
                              '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)
    transform = mapper[i].fit_transform(new_df[i])
    new_df.loc[:,i+'_labels'] = pd.Series(transform, index=new_df.index)
    labeled = new_df[['yearOfReg','powerPS','kilometer','monthOfRegistration']
+ [x+'_labels' for x in labels]]

X = labeled.values.tolist()
print('\n\n', X)
predict = reg_model.predict(X)

#predict = predictions['predictions'][0]['values'][0][0]
print("Final prediction :",predict)

return render_template('predict.html',predict=predict)

if __name__ == '__main__':
    reg_model = load_model()#load the saved model
    app.run(debug=True)

```

## HTML Design Code:

```
<!DOCTYPE html>
<nav lang="en" dir="ltr">
  <head>
    <style>
      :root {

        --typewriterSpeed: 6s;
      }

      body
      {
        margin:0;

        background-color:peachpuff;
      }
      h1
      {
        position:relative;
        font-size:4rem;
        position:relative;
      }
      a{
        text-decoration: none;
      }
      .bg{

        margin:50px 170px;

      }
      button{
        margin-left: 650px;
        background-color: lightgreen;
      }
      h1::before,
      h1::after
      {
        content:"";
        position:absolute;
        top:0;
        bottom:0;
        left:0;
        right:0;
      }

      h1::before
      {
        background:peachpuff;
        animation:typewriter 6s steps(28) 2s forwards;
      }
    </style>
  </head>
  <body>
    <div class="bg">
      <h1>
        <a href="#">
          <button>
            </a>
          </h1>
        </div>
      </body>
    </nav>
```

```

h1::after
{
    width:0.125em;
    background:black;
    animation:
        typewriter 6s steps(28) 2s forwards,
        blink 730ms steps(28) infinite,
        blinks 12s steps(28) forwards;
}
@keyframes blinks
{
    to{
        background-color:peachpuff;
    }
}
@keyframes typewriter
{
    to{left:100%;}
}

@keyframes blink
{
    to{
        background-color:transparent;
    }
}

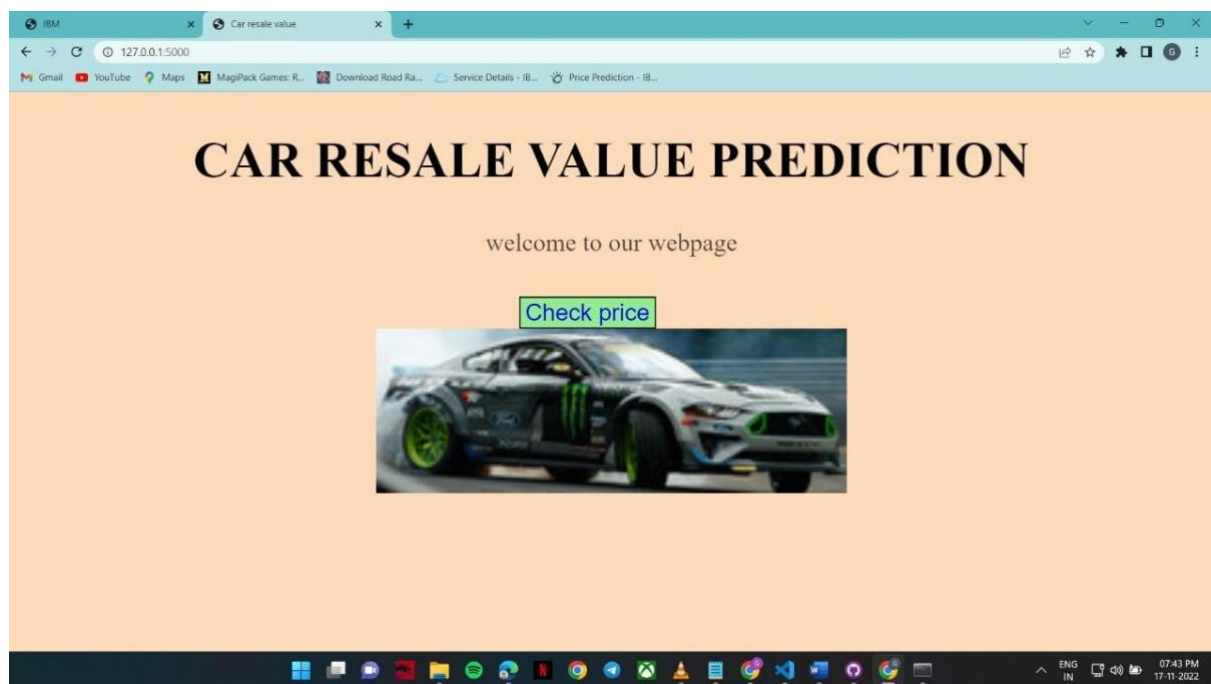
.subtitle
{
    display:grid;
    place-content:center;
    text-align:center;
    color:hsl(0 0% 0%/0.7);
    font-size:2rem;
    font-weight:400;
    opacity:0;
    transform: translateY(3rem);
    animation: fade 2s ease 8s forwards;
}

@keyframes fade {
    to {
        opacity: 1;
        transform: translateY(0);
    }
}


```

```
<meta charset="utf-8">
<title>Car resale value
</title>
<link rel="stylesheet"
href="../static/css/styl
e.css">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-
```

### Application Webpage:



IBM x Car resale value x +

127.0.0.1:5000/predict\_page

Gmail YouTube Maps MagiPack Games: R... Download Road Ra... Service Details - IB... Price Prediction - IB...

## Accurate Resale Value of Your Car

Registration year :

Registration Month :

Power of car in PS:

Kilometers that car have driven :

Gear type : ☒ Manual ☐ Automatic ☐ Not declared

Your car is repaired or damaged : ☐ Yes ☒ No ☐ Not declared

Model Type :

Windows taskbar: ENG IN 07:45 PM 17-11-2022

IBM x Car resale value x +

127.0.0.1:5000/predict\_page

Gmail YouTube Maps MagiPack Games: R... Download Road Ra... Service Details - IB... Price Prediction - IB...

Power of car in PS:

Kilometers that car have driven :

Gear type : ☒ Manual ☐ Automatic ☐ Not declared

Your car is repaired or damaged : ☐ Yes ☒ No ☐ Not declared

Model Type :

Brand :

Fuel Type :

Vehicle type:

Windows taskbar: ENG IN 07:45 PM 17-11-2022



**The Predicted Car Resale Value is**

**₹[16984.07610861]**

