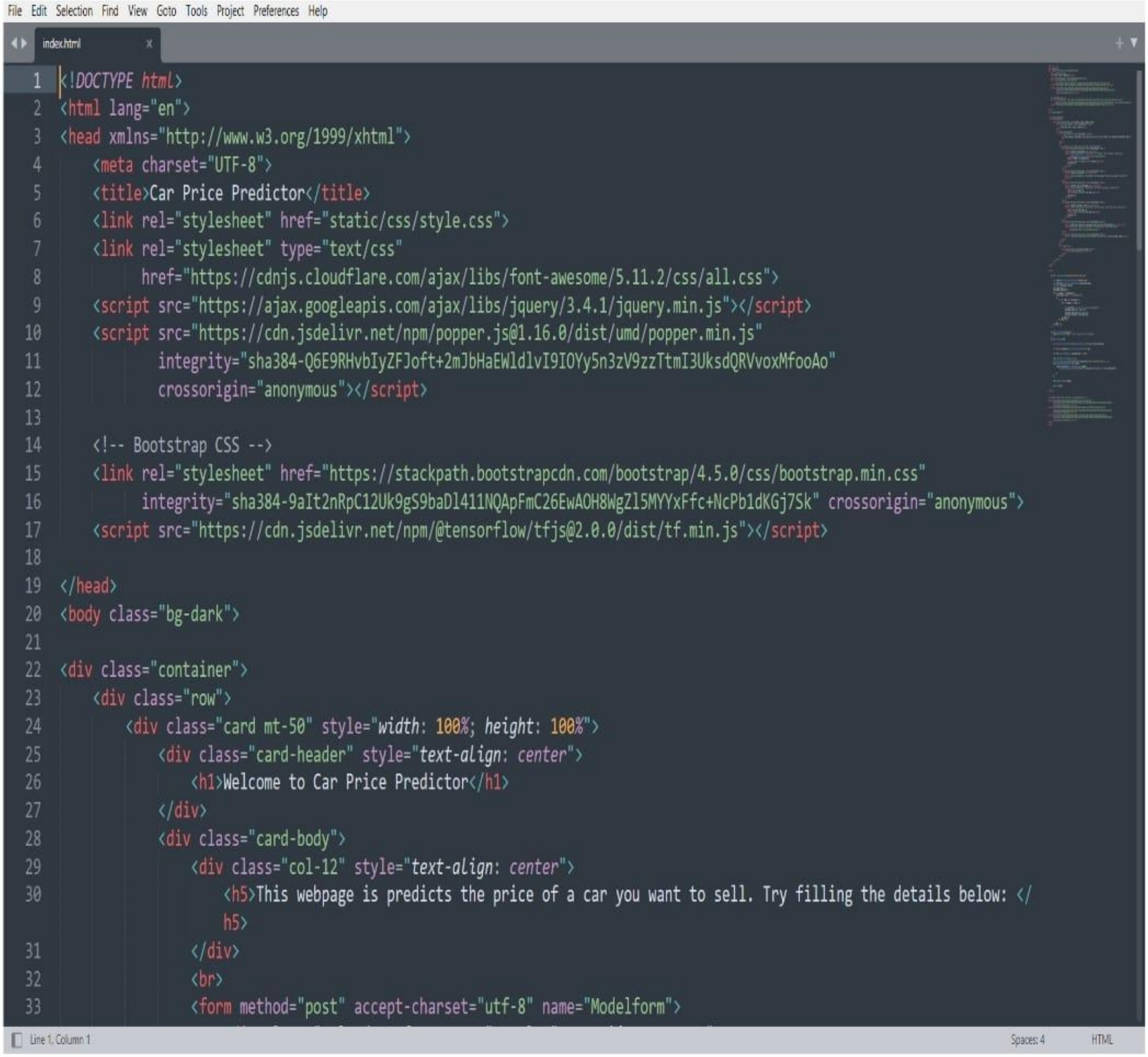
Project Development Phase
Sprint 4

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

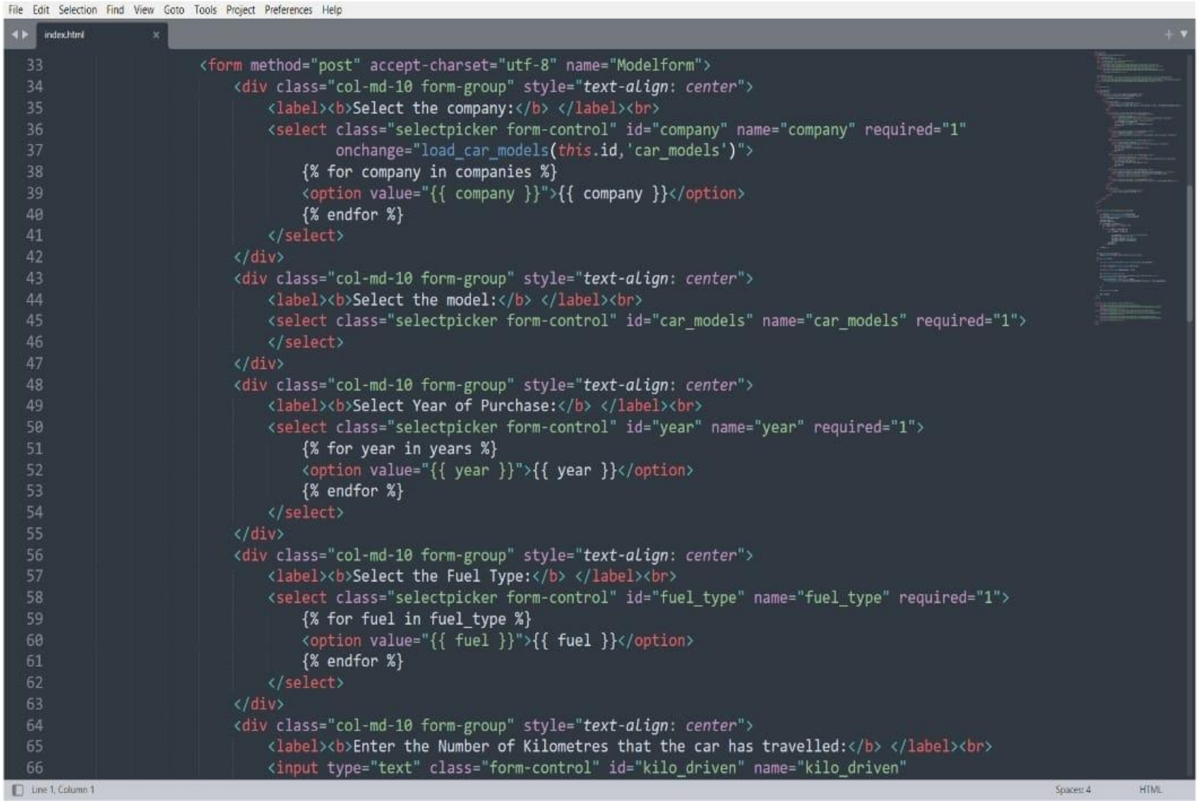
Sprint 4 tasks:

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

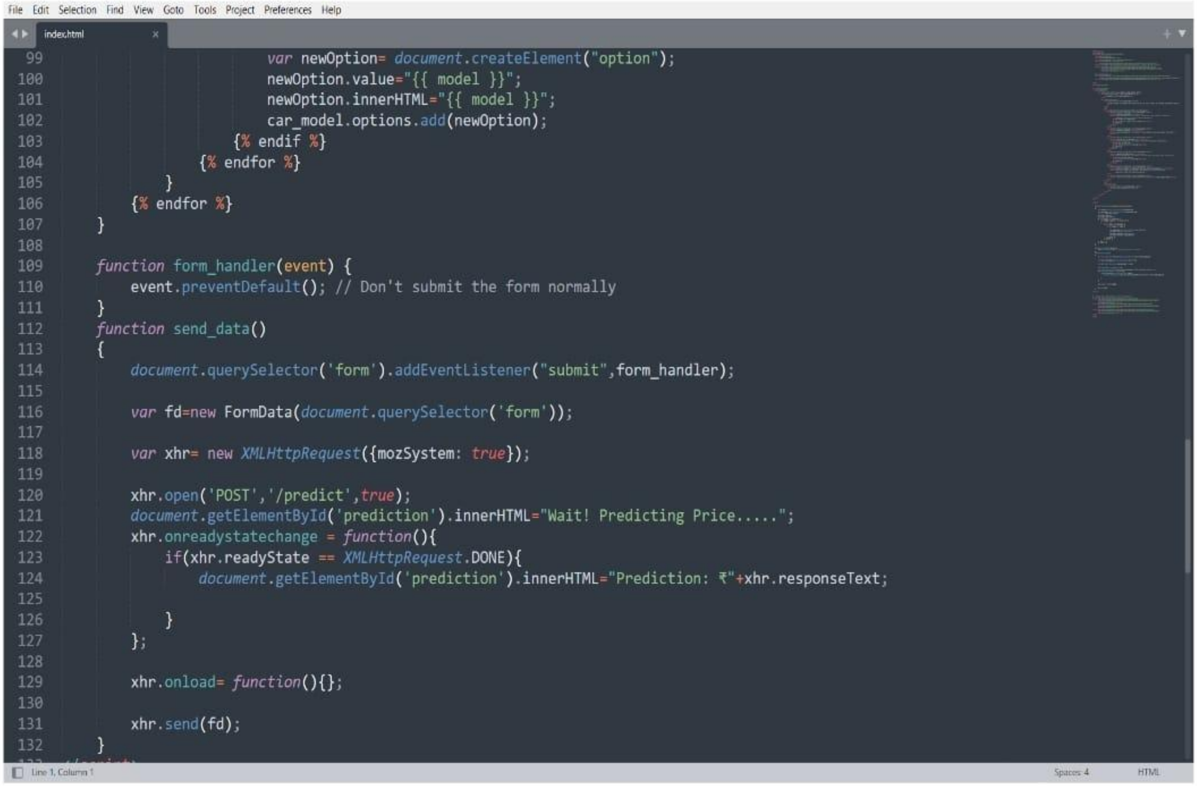
Creating web application:


File Edit Selection Find View Goto Tools Project Preferences Help
index.html
Line 1, Column 1
Spaces: 4
HTML

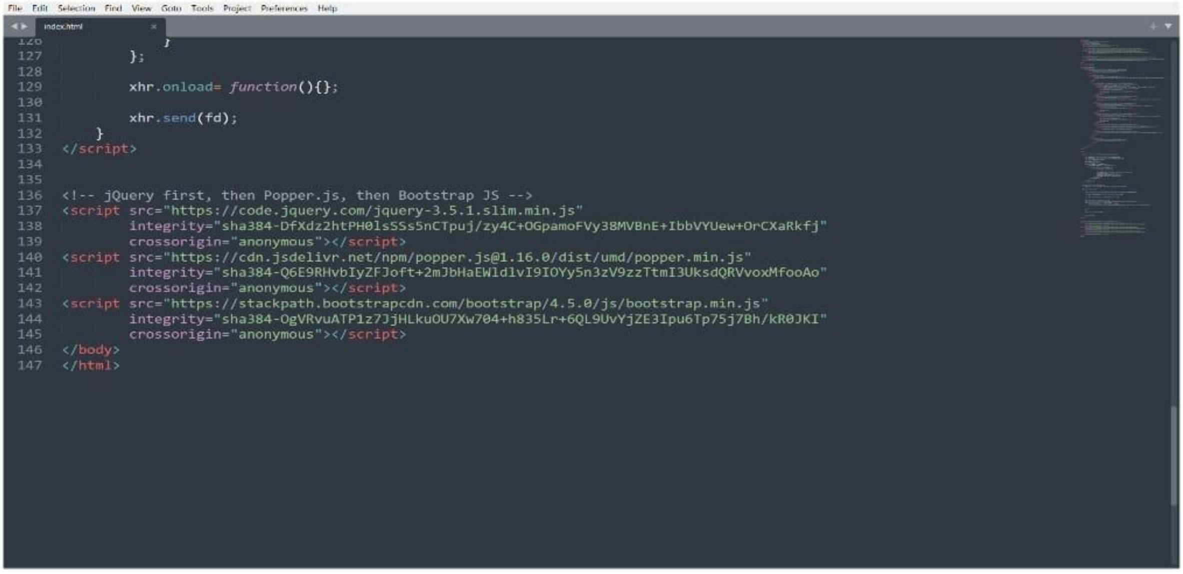
```html
<!DOCTYPE html>
<html lang="en">
<head xmlns="http://www.w3.org/1999/xhtml">
    <meta charset="UTF-8">
    <title>Car Price Predictor</title>
    <link rel="stylesheet" href="static/css/style.css">
    <link rel="stylesheet" type="text/css"
          href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.11.2/css/all.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
            integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
            crossorigin="anonymous"></script>

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
          integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous">
    <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@2.0.0/dist/tf.min.js"></script>

</head>
<body class="bg-dark">

<div class="container">
    <div class="row">
        <div class="card mt-50" style="width: 100%; height: 100%">
            <div class="card-header" style="text-align: center">
                <h1>Welcome to Car Price Predictor</h1>
            </div>
            <div class="card-body">
                <div class="col-12" style="text-align: center">
                    <h5>This webpage is predicts the price of a car you want to sell. Try filling the details below: </
                    h5>
                </div>
                <br>
                <form method="post" accept-charset="utf-8" name="Modelform">
```


File Edit Selection Find View Goto Tools Project Preferences Help
index.html
Line 1, Column 1
Spaces: 4
HTML

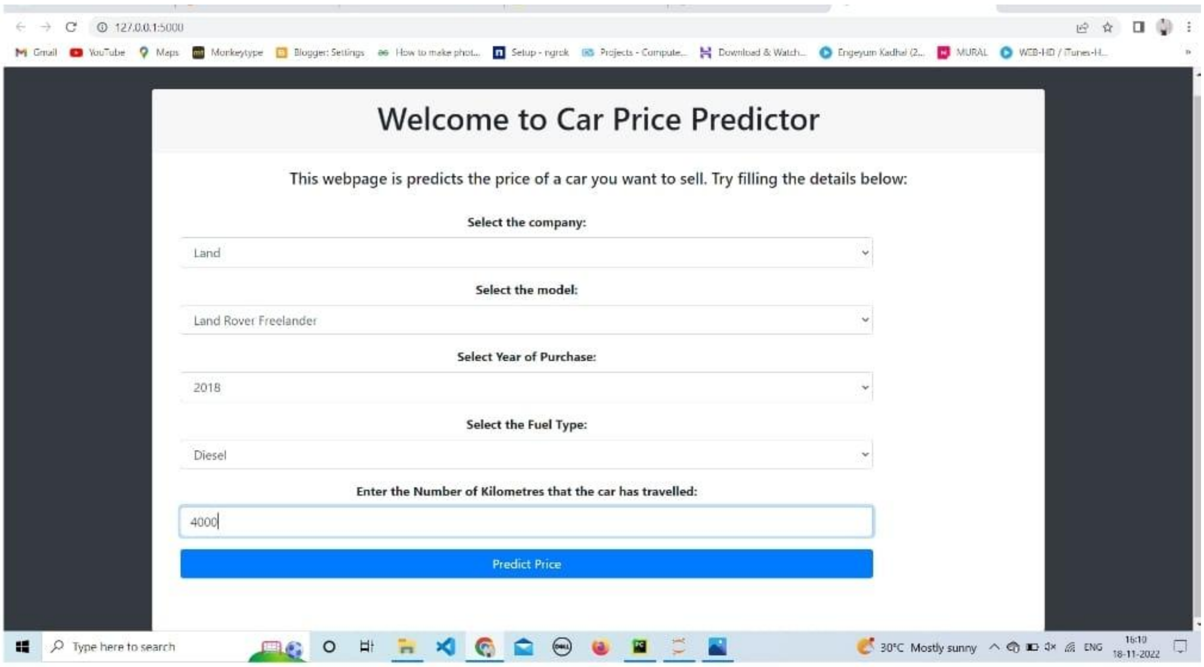
```
<form method="post" accept-charset="utf-8" name="Modelform">
    <div class="col-md-10 form-group" style="text-align: center">
        <label><b>Select the company:</b> </label><br>
        <select class="selectpicker form-control" id="company" name="company" required="1"
                onchange="load_car_models(this.id,'car_models')">
            {% for company in companies %}
            <option value="{{ company }}">{{ company }}</option>
            {% endfor %}
        </select>
    </div>
    <div class="col-md-10 form-group" style="text-align: center">
        <label><b>Select the model:</b> </label><br>
        <select class="selectpicker form-control" id="car_models" name="car_models" required="1">
        </select>
    </div>
    <div class="col-md-10 form-group" style="text-align: center">
        <label><b>Select Year of Purchase:</b> </label><br>
        <select class="selectpicker form-control" id="year" name="year" required="1">
            {% for year in years %}
            <option value="{{ year }}">{{ year }}</option>
            {% endfor %}
        </select>
    </div>
    <div class="col-md-10 form-group" style="text-align: center">
        <label><b>Select the Fuel Type:</b> </label><br>
        <select class="selectpicker form-control" id="fuel_type" name="fuel_type" required="1">
            {% for fuel in fuel_type %}
            <option value="{{ fuel }}">{{ fuel }}</option>
            {% endfor %}
        </select>
    </div>
    <div class="col-md-10 form-group" style="text-align: center">
        <label><b>Enter the Number of Kilometres that the car has travelled:</b> </label><br>
        <input type="text" class="form-control" id="kilo_driven" name="kilo_driven"
```


File Edit Selection Find View Goto Tools Project Preferences Help
index.html
Line 1, Column 1
Spaces: 4
HTML

```
                    var newOption= document.createElement("option");
                    newOption.value="{{ model }}";
                    newOption.innerHTML="{{ model }}";
                    car_model.options.add(newOption);
                {% endif %}
            {% endfor %}
        }
    {% endfor %}
}

function form_handler(event) {
    event.preventDefault(); // Don't submit the form normally
}
function send_data()
{
    document.querySelector('form').addEventListener("submit",form_handler);

    var fd=new FormData(document.querySelector('form'));

    var xhr= new XMLHttpRequest({mozSystem: true});

    xhr.open('POST','/predict',true);
    document.getElementById('prediction').innerHTML="Wait! Predicting Price.....";
    xhr.onreadystatechange = function(){
        if(xhr.readyState == XMLHttpRequest.DONE){
            document.getElementById('prediction').innerHTML="Prediction: ₹"+xhr.responseText;

        }
    };

    xhr.onload= function(){};

    xhr.send(fd);
}
```


File Edit Selection Find View Goto Tools Project Preferences Help
index.html

```
            }
        };

        xhr.onload= function(){};

        xhr.send(fd);
    }
</script>


<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
        integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
        crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
        integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
        crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"
        integrity="sha384-OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI"
        crossorigin="anonymous"></script>
</body>
</html>
```



Welcome to Car Price Predictor

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

Select the company:

Land

Select the model:

Land Rover Freelander

Select Year of Purchase:

2018

Select the Fuel Type:

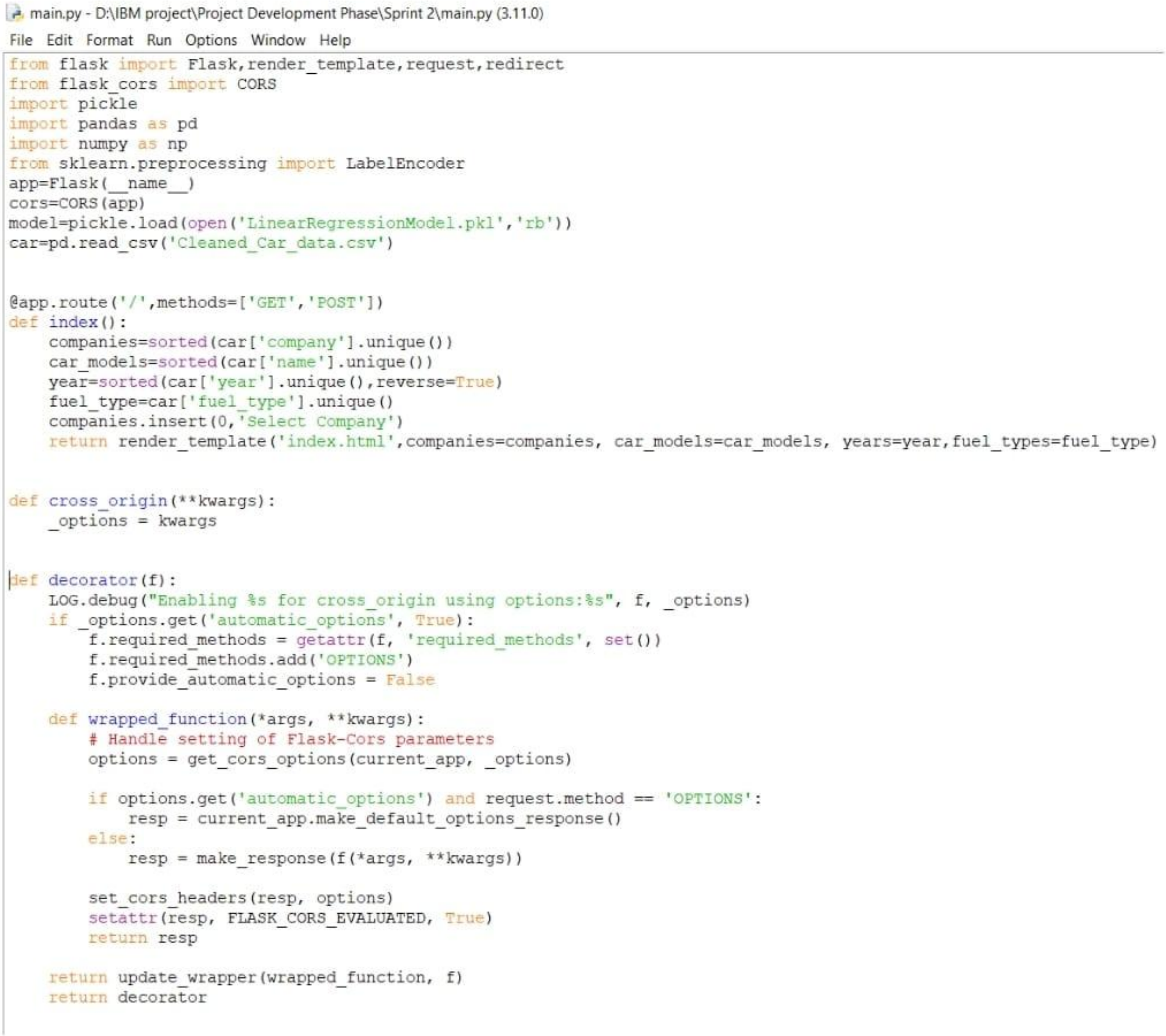
Diesel

Enter the Number of Kilometres that the car has travelled:

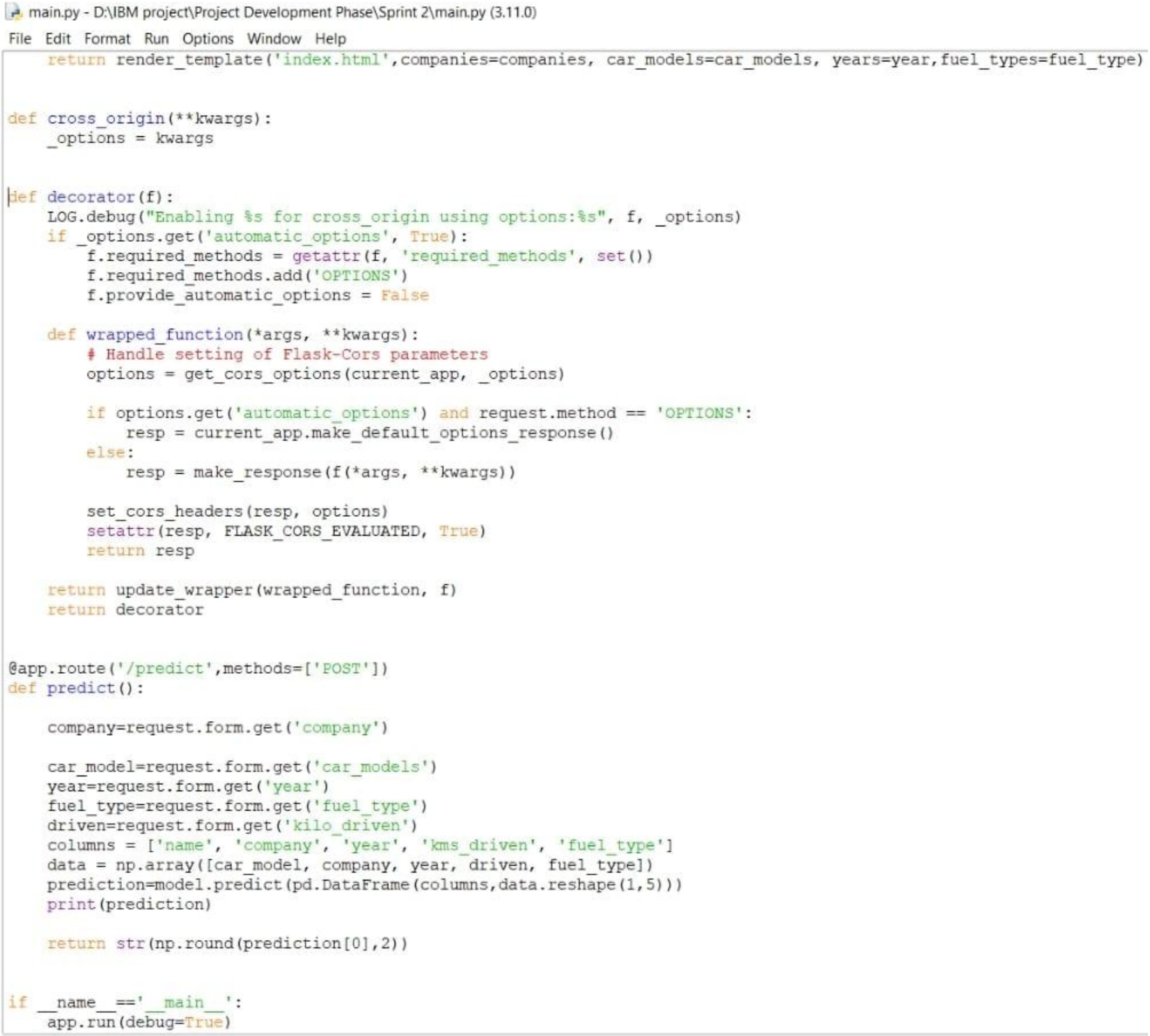
4000

Predict Price

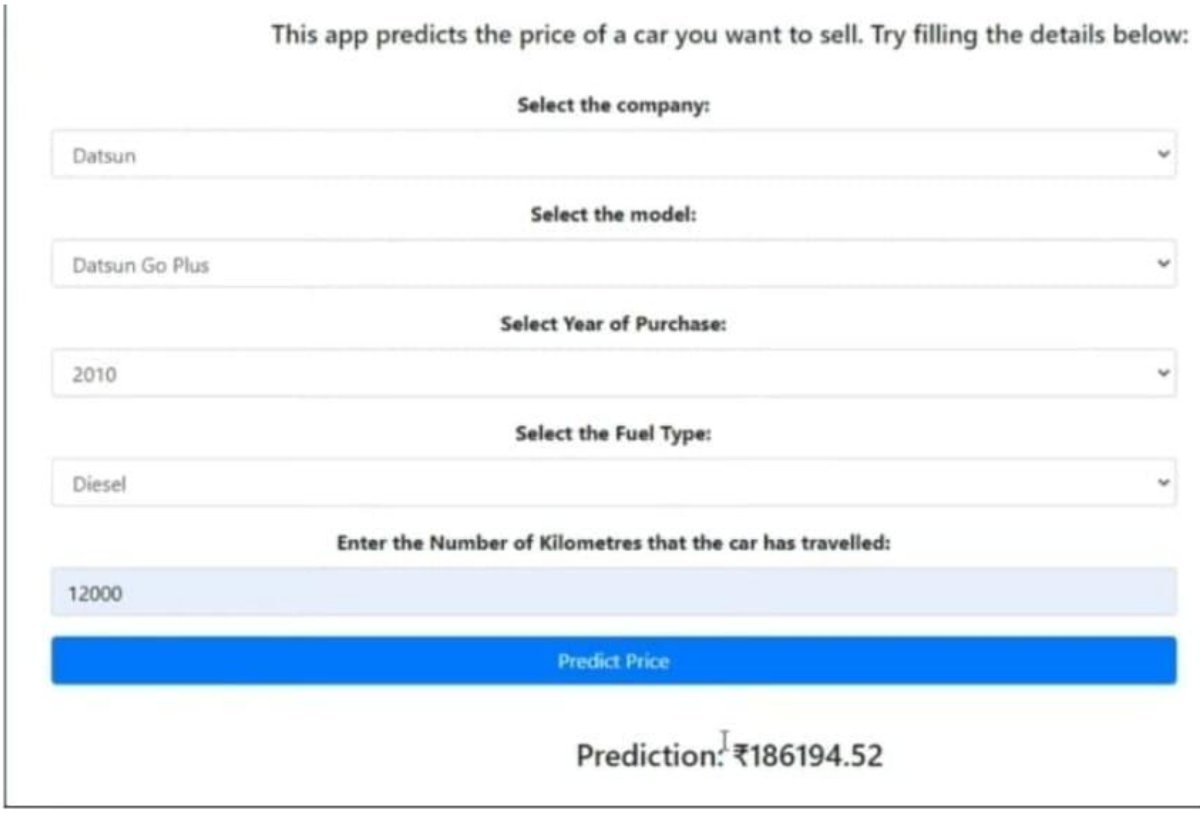
Embedding Story to web application:


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
```


main.py - D:\IBM project\Project Development Phase\Sprint 2\main.py (3.11.0)
File Edit Format Run Options Window Help

```
    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

Prediction: ₹186194.52