Car Resale Value Prediction

Applied Data Science

Submitted by:

Shubhavya K
Sivaranjani S
Parameshwaran T R
Vighnesh A K

Project ID PNT2022TMID21553

Department of Information Technology,
Thiagarajar College of Engineering, Madurai - 625015.

Institution Mentor: Dr.C.Jeyamala
Industrial Mentor: Prof Swetha

1.Introduction:

1.1 Project Overview:

The main objective of this project is to estimate the resale value of any car model using its original cost as well as additional factors like the type of fuel it uses, the number of years it has been driven, etc. Due to the fact that they will be able to find all the information they need on a single website, people will be able to save a lot of time and money as a result.

1.2 Purpose:

The main goal of creating a system to predict car resale value is to practise Python and Data Science. The user-provided parameters form the basis of the system that predicts the amount of resale value for cars. The user enters the car's information into the available form, and the predicted selling price is generated.

2.Literature Survey

2.1 Existing Problem

1. Used Cars Price Prediction using Supervised Learning Techniques
2. Used car price prediction.
3. International Research Journal of Modernization in Engineering Technology and Science

2.2 References

https://www.researchgate.net/publication/343878698_Used_Cars_Price_Prediction_using_Supervised_Learning_Techniques

https://www.irjet.net/archives/V8/i4/IRJET-V8I4278.pdf

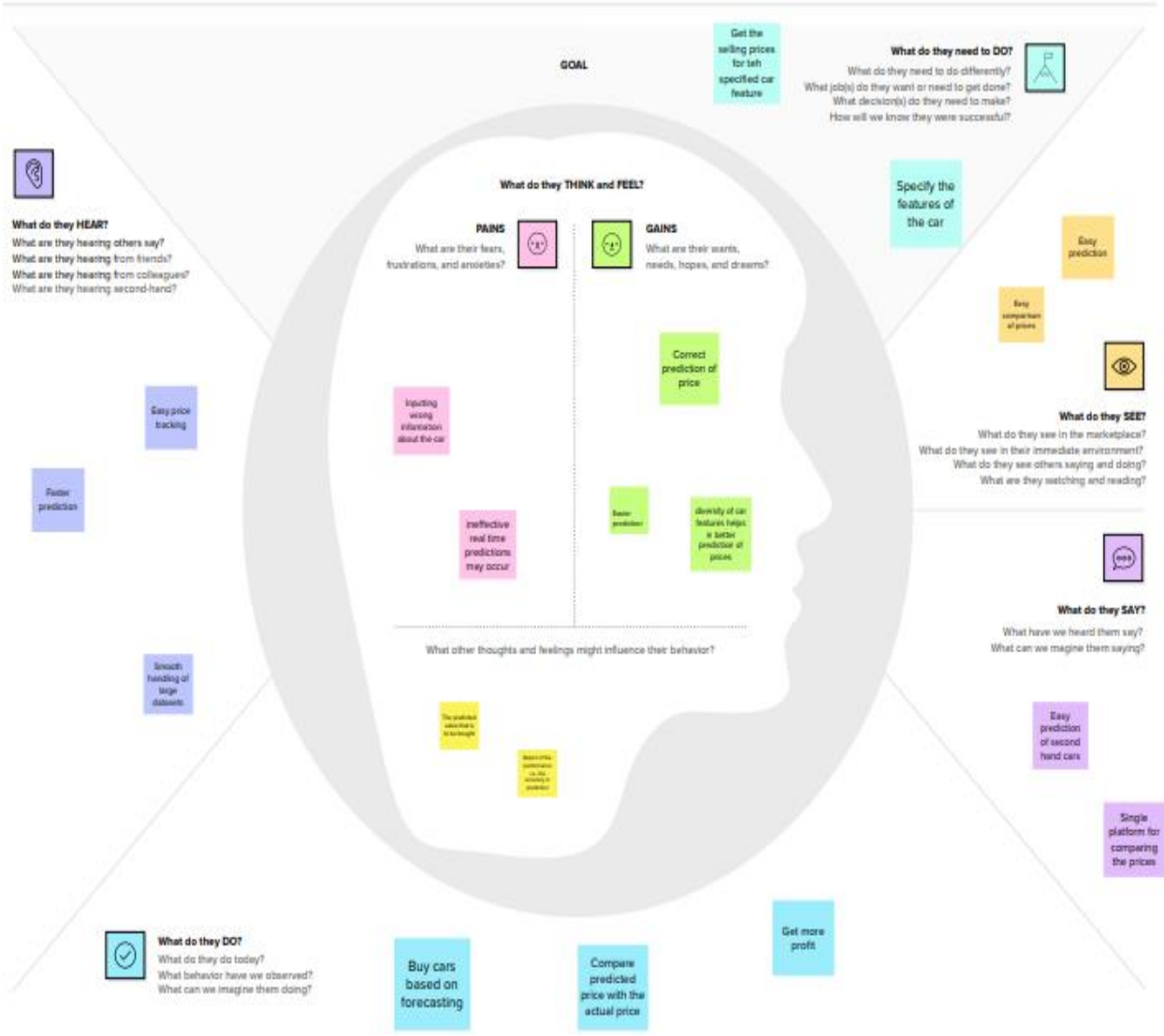
https://www.irjmets.com/uploadedfiles/paper/volume3/issue_6_june_2021/12071/1628083486.pdf

2.3 Problem Statement Definition

- I am a buyer and I am trying to buy a second hand car, but I don't know the actual worth of the car because I don't know if the price fixed by the seller is worth the deliverable, which makes me feel doubtful.
- I am a customer and I am trying to sell my used car and I have no idea about the maximum price that can be fixed for my car. I don't want to incur any loss by selling my car to the lower price, which makes me feel confused

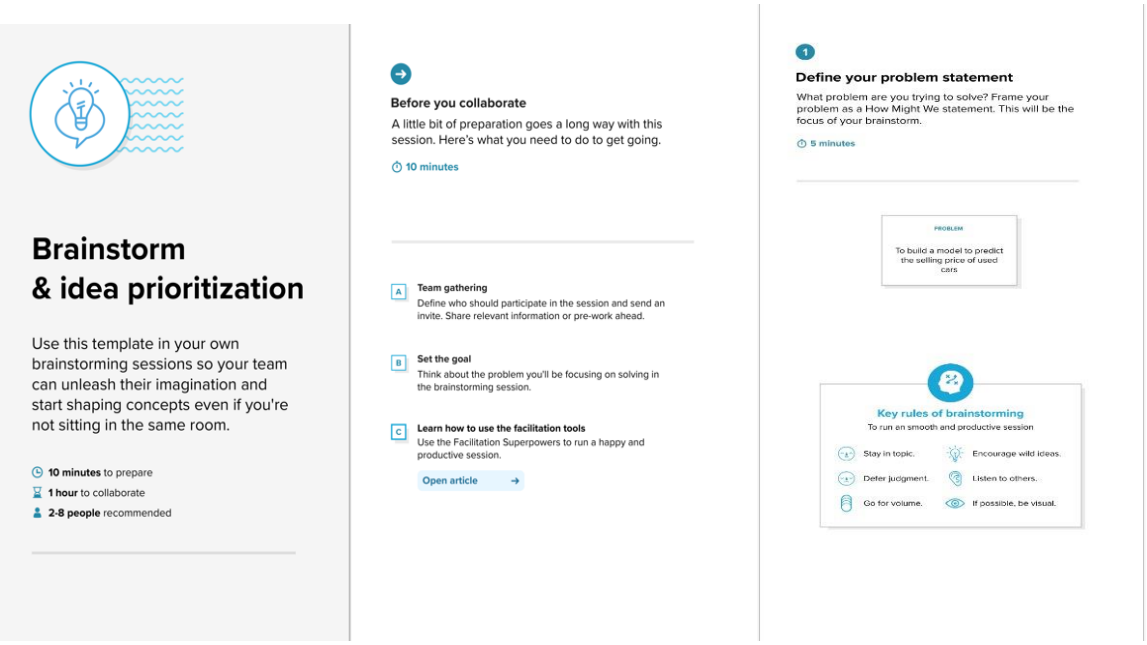
3.Ideation and Proposed Solution:

3.1 Empathy Map Canvas



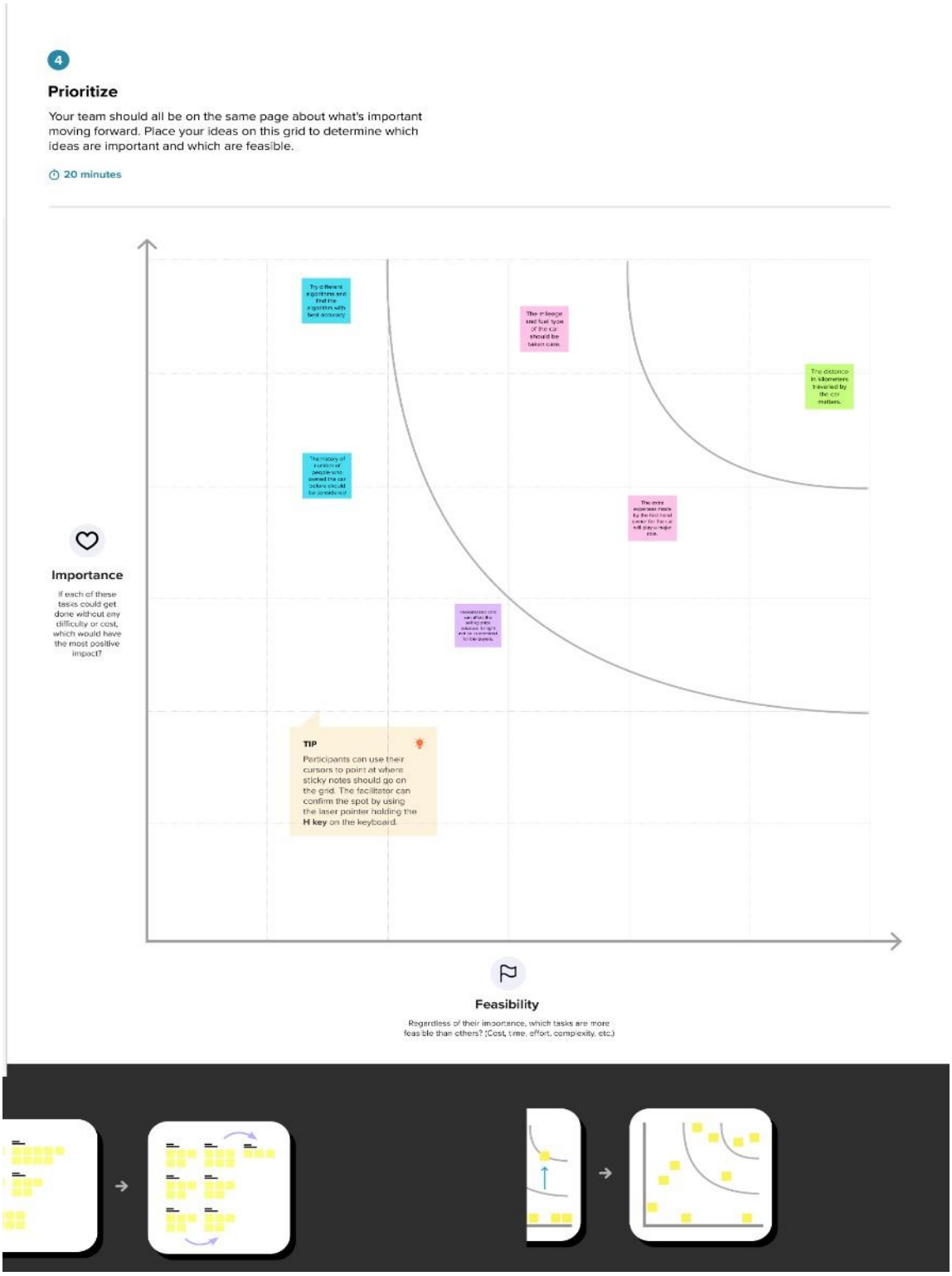
3.2 Ideation and Brainstorming

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping

Step-3: Idea Prioritization



3.3 Proposed Solution

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To predict an accurate selling price value for a second-hand car.
2.	Idea / Solution description	The solution planned for the given problem statement is that we can use any Machine Learning model to accurately predict the selling price values of resale cars with the given existing data.
3.	Novelty / Uniqueness	The uniqueness in the project comes from the dataset where we consider several parameters like the mileage, kilometres covered by the car, etc without omitting them.
4.	Social Impact / Customer Satisfaction	The gap between the seller and buyer is reduced when an accurate selling price is predicted because, no party would incur any loss and this causes immense satisfaction among the customer who turns out to be both the buyer and seller.
5.	Business Model (Revenue Model)	The business model for the algorithm and the ML model created is to create a website and deploy the model as an application so that they can generate revenue like every other website.
6.	Scalability of the Solution	The solution is scalable to a wide variety of different models of cars and the scalability can be tremendously increased when there is such availability in the dataset.

3.4 Problem Solution Fit

4. Requirement Analysis

4.1 Function Requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User login	Login via Google Login via email and Password
FR-4	Car Model input	Input the model of the car Input all the car details
FR-5	Resale price prediction	Prediction using the machine learning model

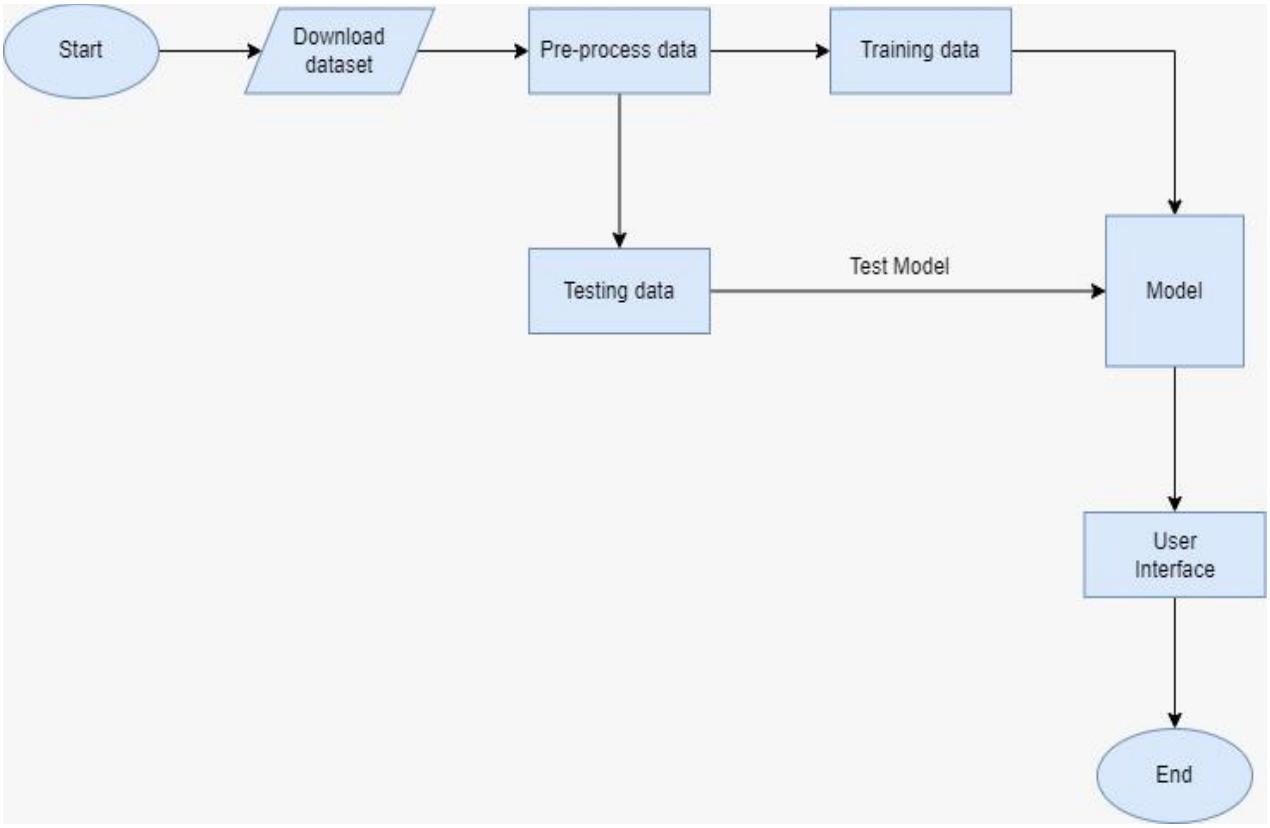
4.2 Non-function Requirement

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Simple Design Easily to navigate inside the website
NFR-2	**Security**	Allow only strong passwords Record User Access and Administrative Privileges
NFR-3	**Reliability**	The system must perform without failure Should not create downtime
NFR-4	**Performance**	The response for any search should be fast There should be good storage capacity
NFR-5	**Availability**	Must be available for all users Resale rates for all models should be predicted
NFR-6	**Scalability**	This should be supported by all the OS. Should be able to cope with an increasing number of users concurrently interacting with the site

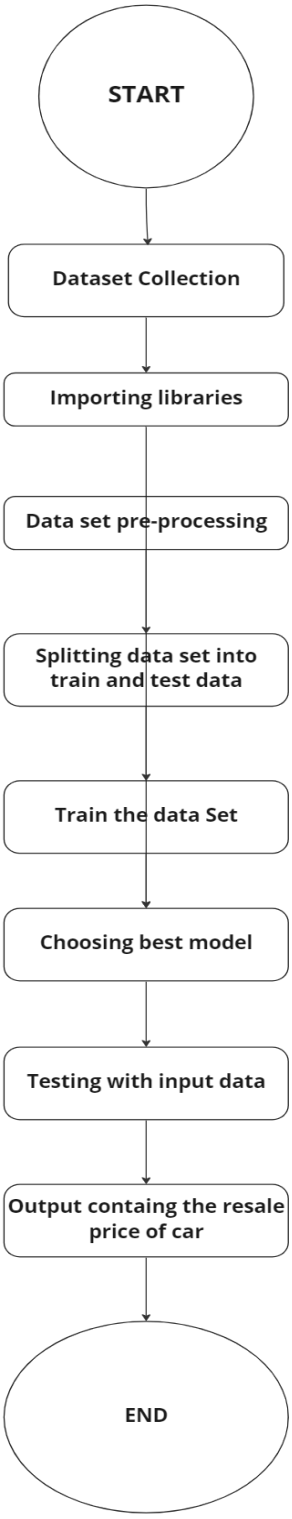
5.Project Design

5.1 Data Flow diagram

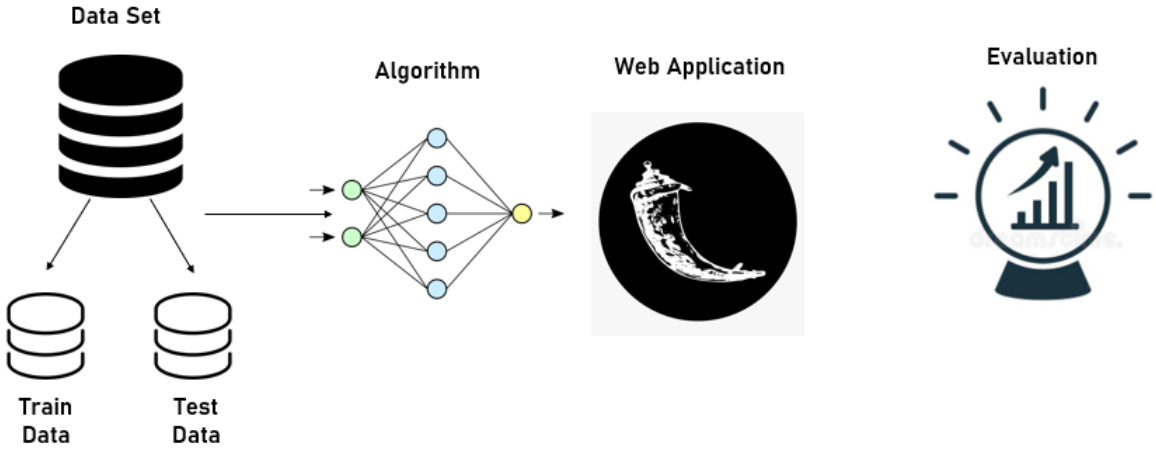


5.2 Solution and Technical Architecture

Solution Architecture



Technical Architecture



Technical Interface

Table-1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	The user interacts with application using Web UI	HTML, CSS, JavaScript etc.
2.	Application Logic-1	Logic for a process in the application	Python
3.	Database	The dataset containing car details is used for training the model to predict the rate	NoSQL.
4.	Cloud Database	The dataset is stored in the IBM cloud	IBM DB2.

5.	File Storage	File storage requirements	IBM Block Storage.
6.	Machine Learning Model	It is responsible for predicting the resale value of the cars.	Regression Model
7.	Infrastructure (Server / Cloud)	Application will be deployed in cloud.	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	**Characteristics**	**Description**	**Technology**
1.	Open-Source Frameworks	Open-source frameworks used	Flask, Python, IBM Cloud
2.	Security Implementations	Security / access controls implemented, use of firewalls etc.	Encryptions
3.	Scalable Architecture	Scalability of architecture consists of 3 tiers	Web Server - HTML, CSS, Javascript Application Server - Python Flask Database Server - IBM Cloud
4.	Availability	User can access our application through cloud all the time	IBM Cloud Hosting.
5.	Performance	Multiple users can access the web application and can perform actions simultaneously	IBM Load Balance

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, 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
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-6	As, a user I can explore the entire website	I should be able to receive notifications based on my interest.	Medium	Sprint-1
Customer (Web user)	Search	USN-7	As a user I can view all the car models and their prices		High	Sprint-2
Customer Care Executive	Chat	USN-8	As a user I can use the chat option to clear all my doubts		low	Sprint-3
Administrator	Data collection and modification	USN-9	As an admin, I can collect the data of all my users without mis using it.		high	Sprint-3

6.Project Planning and Scheduling

6.1 Sprint Planning and Estimation

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project and collecting other information	28 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	19 SEPTEMBER2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	19 SEPTEMBER2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	23 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	30 SEPTEMBER 2022

Solution Architecture	Prepare a solution architecture document.	28 SEPTEMBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	20 OCTOBER 2022
Functional Requirement	Prepare the functional requirement document.	8 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	9 OCTOBER2022
Technology Architecture	Prepare the technology architecture diagram.	10 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	22 OCTOBER 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	IN PROGRESS

6.2 Sprint Delivery Schedule

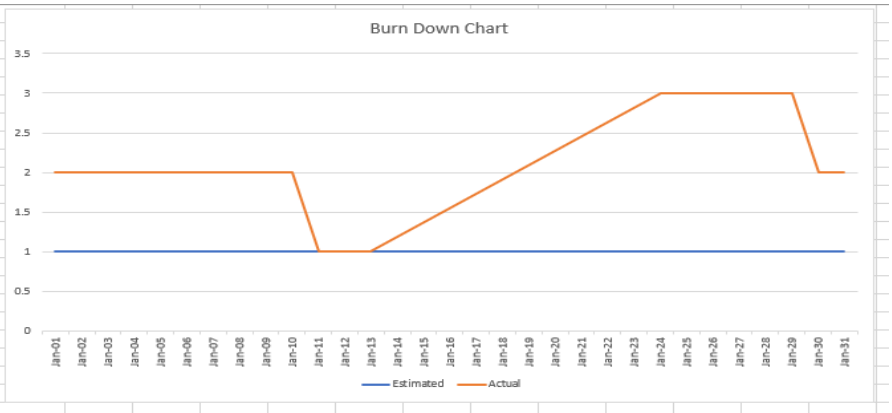
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Search	USN-1	As a user, I can find all the available car models as the result of a search	2	High	Shubhavya K, Sivaranjani S, Parameshwaran T R, Vighnesh A K
Sprint-1	Display	USN-2	As a user, I can get all the details of a selected car model	1	High	Shubhavya K, Sivaranjani S, Parameshwaran T R, Vighnesh A K
Sprint-2	Result	USN-3	As a user, I can see all the predicted results	2	High	Shubhavya K, Sivaranjani S, Parameshwaran T R, Vighnesh A K
Sprint-1	Analyse	USN-4	As an administrator, I can save the details of the users to give them best suggestions	2	Medium	Shubhavya K, Sivaranjani S, Parameshwaran T R, Vighnesh A K

Project Tracker, Velocity & Burndown Chart: (4 Marks)

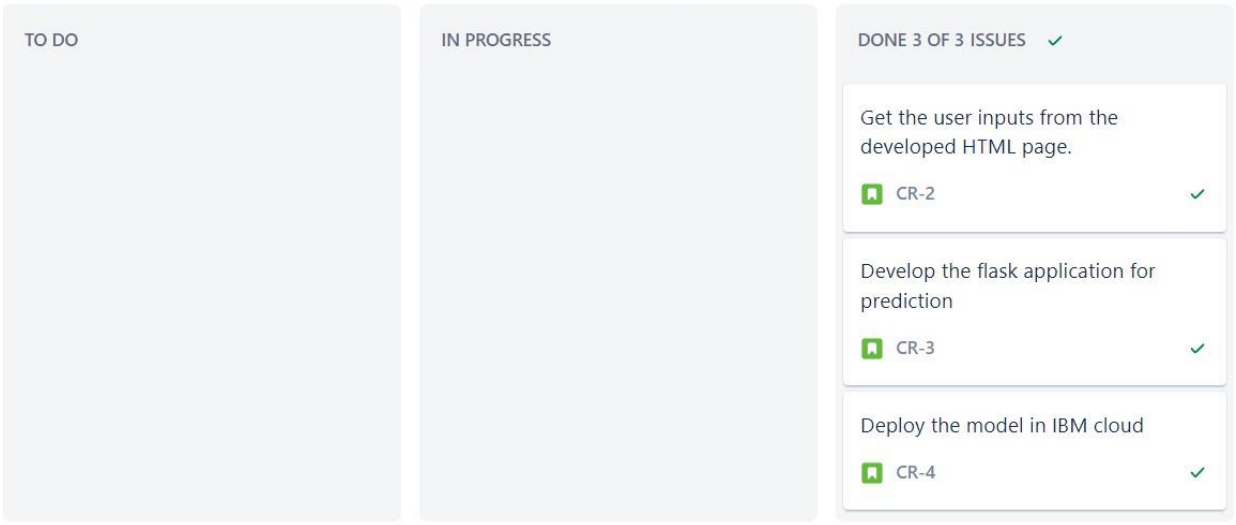
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	5 Days	24 Oct 2022	29 Oct 2022	20	15 Nov 2022
Sprint-2	20	5 Days	31 Nov 2022	05 Nov 2022	20	16 Nov 2022
Sprint-3	20	5 Days	07 Nov 2022	12 Nov 2022	20	17 Nov 2022
Sprint-4	20	5 Days	14 Nov 2022	19 Nov 2022	20	18 Nov 2022

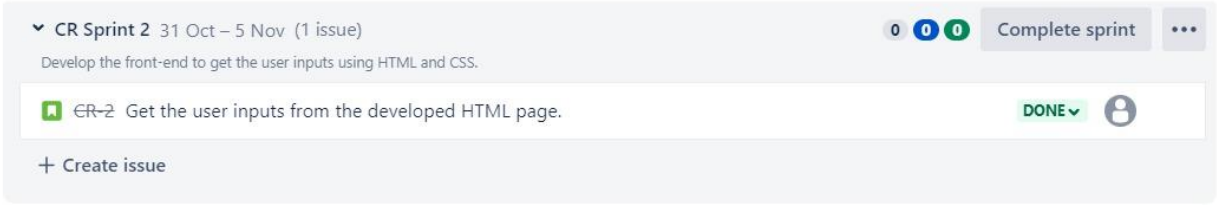
Burntdown Chart

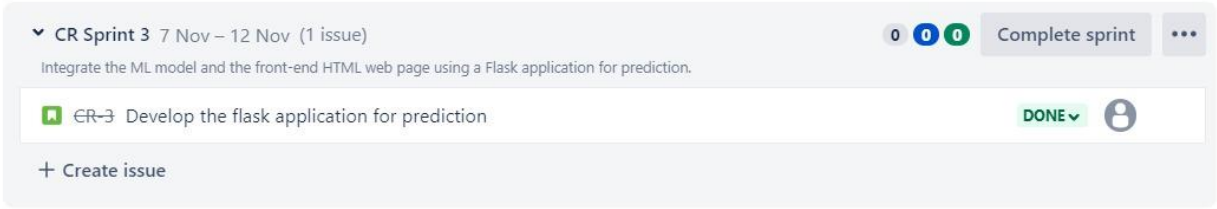
Day	dates	Estimated	Actual
Monday	Oct-24	1	3
Tuesday	Oct-25	1	3
Wednesday	Oct-26	1	3
Thursday	Oct-27	1	3
Friday	Oct-28	1	3
Saturday	Oct-29	1	3
Sunday	Oct-30	1	2
Monday	Oct-31	1	2
Tuesday	Nov-01	1	2
Wednesday	Nov-02	1	2
Thursday	Nov-03	1	2
Friday	Nov-04	1	2
Saturday	Nov-05	1	2
Sunday	Nov-06	1	2
Monday	Nov-07	1	2
Tuesday	Nov-08	1	2
Wednesday	Nov-09	1	2
Thursday	Nov-10	1	2
Friday	Nov-11	1	1
Saturday	Nov-12	1	1
Sunday	Nov-13	1	1
Monday	Nov-14	1	0

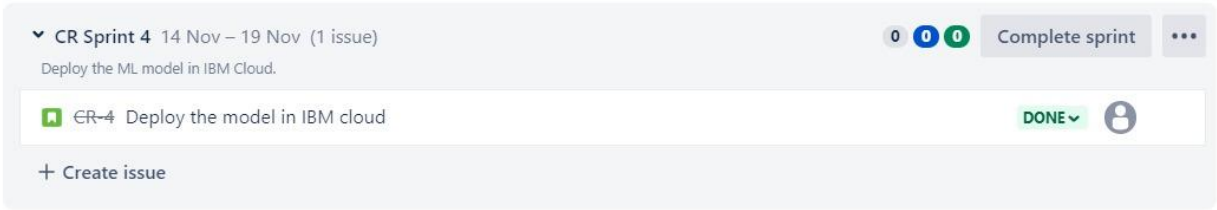


6.3 Reports from JIRA



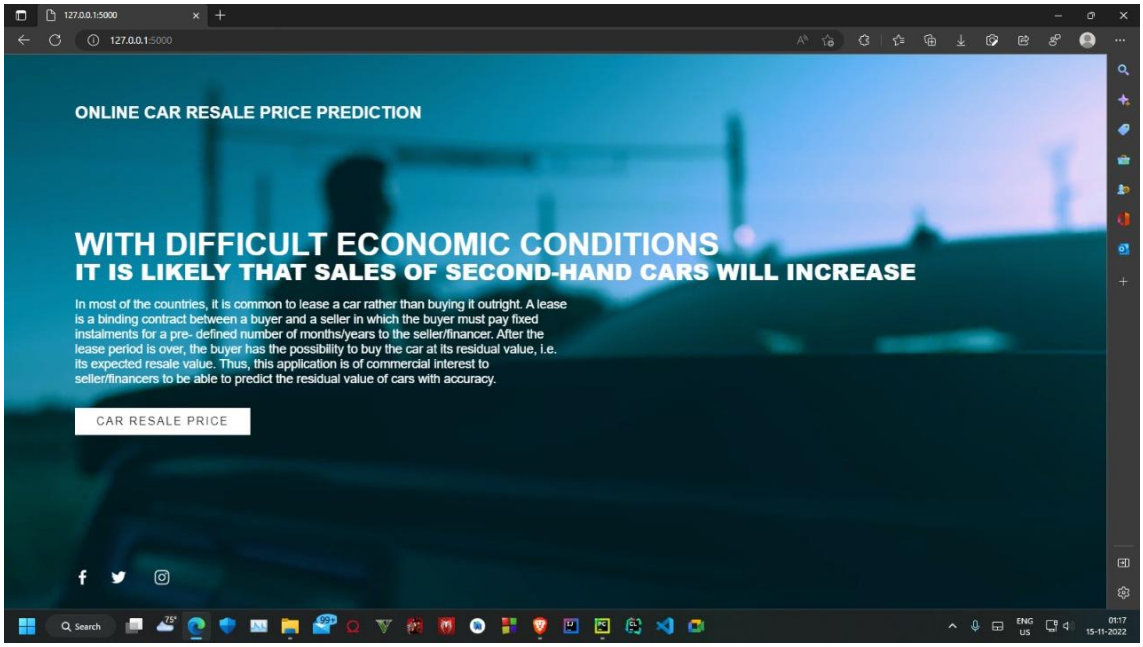




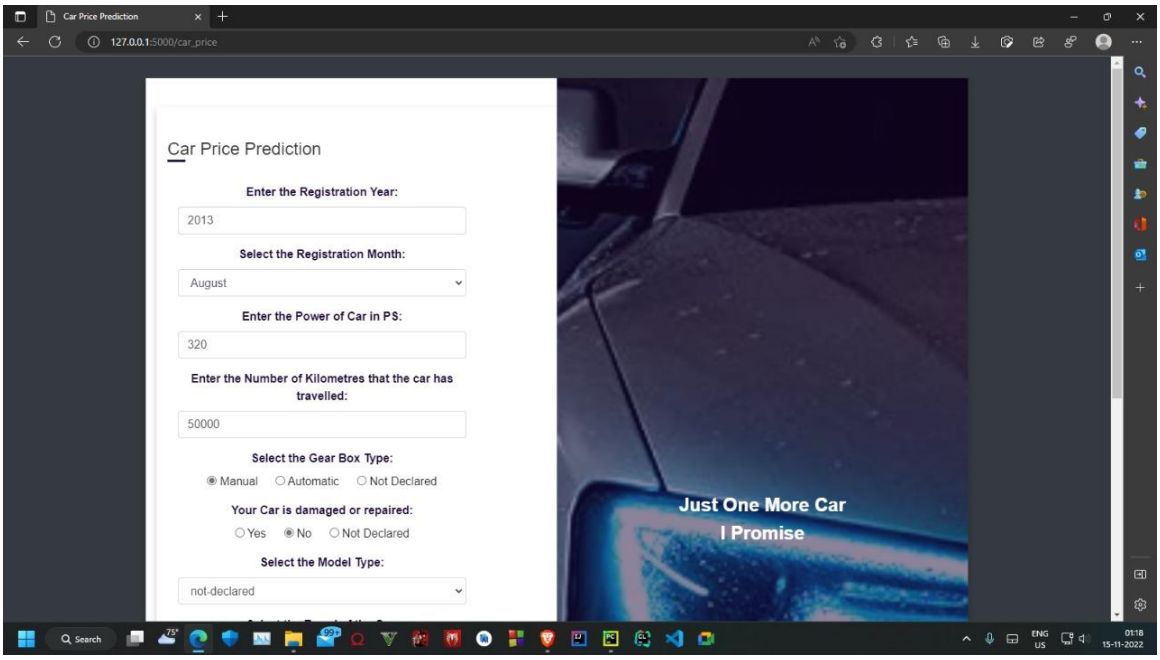


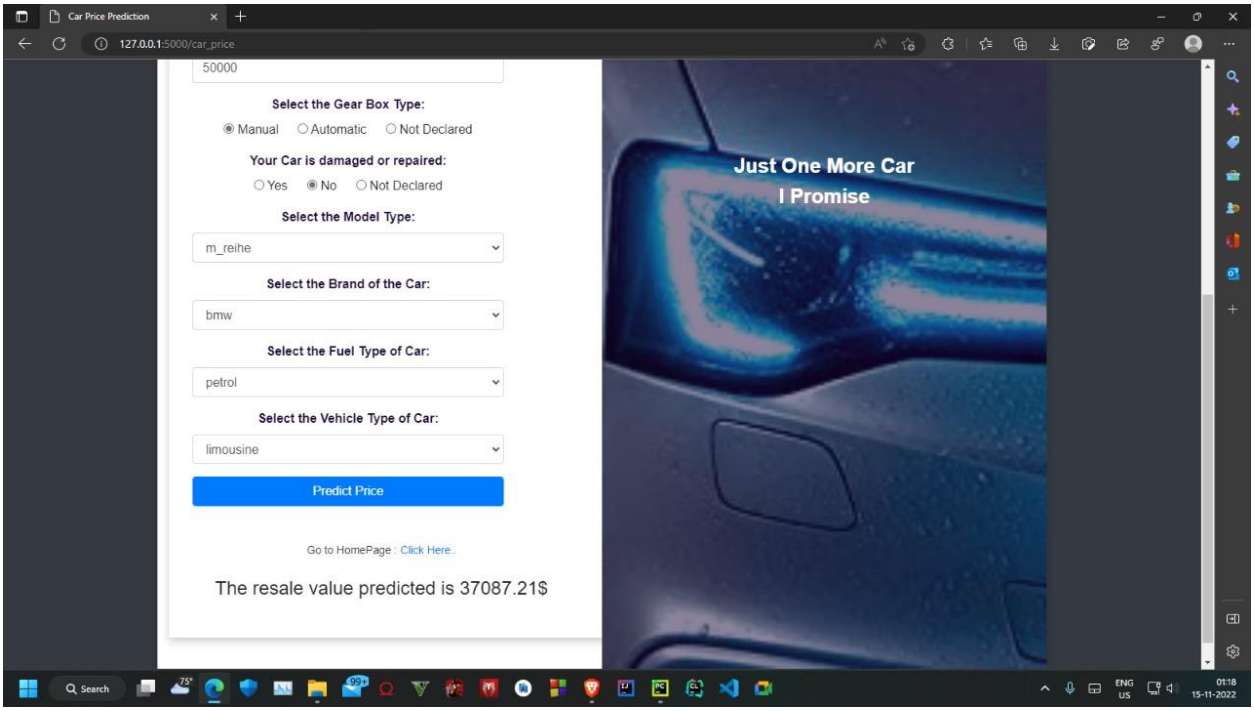
7.Coding and Solutioning

7.1 Home Page:



7.2 Prediction Page





Coding

```
IMPORT REQUIRED LIBRARIES
"""

import pandas as pd
import numpy as np
import matplotlib as plt
from sklearn.preprocessing import LabelEncoder
import pickle

"""READ THE DATASETS"""

from google.colab import drive
drive.mount('/content/drive')

df = pd.read_csv("/content/drive/MyDrive/IBM_Nalaiya Thiran/car_resale.csv", encoding='latin-1')
df.head()

df.tail()

"""CLEANING THE DATASET"""

#different sellers
print(df.seller.value_counts())
```

```
#remove the seller 'gewerblich'
df[df.seller != 'gewerblich']

#all entries of column 'seller' are same
#drop the column 'seller'
df = df.drop('seller', 1)

#different offer types
print(df.offerType.value_counts())

#remove the offertype 'Gesuch'
df[df.offerType != 'Gesuch']

#column 'offerType' has same entires
#drop the column 'offerType'
df = df.drop('offerType', 1)

print(df.shape)

#remove cars having power less than 50p and greater than 900p
df = df[(df.powerPS > 50) & (df.powerPS < 900)]
print(df.shape)

#remove cars with year of registration before 1950 and after 2017
df = df[(df.yearOfRegistration >= 1950) & (df.yearOfRegistration < 2017)]
print(df.shape)

#remove columns that are not relevant
df.drop(['name', 'abtest', 'dateCrawled', 'nrOfPictures', 'lastSeen', 'postalCode', 'dateCreated'],
axis='columns', inplace=True)

#creating a copy of the dataframe and remove the duplicates in the columns
new_df = df.copy()
new_df = new_df.drop_duplicates(['price', 'vehicleType', 'yearOfRegistration', 'gearbox', 'powerPS',
'model', 'kilometer', 'monthOfRegistration', 'fuelType', 'notRepairedDamage'])

#clean the dataset of German words and replace with proper English words
new_df.gearbox.replace(('manuell', 'automatik'), ('manual', 'automatic'), inplace=True)
new_df.fuelType.replace(('benzin', 'andere', 'elektro'), ('petrol', 'others', 'electric'), inplace=True)
new_df.vehicleType.replace(('kleinwagen', 'cabrio', 'kombi', 'andere'), ('small  car', 'convertible',
'combination', 'others'), inplace=True)
new_df.notRepairedDamage.replace(('ja', 'nein'), ('Yes', 'No'), inplace=True)

#Outlier Removal
new_df = new_df[(new_df.price >= 100) & (new_df.price <= 150000)]

#Fill the not declared values of the columns as NaN using fillna function
new_df['notRepairedDamage'].fillna(value='not-declared', inplace=True)
new_df['fuelType'].fillna(value='not-declared', inplace=True)
new_df['gearbox'].fillna(value='not-declared', inplace=True)
```

```
new_df['vehicleType'].fillna(value='not-declared', inplace=True)
new_df['model'].fillna(value='not-declared', inplace=True)

#save the dataframe as csv
new_df.to_csv('car_resale_preprocessed.csv')

#label encode the categorical data
labels = ['gearbox', 'notRepairedDamage', 'model', 'brand', 'fuelType', 'vehicleType']

mapping = {}
for i in labels:
  mapping[i] = LabelEncoder()
  mapping[i].fit(new_df[i])
  tr = mapping[i].transform(new_df[i])
  np.save(str('classes'+i+'.npy'), mapping[i].classes_)
  print(i, ":", mapping[i])
  new_df.loc[:, i+'_labels'] = pd.Series(tr, index=new_df.index)

#'labeled' dataframe contains the final data

labelled = new_df[ ['price', 'yearOfRegistration', 'powerPS', 'kilometer', 'monthOfRegistration'] +
[x+"_labels" for x in labels]]
print(labelled.columns)

"""SPLITTING DATA INTO INDEPENDENT AND DEPENDENT VARIABLES"""

#split price and other data into Y and X respectively
Y = labelled.iloc[:, 0].values
X = labelled.iloc[:, 1:].values
Y = Y.reshape(-1, 1)

#split dataset into train and test dataset
from sklearn.model_selection import cross_val_score, train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=3)

"""MODEL BUILDING

CHOOSE THE APPROPRIATE MODEL AND CHECK THE METRICS OF THE MODELS
"""

from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score

regressor = RandomForestRegressor(n_estimators=1000, max_depth=10, random_state=34)

regressor.fit(X_train, np.ravel(Y_train, order='C'))

pred_1 = regressor.predict(X_test)

print(r2_score(Y_test, pred_1))
```

```
from sklearn.tree import DecisionTreeClassifier

ds = DecisionTreeClassifier(max_depth=5000, max_features=0.9, max_leaf_nodes=5000,
random_state=2, splitter='best')

ds.fit(X_train,  np.ravel(Y_train, order='C'))

pred_3 =ds.predict(X_test)

print(r2_score(Y_test, pred_3))

"""SAVE THE MODEL"""

file_name = 'resale_model.pkl'
pickle.dump(regressor, open(file_name, 'wb'))
```

8. Testing

8.1 Test cases

				Date	07-Nov-22								
				Team ID	PNT2022TMID21553								
				Project Name	Project - Car Resale Value Predictio								
				Maximum Marks	4 marks								
Test case ID	**Feature Type**	**Component**	**Test Scenario**	**Pre-Requisite**	**Steps To Execute**	**Test Data**	**Expected Result**	**Actual Result**	**Status**	**Commnets**	**TC for Automation(Y/N**	**BUG ID**	**Executed By**
HomePage_TC_OO1	Functional	Home Page	Verify if the user is able to click and navigate through all the elements present in the home page	Nil	1.Enter URL and click go 2.Click on the elements avilable in the home page.		The user should be able to go to other pages from the home page	Working as expected	Pass		N		Shubhavya K
HomePage_TC_OO2	UI	Home Page	Verify if all the elements in the home page are properly arranged	Nil	1.Enter URL and click go 2.Viewthe elements avilable in the home page.		All the elements in the home page should be visible to the user	Working as expected	Pass		N		Parameshwaran T R
InputPage_TC_OO1	Functional	input page	Check to see if the user can choose anything from the drop-down menus on the input page.	Nil	1.Enter URL and click go 2.Go to the input page 3. select anything from the drop down list		User should be able to select anything from the drop down list displayed	Working as expected	Pass		N		Sivaranjani S
InputPage_TC_OO2	Functional	input page	Check if the user can select one of the radio buttons	Nil	1.Enter URL and click go 2.Go to the input page 3. select one of the options given in the radio button		User should be able to select one option from the radio button	Working as expected	Pass		N		Vighnesh A K
InputPage_TC_OO3	UI	Input page	Verify if all the elements in the home page are properly displayed	Nil	1.Enter URL and click go 2.Go to the input page 3. View and check if the elements are displayed properly		User should be able to view all the elemets in the input page	Working as expected	Pass		N		Parameshwaran T R
OutputPage_TC_OO1	Functional	Output page	Check if the user is taken to the output page from the input page	Nil	1.Enter URL and click go 2.Go to the input page 3. Enter all the input data 4. Check if you are directed to the output page		User should be directed ro the output page	Working as expected	Pass		N		Shubhavya K

8.2 User Acceptance Testing

Defect Analysis

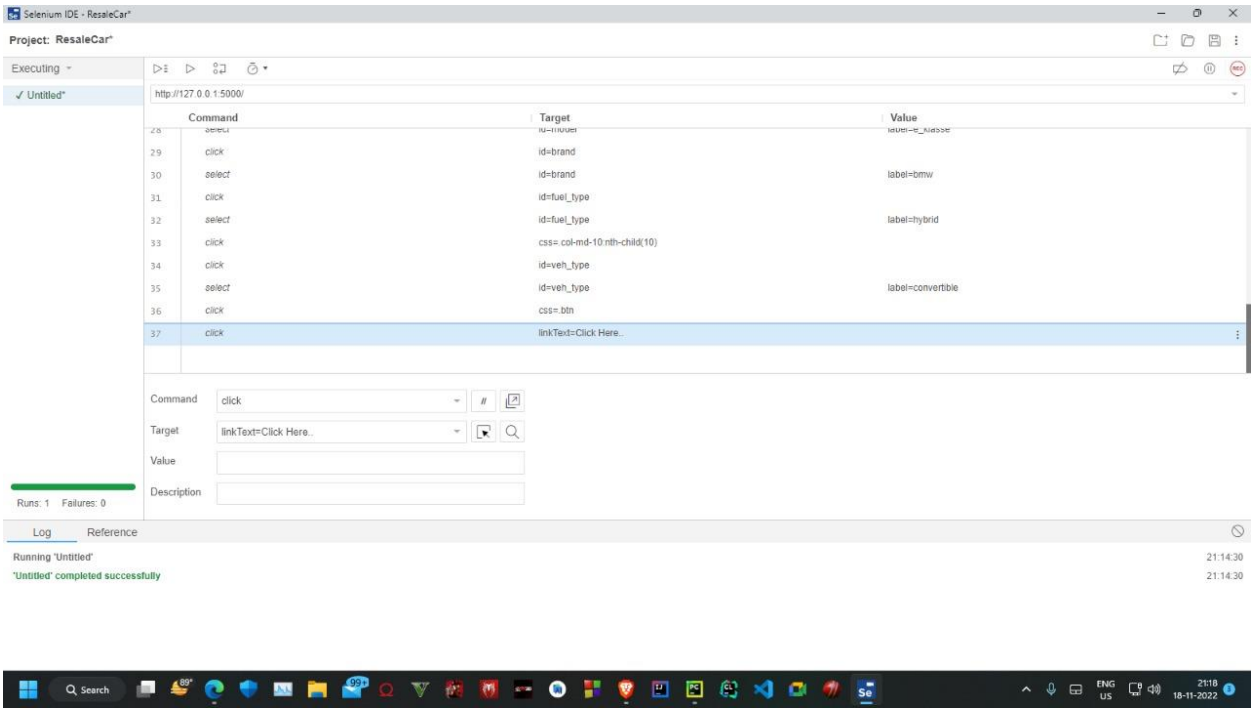
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	2	18
Duplicate	1	0	3	0	4
External	1	3	0	1	5
Fixed	10	2	3	20	35
Not Reproduced	0	0	1	0	1
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	22	9	6	23	63

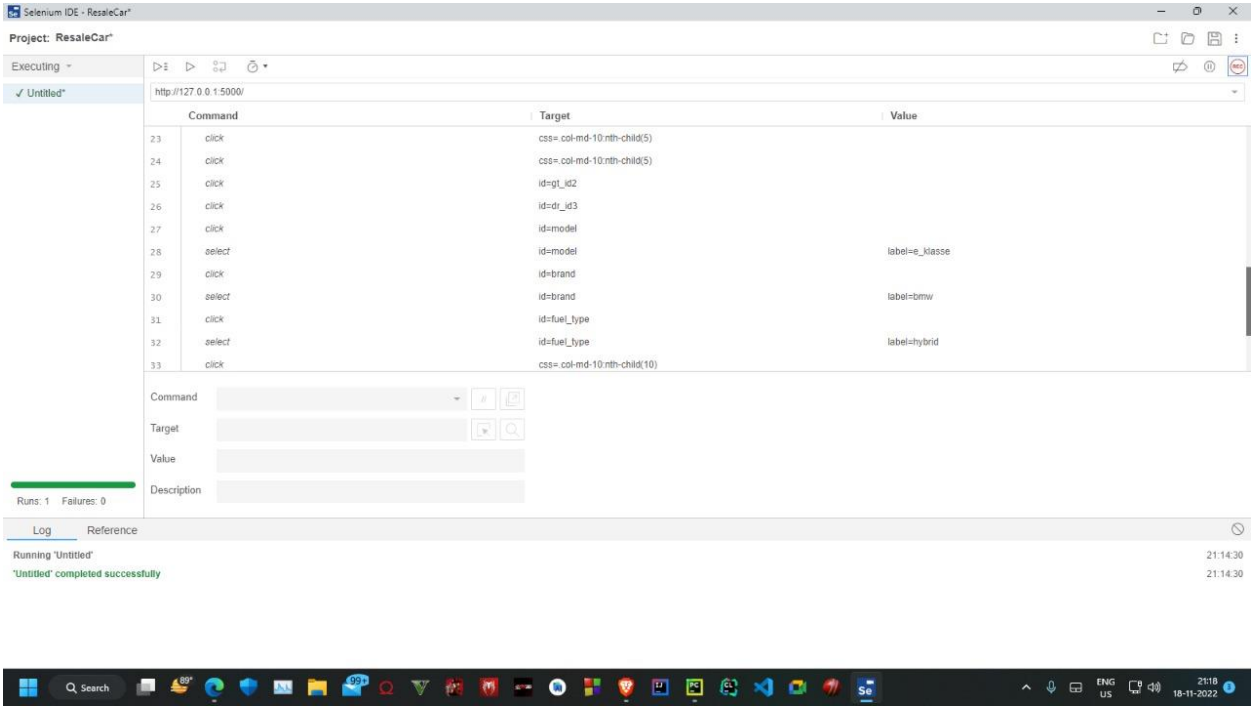
Test Case Analysis

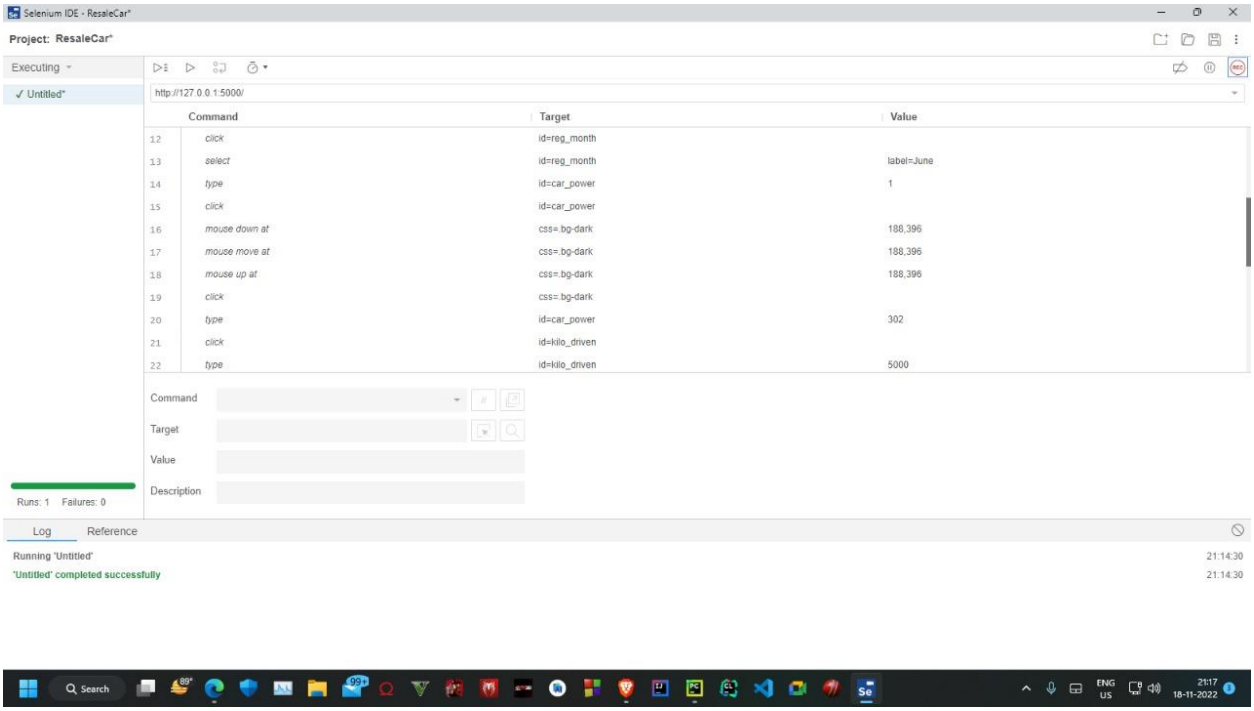
Section	Total Cases	Not Tested	Fail	Pass
Checking the structures	3	0	0	3
Predicting output	1	0	0	1
Giving input	2	0	0	2

9.Results

9.1 Performance Metrics






Selenium IDE - ResaleCar*
Project: ResaleCar*
Executing
✓ Untitled*
http://127.0.0.1:5000/
Command
Target
Value
12 click id=reg_month
13 select id=reg_month label=June
14 type id=car_power 1
15 click id=car_power
16 mouse down at css=.bg-dark 188,396
17 mouse move at css=.bg-dark 188,396
18 mouse up at css=.bg-dark 188,396
19 click css=.bg-dark
20 type id=car_power 302
21 click id=kilo_driven
22 type id=kilo_driven 5000
Command
Target
Value
Description
Runs: 1 Failures: 0
Log
Reference
Running 'Untitled' 21:14:30
'Untitled' completed successfully 21:14:30
Search
ENG
US
21:17
18-11-2022

10.Advantages and Disadvantages

Advantages

- This website makes the buying of resale cars very easy to the users.
- Users will not waste time by asking and collecting information from various places

Disadvantages

- This model can only give an idea to the user about which car they can buy according to their budget, so this prediction may not be 100% true.

11.Conclusion

Since many of the customers can afford to purchase brand-new cars, they will undoubtedly spend a lot of money and time gathering information about used cars, and even after wasting all of this time, it is still uncertain whether the price at which they are purchasing a used car will be accurate or not. As a result, this system will be very helpful in assisting the users to obtain an overall understanding of the prices of all used cars. Additionally, this system can be used by sellers who are unsure of the exact price at which they should list their used cars.

12. Future Scope

In the future, we're going to add a chat feature to our website to quickly answer user questions, and we also have plans to use user data to suggest cars based on past searches.

13. Appendix

GitHub :
https://github.com/IBM-EPBL/IBM-Project-26948-1660041142

Video Link:
https://youtu.be/0LMUuryrkoI

Source Code Link:
https://colab.research.google.com/drive/1qepEk_vjLUnL8TKCHiF9CezhWnB1re7S?usp=sharing