TEAM ID-PNT2022TMID15996

# 

# 1.INTRODUCTION

* 1. **PROEJCT OVERVIEW**

Determining whether the listed price of a used car is a challenging task, due to the many factors that drive a used vehicle’s price on the market. The focus of this project is developing machine learning models that can accurately predict the price of a used car based on its features, in order to make informed purchases. We implement and evaluate various learning methods on a dataset consisting of the sale prices of different makes and models . We will compare the performance of various machine learning algorithms like Linear Regression, Ridge Regression, Lasso Regression, Elastic Net, Decision Tree Regressor and choose the best out of it. Depending on various parameters we will determine the price of the car. Regression Algorithms are used because they provide us with continuous value as an output and not a categorized value because of which it will be possible to predict the actual price a car rather than the price range of a car. User Interface has also been developed which acquires input from any user and displays the Price of a car according to user’s inputs.

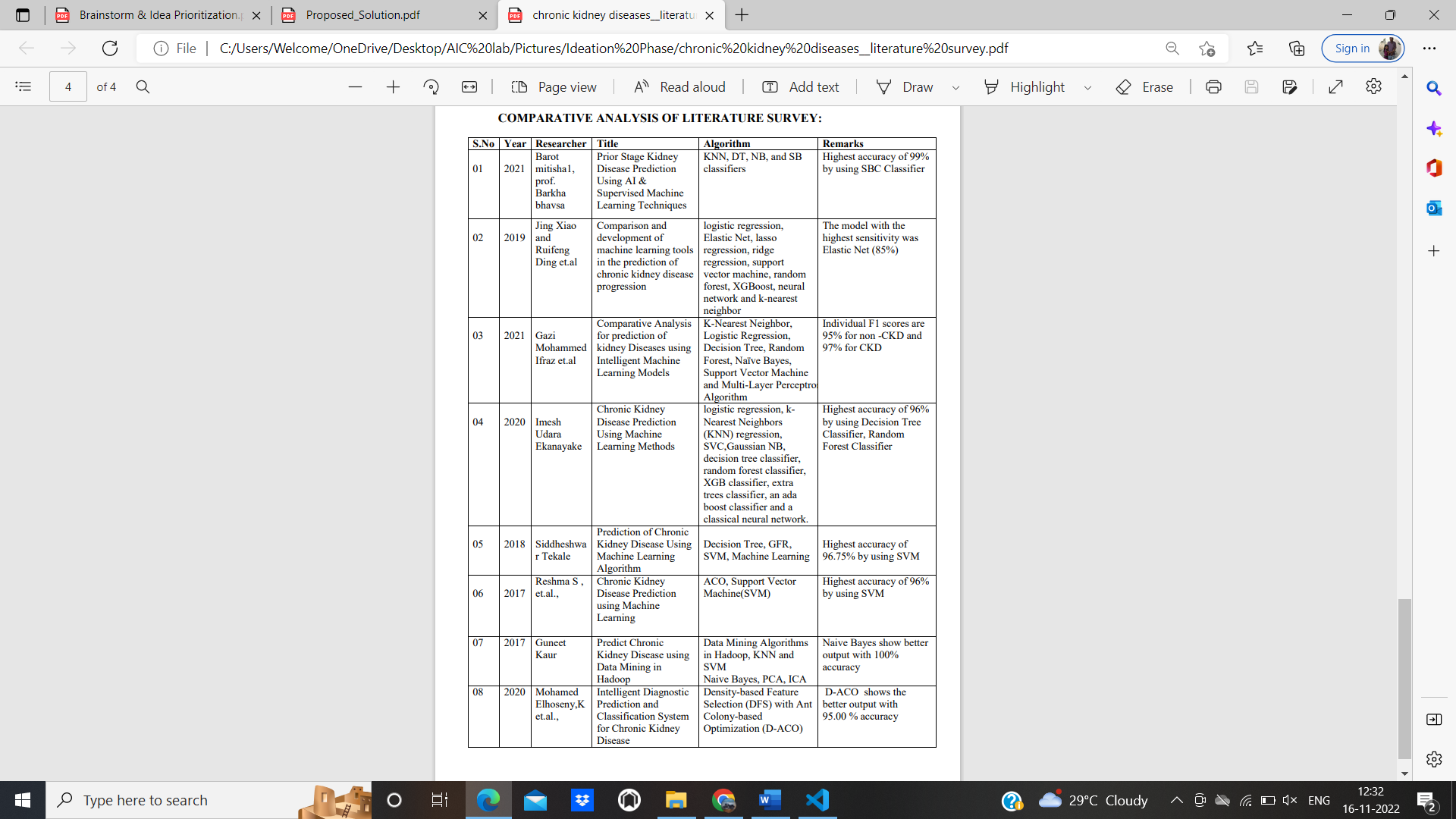
# PURPOSE

The vehicle price estimate has been made with accuracy using a variety of features and criteria as well as the experience of industry professionals. The most essential component of Predictions include car make and model, usage time, and mileage. Due to frequent changes in fuel prices, the gasoline type used in the car and fuel consumption per mile have a significant impact on the price of a vehicle. The price of a car will also be influenced by many aspects such as the outside colour, door count, type of transmission, dimensions, safety, air conditioning, interior, and whether or not it includes navigation. We used a variety of strategies and techniques in this paper to increase the accuracy of the used vehicle price prediction.

# 2.LITERATURE SURVEY

* 1. **EXISITING PROBLEM**

The below table contains the list of articles that includes the existing problem.



# Table 2.1

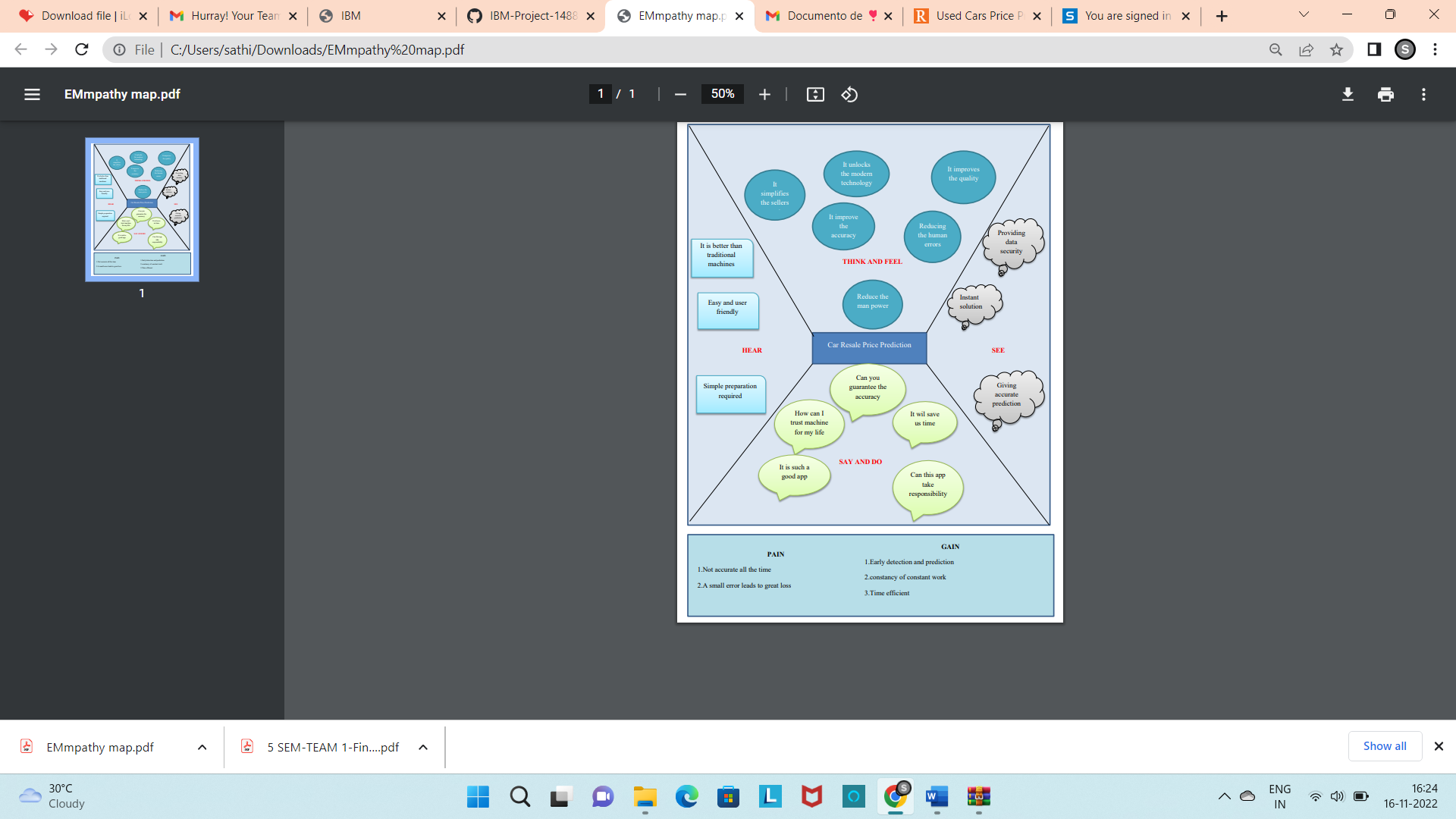
* 1. **REFERENCES**
     1. Saurabh Pal - Chronic Kidney Disease Prediction Using Machine Learning Methods.Issue:16,August 2022
     2. Vineeta Gulatiand Neeraj Raheja - Comparative Analysis for prediction of kidney Diseases using Intelligent Machine Learning Models.Issue:2021
     3. Barot mitishal - Prior Stage Kidney Disease Prediction Using AI & Supervised Machine Learning.Issue:12,Dec 2021
     4. Jing Xiao and Ruifeng Ding et.al - Comparison and development of machine learning tools in the prediction of chronic kidney disease progression.Issue:2019
     5. Siddheshwar Tekale, Pranjal Shingavi et.al - Prediction of Chronic Kidney Disease Using Machine Learning Algorithm.Issue:10.Oct 2018
     6. 6.Sameerchand Pudaruth, “Predicting the Price of Used Cars using Machine Learning Techniques” ;(IJICT 2014).
     7. 7. Enis Gegic, Becir Isakovic, Dino Keco, Zerina Masetic, Jasmin Kevric,” Car Price Prediction Using Machine

# PROBLEM STATEMENT DEFINITION

The research objective of this study is to predict used cars prices in Dubai using data mining techniques, by scraping data from websites that sell used cars, and analysing the different aspects and factors that lead to the actual used car price valuation. To enable consumers to know the actual worth of their car or desired car, by simply providing the program with a set of attributes from the desired car to predict the car price.The purpose of this study is to understand and evaluate used car prices in the UAE, and to develop a strategy that utilizes data mining techniques to predict used car prices.

# 3.IDEATION AND PROPOSED SOLUTION

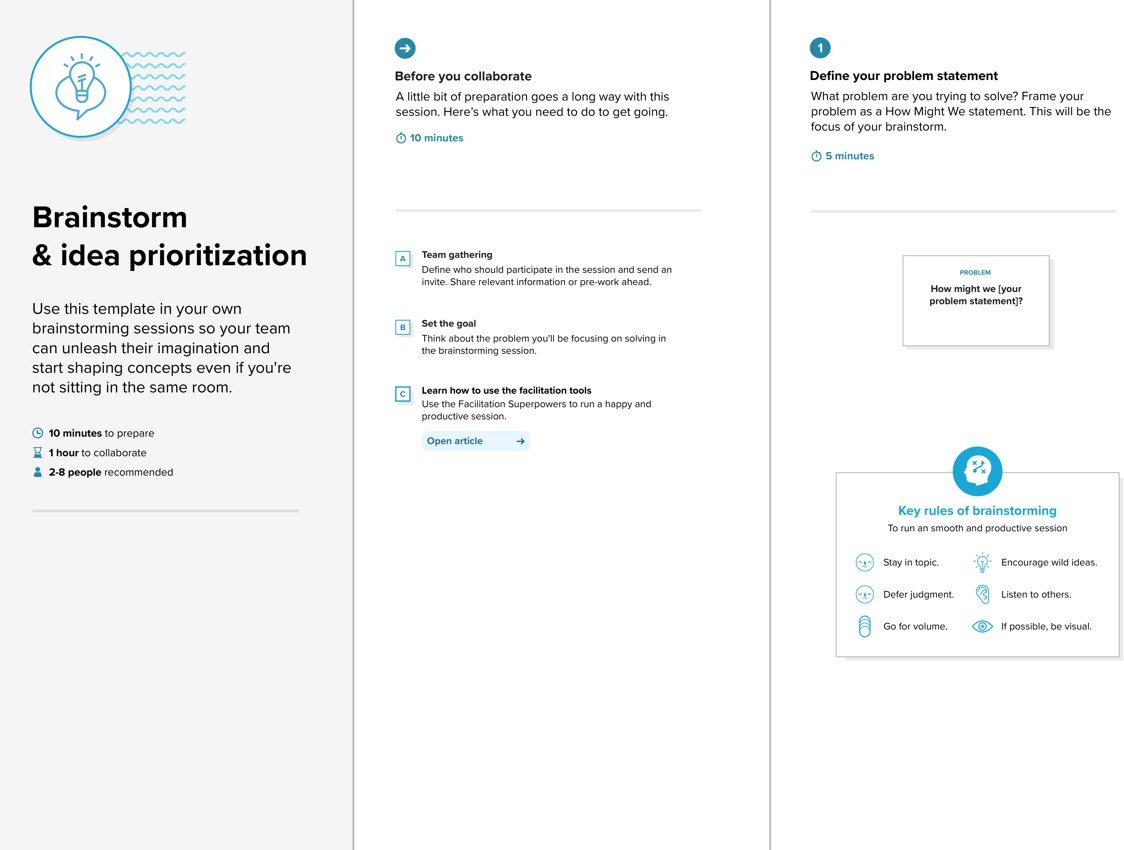
* 1. **EMPATHY MAP CANVAS**



# Fig 3.1

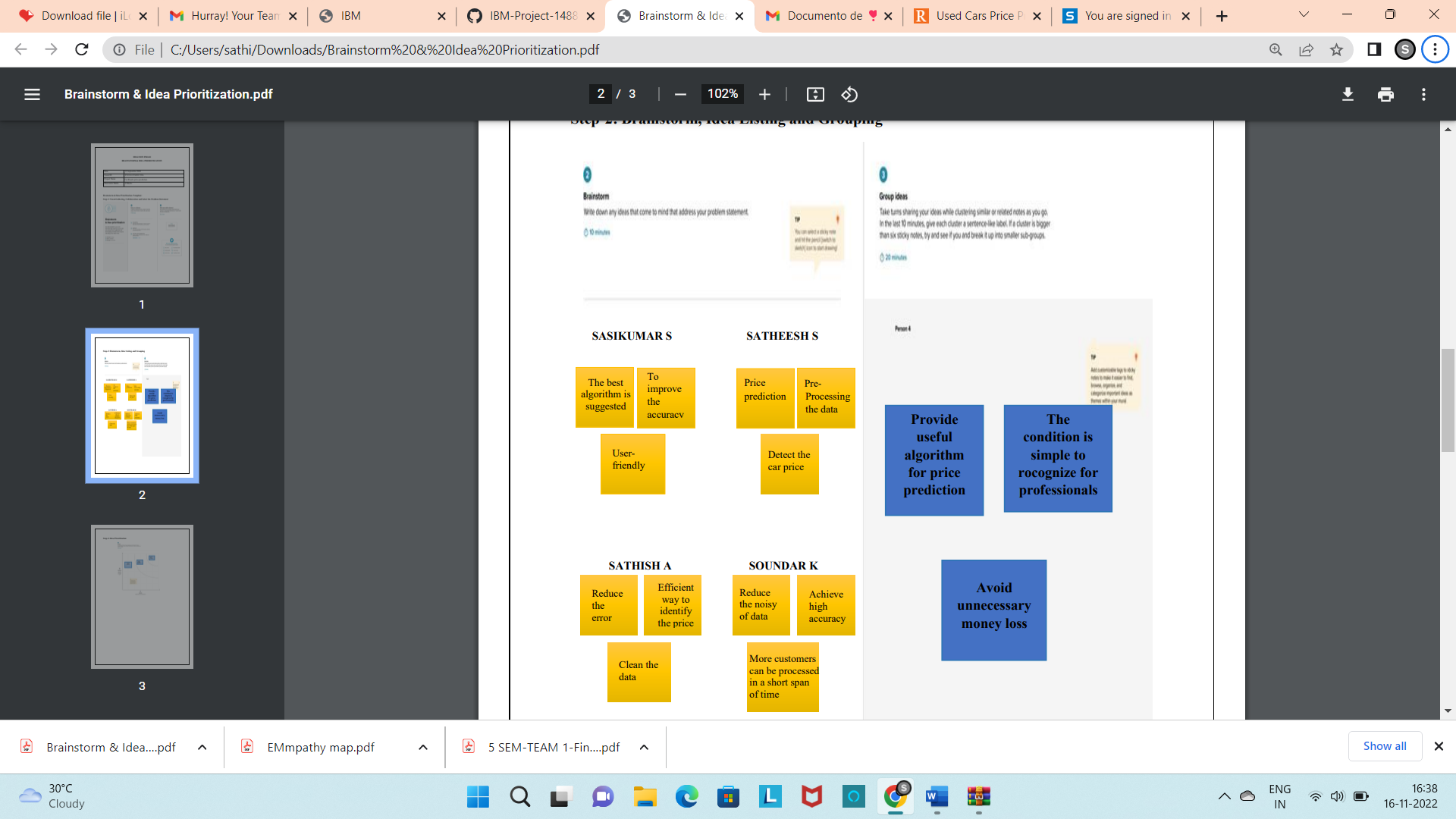
* 1. **IDEATON AND BRAINSTORMING Brainstorm and Idea Prioritization Template:**

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



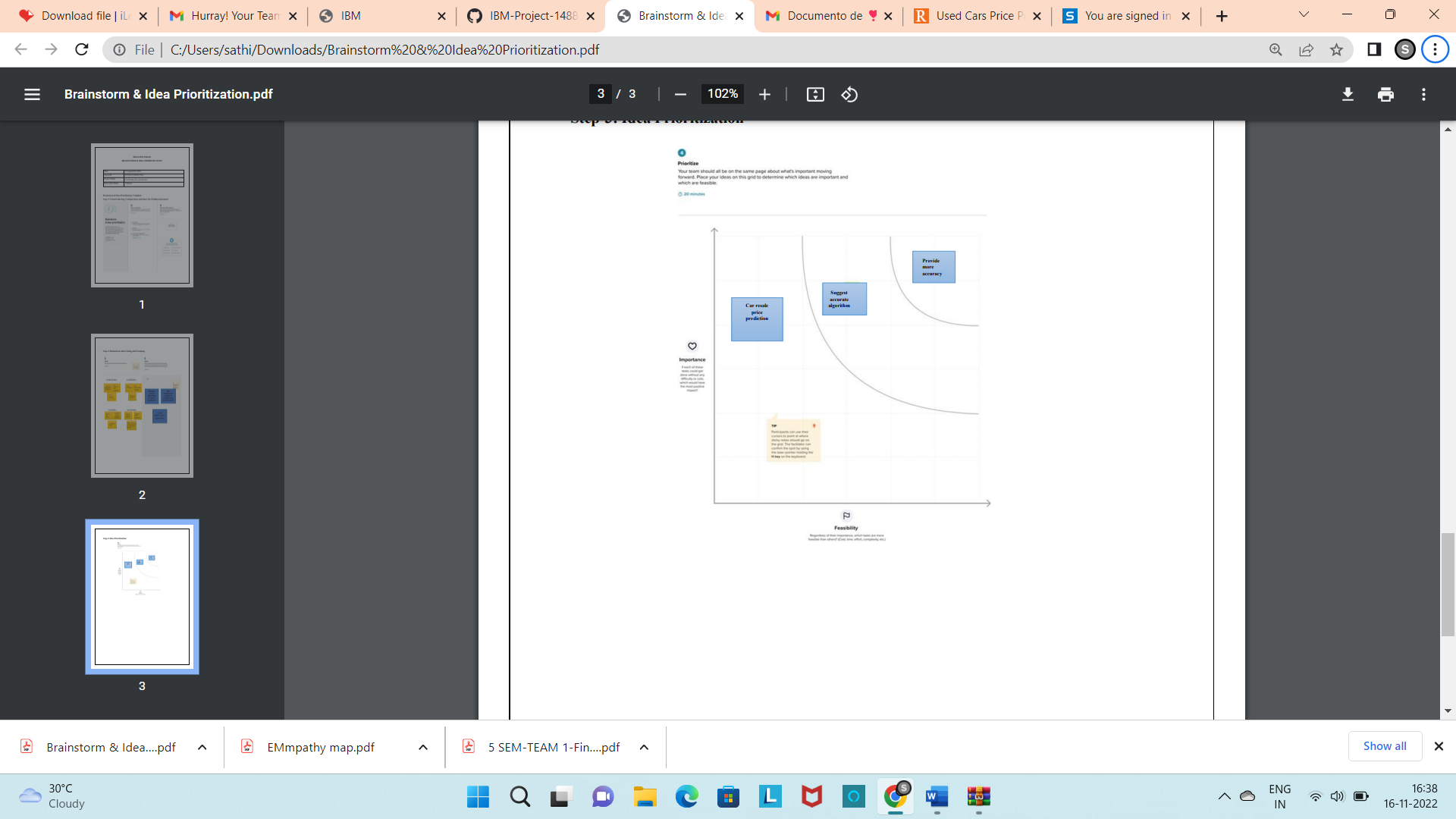
**Fig 3.2.1**

# Step-2: Brainstorm, Idea Listing and Grouping



**Fig 3.2.2**

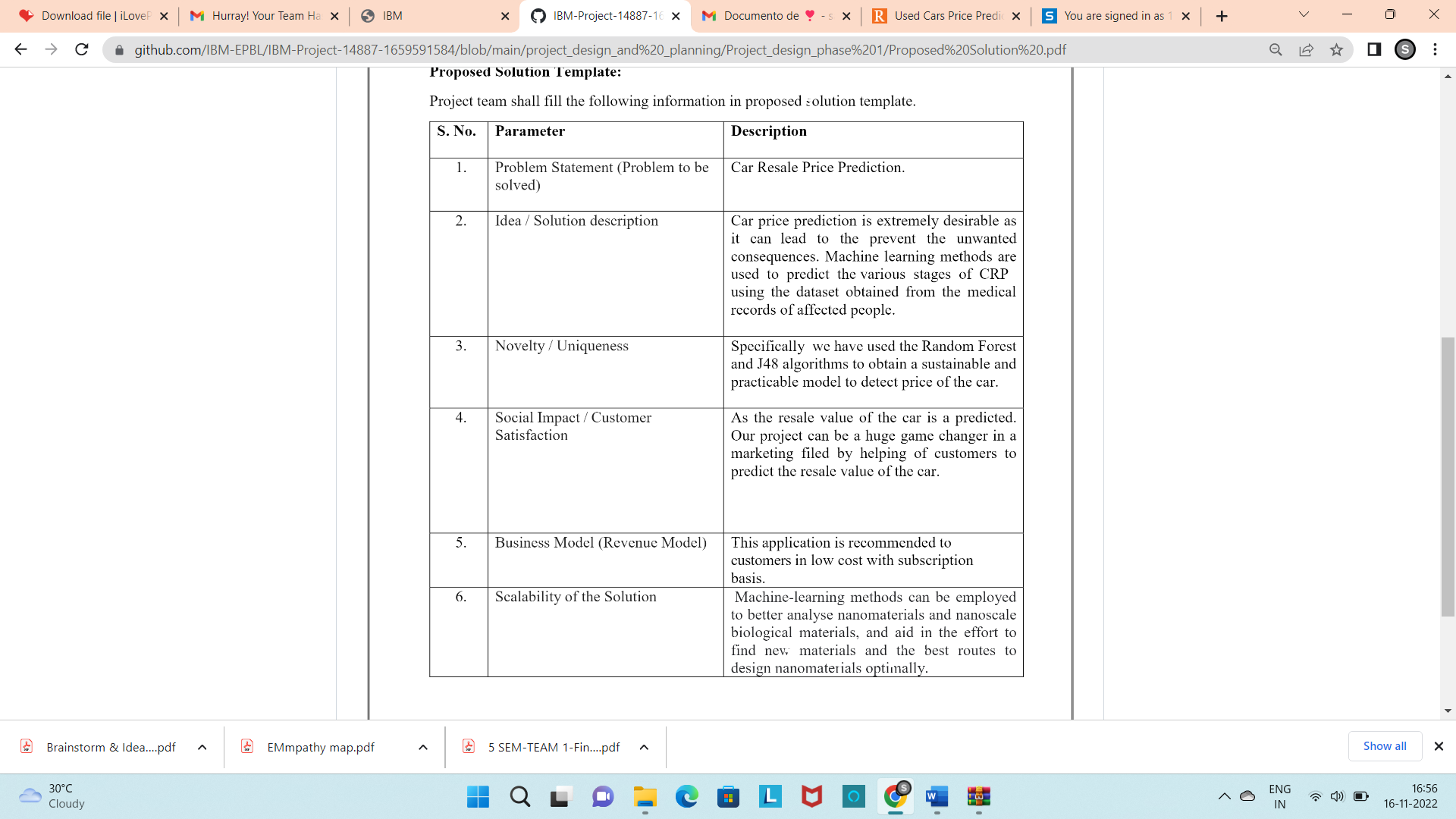
# Step-3: Idea Prioritization



**Fig 3.2.3**

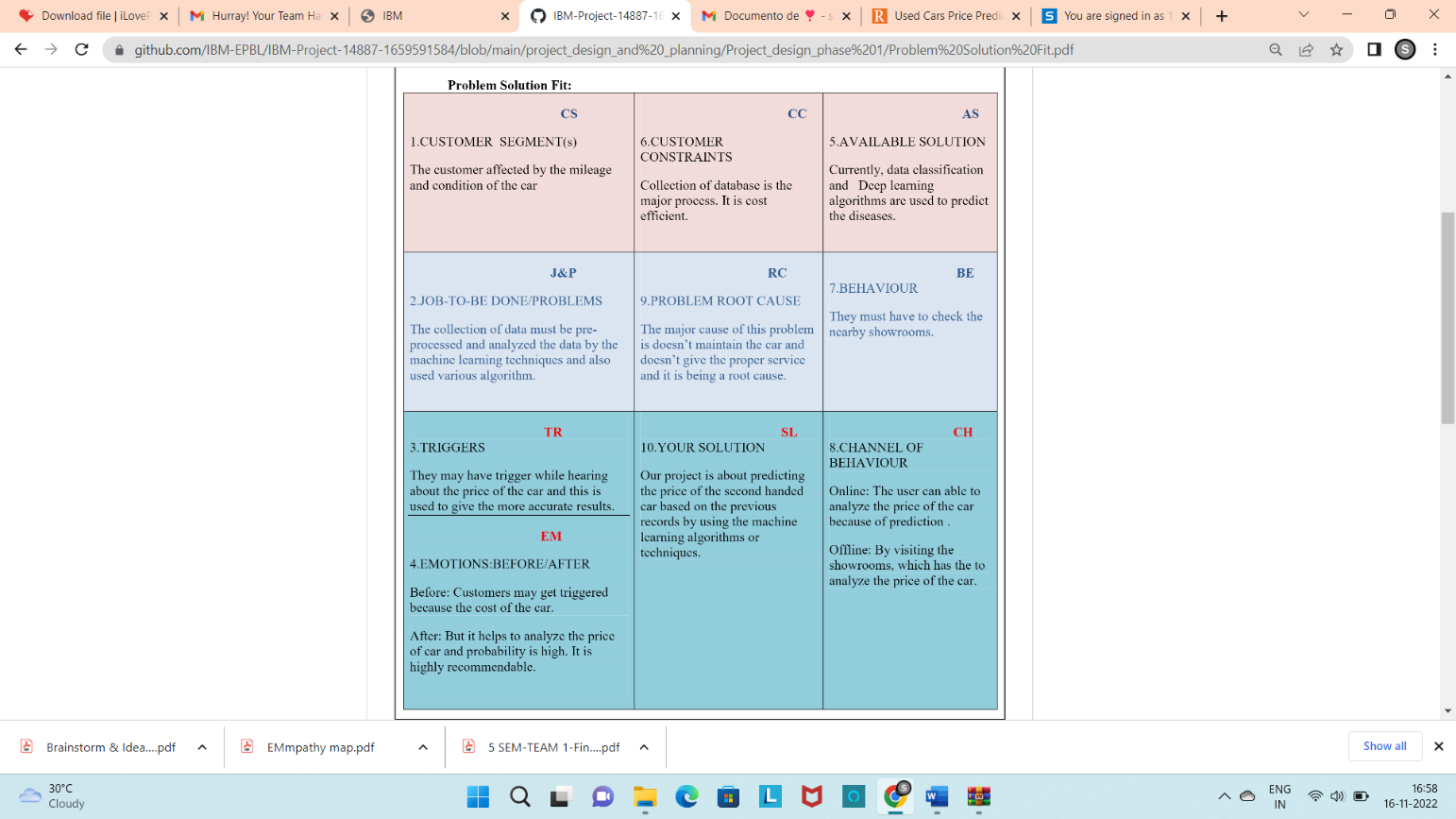
# PROPOSED SOLUTION

Project team shall ﬁll the following information in proposed solution template.



# Table 3.3

* 1. **PROPOSED SOLUTION FIT**

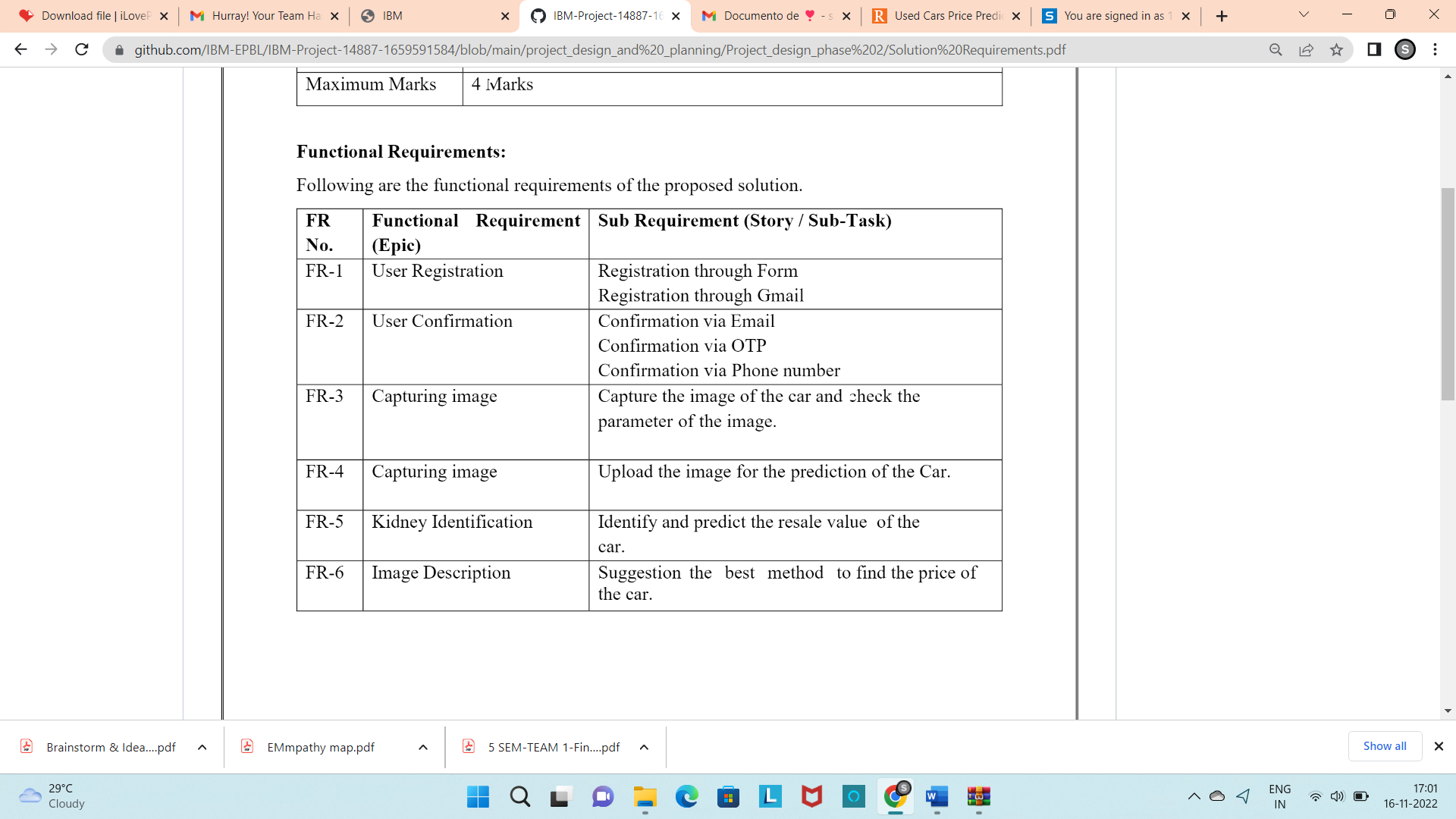


# Fig 3.4

**4.REQUIREMENT ANALYSIS**

# FUNCTIONAL REQUIREMENTS:

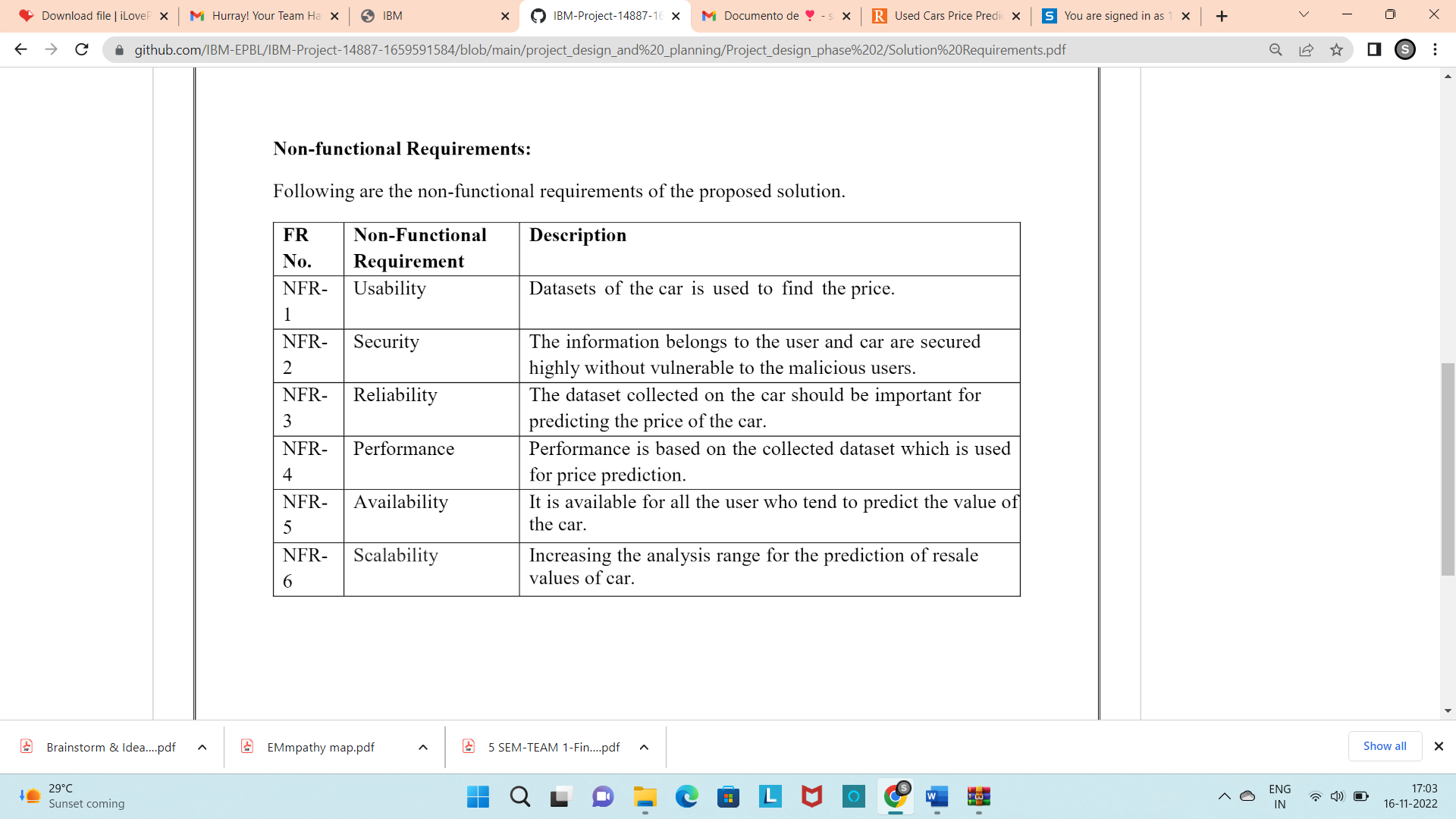
Following are the functional requirements of the proposed solution.



# Fig 4.1

* 1. **NON-FUNCTIONAL REQUIREMENTS:**

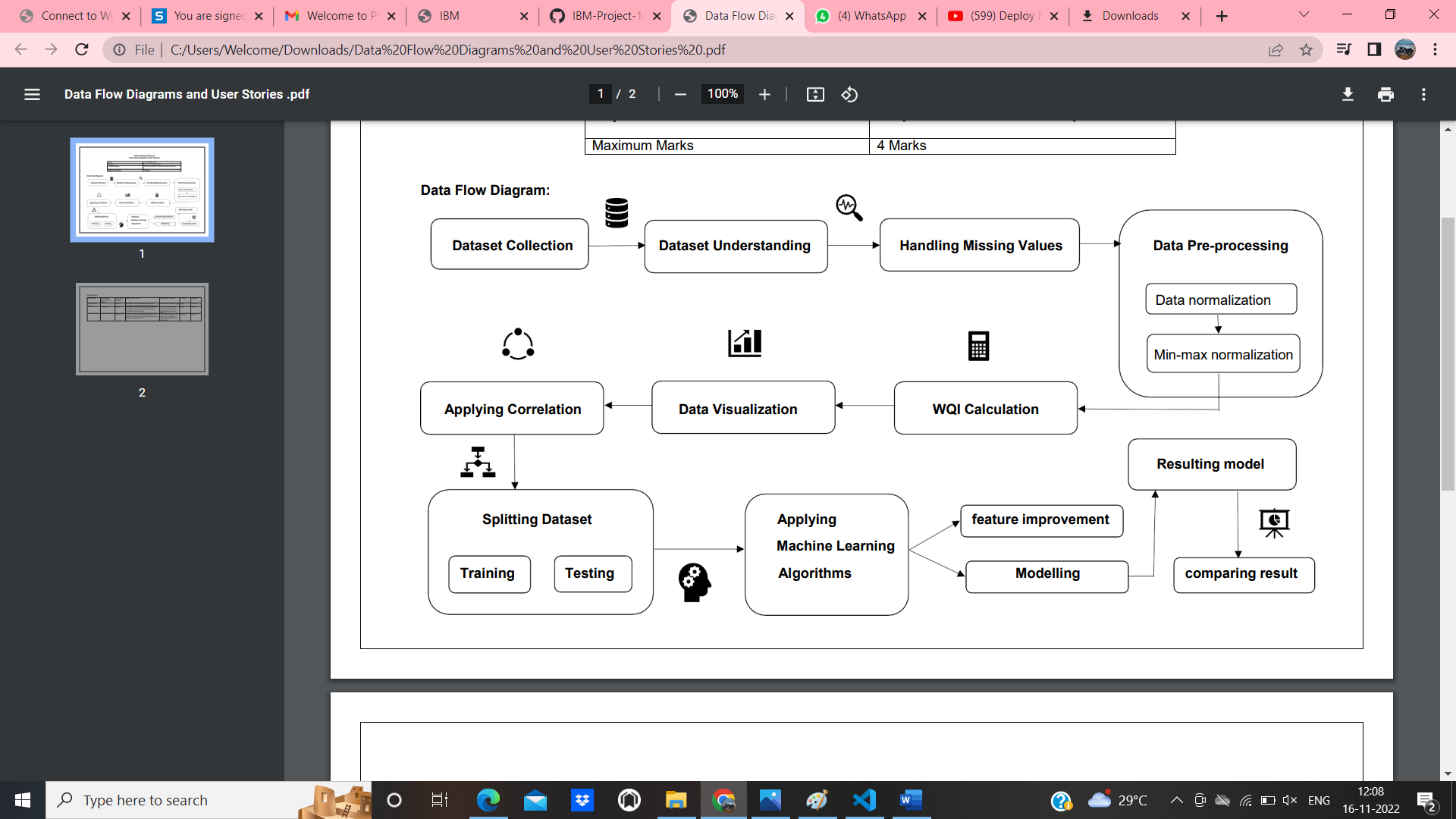
Following are the non-functional requirements of the proposed solution.



# Fig 4.2

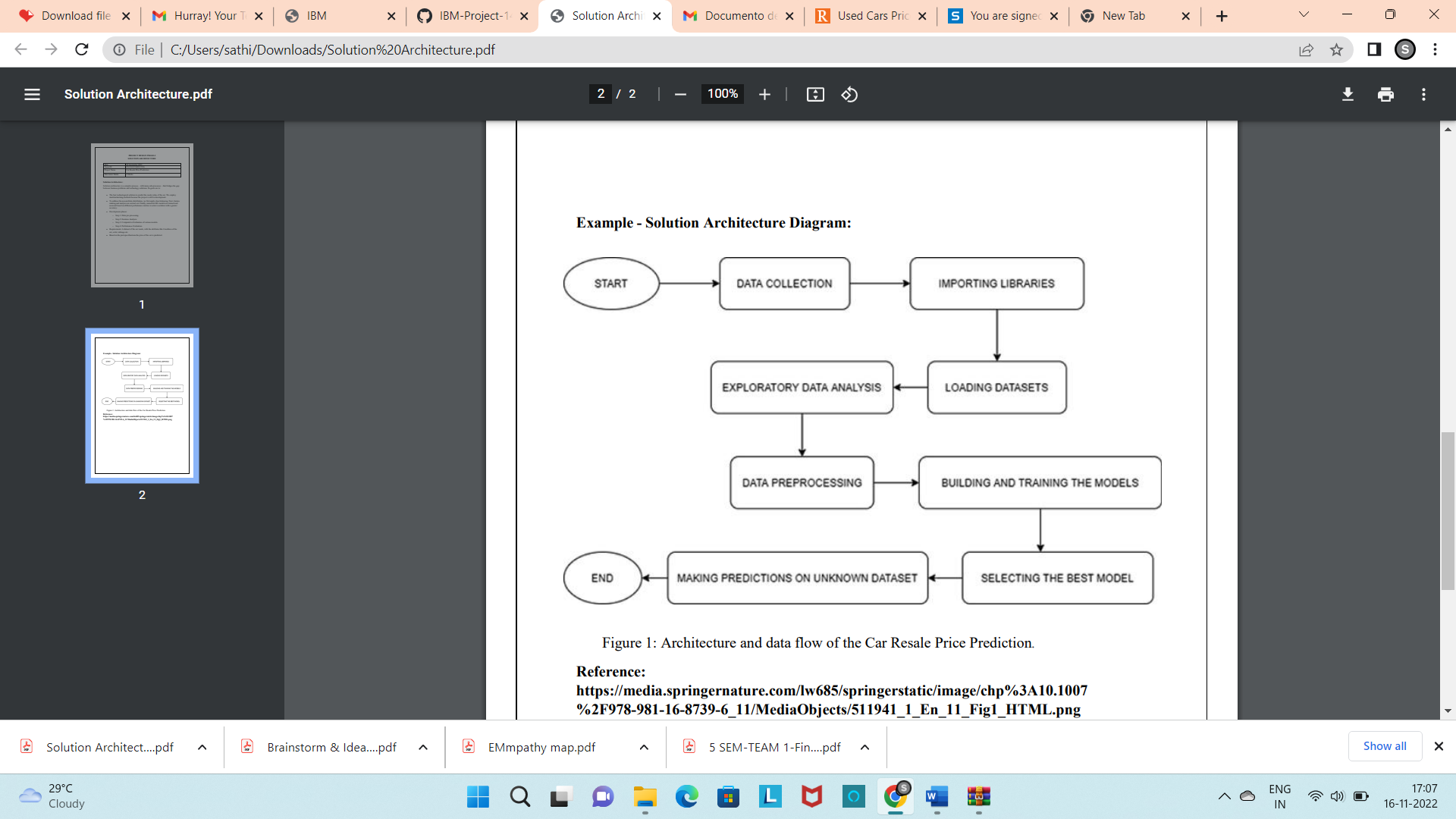
**5.PROJECT DESIGN**

# DATA FLOW DIAGRAMS



**Fig 5.1**

# SOLUTION AND TECHNICAL ARCHITECTURE



**Fig 5.2**

# Table-1: Components & Technologies:

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Component** | **Description** | **Technology** |
| 1. | User Interface | User interacts by using web user interface. | HTML, CSS and  Python Flask |
| 2. | Application Logic-1 (Login) | User can able to login if that person is already registered to the site. | HTML, CSS and  Python Flask |
| 3. | Application Logic-2 (Register) | User needs to be registered if that person is new to the site. | HTML, CSS and  Python Flask. |
| 4. | Application Logic- 3(Reporting Form) | User needs to click on the reporting form in order to get the prediction result | Front end- HTML, CSS and Python Flask.  Back end – Query Languages, Python. |
| 5. | Database | Data Type-String, Numeral values. | Query Languages such as MySQL, NoSQL etc. |
| 6. | Cloud Database | Database Service on Cloud. | IBM DB2, IBM Cloud  ant etc. |
| 7. | File Storage | File storage requirements. | Local File-system. |
| 8. | External API-1 | Anyone can access the details with some restrictions to the personal details of other users. | Web API. |
| 9. | External API-2 | Accessibility. | Aadhar API. |
| 10. | Machine Learning Model | Predict the result based on the training and testing dataset. | Data Recognition Model, etc. |
| 11. | Infrastructure (Server / Cloud) | Application Deployment on Local System. | Local System. |

**Table 5.2.1**

# Table-2: Application Characteristics:

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Characteristics** | **Description** | **Technology** |
| 1. | Open-Source Frameworks | Frameworks are used for predictive data analysis, providing clear and actionable error messages. | Tensor ﬂow, Sci-kit learn, Keras. |
| 2. | Security Implementations | OTP will be sent to the registered email id. Unauthorized users could not access the user’s details. | Email Veriﬁcation. |
| 3. | Scalable Architecture | Scalability is improved for implementing the three-tier architecture. | Three tier architecture. |
| 4. | Availability | For enhancing the high availability, load balancer is needed. | Load Balancer. |
| 5. | Performance | The model could be able to process large number of datasets. | Load Balancer. |

**Table 5.2.2**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **5.3 USER STORIES** | | | | | | | |
|  | **User Type** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Release** |
| Customer (Web user) | Registration | USN-1 | As a user, I can register for the site by entering my email, password, and conﬁrming my  email id. | I can access my account and view the details. | High | Sprint-1 |
|  | Conﬁrmation | USN-2 | As a user, I will receive conﬁrmation email once I have registered for the site. | I can receive conﬁrmati on email and click  conﬁrm. | High | Sprint-1 |
|  | Login | USN-3 | As a user, I can login into the site by clicking onto the login link. | I can successful ly login to the page and my details will  be shown. | Low | Sprint-2 |
|  | Dashboard | USN-4 | As a user, I can access my dashboard. | I can modify the details in the  dashboard. | Medi um | Sprint-2 |
|  | Homepage | USN-5 | As a user, I can view the contact details and required information. | Based on user requirement s,Contents are categorize  d. | High | Sprint-3 |
| Customer Care Executive | Help | USN-6 | As a user, I could contact the site owner if I faced any issues. | Report issues option will be  provided. | High | Sprint-3 |
| **Table 5.3** | | | | | | | |

# 6.PROJECT PLANNING AND SCHEDULING

* 1. **SPRINT PLANNING AND ESTIMATION**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement**  **(Epic)** | **User Story**  **Number** | **User Story/Task** | **Story Points** | **Priority** | **Team Members** |
| Sprint-1 | Data Collection | USN-1 | Collect the suitable dataset for predicting the price of the car. | 10 | High | Sathish A |
| Sprint-1 | Data Pre- Processing | USN-2 | Datasets are transformed into useful  format. | 7 | Medium | Sathish A |
| Sprint-2 | Model Building | USN-3 | Calculate the  Index values | 10 | High | Satheesh S |
| Sprint-2 |  | USN-4 | Splitting the Model into Training and Testing from the overall  dataset. | 7 | Medium | Satheesh S |
| Sprint-3 | Training and Testing | USN-5 | Train the Model using Regression algorithm and testing the performance of  the model. | 10 | High | Sasikumar S |
| Sprint-3 | Application Building | USN-6 | Build the HTML and python  code | 7 | Medium | Sasikumar S |
| Sprint-4 |  | USN-7 | Run Flask App | 10 | High | Soundar K |
| Sprint-4 | Implementation of the  Application | USN-8 | Deploy the model on IBM  cloud. | 7 | Medium | Soundar K |

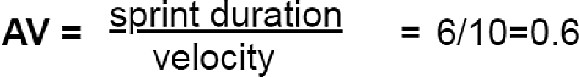
# Fig 6.1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 | 10 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 8 | 29 Oct 2022 |
| Sprint-2 | 10 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 7 | 05 Nov 2022 |
| Sprint-3 | 10 | 6 Days | 06 Oct 2022 | 12 Nov 2022 | 8 | 12 Nov 2022 |
| Sprint-4 | 10 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 7 | 19 Nov 2022 |

**Fig 6.2**

# VELOCITY:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit (story points per day.



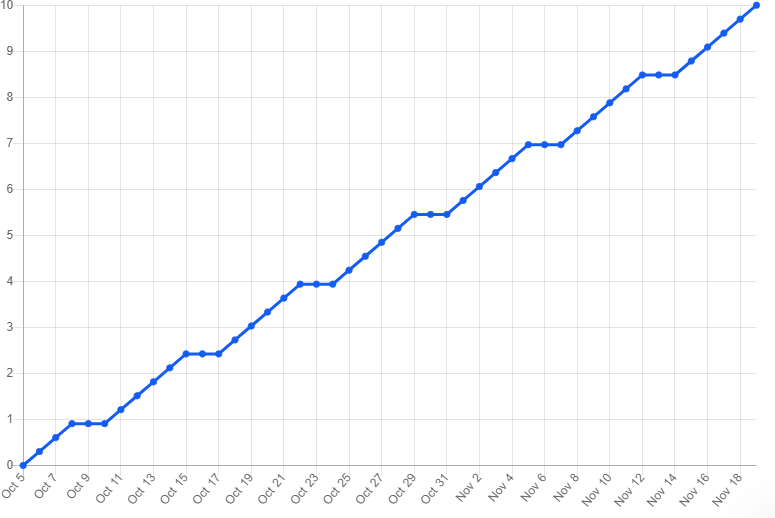
# SPRINT DELIVERY SCHEDULE

|  |  |  |
| --- | --- | --- |
| **TITLE** | **DESCRIPTION** | **DATE** |
| **Literature Survey & Information Gathering** | Literature survey on the selected project & gathering information by referring the technical papers, research publications, journals etc. | 1 SEPTEMBER 2022 |
| **Prepare Empathy Map** | Prepare Empathy Map Canvas to capture the user Pains and Gains, prepare list of problem Statements that are to be solved by this project. | 7 SEPTEMBER 2022 &  9 SEPTEMBER 2022 |
| **Ideation** | List the ideas by organizing a brain storming session and prioritize the top three ideas based on the feasibility and importance. | 15 SEPTEMBER 2022 |
| **Proposed Solution** | Prepare the proposed solution document, which includes novelty, feasibility of idea, revenue model, social impact, scalability of solution, etc. | 22 SEPTEMBER 2022 |
| **Problem Solution Fit** | Prepare problem - solution ﬁt document. | 30 SEPTEMBER 2022 |
| **Solution Architecture** | Prepare solution architecture document. | 30 SEPTEMBER 2022 |
| **Customer Journey** | Prepare the customer journey maps to understand the user interactions and experiences with the application (entry to exit). | 6 OCTOBER2022 |
| **Functional Requirement** | Prepare the functional requirement document. | 11 OCTOBER 2022 |

|  |  |  |
| --- | --- | --- |
| **Data Flow Diagrams** | Draw the data ﬂow diagrams and submit for review. | 11 OCTOBER 2022 |
| **Technology Architecture** | Prepare the technology architecture diagram. | 14 OCTOBER 2022 |
| **Prepare Milestone &Activity List** | Prepare the milestones and activity list of the project. | 21 OCTOBER 2022 |
| **Project Development - Delivery of Sprint-1, 2, 3 &4** | Develop and submit the developed code by testing it. | IN PROGRESS... |

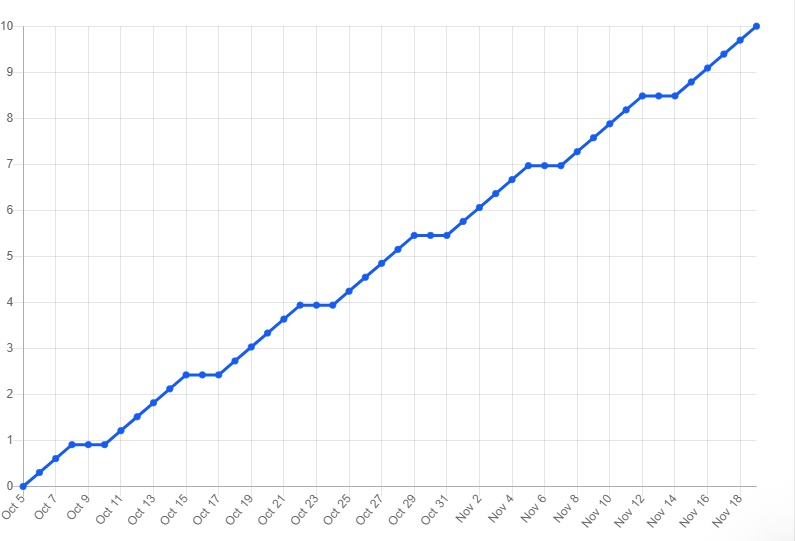
**Fig 6.2**

# REPORTS FROM JIRA BURNDOWN CHART



**Fig 6.3.1**

# BURNUP CHART



**Fig 6.3.2**

# 7.CODING AND SOLUTIONING

* 1. **FEATURE 1(RANDOM FOREST ALGORITHM MODEL)**

Random Forest Classiﬁer is used to train and test the model for detecting the price of the used car with the help of collected and pre- processed dataset collections. NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. Moreover, NumPy forms the foundation of the Machine Learning stack. Pandas is an open-source Python package that is most widely used for data science/data analysis and machine learning tasks. Sea born is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. For a brief introduction to the ideas behind the library, you can read the introductory notes or the paper. Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Create publication quality plots. Make interactive ﬁgures that can zoom, pan, update. EDA is applied to investigate the data and summarize the key insights. It will give you the basic understanding of your data, it is distribution, null values and much more. You can either explore data using graphs or through some python functions.

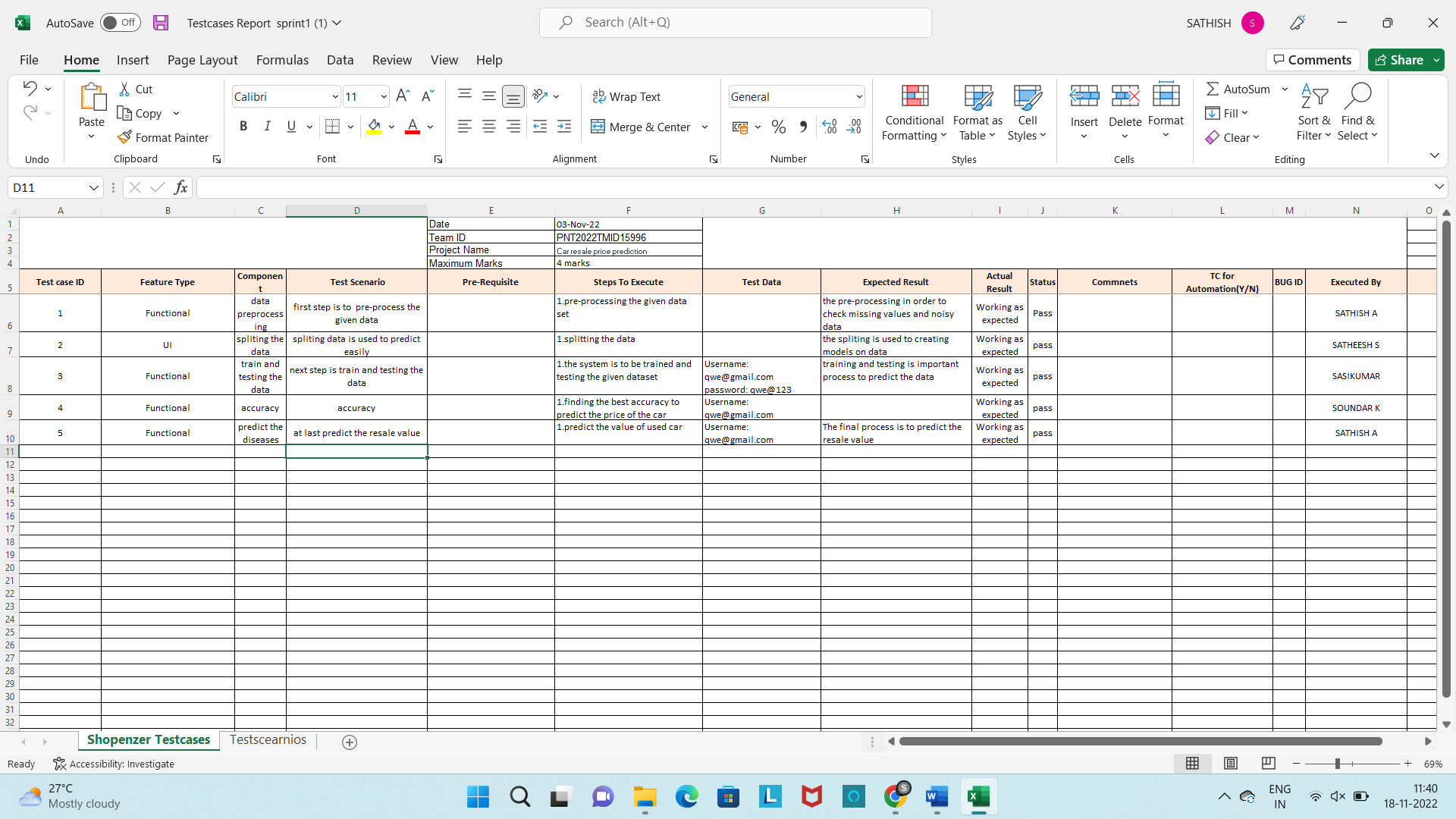
# FEATURE 2(FLASK CONNECTIVITY)

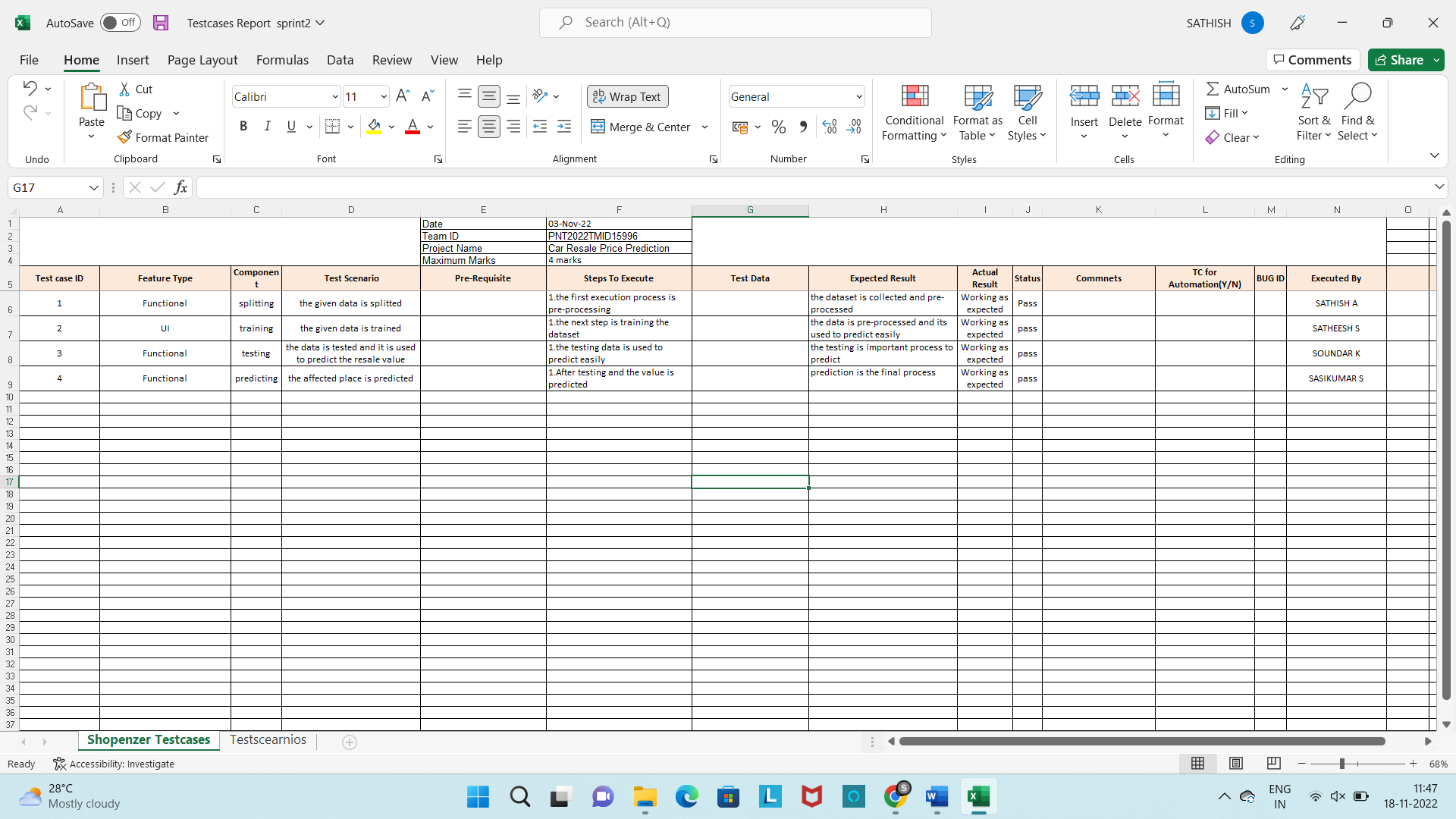
The framework is the basis upon which software programs are built. It serves as a foundation for software developers, allowing them to create a variety of applications for certain platforms. It is a set of functions and predeﬁned classes used to connect with the system software and handle inputs and outputs. It simpliﬁes the life of a developer while giving them the ability to use certain extensions and makes the online applications scalable and maintainable. Flask is a web application framework written in Python. A Web Application Framework or a simply a Web Framework represents a collection of libraries and modules that enable web application developers to write applications without worrying about low-level details such as protocol, thread management, among other examples. Flask is a web application framework written in Python. It was developed by Armin Ronacher, who led a team of international Python enthusiasts called Poocco. Flask is based on the Werkzeg WSGI toolkit and the Jinja2 template engine. Both are Pocco projects. The Web Server Gateway Interface (Web Server Gateway Interface, WSGI) has been used as a standard for Python web application development. WSGI is the speciﬁcation of a common interface between web servers and web applications. Flask is often referred to as a micro-framework. It is designed to keep the core of the application simple and scalable. Instead of an abstraction layer for database support, Flask supports extensions to add such capabilities to the application.

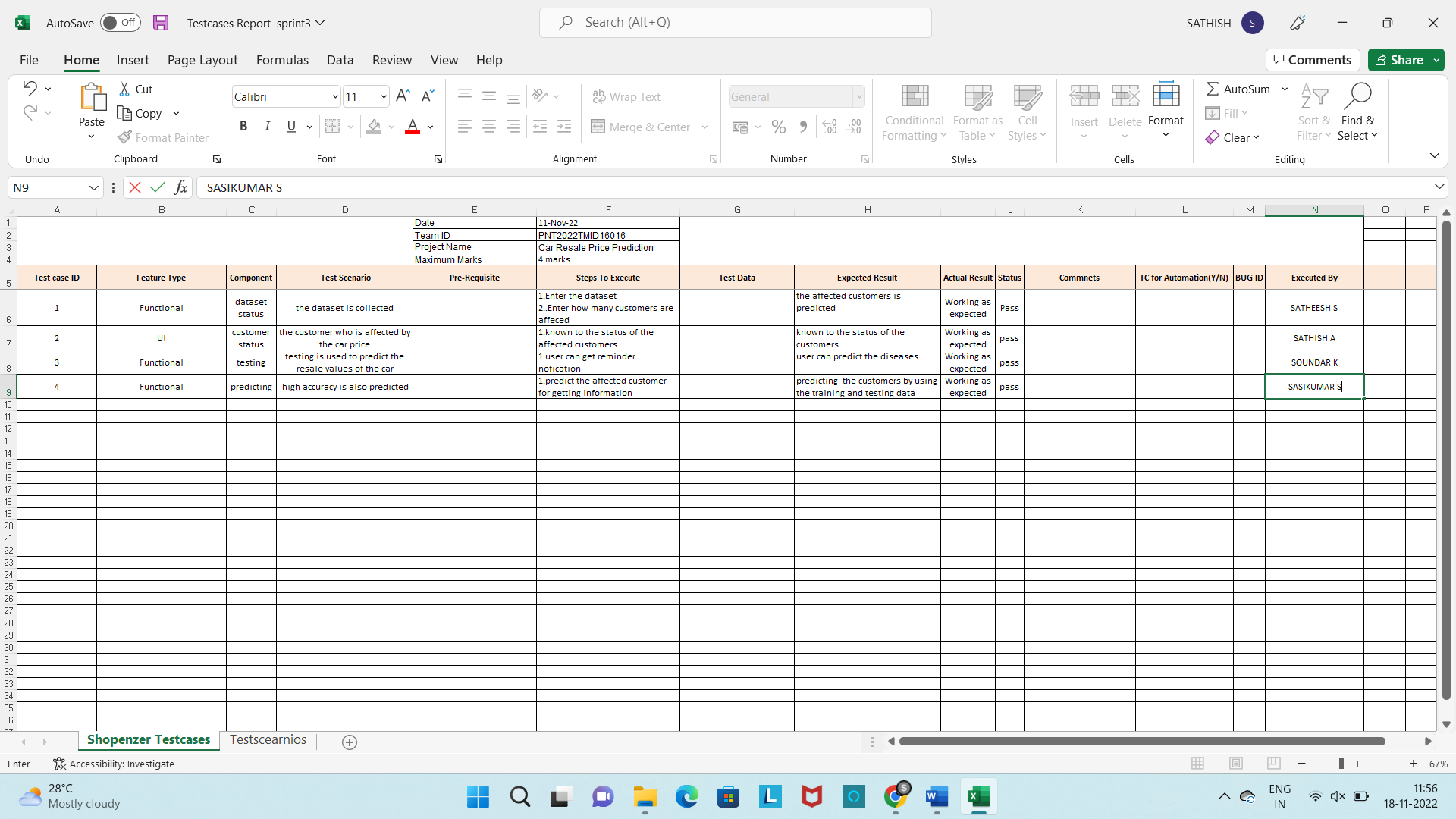
developing web applications using python, implemented on Werkzeug and Jinja2. Advantages of using Flask framework are: There is a built-in development server and a fast debugger provided. The model deployed using Flask is used to predict the price of the car. Hypertext markup language (HTML) is the basic language used to create documents for the [Web](http://www.linfo.org/web.html) and, along with HTTP (hypertext transfer protocol) and URLs (universal resource locators), is one of the three main [protocols](http://www.linfo.org/protocol.html) of the Web. Hypertext is text that contains [hyperlinks](http://www.linfo.org/hyperlink.html). A hyperlink is an automated cross-reference to another location on the same document or to another document which, when selected by a user, causes the [computer](http://www.linfo.org/computer.html) to display the linked location or document within a concise period. A [markup](http://www.linfo.org/markup_language.html) [language](http://www.linfo.org/markup_language.html) is a set of tags that can be embedded in digital text to provide additional [information](http://www.linfo.org/information.html) about it, including its [content](http://www.linfo.org/content.html), structure and appearance. This information facilitates automated operations on the text, including formatting it for display, searching it and even modifying it. Some type of markup language is employed by every word processing [program](http://www.linfo.org/program.html) and by nearly every other program that displays text, although such languages and their tags are typically hidden from the user.HTML consists of a set of predeﬁned tags that can be embedded in text by web site designers in order to indicate the details of how web pages are rendered (i.e., converted into a ﬁnal, easily usable, form) by [web browsers](http://www.linfo.org/browser.html).

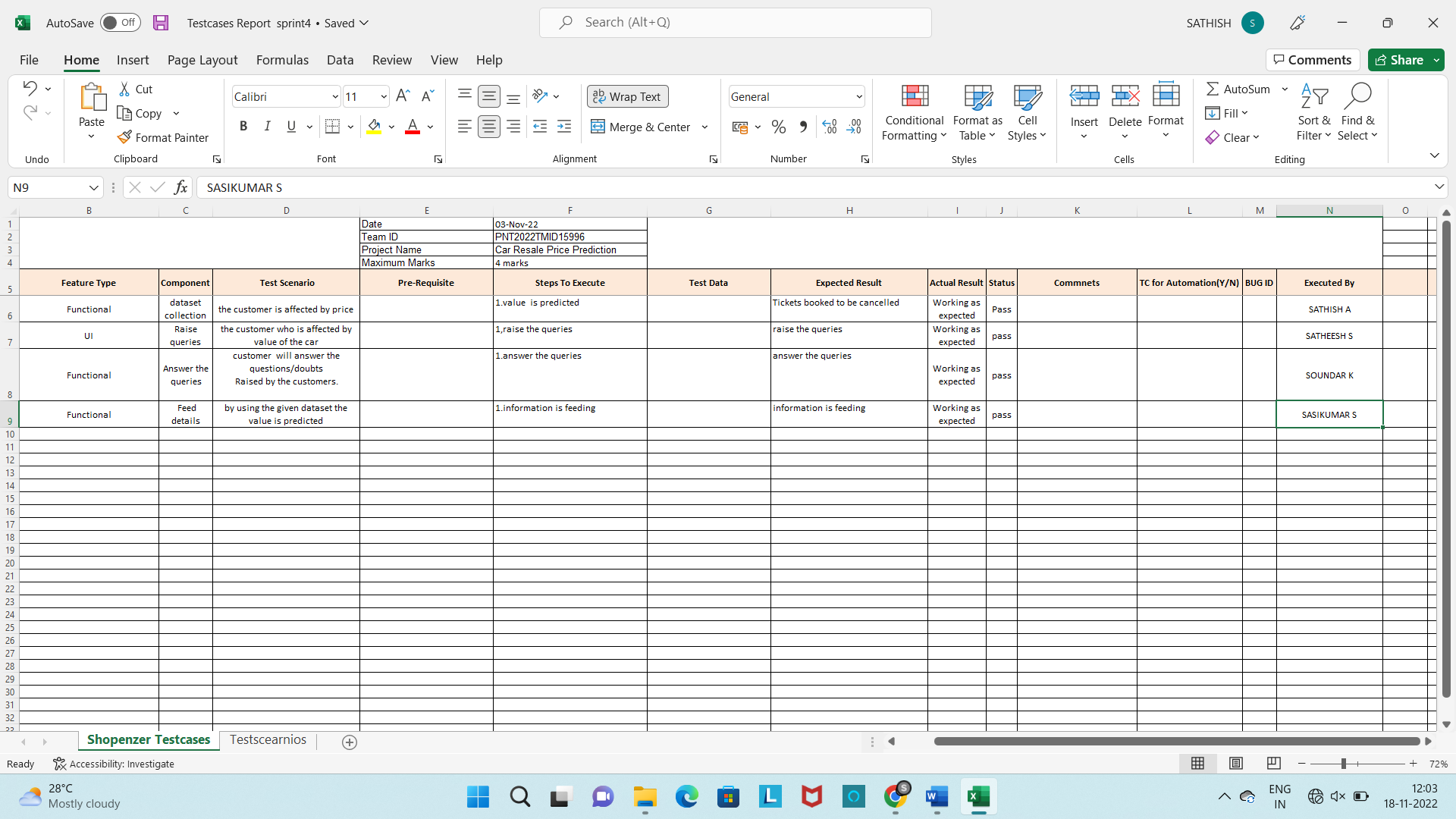
**8.TESTING**

# TEST CASES

****

****





**Fig 8.1**

* 1. **USER ACCEPTANCE TESTING**

# 9.RESULTS

**9.1 PERFORMANCE METRICS**

|  |  |  |
| --- | --- | --- |
| **TITLE** | **DESCRIPTION** | **DATE** |
| **Literature Survey & Information Gathering** | Literature survey on the selected project & gathering information by referring the technical papers, research publications, journals  etc. | 1 SEPTEMBER 2022 |
| **Prepare Empathy Map** | Prepare Empathy Map  Canvas to capture the user Pains & Gains, prepare list of problem Statements that are to be solved by this project. | 7 SEPTEMBER 2022 &  9 SEPTEMBER 2022 |
| **Ideation** | List the ideas by organizing a brainstorming session and prioritize the top 3 ideas based on the feasibility & importance. | 15 SEPTEMBER 2022 |
| **Proposed Solution** | Prepare the proposed  Solution document, which includes novelty, feasibility of idea, revenue model, social impact, scalability of  solution, etc. | 22 SEPTEMBER 2022 |
| **Problem Solution Fit** | Prepare problem -  solution fit document. | 30 SEPTEMBER 2022 |
| **Solution Architecture** | Prepare solution  Architecture document. | 30 SEPTEMBER 2022 |

|  |  |  |
| --- | --- | --- |
| **Customer Journey** | Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to  exit). | 6 OCTOBER 2022 |
| **Functional Requirement** | Prepare the functional requirement  document. | 11 OCTOBER 2022 |
| **Data Flow Diagrams** | Draw the data flow diagrams and submit for review. | 11 OCTOBER 2022 |
| **Technology architecture** | Prepare the technology architectu re diagram. | 14 OCTOBER 2022 |
| **Prepare Milestone &Activity List** | Prepare the milestones &activity list of the  project. | 21 OCTOBER 2022 |
| **Project Development - Delivery of Sprint-1, 2, 3 &4** | Develop & submit the developed code by testing it. | IN PROGRESS... |

# Table 9.1

**10.ADVANTAGES AND DISADVANTAGES**

# ADVANTAGES:

* You can get used cars at a lower price You may not have to go into debt for used cars
* You can pay off used cars quicker than new cars • Used cars have lower depreciation
* You don't have to be afraid of minor damages that much • Insurance premiums for used cars are often lower
* More unlikely that people steal used cars
* Buying used cars is good for our planet
* Used cars can give you a vintage feeling Buying antique cars can make sense as an investment
* Buying a used car makes sense in college
* Buying a used car is great if you have a low income Used cars are quite popular among retirees

**DISADVANTAGES:**

* Used cars are less reliable
* Higher maintenance costs for used cars
* Used cars often don't look nice
* You may have to spend lots of time on repairs
* Used cars are not that popular in society
* People may make fun of you
* Used cars may smell bad
* Often no warranty for used cars
* Used cars may make strange sounds
* Older cars are often quite loud
* Used cars are often not as comfortable as new ones
* You may miss the new-car feeling Driving a used car may be bad for your ego

**11.CONCLUSION**

This research offered a scalable framework for used car price prediction in Dubai using data mining and machine learning techniques. To get the benchmark data, the Parse Hub scraping tool was used to crawl the Buyanycar.com website. Training, testing, and assessing a machine learning model results in an effective model.

Random Forest Regressor, Linear Regression, and Bagging Regressor are three machine learning regressors. After pre-processing and transformation, the Random Forest Regressor, which had an accuracy of 95%, triumphed, followed by the Bagging Regressor, which had an accuracy of 88%. Each experiment was carried out in the Google Colab environment in real time. Algorithms in Google Colab required less training time than those in the system's integrated Jupyter notebook and the Anaconda platform.

**12.FUTURE SCOPE**

Deep learning classifiers will be explored in the future when additional data is gathered using various web-scraping methods. We'll test algorithms like Quantile Regression, ANN, and SVM.

The intelligent model will then be included in online and mobile applications for general use.

In addition, after the data gathering phase, the pandemic-related shortages of semiconductors caused a rise in automobile costs and had a significant impact on the used-car market. Therefore, it is necessary to periodically gather and analyse data; ideally, we would use a real-time processing tool.

# 13.APPENDIX

**Source Code:**

# Random Forest Alogirhtm:

import pandas as pd

import numpy as np

import matplotlib as plt

from sklearn. preprocessing import LabelEncoder

import pickle

pip install pandas

df = pd.read\_csv("C:/Users/sathi/Downloads/autos - Copy.csv",encoding='Latin1')

df.head()

df[df.seller != 'gewerblich']

df=df.drop( 'seller', 1)

df[df.offerType != 'Gesuch']

df=df.drop( 'offerType', 1)

df = df[ (df.powerPS > 50) & (df.powerPS < 900) ]

df = df[ (df.yearOfRegistration >= 1950) & (df.yearOfRegistration < 2017)]

df.drop(['name','abtest','dateCrawled','nrOfPictures','lastSeen','postalCode','dateCreated'],axis='columns',inplace=True)

new\_df = df.copy()

new\_df = new\_df.drop\_duplicates(['price','vehicleType','yearOfRegistration','gearbox','powerPS','model','kilometer','monthOfRegistration','fuelType','notRepairedDamage'])

new\_df.gearbox.replace(('manuell','automatik'),('manual','automatic'),inplace=True)

new\_df.fuelType.replace(('benzin','andere','elektro'),('petrol','others','electirc'),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)

mapper = {}

for i in labels:

mapper[i] = LabelEncoder()

mapper[i].fit(new\_df[i])

tr = mapper[i].transform(new\_df[i])

np.save(str('classes'+i+'.npy'),mapper[i].classes\_)

print(i,";",mapper[i])

new\_df.loc[:,i+'\_labels'] = pd.Series(tr,index = new\_df.index)

labeled = new\_df[ ['price','yearOfRegistration','powerPS','kilometer','monthOfRegistration'] + [x+"\_labels" for x in labels]]

print(labeled.columns)

Y = labeled.iloc[:,0].values

X = labeled.iloc[:,1:].values

Y = Y.reshape(-1,1)

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)

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'))

y\_pred = regressor.predict(X\_test)

print(r2\_score(Y\_test,y\_pred))

filename = 'resale\_value\_pickle\_file.sav'

pickle.dump(regressor,open(filename,'wb'))

HTML

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Car Resale Value Prediction</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z" crossorigin="anonymous">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<style>

body{

background-color: #fcde67;

}

.container{

background-color: white;

margin-top: 10px;

padding: 20px;

border-radius: 0px;

}

.form-group{

margin-top: 20px;

}

.form-control{

border-radius: 10px;

}

.btn{

border-radius: 10px;

margin-top: 20px;

}

.fa{

font-size: 30px;

color: #f2f2f2;

}

</style>

</head>

<body>

<div class="container">

<h1 class="text-center">🚙 Obtain the Precise Resale Value of Your Vehicle 🚗</h1><br><br>

<h3 class="text-center"><i>Predicted Price for Your Car:</i> ₹ {{prediction}}</h3>

</div>

</body>

</html>

**FLASK CONNECTIVITY:**

from flask import Flask,render\_template,request

import pickle

import numpy as np

app = Flask(\_\_name\_\_)

model=pickle.load(open("resale\_value\_pickle\_file.sav","rb"))

@app.route("/")

def home():

return render\_template("cars-index.html")

@app.route('/submit',methods=["POST","GET"])

def prediction():

if request.method=="POST":

yearofRegistration=request.form["yearofRegistration"]

monthofRegistration=request.form["monthofRegistration"]

powerPS = request.form["powerPS"]

kilometer = request.form["kilometer"]

notRepairedDamage = request.form["notRepairedDamage"]

if notRepairedDamage == "Yes":

notRepairedDamage =1

elif notRepairedDamage == "No":

notRepairedDamage =0

elif notRepairedDamage == "not-declared":

notRepairedDamage =2

brand = request.form["brand"]

if brand == "audi":

brand = 1

elif brand == "jeep":

brand = 14

elif brand == "volkswagen":

brand = 38

elif brand == "skoda":

brand = 31

elif brand == "bmw":

brand = 2

elif brand == "nissan":

brand = 23

elif brand == "renault":

brand = 27

elif brand == "ford":

brand = 10

elif brand == "honda":

brand = 11

elif brand == "mercedes\_benz":

brand = 20

elif brand == "toyota":

brand = 36

elif brand == "hyundai":

brand = 12

elif brand == "kia":

brand = 15

elif brand == "peugeot":

brand = 25

elif brand == "mitsubishi":

brand = 22

elif brand == "fiat":

brand = 9

elif brand == "volvo":

brand = 39

elif brand == "suzuki":

brand = 35

elif brand == "porsche":

brand = 26

elif brand == "dacia":

brand = 6

gearbox\_feat=request.form["gearbox\_feat"]

if gearbox\_feat == "manual":

gearbox\_feat =1

elif gearbox\_feat == "automatic":

gearbox\_feat =0

fuelType\_feat = request.form["fuelType\_feat"]

if fuelType\_feat=="petrol":

fuelType\_feat=1

elif fuelType\_feat=="diesel":

fuelType\_feat=3

elif fuelType\_feat=="lpg":

fuelType\_feat=4

elif fuelType\_feat=="hybrid":

fuelType\_feat=6

elif fuelType\_feat=="cng":

fuelType\_feat=7

vehicleType = request.form["vehicleType"]

if vehicleType == "coupe":

vehicleType = 3

elif vehicleType == "suv":

vehicleType = 8

elif vehicleType == "small car":

vehicleType = 7

elif vehicleType == "limousine":

vehicleType = 4

elif vehicleType == "bus":

vehicleType = 0

elif vehicleType == "combination":

vehicleType = 1

elif vehicleType == "others":

vehicleType = 6

elif vehicleType == "convertible":

vehicleType = 7

int\_features = [yearofRegistration, powerPS,kilometer,monthofRegistration,notRepairedDamage,brand,gearbox\_feat,fuelType\_feat,vehicleType]

features = [np.array(int\_features, dtype=int)]

prediction=model.predict(features)

return render\_template("cars-submit.html",prediction=round(prediction[0],2))

if \_\_name\_\_=="\_\_main\_\_":

app.run(debug=True)