# PROJECT REPORT

**PROJECT TITLE** : **Car Resale value Prediction**

**TEAM ID : PNT2022TMID45616**

**TEAM MEMBERS :** Aravindhan A(TEAMLEAD)
Rakesh C
Hemanth Kumar M
Akash Yovel A
Vigneswar C

# 1.INTRODUCTION

## 1.1 Project overview

The Car Valuation Tool is a free tool designed to help you get the estimated resale value of your car within seconds.Our automobile valuation algorithm is real-time updated, so it keeps up with the most recent modifications and market trends. However, the amounts displayed during the online assessment are only estimates and might alter when the retailer inspects your automobile. You don't even need to register to have your automobile valued; all you need to do is provide some basic information about it, such as its make, model, amount of miles driven, city of residence, and contact information.

## 1.2 Purpose

In 2019, the Indian used automobile resale industry was valued at $24.2 billion USD. There is a critical need to close this gap between sellers and buyers due to the enormous demand for used automobiles and the shortage of professionals who can evaluate the proper valuation. The goal of this research is to create a system that can impartially forecast a car's resale value based on little information such as the number of miles travelled and the year of purchase.The process of determining the current used automobile pricing in a certain location is known as used car value. By selecting the brand, model, year, trim, and the number of kilometers travelled, a user of OBV may quickly determine the used car's price. The value of a used automobile is based on a number of variables, including its state right now, when it was bought, etc. Used automobile valuation will never have a precise price; instead, it will always fall within a reasonable price range.

# 2.LITERATURE SURVEY

## 2.1Existing problem

Car Resale value prediction is one of the best to sell our in this market for an best and better price.Rather than giving our car to an less price , the customer those who uses the car will be benifitted and the seller will also be benefitted.The goal of this research is to create a system that can impartially forecast a car's resale value based on little information such as the number of miles travelled and the year of purchase.You don't even need to register to have your automobile valued; all you need to do is provide some basic information about it, such as its make, model, amount of miles driven, city of residence, and contact information.

## 2.2 References

[1] Pudaruth Sameerchand , Pudaruth Sameerchand , Predicting the price of Used Car Using Machine Learning Techniques

[2] Enis gegic, Becir ,Isakovic, Dino Keco, ,Zerina Masetic,Jasmin Kevric Car Price Prediction Using Machine Learning

[3] Ning sun,Hongxi Bai,Yuxia Geng,Huizhu Shi Price Evaluation model in second hand car system

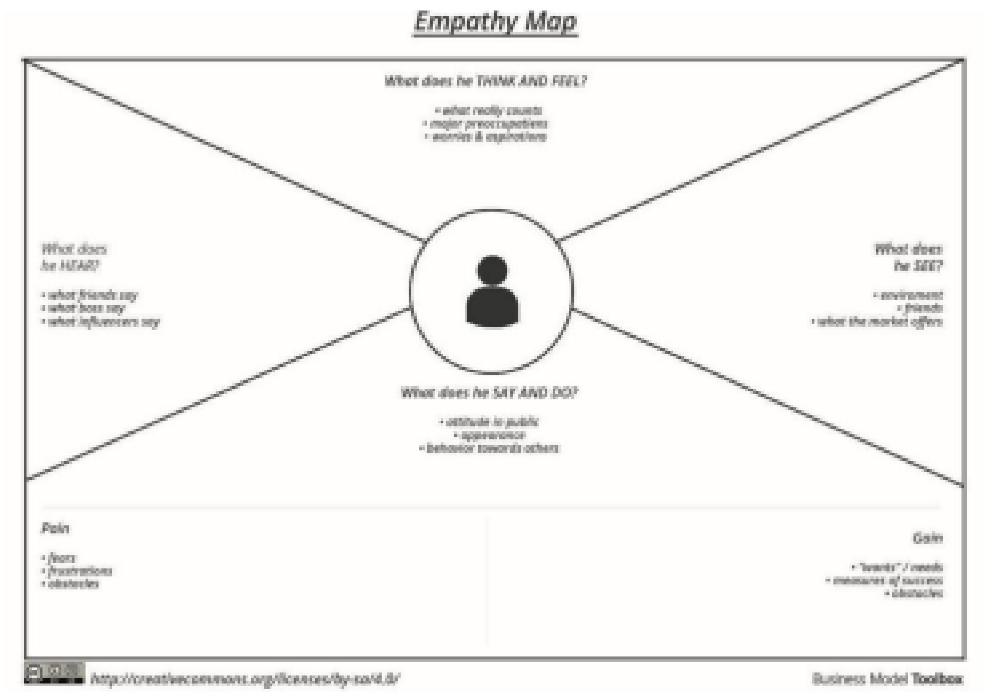
[4] Doan Van Thai, Luong Ngoc Son, Pham Vu Tien, Nguyen Nhat Anh, Nguyen Thi Ngoc Anh Prediction car prices using qualify qualitative data and knowledge-based system

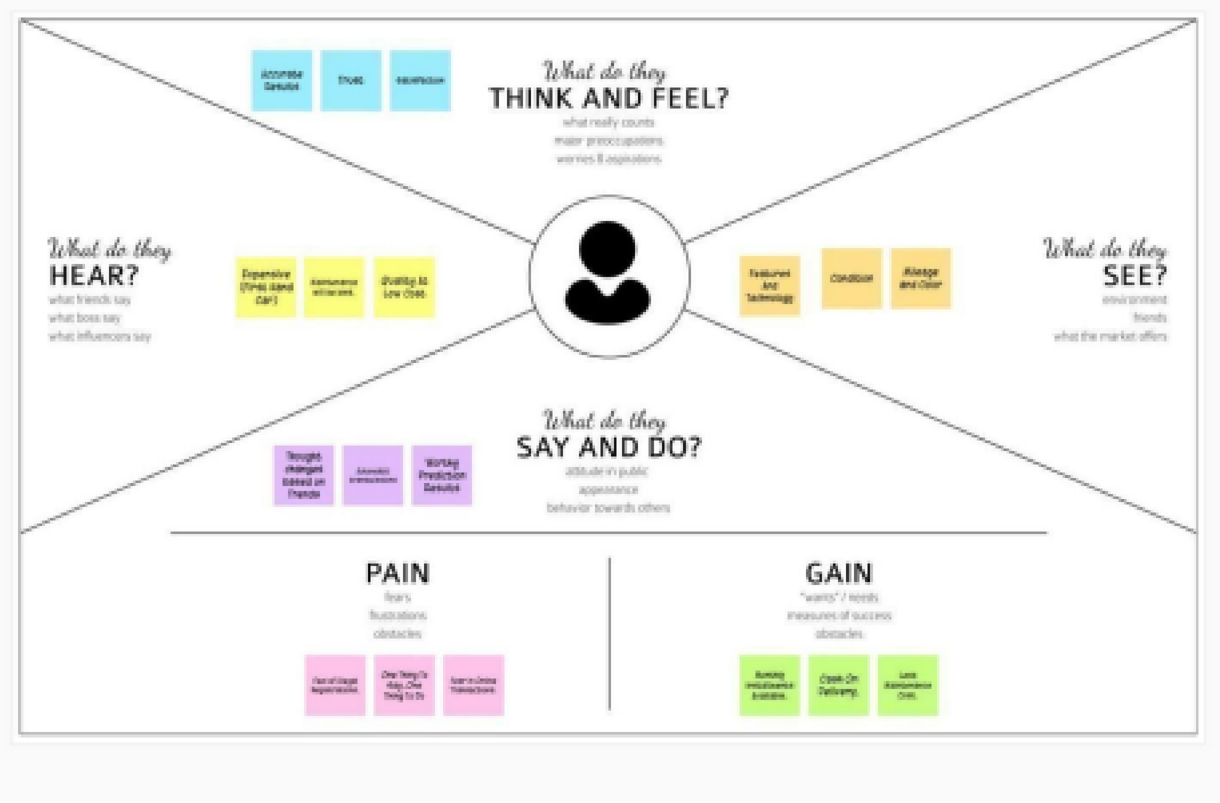
## 2.3 Problem Statement Definition

Car Resale value prediction is used to predict the value of the used cars to an reasonable price which satisfies the customer.

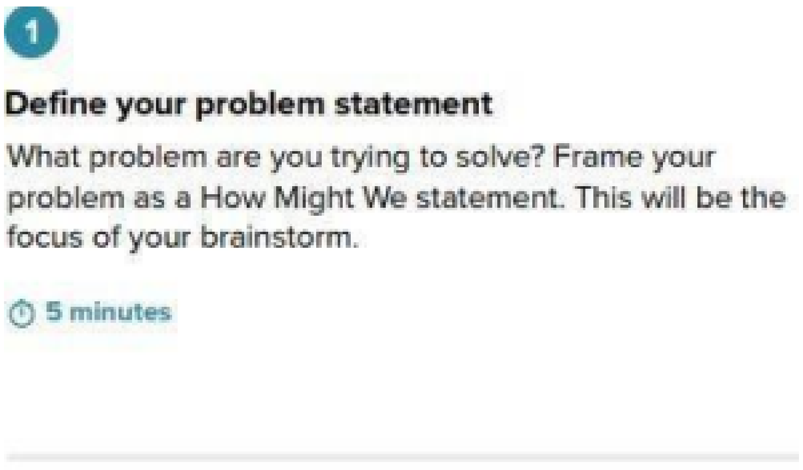
# 3 IDEATIOIN AND PROPOSED SOLUTION

## 3.1 Empathy Map Canvas





## 3.2 Ideation & Brainstroming

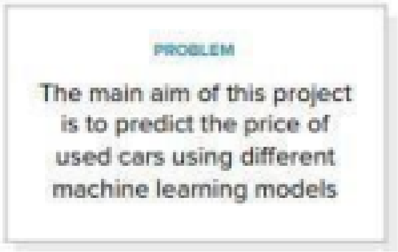


1

**Define your problem statement**

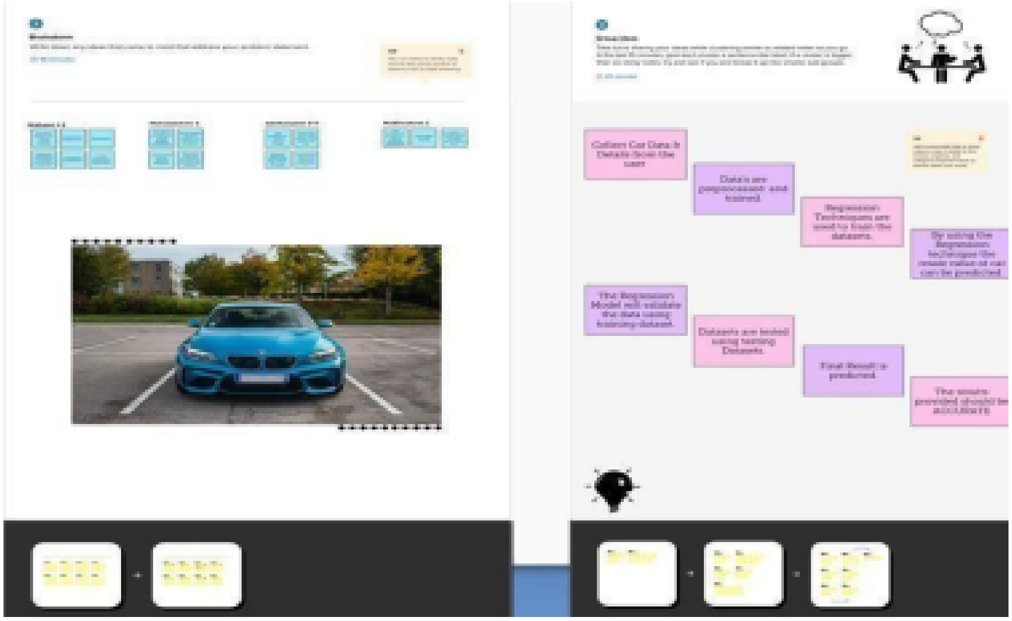
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes



PROBLEM

The main aim of this project is to predict the price of used cars using different machine learning models



Collect Car Data & Details from the user

Data's are preprocessed and trained

Regression Techniques are used to train the datasets

By using the Regression technique the resale value of car can be predicted

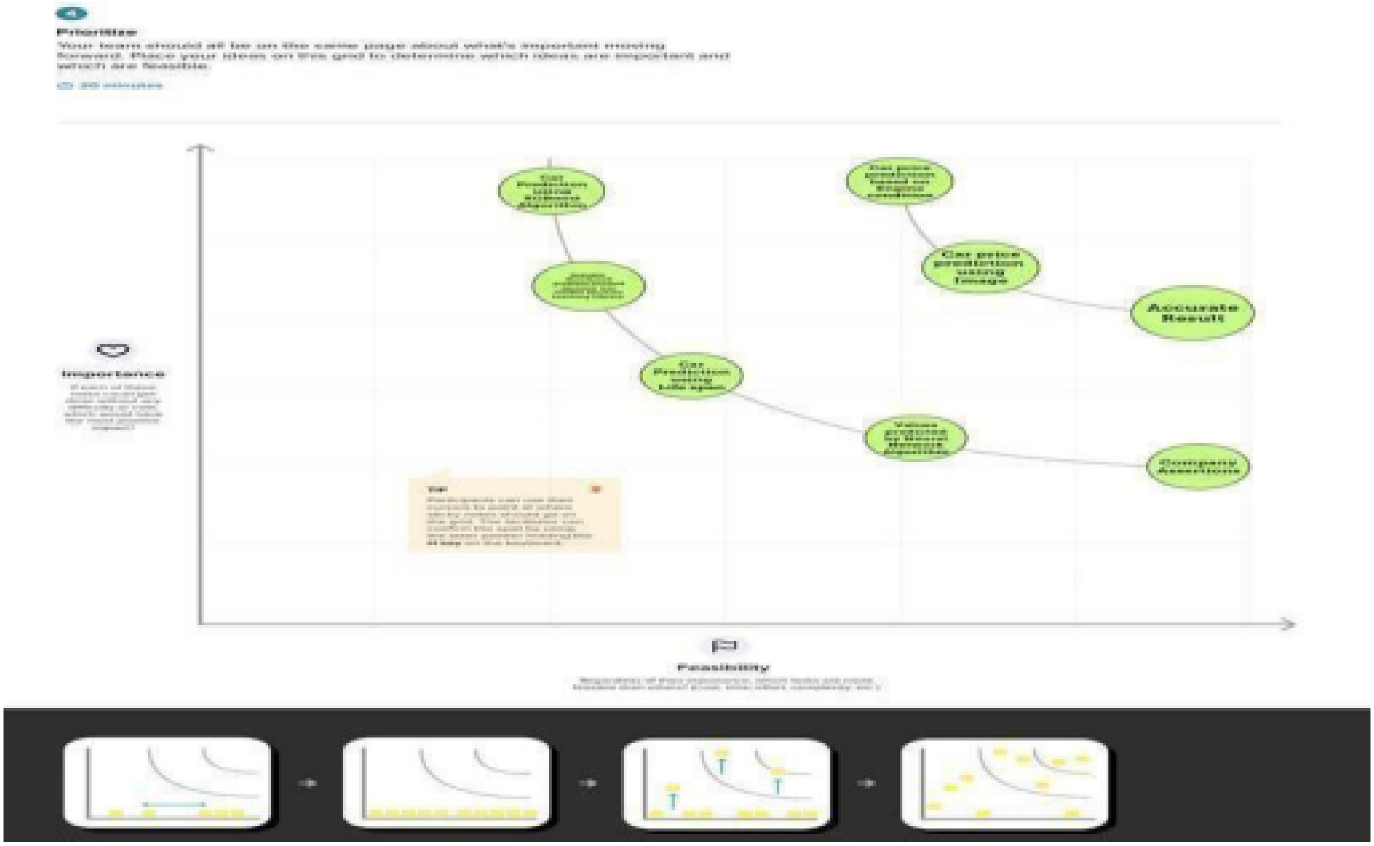
The Regression Model will validate the data using training dataset

Datasets are tested using testing Datasets

Final Result is predicted

The results provided should be ACCURATE

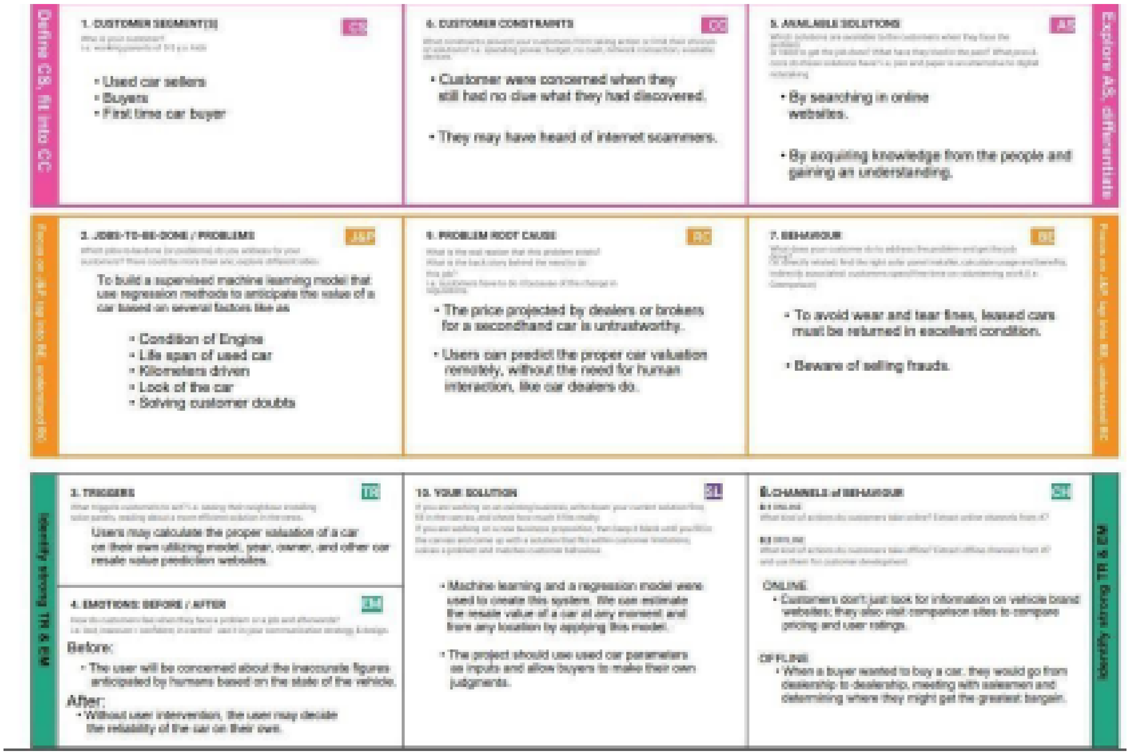
Idea prioritation :


Prioritize
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.
20 minutes
Car Prediction using
Car price prediction using Image
Accurate Result
Car Prediction using Life span
Values predicted
Company Assertions
Importance
Tip
Feasibility

## 3.3 Proposed Solution

| S.No. | Parameter | Description |
|---|---|---|
| 1. | Problem Statement(Problem to be solved) | Topredict the resale value ofsecond hand car orused car considering itsfeatures. |
| 2. | Idea / Solution description | To develop a Machine learning algorithm which predicts the resale value of any used car which is shown in web design. |
| 3. | Novelty / Uniqueness | The model predicts the resale value of carwith high accuracy. |
| 4. | Social Impact/ Customer Satisfaction | A good platform with more reliability and portability. |
| 5. | Business Model (Revenue Model) | Themodel deployed in cloud so anyone can access it anywhere and anytime. |
| 6. | Scalability ofthe Solution | Itis a web page model so it can be viewed and accessedin both computer as well as mobile phones. |

## 3.4 Problem Solution Fit



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

- Used car sellers
- Buyers
- First time car buyer

6. CUSTOMER CONSTRAINTS CC

- Customer were concerned when they still had no clue what they had discovered.
- They may have heard of internet scammers.

5. AVAILABLE SOLUTIONS AS

- By searching in online websites.
- By acquiring knowledge from the people and gaining an understanding.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

To build a supervised machine learning model that use regression methods to anticipate the value of a car based on several factors like as

- Condition of Engine
- Life span of used car
- Kilometers driven
- Look of the car
- Solving customer doubts

9. PROBLEM ROOT CAUSE RC

- The price projected by dealers or brokers for a secondhand car is untrustworthy.
- Users can predict the proper car valuation remotely, without the need for human interaction, like car dealers do.

7. BEHAVIOUR BE

- To avoid wear and tear fines, leased cars must be returned in excellent condition.
- Beware of selling frauds.

Identify strong TR & EM

3. TRIGGERS TR

Users may calculate the proper valuation of a car on their own utilizing model, year, owner, and other car resale value prediction websites.

4. EMOTIONS: BEFORE / AFTER EM

Before:

- The user will be concerned about the inaccurate figures anticipated by humans based on the state of the vehicle.

After:

- Without user intervention, the user may decide the reliability of the car on their own.

10. YOUR SOLUTION SL

- Machine learning and a regression model were used to create this system. We can estimate the resale value of a car at any moment and from any location by applying this model.
- The project should use used car parameters as inputs and allow buyers to make their own judgments.

8. CHANNELS of BEHAVIOUR CH

ONLINE

- Customers don't just look for information on vehicle brand websites; they also visit comparison sites to compare pricing and user ratings.

OFFLINE

- When a buyer wanted to buy a car, they would go from dealership to dealership, meeting with salesmen and determining where they might get the greatest bargain.

Identify strong TR & EM

# 4 REQUIREMENT ANALYSIS

## 4.1 Functional requirement

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|---|---|---|
| FR-1 | User Registration | Registration through Website |
| FR-2 | User Confirmation | Confirmation via Website |
| FR-3 | Car Registration | Registration through Website |
| FR-4 | Car Information | Getting the car details through Website |
| FR-5 | Value Prediction | Shows the resale value of the car through website |

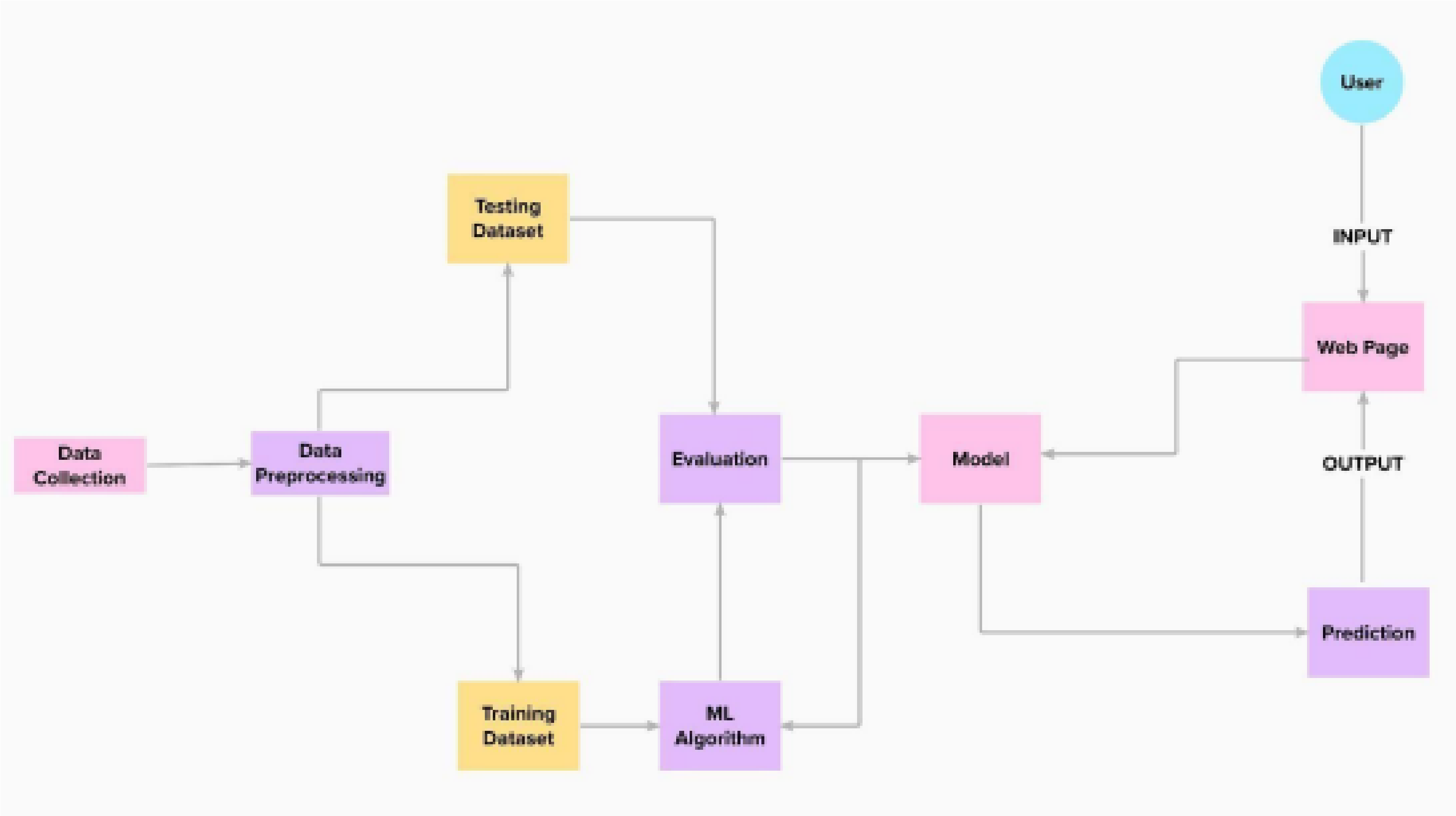
## 4.2 Non-Functional requirement

| FR No. | Non-Functional Requirement | Description |
|---|---|---|
| NFR-1 | **Usability** | The model predicts the resale value of the car with more accuracy. |
| NFR-2 | **Security** | Protect the user information as well as their car details. |
| NFR-3 | **Reliability** | The model performs consistently well and also it begins trust to the user. |
| NFR-4 | **Performance** | The model performance has high accuracy and with portable from one machine to another machine. |
| NFR-5 | **Availability** | The model can be available anywhere at anytime. |

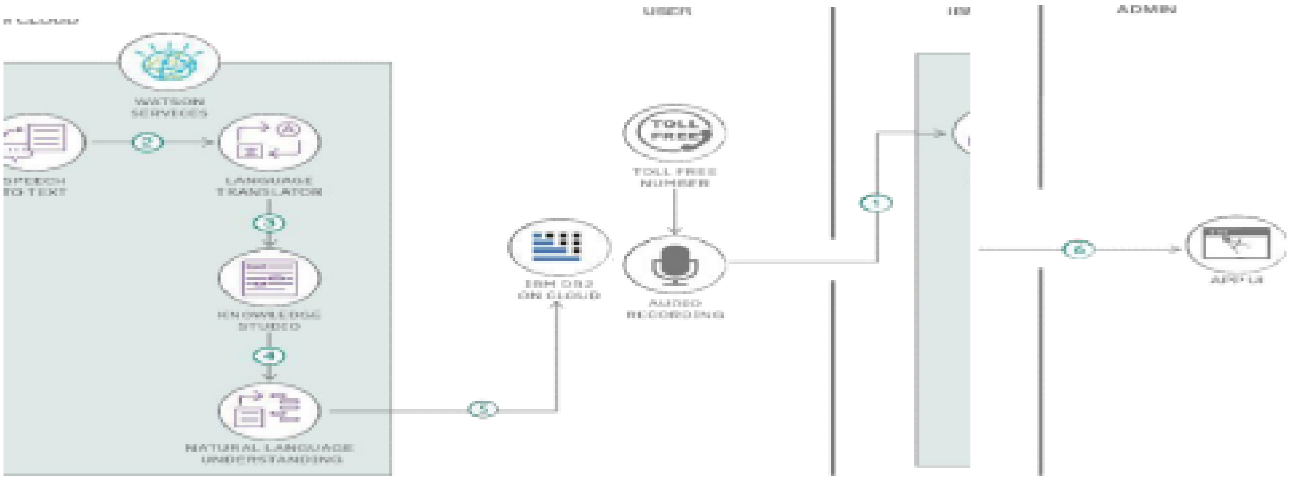
|  |  |  |
|---|---|---|
| NFR-6 | **Scalability** | The model can be viewed and accessed in both computer as well as mobile phone. |

# 5 PROJECT DESIGN

## 5.1 Data Flow Diagram



## 5.2 Solution & Technical Architecture



## User Stories

| User Type | Functional requirement | User story | User story/task | Acceptance | Priority | Release |
|---|---|---|---|---|---|---|

|  |  | number |  | criteria |  |  |
|---|---|---|---|---|---|---|
| Customer (Mobile user, Web user, Care executive, Administrator) | Registration | USN-1 | As a user, I can register for the application by entering my mail, password, and confirming my password | I can access my account/ dashboard | High | Sprint-1 |
|  |  | USN-2 | As a user, I will receive confirmation email once I have registered for the application | I can receive confirmation email & click confirm | High | Sprint-1 |
|  | Dashboard | USN-3 | As a user, I can register for the application through internet | I can register & access the dashboard with Internet login | Low | Sprint-2 |
|  |  | USN-4 | As a user, I can register for the application through Gmail | I can confirm the registration in Gmail | Medium | Sprint-1 |
|  | Login | USN-5 | As a user, I can log into the application by entering email & password | I can login with my id and password | High | Sprint-1 |

# 6 PROJECT PLANNING & SCHEDULING

## 6.1 Sprint Planning & Estimation

| Sprint | Functional Requirement (Epic) | User Story / Task | Story Point s | Priority | Team Members |
|---|---|---|---|---|---|
| Sprint-1 | Resources Initialization | We have to create and initializeaccounts in various public APIslike OpenWeatherMap API. | 1 | LOW | Devadharshini P, Durga G |
| Sprint-1 | Local Server/Software Run | Write a Python program thaoutputs results given the inputslike weather and locationthrough the software | 1 | MEDIU M | Devadharshini P, Durga G |
| Sprint-2 | Push the server/softwar e to cloud | Push the code from Sprint 1 to cloud so it can be accessed from anywhere | 2 | MEDIU M | RajaRajeswari R, Gowshe A |
| Sprint-3 | Hardware initialization | Integrate the hardware to beable to access the cloudfunctions and provide inputs tothe same. | 2 | HIGH | Gowshe A, Vasuki R |
| Sprint-4 | UI/UX Optimization & Debugging | Optimize all the shortcomings and provide better user experience. | 2 | LOW | Devadharshini P, RajaRajeswar iR |

## 6.2 Sprint Delivery Schedule

| Sprint | Functional Requirem ent(Epic) | User Story Number | User S |
|---|---|---|---|
| Sprint-3 | Home Page | USN-1 | Descrip resale p |
| Sprint-3 | Registration/Login | USN-2 | As a user, I car for the applicat byentering my username, ema number, and password a it. As a user, I c to the web app by entering my Username &pa |
| Sprint-3 | Form Page | USN-3 | As a user, I sub car details. |

| | | | | |
|---|---|---|---|---|
| Sprint-3 | Result | USN-4 | The pre price fo carmode displaye | RajaRajeswari R Gowshe A |
| Sprint-1 | Data collection andData preprocessing | USN-5 | Collect data anc data. | Vasuki R |
| Sprint-1 | Data collection andData preprocessing | USN-6 | Clean a analyse data to a duplicat | Durga G |

**Team Members**

Devadharshini P Durga G

| | | | | | | |
|---|---|---|---|---|---|---|
| Sprint-1 | Data collection andData preprocessing | USN-7 | Split the data into Dependent and Independentvariables | 6 | High | Durga G, RajaRajeswariR |
| Sprint-2 | Model Building | USN-8 | Build the model using a Random Forestregression to classify the data. | 9 | High | Gowshe A, Vasuki R |
| Sprint-2 | Model Building | USN-9 | Check the metrics | 7 | High | Durga G |
| Sprint-2 | Model Building | USN-10 | Save the model | 5 | High | Devadharshini P |

| | | | | | | |
|---|---|---|---|---|---|---|
| Sprint-4 | Deploy the model | USN-11 | Deployment of ML model using IBM WatsonStudio, object storage. | 13 | High | RajaRajeswari R |
| Sprint-4 | Integrate the webapp with the IBM model | USN-12 | Use flask for the integrationpurpose. | 8 | Medium | Devadharshini P |

# 7 CODING & SOLUTIONING

## 7.1 Feature 1

- IoT device
- IBM Watson Platform
- Node red
- Cloudant DB
- Web UI
- MIT App Inventor
- Python code

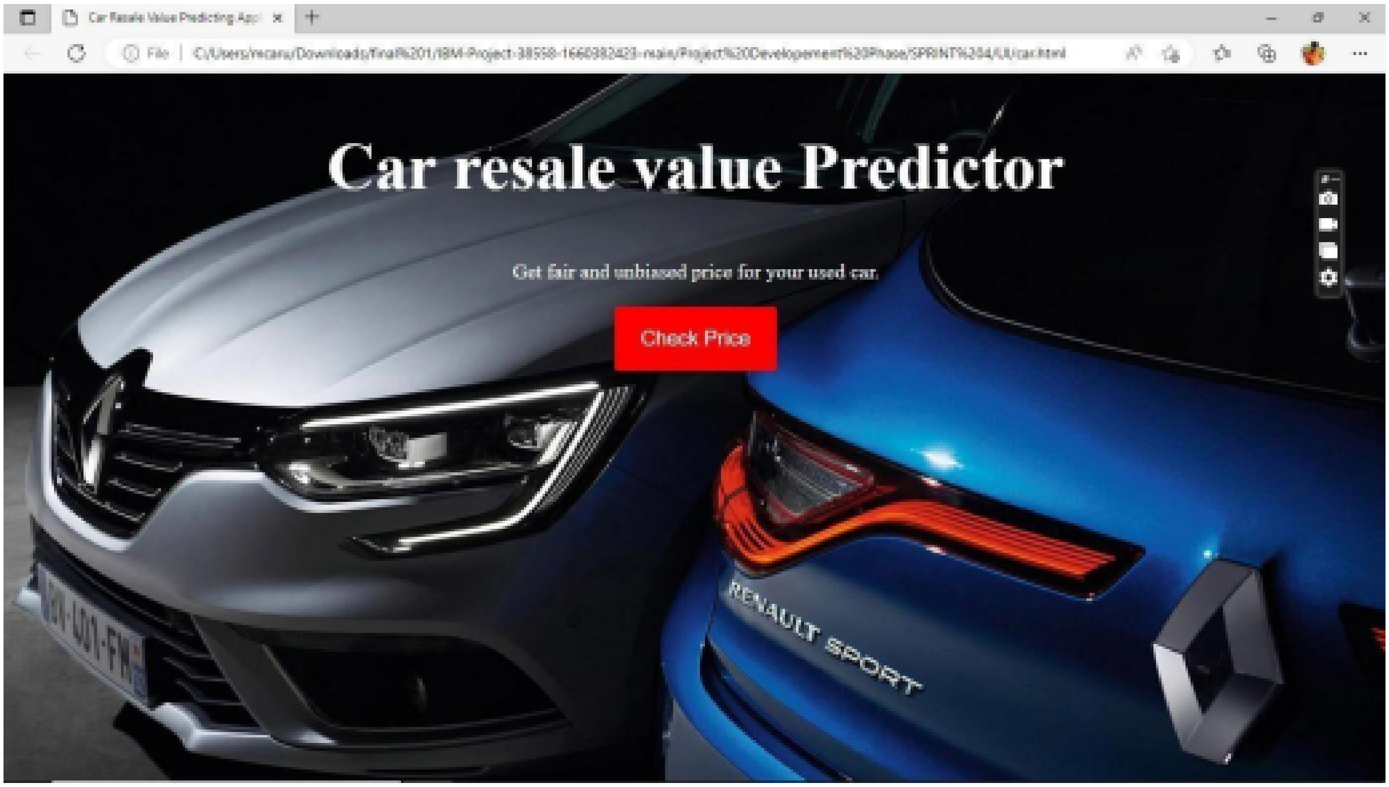
## 7.2 Feature 2

- Login
- Wokwi

# 8 TESTING AND RESULTS

## 8.1 Test Cases

### Test case 1:

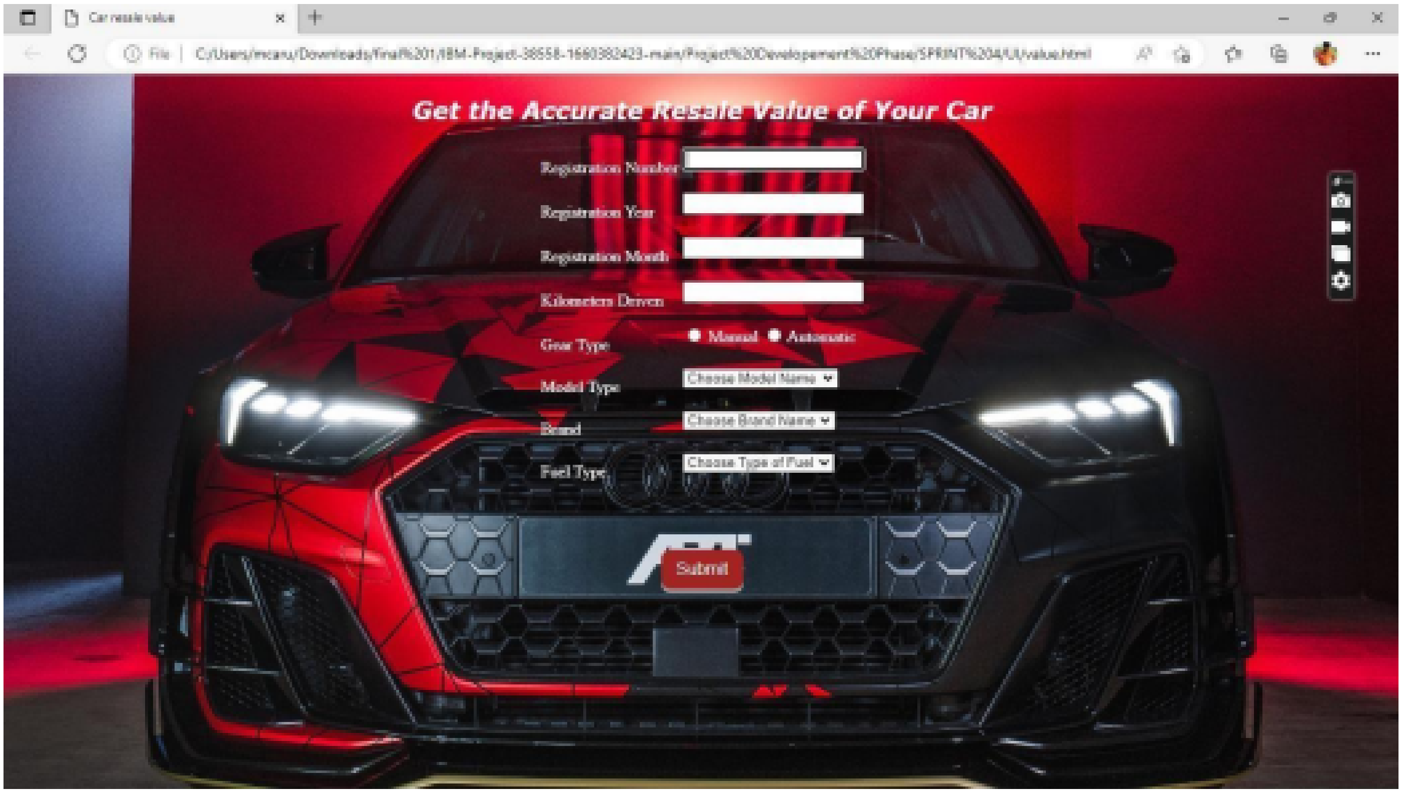

Car Resale Value Predicting App

C:/Users/mcanu/Downloads/final%201/IBM-Project-38558-1660382423-main/Project%20Developement%20Phase/SPRINT%204/UI/car.html

Car resale value Predictor

Get fair and unbiased price for your used car.

**Test case 2:**


Car resale value

C:/Users/mcanu/Downloads/final%201/IBM-Project-38558-1660382423-main/Project%20Developement%20Phase/SPRINT%204/UI/value.html

Get the Accurate Resale Value of Your Car

Registration Number

Registration Year

Registration Month

Kilometers Driven

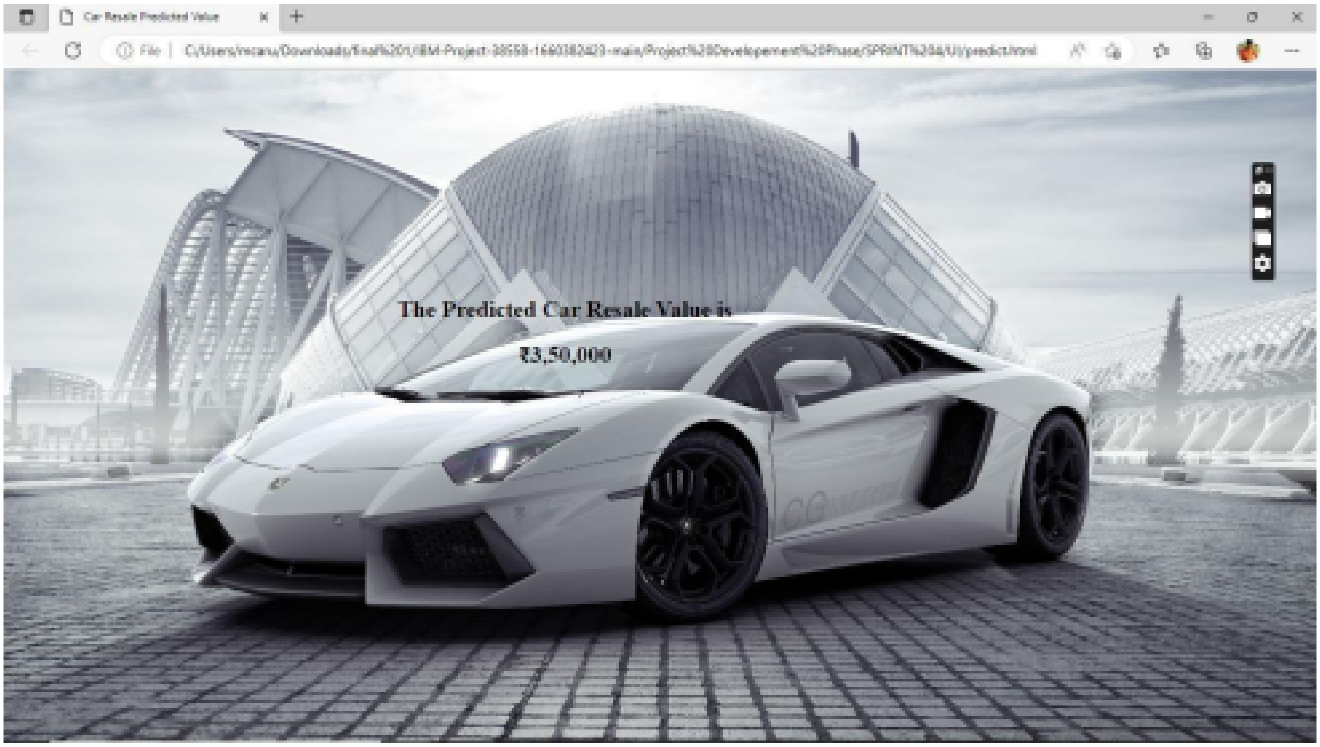
Gear Type ○ Manual ○ Automatic

Model Type Choose Model Name

Brand Choose Brand Name

Fuel Type Choose Type of Fuel

# 9 Result



Car Resale Predicted Value
File | C:/Users/mcanu/Downloads/final%201/IBM-Project-38558-1660382423-main/Project%20Developement%20Phase/SPRINT%204/UI/predict.html

The Predicted Car Resale Value is

₹3,50,000

**10.**

## 10.1 ADVANTAGES

- This will reduced installation cost.
- It will monitor 24/7.
- Very useful to sale the car for reasonable price

## 10.2 DISADVANTAGES

- Car Resale value can not be used by the person who doesn’t have access to the internet.
- Very hard to use for targeted range of people

## 11 CONCLUSION

Price prediction analyses a good or service based on its attributes, demand, and current market trends using an algorithm. The pricing is then adjusted by the programme at a level that it believes would both draw people and optimise sales. The method is known as price forecasting or predictive pricing in some quarters.

## 12 FUTURE SCOPE

When compared to February 2020, average prices were up 42.5% in September 2022. While it's possible that used vehicle prices have peaked, new car prices are expected to be high through the end of 2022. Prices are anticipated to drop for both newand used automobiles in 2023, by 2.5% to 5% for new cars and 10% to 20% for used cars.

# 13 APPENDIX

Source Code

HTML FILES:

home.hmtl

```
<!DOCTYPE html>
<html lang="en">
  <style>
     a:link, a:visited
       { background-color:
       white;color: black;
      border: 2px solid
      black;border-radius:
       25px; padding:
       10px 20px;
      text-align: center;
      text-decoration:
      none;display:
      inline-block;
     }

     a:hover, a:active
       { background-color:
      beige;color: black;
      text-decoration: none;
     }
```

```
      </style>

<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
                                                                        <link
                                                                                        rel="styleshe
et"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRx
T2MZw1T" crossorigin="anonymous">
  <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='style.css') }}" />

   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Home</title>
</head>

<body>
   <div class="col-md-8">
      {% with messages = get_flashed_messages(with_categories=true)
      %} {% if messages %}
  {% for category, message in messages %}
      <div class="alert alert-{{category}}">
                  {{ message }}
              </div>
          {% endfor %}
      {% endif %}
   {% endwith %}
      {% block content %} {% endblock %}
   </div>
   <center>
   <h1>Car Resale Value Predictor</h1> <br>
   <a href="{{ url_for('login') }}">Click to Login</a><br> <br> <a
   href="{{ url_for('register') }}">Register here</a><br> <br> <a
   href="{{ url_for('update') }}">Update Password</a><br> <br>
   </center>

                                   <script src="https://code.jquery.com/jquery
3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi
6jizo" crossorigin="anonymous"></script>
           <script
                     src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/po
pper.min.js" integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86
dIHNDz0W1" crossorigin="anonymous"></script>
            <script
                      src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/boots
t rap.min.js"integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B0
7jRM" crossorigin="anonymous"></script>
</body>

</html>
```

Welcome.html(taking in inputs from the user)

```
<!DOCTYPE html>
<html lang="en">

<head>
                                                            <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"
```

```
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRx
T2MZw1T" crossorigin="anonymous">
 <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='style.css') }}" />
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge"> <meta
  name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Welcome Page</title>
</head>

<body>
  <marquee>Welcome!</marquee> <br>
  <center>
  <h3>Enter Car details</h3>
  </center>

  <center>
  <form method="POST" action="/predict">
    <label for="sell">Choose a Seller:</label>
    <select id="sell" name="sell">
    <option value=0>Commercial</option>
    <option value=1>Private</option>
    </select>
    <br>
    <label for="ot">Choose a OfferType:</label>
    <select id="ot" name="ot">
    <option value=0>Offer</option>
    <option value=1>Request</option>
    </select>
    <br>
    <label for="vt">Choose a Vehicle type:</label>
    <select id="vt" name="vt">
```

```
<option value=3>Coupe</option>
<option value=1>Combination</option>
<option value=0>Bus</option>
<option  value=2>Convertible</option>
<option
value=4>Limousine</option>
<option value=7>Small</option>
<option value=8>SUV</option>
<option value=6>Others</option>
<option value=5>Not Declared</option>
</select>
<br>
<label for="gb">Choose a
Gearbox:</label> <select id="gb"
name="gb">
<option value=0>Automatic</option>
<option value=1>Manual</option>
<option value=2>Not Declared</option>
</select>
<br>
<p>PowerPs</p>
<input name="pps" required>
<p>kilometers</p>
<input name="km" required>
<br>
<label for="ft">Choose a
FuelType:</label> <select id="ft"
name="ft">
<option value=7>Petrol</option>
<option
value=1>Diesel</option>
<option value=0>CNG</option>
```

```
<option value=4>LPG</option>
<option value=3>Hybrid</option>
<option value=2>Electric</option>
<option value=5>Not Declared</option>
<option values=6>Others</option>
</select>
<br>
<label for="brand">Choose a
Brand:</label> <select id="brand"
name="brand">
<option value=0>Alpha
Romeo</option> <option
value=1>Audi</option> <option
value=2>BMW</option>
<option value=3>Chevrolet</option>
<option value=4>Chrysler</option>
<option value=5>Citroen</option>
<option value=6>Dacia</option>
<option value=7>Daewoo</option>
<option value=8>Daihatsu</option>
<option value=9>Fiat</option> <option
value=10>Ford</option> <option
value=11>Honda</option> <option
value=12>Hyundai</option> <option
value=13>Jaguar</option> <option
value=14>Jeep</option> <option
value=15>Kia</option> <option
value=16>Lada</option> <option
value=17>Lancia</option> <option
value=18>Land Rover</option>
<option value=19>Mazda</option>
<option value=20>Mercedes</option>
```

```
  <option value=21>Mini</option>
  <option value=22>Mitsubishi</option>
  <option value=23>Nissan</option>
  <option value=24>Opel</option>
  <option value=25>Peugeot</option>
  <option value=26>Porsche</option>
  <option value=27>Renault</option>
  <option value=28>Rover</option>
  <option value=29>Saab</option>
  <option value=30>Seat</option>
  <option value=31>Skoda</option>
  <option value=34>Subaru</option>
  <option value=35>Suzuki</option>
  <option value=36>Toyota</option>
  <option
  value=38>Volkswagen</option>
  <option value=39>Volvo</option>
  </select>
  <br>
  <label for="nr">Repaired?</label>
  <select id="nr" name="nr">
  <option value=1>Yes</option>
  <option value=0>No</option>
  <option value=2>Not Declared</option>
  </select>
  <br>
  <p>Age(in months)</p>
  <input name="age" required>
  <br>
  <br>
  <button type="submit">Submit</button>
</form>
```

```
   </center>

   <br>
   <br>
   <a href="{{url_for('logout')}}">Press here to logout</a>

                                        <script src="https://code.jquery.com/jquery
3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi
6jizo" crossorigin="anonymous"></script>
             <script
                         src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/po
pper.min.js" integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86
dIHNDz0W1" crossorigin="anonymous"></script>
              <script
                          src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/boots
t rap.min.js"integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B0
7jRM" crossorigin="anonymous"></script>
</body>

</html>
```

Predict.html(For displaying the output)

```
<!DOCTYPE html>
<html lang="en">
   <style>
      body{
         background-color:
         pink;color:black;
       }
```

```
    </style>
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Car Value</title>
</head>
<body>
    <h1>Car Value Predicted is</h1>

    <h1>{{predict1}}</h1>
   <br>
   <br>


</body>
</html>
```

app-ibm.py(FLASK FRAMEWORK)

```
from flask import Flask,request, render_template, url_for, redirect, flash from
flask_login import UserMixin, login_user, LoginManager, login_required,
logout_user,current_user
from wtforms.validators import InputRequired,
Length, ValidationErrorfrom flask_bcrypt import
Bcrypt
from wtforms import StringField, PasswordField, SubmitField,
IntegerFieldfrom flask_wtf import FlaskForm
import
sqlite3
import
joblib
```

```
from sklearn.preprocessing import
OrdinalEncoderimport pandas as pd
import requests

# NOTE: you must manually set API_KEY below using information retrieved
from your IBM Cloud account.
API_KEY = "iX5xF0JyPhYIfWGg37VPe14p1D7OMDPgPG1cl1yQJfJg"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token',
data={"apikey":API_KEY, "grant_type": 'urn:ibm:params:oauth:grant
type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' +

mltoken} app = Flask(_name_)
bcrypt = Bcrypt(app)
app.config['SECRET_KEY'] = 'B7-
1A3E'



login_manager = LoginManager()
login_manager.init_app(app)
login_manager.login_view = 'login'
@login_manager.user_loa
derdef load_user(user_id):
   conn = connect_db()
     user = conn.execute('SELECT * FROM user WHERE id =
                              ?', (user_id,)).fetchone()
   usr_obj = User(user[0], user[1],
   user[2])return usr_obj
def connect_db():
   conn =
  sqlite3.connect('database.db'
  ) return conn

class User:

   def_init_(self, id, email,
    username):self.id = id
    self.username =
    usernameself.email =
    email

  def to_json(self):
    return {"username":
        self.username,"email":
        self.email}

  def
    is_authenticated(self):
    return True

  def
    is_active(sel
    f):return
    True

  def
    is_anonymous(self):
    return False
  def get_id(self):
    return
    str(self.id)
```

```
class
 RegisterForm(FlaskForm):
 email =
 StringField(validators=[
   InputRequired(), Length(min=4, max=50)],
 render_kw={"placeholder": "Email"})username =
 StringField(validators=[
                  InputRequired(), Length(min=4, max=20)], render_kw={"placeholder":
"Username"})
 rollnumber = StringField(validators=[
   InputRequired(), Length(min=5, max=10)],
 render_kw={"placeholder": "RollNumber"})password =
 PasswordField(validators=[
                  InputRequired(), Length(min=8, max=20)], render_kw={"placeholder":
"Password"})

 submit = SubmitField('Register')

 def validate_username(self,
   username):conn = connect_db()
        existing_user_username = conn.execute('SELECT * FROM user
              WHERE username = ?',(username.data,)).fetchone()
   conn.comm
   it()
   conn.close()
   if
     existing_user_usern
     ame:raise
     ValidationError(
       'That username already exists. Try another one.')
```

```
class LoginForm(FlaskForm):
   username =
   StringField(validators=[
                           InputRequired(), Length(min=4, max=20)], render_kw={"placeholder":
"Username"})

   password = PasswordField(validators=[
                           InputRequired(), Length(min=8, max=20)], render_kw={"placeholder":
"Password"})

   submit = SubmitField('Login')


class UpdateForm(FlaskForm):
   username = StringField(validators=[
                           InputRequired(), Length(min=4, max=20)], render_kw={"placeholder":
"Username"})

   oldpassword = PasswordField(validators=[
              InputRequired(), Length(min=8, max=20)],
render_kw={"placeholder": "PreviousPassword"})

   password = PasswordField(validators=[
                           InputRequired(), Length(min=8, max=20)], render_kw={"placeholder":
"Password"})

   submit = SubmitField('Update')



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


@app.route('/login', methods=['GET',
'POST'])def login():
  form = LoginForm()
  if
    form.validate_on_submit
    ():conn = connect_db()
    user = conn.execute('SELECT * FROM user WHERE username = ?',
              (form.username.data,)).fetchone()
    conn.comm
    it()
    conn.close()
    if user:
      if bcrypt.check_password_hash(user[4],
        form.password.data):usr_obj = User(user[0],
        user[1], user[2]) login_user(usr_obj) return
        redirect(url_for('welcome'))
      else:
        print('Hi')
        flash(f'Invalid credentials, check and try logging in
        again.', 'danger')return redirect(url_for('login'))

  return render_template('login.html', form=form)


@app.route('/welcome', methods=['GET',
'POST'])@login_required
def welcome():
  return render_template('welcome.html')
```

```
@app.route('/predict',
methods=['POST'])def
predictSpecies():
   sell =
   float(request.form['sell'])ot
   = float(request.form['ot'])
   vt =
   float(request.form['vt'])
   gb =
   float(request.form['gb'])
   pps=float(request.form['p
   ps'])
   km=float(request.form['k
   m'])
   ft=float(request.form['ft'])
   brand=float(request.form['brand'])
   nr=float(request.form['nr'])
   age=float(request.form['age'])
   arr = [[sell, ot, vt, gb,pps,km,ft,brand,nr,age]]


                 payload_scoring = {"input_data": [{"field": [['sell', 'ot', 'vt',
'gb','pps','km','ft','brand','nr','age']], "values":arr}]}


                                                  response_scoring


=requests.post('https://us
south.ml.cloud.ibm.com/ml/v4/deployments/a4a92034-8fcd-4e79-ab7c-
521a5d8cb7d5/predictions?version=202
2-11-15',
ing,headers={'Authorization': 'Bearer ' +
mltoken}) print(response_scoring)
predictions = response_scoring.json()
pr =
predictions['predictions'][0]['values'][0][0
] print("final prediction",pr)
return
render_template('predict.html',predict1=
pr)

@app.route('/logout', methods=['GET',
'POST'])@login_required
def logout():
```

```
logout_use
r()
return redirect(url_for('login'))

@ app.route('/register', methods=['GET',
'POST'])def register():
form =
RegisterForm()
conn =
connect_db()
if form.validate_on_submit():

json=payload_scor
```

```
    hashed_password = bcrypt.generate_password_hash(form.password.data)
    conn.execute('INSERT INTO user (email, username, roll_number,
    pass_word) VALUES (?,
?, ?, ?)',
            (form.email.data, form.username.data,
    form.rollnumber.data, hashed_password))conn.commit()
```

```
        conn.close()
        return redirect(url_for('login'))
      return render_template('register.html', form=form)


    @ app.route('/update', methods=['GET',
    'POST'])def update():
form =
UpdateForm()conn
= connect_db()
if form.validate_on_submit():
conn = connect_db()
user = conn.execute('SELECT * FROM user WHERE username = ?',
(form.username.data,)).fetchone()
if user:
if bcrypt.check_password_hash(user[4],
form.oldpassword.data):print(user)
hashed_password1 =
bcrypt.generate_password_hash(form.passw
ord.data)
conn.execute('UPDATE user set pass_word = ? where username = ?',
(hashed_password1, form.username.data))
conn.commit()
conn.close()
flash(f'Password changed successfully.',
'success')return redirect(url_for('home'))
else:
flash(f'Invalid password, Enter valid password.',
'danger')return redirect(url_for('update'))
else:
flash(f'Invalid user, Enter valid User.',
'danger')return redirect(url_for('update'))
return render_template('update.html', form=form)


if name == " main ":
app.run(debug=True)
```

### 13.1 GitHub & Project Demo

**Link:**

https://github.com/IBM-EPBL/IBM-Project-20951-1659767825

**Demo Link :**

https://drive.google.com/file/d/1G4O0Bz51JEf0kJxE_bKqp_sAr
h FIoz6W/view?usp=drivesdk