

## Project Development Phase

### Sprint 4

Date	14 November 2022
Team ID	PNT2022TMID41243
Project Name	Car Resale Value Prediction

#### Sprint 4 tasks:

- Creating web application
- Embedding Dashboard to web application
- Embedding Report to web application
- Embedding Story to web application

#### Creating web application:

```
File Edit Selection Find View Goto Tools Project Preferences Help
index.html x
1 <!DOCTYPE html>
2 <html lang="en">
3 <head xmlns="http://www.w3.org/1999/xhtml">
4   <meta charset="UTF-8">
5   <title>Car Price Predictor</title>
6   <link rel="stylesheet" href="static/css/style.css">
7   <link rel="stylesheet" type="text/css"
8     href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.11.2/css/all.css">
9   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
10  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
11    integrity="sha384-Q6E9RHvbiYzFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
12    crossorigin="anonymous"></script>
13
14  <!-- Bootstrap CSS -->
15  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
16    integrity="sha384-9aIt2nRpC12Uk9gS9baD1411NQApFmC26EwAOH8WgZ15MYxXfFc+NcPb1dK6j7Sk" crossorigin="anonymous">
17  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@2.0.0/dist/tf.min.js"></script>
18
19 </head>
20 <body class="bg-dark">
21
22 <div class="container">
23   <div class="row">
24     <div class="card mt-50" style="width: 100%; height: 100%">
25       <div class="card-header" style="text-align: center">
26         <h1>Welcome to Car Price Predictor</h1>
27       </div>
28       <div class="card-body">
29         <div class="col-12" style="text-align: center">
30           <h5>This webpage is predicts the price of a car you want to sell. Try filling the details below: </
31             h5>
32           </div>
33           <br>
34           <form method="post" accept-charset="utf-8" name="Modelform">
```

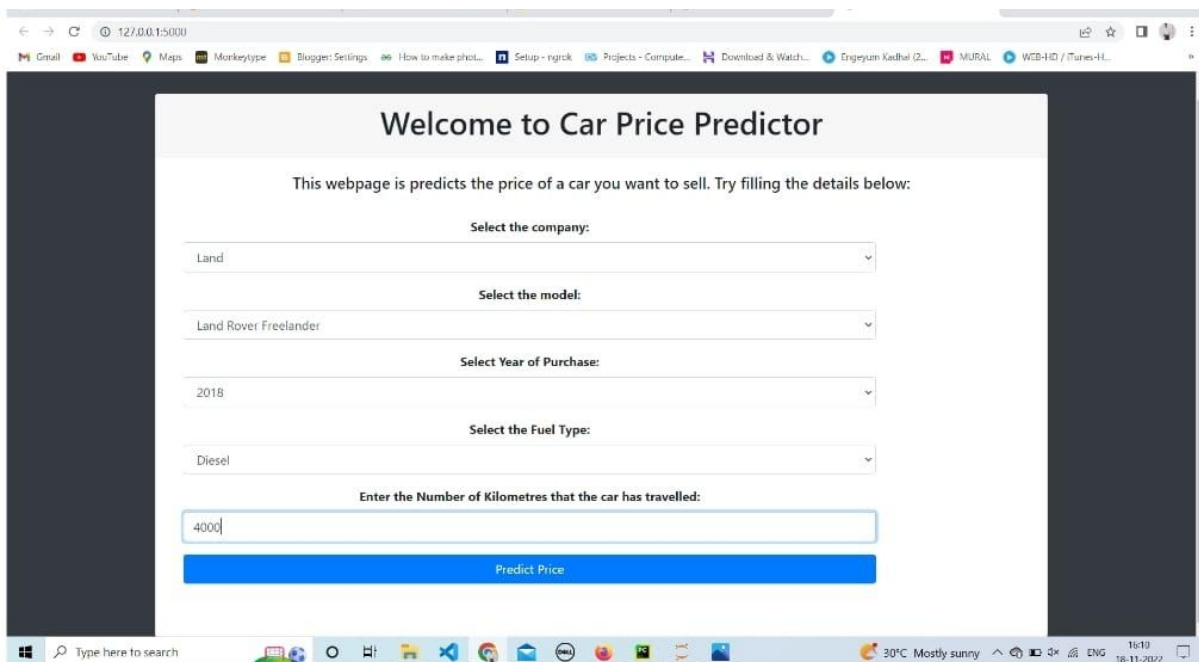
```
File Edit Selection Find View Goto Tools Project Preferences Help
index.html
33 <form method="post" accept-charset="utf-8" name="Modelform">
34   <div class="col-md-10 form-group" style="text-align: center">
35     <label><b>Select the company:</b></label><br>
36     <select class="selectpicker form-control" id="company" name="company" required="1"
37       onchange="load_car_models(this.id,'car_models')">
38       {% for company in companies %}
39         <option value="{{ company }}">{{ company }}</option>
40       {% endfor %}
41     </select>
42   </div>
43   <div class="col-md-10 form-group" style="text-align: center">
44     <label><b>Select the model:</b></label><br>
45     <select class="selectpicker form-control" id="car_models" name="car_models" required="1">
46     </select>
47   </div>
48   <div class="col-md-10 form-group" style="text-align: center">
49     <label><b>Select Year of Purchase:</b></label><br>
50     <select class="selectpicker form-control" id="year" name="year" required="1">
51       {% for year in years %}
52         <option value="{{ year }}">{{ year }}</option>
53       {% endfor %}
54     </select>
55   </div>
56   <div class="col-md-10 form-group" style="text-align: center">
57     <label><b>Select the Fuel Type:</b></label><br>
58     <select class="selectpicker form-control" id="fuel_type" name="fuel_type" required="1">
59       {% for fuel in fuel_type %}
60         <option value="{{ fuel }}">{{ fuel }}</option>
61       {% endfor %}
62     </select>
63   </div>
64   <div class="col-md-10 form-group" style="text-align: center">
65     <label><b>Enter the Number of Kilometres that the car has travelled:</b></label><br>
66     <input type="text" class="form-control" id="kilo_driven" name="kilo_driven">

```

```
File Edit Selection Find View Goto Tools Project Preferences Help
index.html
99   var newOption= document.createElement("option");
100   newOption.value="{{ model }}";
101   newOption.innerHTML="{{ model }}";
102   car_model.options.add(newOption);
103   {% endif %}
104   {% endfor %}
105 }
106 {% endfor %}
107 }
108
109 function form_handler(event) {
110   event.preventDefault(); // Don't submit the form normally
111 }
112 function send_data()
113 {
114   document.querySelector('form').addEventListener("submit",form_handler);
115
116   var fd=new FormData(document.querySelector('form'));
117
118   var xhr= new XMLHttpRequest({mozSystem: true});
119
120   xhr.open('POST','/predict',true);
121   document.getElementById('prediction').innerHTML="Wait! Predicting Price.....";
122   xhr.onreadystatechange = function(){
123     if(xhr.readyState == XMLHttpRequest.DONE){
124       document.getElementById('prediction').innerHTML="Prediction: ₹"+xhr.responseText;
125     }
126   }
127   };
128
129   xhr.onload= function(){};
130
131   xhr.send(fd);
132 }

```

```
File Edit Selection Find View Goto Tools Project Preferences Help
index.html
120
121
122     };
123
124     xhr.onload= function(){};
125
126     xhr.send(fd);
127 }
128 </script>
129
130 <!-- jQuery first, then Popper.js, then Bootstrap JS -->
131 <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
132       integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MV8nE+IbbVYUew+OrCXaRkfj"
133       crossorigin="anonymous"></script>
134 <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
135       integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
136       crossorigin="anonymous"></script>
137 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"
138       integrity="sha384-OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI"
139       crossorigin="anonymous"></script>
140 </body>
141 </html>
```



## Embedding Story to web application:

main.py - D:\IBM project\Project Development Phase\Sprint 2\main.py (3.11.0)

File Edit Format Run Options Window Help

```
from flask import Flask, render_template, request, redirect
from flask_cors import CORS
import pickle
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
app=Flask(__name__)
cors=CORS(app)
model=pickle.load(open('LinearRegressionModel.pkl','rb'))
car=pd.read_csv('Cleaned_Car_data.csv')

@app.route('/',methods=['GET','POST'])
def index():
    companies=sorted(car['company'].unique())
    car_models=sorted(car['name'].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)

def cross_origin(**kwargs):
    _options = kwargs

def decorator(f):
    LOG.debug("Enabling %s for cross_origin using options:%s", f, _options)
    if _options.get('automatic_options', True):
        f.required_methods = getattr(f, 'required_methods', set())
        f.required_methods.add('OPTIONS')
        f.provide_automatic_options = False

    def wrapped_function(*args, **kwargs):
        # Handle setting of Flask-Cors parameters
        options = get_cors_options(current_app, _options)

        if options.get('automatic_options') and request.method == 'OPTIONS':
            resp = current_app.make_default_options_response()
        else:
            resp = make_response(f(*args, **kwargs))

        set_cors_headers(resp, options)
        setattr(resp, FLASK_CORS_EVALUATED, True)
        return resp

    return update_wrapper(wrapped_function, f)
return decorator
```

```

    return render_template('index.html', companies=companies, car_models=car_models, years=year, fuel_types=fuel_type)

def cross_origin(**kwargs):
    _options = kwargs

def decorator(f):
    LOG.debug("Enabling %s for cross_origin using options:%s", f, _options)
    if _options.get('automatic_options', True):
        f.required_methods = getattr(f, 'required_methods', set())
        f.required_methods.add('OPTIONS')
        f.provide_automatic_options = False

    def wrapped_function(*args, **kwargs):
        # Handle setting of Flask-Cors parameters
        options = get_cors_options(current_app, _options)

        if options.get('automatic_options') and request.method == 'OPTIONS':
            resp = current_app.make_default_options_response()
        else:
            resp = make_response(f(*args, **kwargs))

        set_cors_headers(resp, options)
        setattr(resp, FLASK_CORS_EVALUATED, True)
        return resp

    return update_wrapper(wrapped_function, f)
    return decorator

@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')
    driven=request.form.get('kilo_driven')
    columns = ['name', 'company', 'year', 'kms_driven', 'fuel_type']
    data = np.array([car_model, company, year, driven, fuel_type])
    prediction=model.predict(pd.DataFrame(columns,data.reshape(1,5)))
    print(prediction)

    return str(np.round(prediction[0],2))

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

```

This app predicts the price of a car you want to sell. Try filling the details below:

Select the company:

Datsun

Select the model:

Datsun Go Plus

Select Year of Purchase:

2010

Select the Fuel Type:

Diesel

Enter the Number of Kilometres that the car has travelled:

12000

Predict Price

Prediction: ₹186194.52