# PROJECT REPORT

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

**TEAM ID : PNT2022TMID18385**

**TEAM MEMBERS : Kiruthiga.N** (TEAMLEAD)

Sarukiruthiga.M

Jothi Lakshmi.G

Jeyaranjani.R

## INTRODUCTION

* 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.

## 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.

## LITERATURE SURVEY

* 1. **Existing 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.

## References

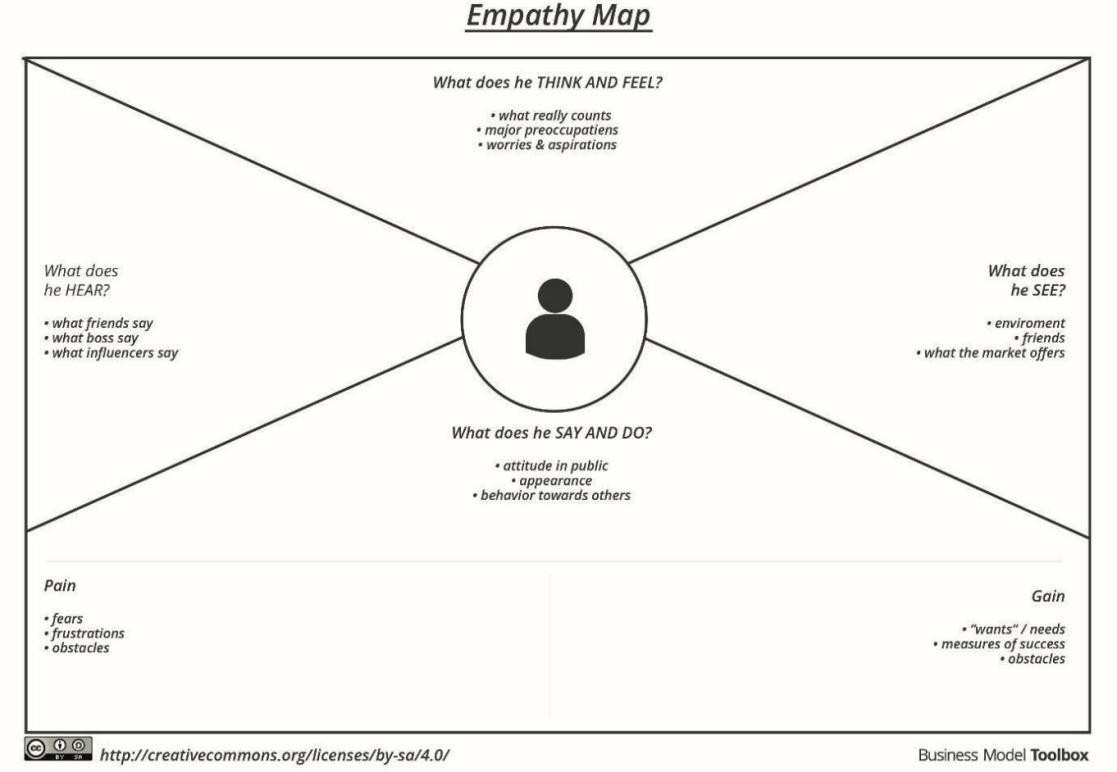
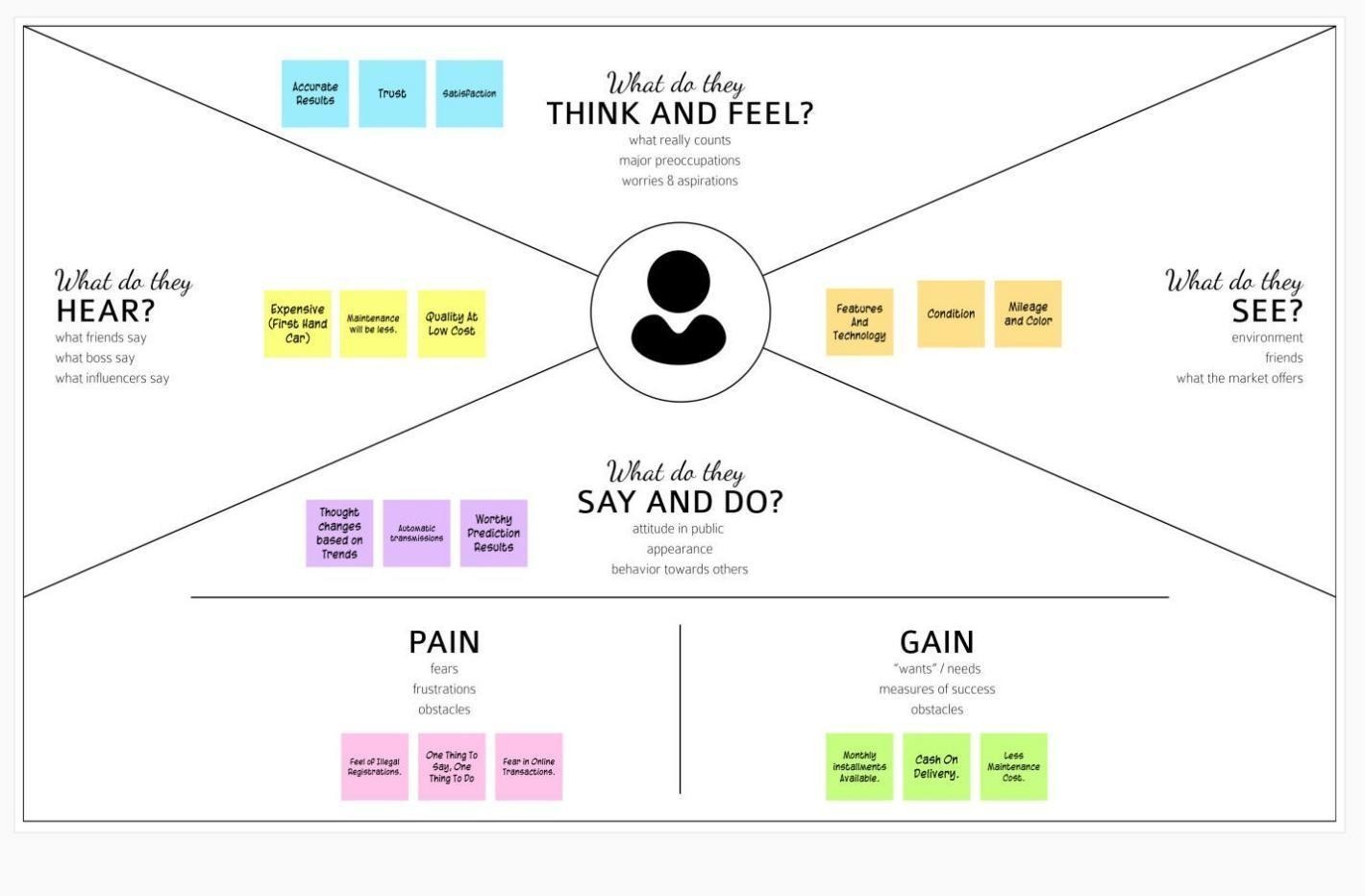
* + 1. Pudaruth Sameerchand , Pudaruth Sameerchand , Predicting the price of Used Car Using Machine Learning Techniques
    2. Enis gegic, Becir ,Isakovic, Dino Keco, ,Zerina Masetic,Jasmin Kevric Car Price Prediction Using Machine Learning
    3. Ning sun,Hongxi Bai,Yuxia Geng,Huizhu Shi Price Evaluation 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

## Problem Statement Definition

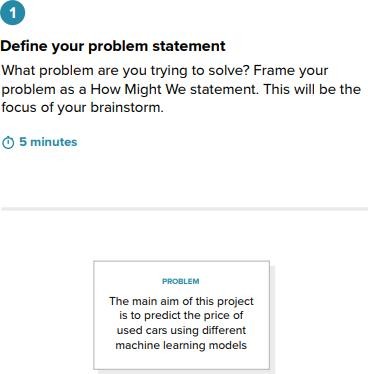
Car Resale value prediction is used to predict the value of the used cars to an reasonable price which satisfies the customer.

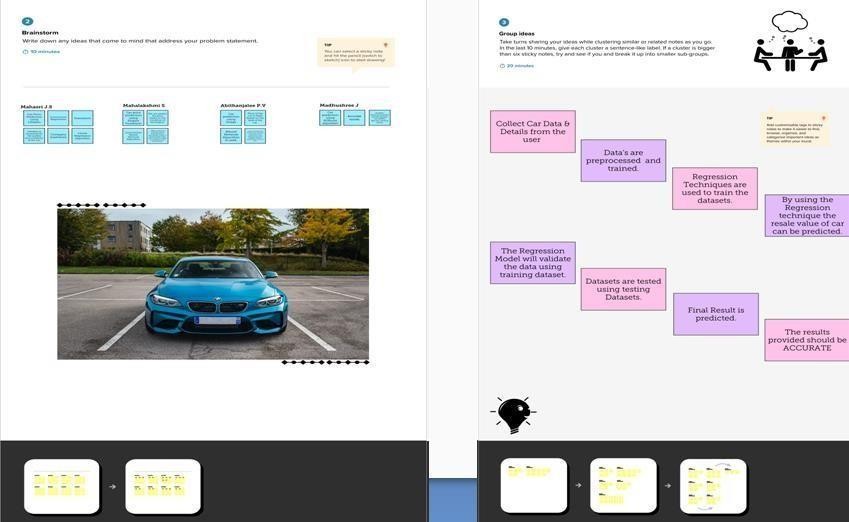
## IDEATIOIN AND PROPOSED SOLUTION

* 1. **Empathy Map Canvas**

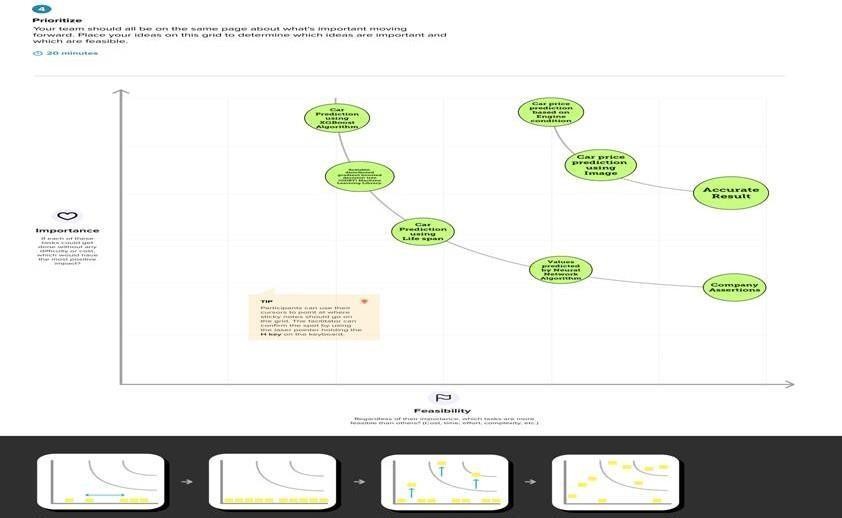


## Ideation & Brainstroming



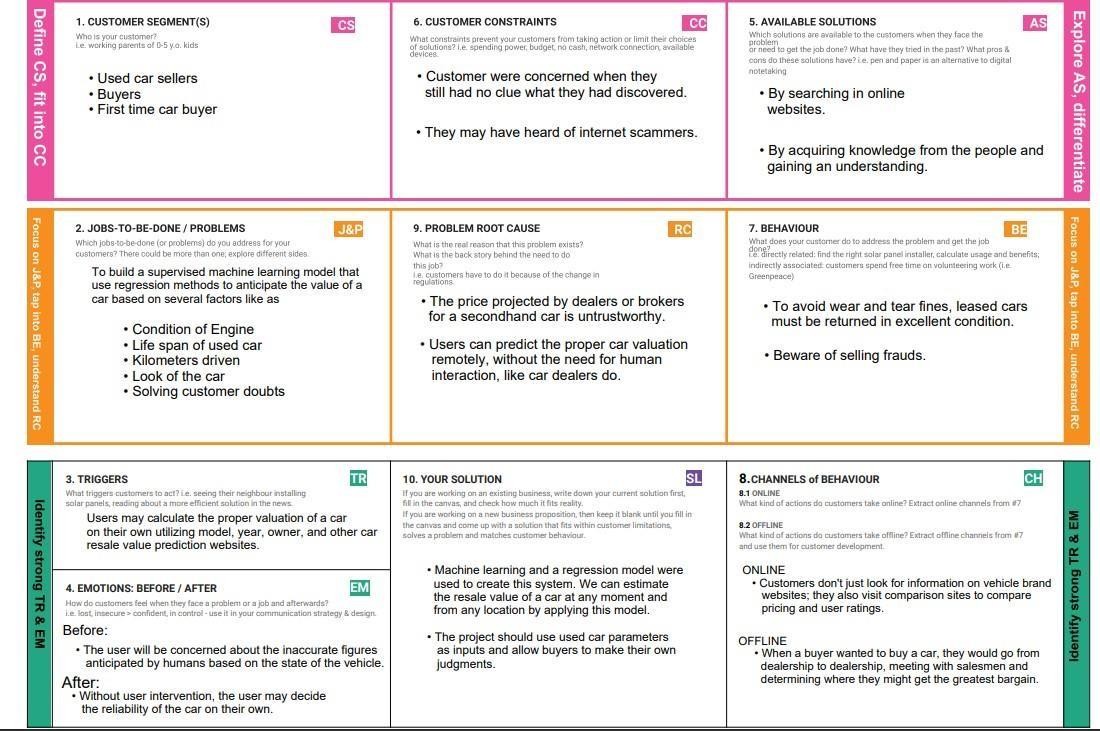


Idea prioritation :



## Proposed Solution

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Parameter** | **Description** |
| 1. | Problem Statement (Problem to be solved) | To predict the resale value of second hand car or used car considering its features. |
| 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 car with high accuracy. |
| 4. | Social Impact / Customer Satisfaction | A good platform with more reliability and portability. |
| 5. | Business Model (Revenue Model) | The model deployed in cloud so anyone can access it anywhere and anytime. |
| 6. | Scalability of the Solution | It is a web page model so it can be viewed and accessed in both computer as well as mobile  phones. |

* 1. **Problem Solution Fit**

## REQUIREMENT ANALYSIS

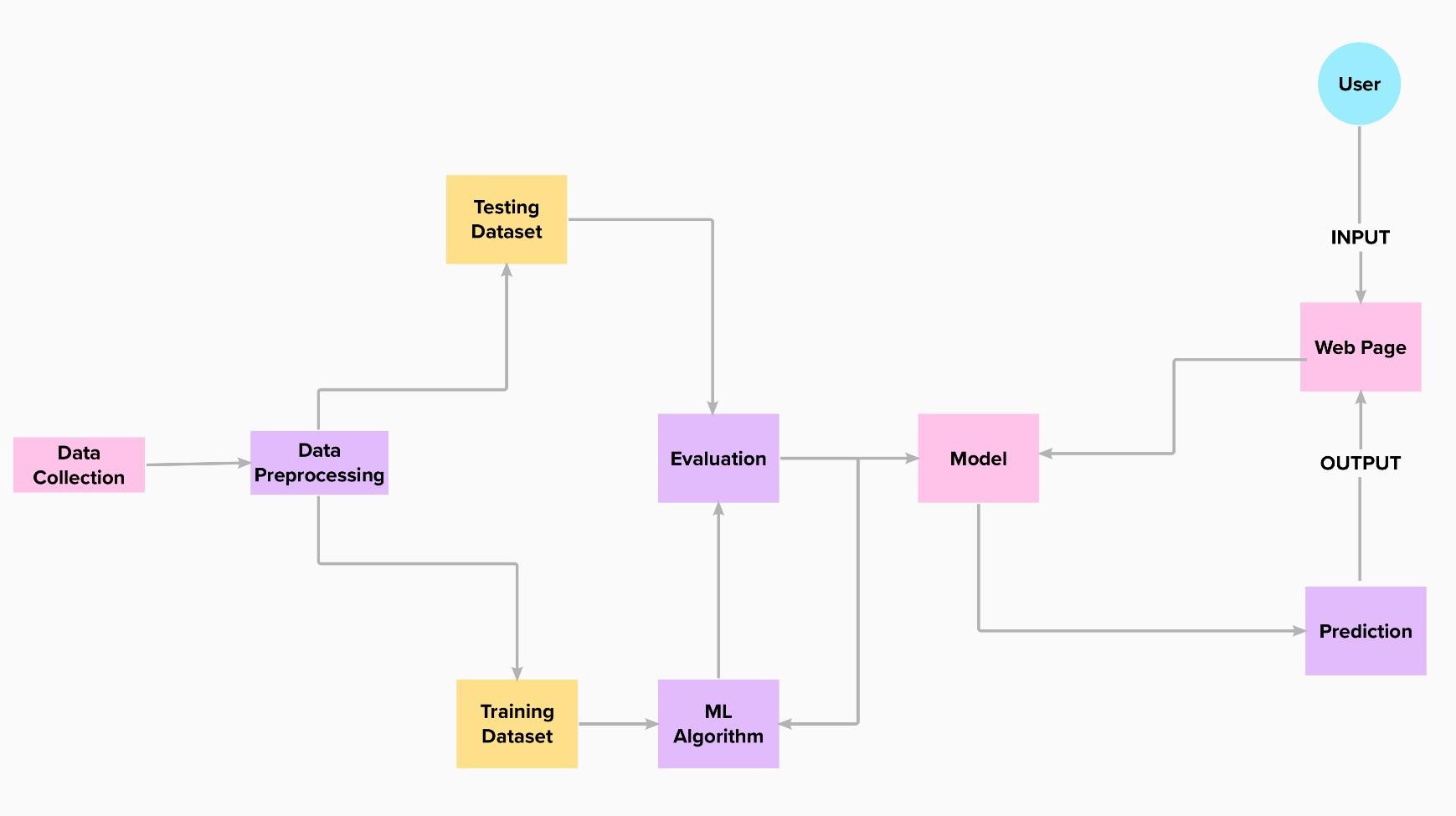
* 1. **Functional requirement**

|  |  |  |
| --- | --- | --- |
| **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 |

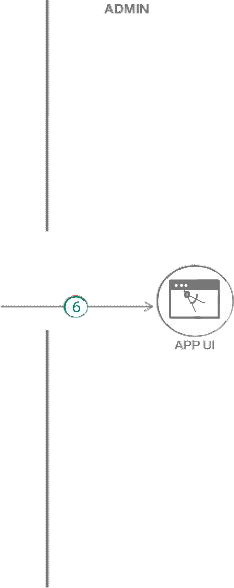
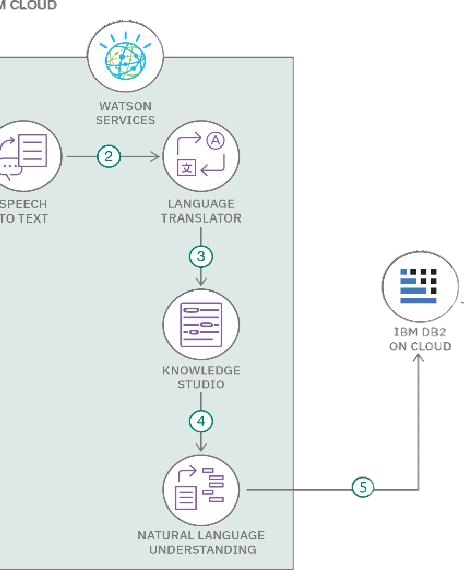
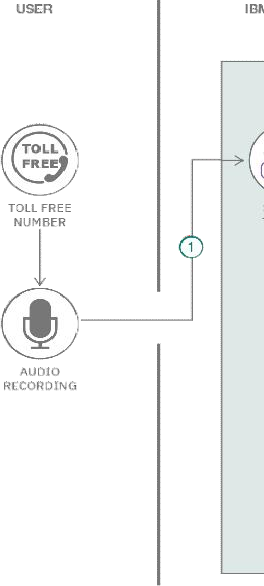
* 1. **Non-Functional requirement**

|  |  |  |
| --- | --- | --- |
| **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. |

## PROJECT DESIGN

* 1. **Data Flow Diagram**

## Solution & Technical Architecture



**User Stories**

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

## PROJECT PLANNING & SCHEDULING

* 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 initialize accounts in various public APIs like OpenWeatherMap API. | 1 | LOW | Arunprasad C Vasudevan V |
| Sprint-1 | Local Server/Software Run | Write a Python program that outputs results given the inputs like weather and location through the software | 1 | MEDIUM | Arunprasad C Vasudevan V |
| Sprint-2 | Push the server/software to cloud | Push the code from Sprint 1 to cloud so it can be accessed from anywhere | 2 | MEDIUM | Manikandan K Ragul K |
| Sprint-3 | Hardware initialization | Integrate the hardware to be able to access the cloud functions and provide inputs to the same. | 2 | HIGH | Kaviyarasan M Vasudevan V |
| Sprint-4 | UI/UX Optimization & Debugging | Optimize all the shortcomings and provide better user experience. | 2 | LOW | Arunprasad C Manikandan K |

## Sprint Delivery Schedule

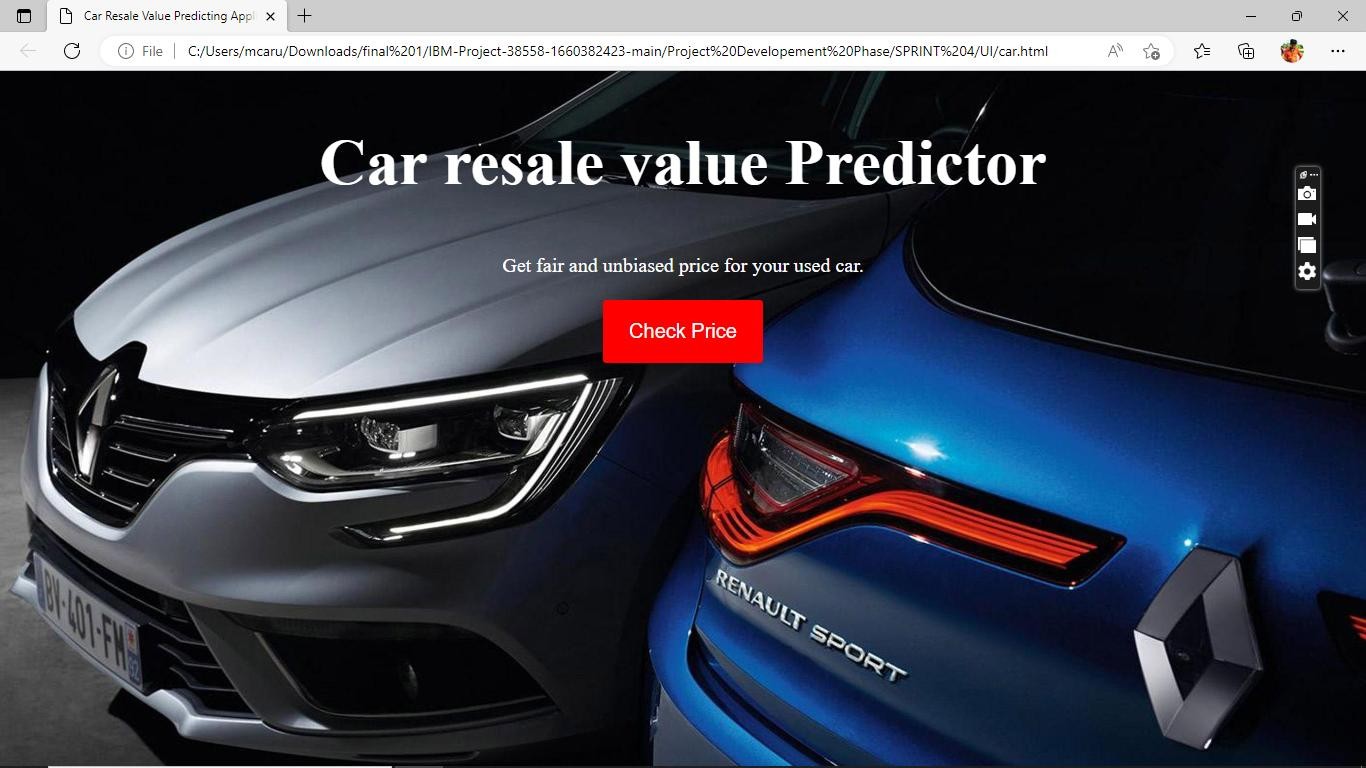
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirem ent(Epic)** | **User Story Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Membe rs** |
| Sprint-3 | Home Page | USN-1 | Description about car resale process | 2 | Low | Arunprasad C |
| Sprint-3 | Registration/Login | USN-2 | As a user, I can register for the application byentering my username, email, phone number,  and password and verify it.  As a user, I can log in to the web application  by entering my Username &password. | 5 | Medium | Vasudevan V |
| Sprint-3 | Form Page | USN-3 | As a user, I submit my car details. | 5 | Medium | Ragul K |
| Sprint-3 | Result | USN-4 | The predicted resale price for the given carmodel will be displayed. | 9 | High | Manikandan K |
| Sprint-1 | Data collection  andData preprocessing | USN-5 | Collect the required data and read the data. | 6 | High | Vasudevan V |
| Sprint-1 | Data collection andData preprocessing | USN-6 | Clean and analyse the data to avoid  duplications | 9 | High | Arunprasad C |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint-1 | Data collection andData preprocessi ng | USN-7 | Split the data into Dependent and Independentvariables | 6 | High | MAnikandan K |
| Sprint-2 | Model Building | USN-8 | Build the model using a Random Forestregression to classify the data. | 9 | High | Kaviyarasan M |
| Sprint-2 | Model Building | USN-9 | Check the metrics | 7 | High | Vasudevan V Manikandan K |
| Sprint-2 | Model Building | USN-10 | Save the model | 5 | High | Ragul K Kaviyarasan M |
| Sprint-4 | Deploy the model | USN-11 | Deployment of ML model using IBM WatsonStudio, object storage. | 13 | High | Vasudevan V Kaviyarasan M |
| Sprint-4 | Integrate the webapp with the IBM model | USN-12 | Use flask for the integrationpurpose. | 8 | Medium | Arunprasad C Ragul K |

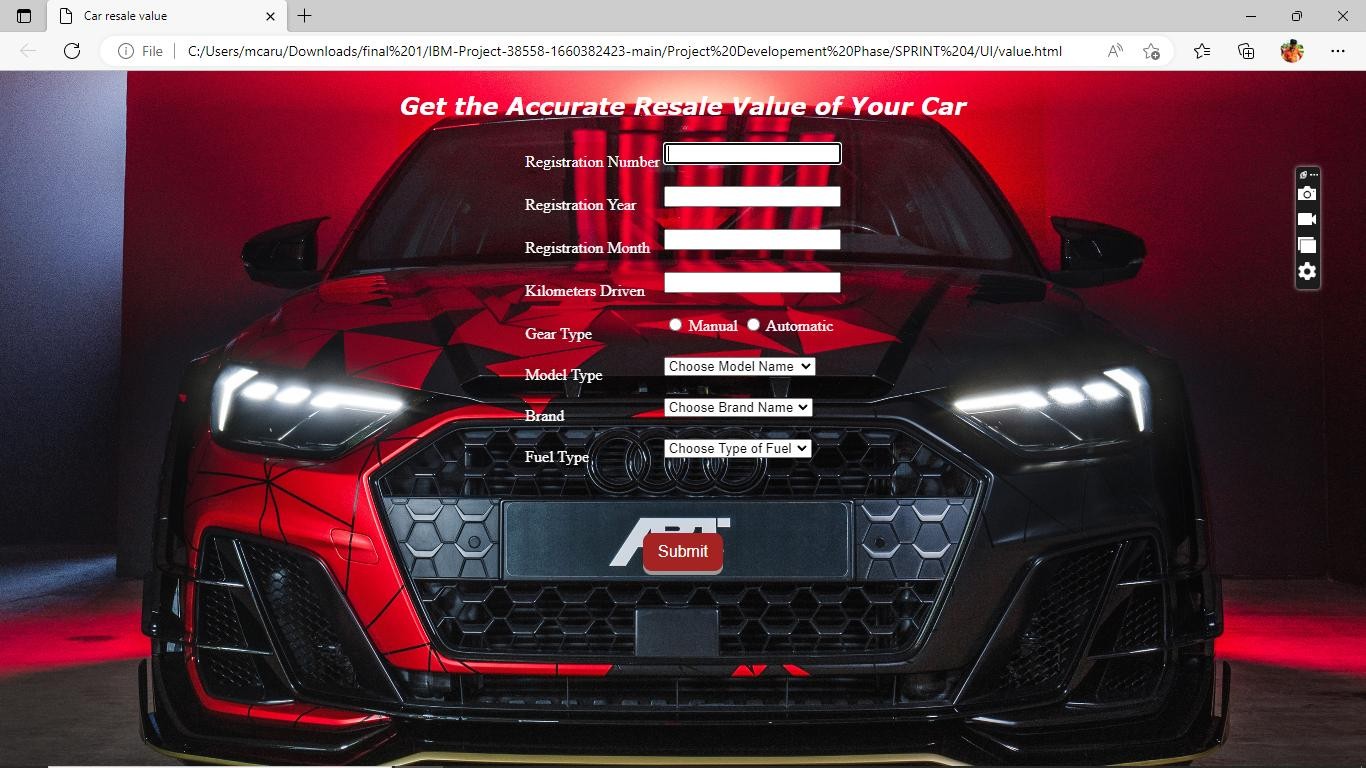
1. **CODING & SOLUTIONING**
   1. **Feature 1**
   2. **Feature 2**

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

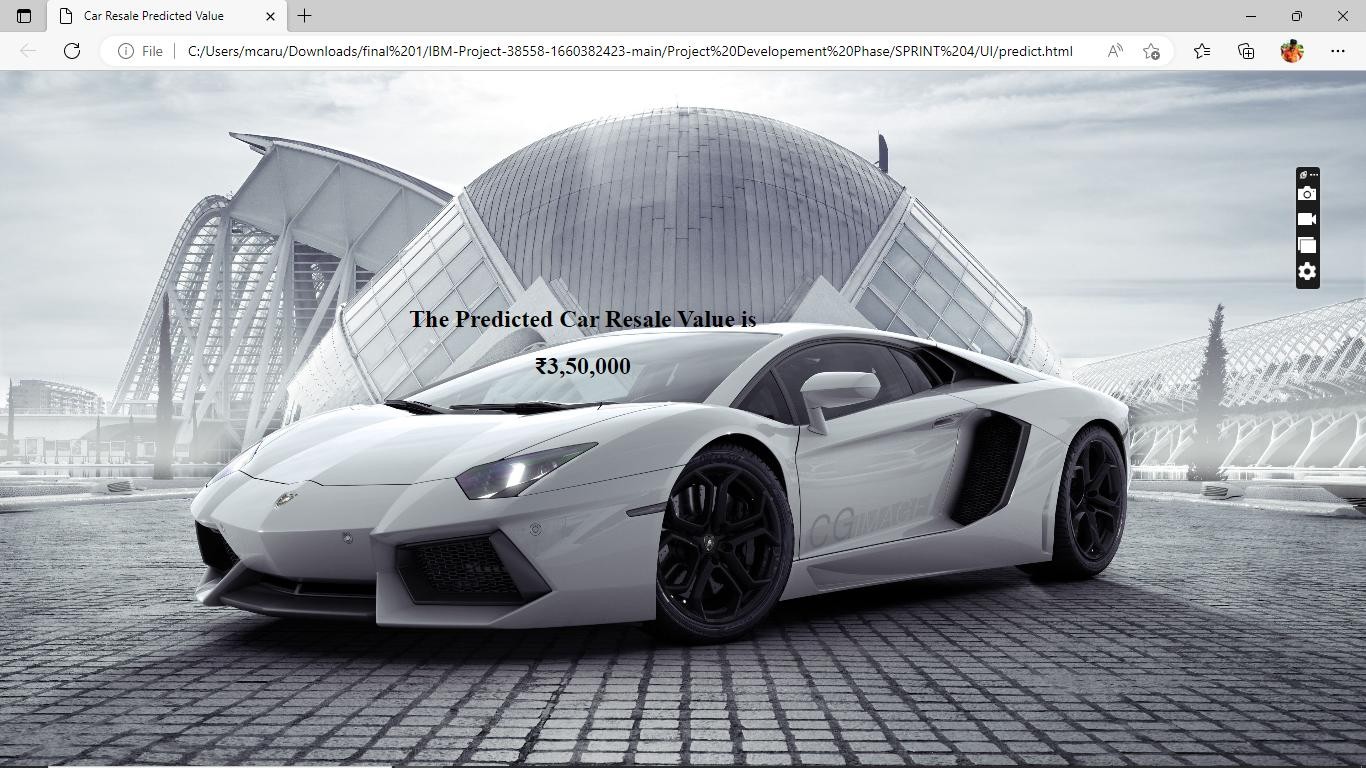
## TESTING AND RESULTS

* 1. **Test Cases Test case 1:**

**Test case 2:**



# Result



**10.**

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

## 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

## 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.

## 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.

1. **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](https://cdn.jsdelivr.net/npm/bootstrap%404.3.1/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](https://cdn.jsdelivr.net/npm/popper.js%401.14.7/dist/umd/popper.min.js) pper.min.js" integrity="sha384- UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86 dIHNDz0W1" crossorigin="anonymous"></script>

<script

src="[https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootst](https://cdn.jsdelivr.net/npm/bootstrap%404.3.1/dist/js/bootstrap.min.js) 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](https://cdn.jsdelivr.net/npm/bootstrap%404.3.1/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](https://cdn.jsdelivr.net/npm/popper.js%401.14.7/dist/umd/popper.min.js) pper.min.js" integrity="sha384- UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86 dIHNDz0W1" crossorigin="anonymous"></script>

<script

src="[https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootst](https://cdn.jsdelivr.net/npm/bootstrap%404.3.1/dist/js/bootstrap.min.js) 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=2022-11-15',

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'))

json=payload\_scor

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

form = RegisterForm() conn = connect\_db()

if form.validate\_on\_submit():

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)