Project On

Intelligent Vehicle Damage Assessment & Cost Estimator for Insurance Companies

powered By IBM India

Submitted By

Aravindh.S
Dinesh.A
Balamurugan.S
Jaganathan.M

Project ID: PNT2022TMID16910

of

Department of ELECTRONICS AND COMMUNICATION ENGINEERING
MAHENDRA ENGINEERING COLLEGE
NAMAKKAL DT

College Mentor: **SENTHILKUMARAN M**
Industrial Mentor: **SWATHI**
Evaluation Mentor: **KATHIRVEL.R**

INDEX

1. INTRODUCTION

1.1. Project Overview

The project "Intelligent Vechile Damage Assessment and Cost Estimator for Insurance Companies" is a responsive web application powered by aritifical Intelligence and IBM Watson Cloud. Deep Learning model is trained with the various damaged car images in various views and the VGG16 from the TensorFlow library is used for the better Deep Learning model architecture. An attractive front end can be developed using HTML and CSS. The pages such as Index.html , login.html, logout.html, register.html and prediction.html are created and embedded with the IBM cloud databse using python framework called flask. The web application takes the image input and estimate the cost for the insurance companies based on the damages in the car.

1.2. Purpose

The project is based on the domain of Artificial Intelligence and powered by the IBM watson cloud. A responsive web application can be developed using the HTML and CSS which is connected to waston cloud. In the cloud, a database service by availing the service Instance of the IBM cloud and the database API key is collected and connected with the front-end using flash which is an python framework for desiging the backend. Pages such as index.html, login.html, logout.html and prediction.html are used to interact with the web application. The user can register and the data of the user is saved in the databse of the IBM cloud, during the time of login, the login ID is compared with the ID in the databse and allow the user to the next page. The Deep Learning model is build using the VGG16 which is present in the keras library and the model is trained with the images of mulitple car with various level cum types of damages. The model is deployed in the back-end using the flask and the prediction.html page is setted to collect the image from the user. The prediction algorithm is used treat the image and estimated the cost for the user. The project is based on the various components which helps to handle the back - end and Front - end. Then front - end is build using html and css which is connectedback - end which is build using the python and IBM cloud. The project is powered by the IBM Watson cloud and is based in the artificial intelligence field. With the use of HTML and CSS and the Waston Cloud, a responsive web application may be created. The database API key is gathered and connected with the front-end using flash, which is a python framework for designing the backend.

2. LITERATURE SURVEY

2.1 Existing problem

1. Damage Assessment of a vehicle and Insurance Reclaim: This paper presents a system using CNN and image classification to assess the severity of damage to an automobile, which takes a user's input as an image to test the severity of the damage, which happens in two steps. The first step is image classification, where the user's input is used by the neural network to determine whether or not an automobile is damaged. the region and severity of the damage are determined in the second step using object detection on the flattened input that was received as the output in step one. The area may be the back, the front, or the side, and the severity may be classified as minor, moderate, or major. A report is filed and delivered to the user and the insurance company when the R-CNN network determines the extent of the damage. With little human contact, the user will be able to receive payment based on the results of the models.

2. Convolutional Neural Networks for vehicle damage detection: In this paper, a model for detecting vehicle damage is created, and it is divided into twelve categories. A deep learning model that can accurately detect and classify vehicle damages is created and evaluated in a specially designed light street, indicating that strong reflections complicate the detection performance. The proposed model outperforms other existing models in the classes Bend and Cover Damage. FSSD with Darknet-53 and YOLO v3 with Darknet-53 yield the best results. The drawback of the proposed approach is the robustness against different light conditions

3. Car Damage Assessment for Insurance Companies: In this paper a neural network-based solution for car detection, managing the problem of car damage analysis, prediction of car damage location and severity of the damage is proposed. The proposed system is intended to help insurance companies to analyze car damage a lot more successfully and well organized, and it quickly performs car damage detection by sending the image containing a damaged car for visual inspection. This system utilizes a machine learning approach along with computer vision to decide the damage analysis, the location of the damage as well as

the severity of the damage.

2.2 PROBLEM STATEMENT DEFINITION

4. Assessing Car Damage with Convolutional Neural Networks: This study focuses on automotive damage estimation, with auto insurers as their main potential clients. Three different Transfer Learning techniques are employed to do this, each of which identifies the existence, location, and degree of damage. Convolutional Neural Networks, which are adapted to maximize accuracy, serve as the foundation for the algorithms used. Each approach is analyzed and varying degrees of accuracy were achieved across different models deployed ranging from 68% to 87%. In this work, accuracy as high as 87.9% was attained. This study improves a number of existing methods and creates opportunities for collaboration in image recognition, notably in the field of auto insurance.

5. Vehicle Damage Classification and Fraudulent Image Detection Including Moiré Effect Using Deep Learning: This paper proposes deep learning-based methods for the classification of car damage types - MobileNet to classify vehicle damage into three groups: medium damage, enormous damage, and no damage. The extent of the damage to the vehicle determines its severity, ranging from medium to huge. The damage categories are based on typical damage kinds including shattered glass, dents on the front or back, damaged lamps or bumpers, etc. Automation in real-time applications, however, faces several challenges. Instead of capturing a picture of a car in real time, users can upload fake pictures. Making fake photos can involve using image-editing software to cover up flaws, getting images from the internet, or even taking screenshots of other devices' screens. To deal with these kinds of fake photographs, a hybrid strategy is also suggested in this research. To determine whether an image has been altered or is a screenshot, metadata analysis, and image editing software signature detection are used. It is suggested that moiré effect detection be used to determine whether an image was captured from the screen of another device, such as a computer screen when a mobile phone was used to snap a photo of an automobile.

6. Deep Learning Based Car Damage Classification and Detection: In this paper, they address the problem of vehicle damage classification/detection, which can be used by insurance companies to automate the process of vehicle insurance claims. With the adoption of fast, scalable, and end-to-end trainable convolutional neural networks, it is now technically feasible to recognize vehicle damages using deep convolutional networks. Various online sources containing different types of vehicle damage were manually collected and annotated. Using CNN models pre-trained on the ImageNet dataset and other techniques to improve the performance of the system, we achieved top accuracy of 96.39%, significantly better than the current results. In addition, they used a state-of-the-art YOLO object detector to detect the damaged region, achieving a maximum map score of 77.78% on the held-out test set, demonstrating the model's ability to recognize different vehicle damages. Furthermore, the paper proposes a pipeline for more robustly identifying vehicle damage by combining classification and detection tasks.

7. Car damage detection and classification: In this paper, a CNN model is developed and trained on the ImageNet dataset. After fine-tuning the dataset, transfer learning with L2 regularization is applied. In the proposed system, a Pre-trained VGG model not only detects the damaged part of a car but also assesses its location and severity. With the use of transfer learning and L2 regularisation, the proposed system achieves an accuracy of 95.22% of VGG19 and 94.56% of VGG16 in damaged detection, 76.48% of VGG19 and 74.39% of VGG16 in damage localization, and 58.48% of VGG19 and 54.8% of VGG16 in damage severity.

2.3.REFERENCES

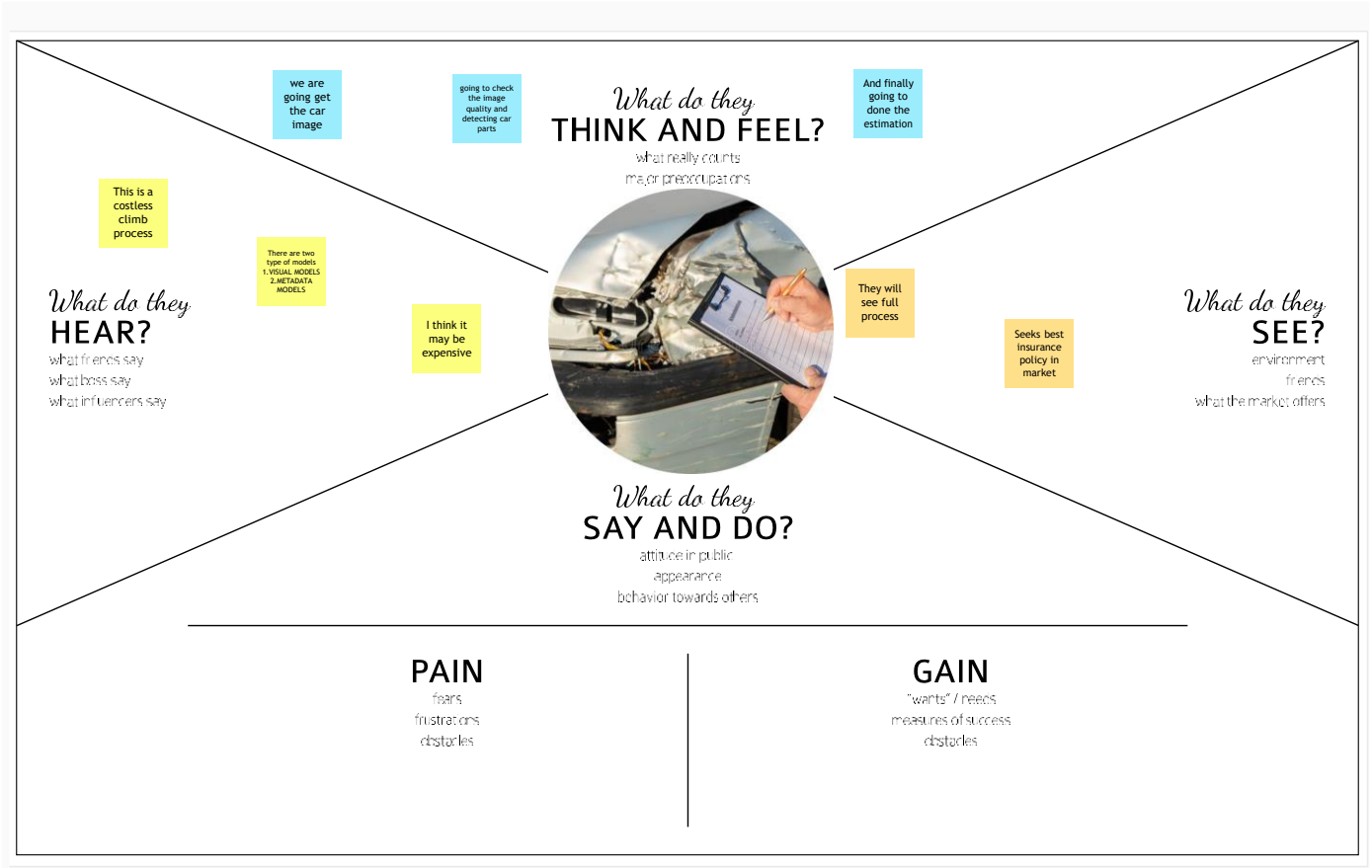
8. Vaibhav Agarwal, Utsav Khandelwal, Shivam Kumar, Raja Kumar, Shilpa M, "Damage Assessment Of A Vehicle And Insurance Reclaim", International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.10, Issue 4, pp.e197-e201, April 2022, Available at :http://www.ijcrt.org/papers/IJCRT2204483.pdf 2. R. E. van Ruitenbeek and S. Bhulai, “Convolutional Neural Networks for vehicle damage detection,” Machine

Learning with Applications, vol. 9. Elsevier BV, p. 100332, Sep. 2022. doi: 10.1016/j.mlwa.2022.100332. 3. Mandara G S and Prashant Ankalkoti, “Car Damage Assessment for Insurance Companies,” International Journal of Advanced Research in Science, Communication and Technology. Naksh Solutions, pp. 431–436, Jun. 23, 2022. doi: 10.48175/ijarsct-5048. 4. H. Bandi, S. Joshi, S. Bhagat, and A. Deshpande, “Assessing Car Damage with Convolutional Neural Networks,” 2021 International Conference on Communication information and Computing Technology (ICCICT). IEEE, Jun. 25, 2021. doi: 10.1109/iccict50803.2021.9510069. 5. U. Waqas, N. Akram, S. Kim, D. Lee and J. Jeon, "Vehicle Damage Classification and Fraudulent Image Detection Including Moiré Effect Using Deep Learning," 2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 2020, pp. 1-5, doi: 10.1109/CCECE47787.2020.9255806. 6. M. Dwivedi et al., “Deep Learning-Based Car Damage Classification and Detection,” Advances in Intelligent Systems and Computing. Springer Singapore, pp. 207–221, Aug. 14, 2020. doi: 10.1007/978-981-15-3514-7_18. 7. P. M. Kyu and K. Woraratpanya, “Car Damage Detection and Classification,” Proceedings of the 11th International Conference on Advances in Information Technology. ACM, Jul. 2020. doi: 10.1145/3406601.3406651.

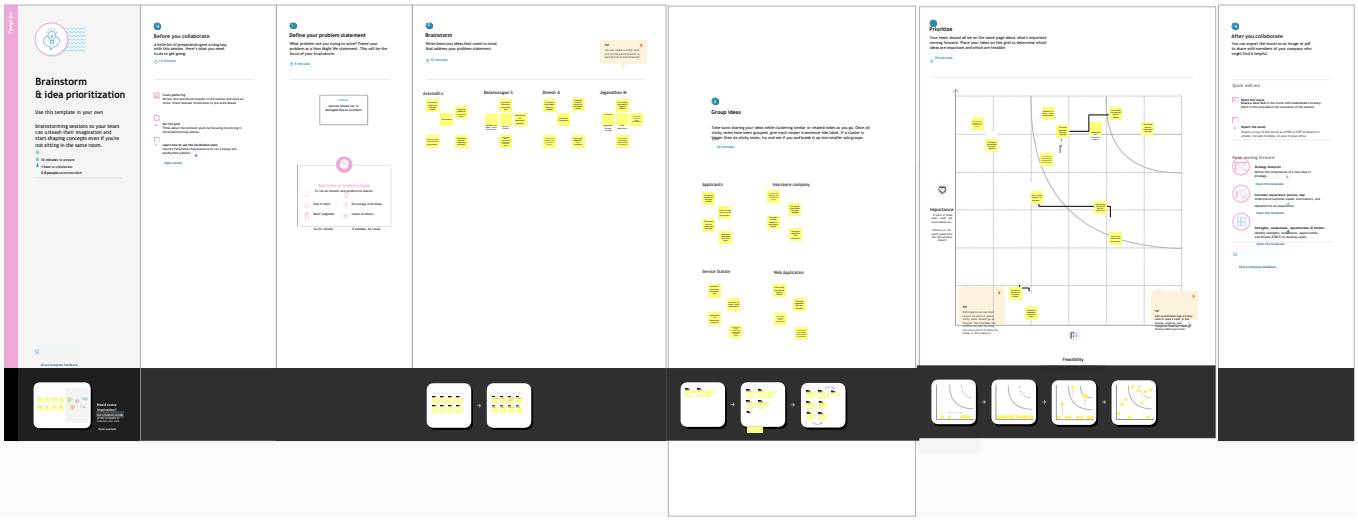
3. IDEATION & PROPOSED SOLUTION

S.NO	PARAMETER	DESCRIPTION
1	Problem Statement	1. The developing application must be very efficient and useful for the user 2. The agenda of this proposed system is to automatically identifying and locating damages in images of the vehicles
2	Idea/solution description	1. This will help the customer to climb the insurance when the vehicle gets damaged 2. We proposed employing convolution neural network to build a mask R-CNN model that can detect the area of damage.
3	Novelty/uniqueness	1. Will give more support to the customer 2. Detect only the damaged areas
4	Social Impact/Customer Satisfaction	1. Very useful as like user friendly 2. The time consumption for the clients we be very less by using this proposed system
5	Business model	1. By creating the ads over all the online platforms will be increase the business model 2. By using this technique the application usage can be understood by everyone
6	Scalability of the solution	1. It is very essential one for all the persons 2. So everyone will use this application to estimate their damage of vehicle.

3.1 Empathy Map



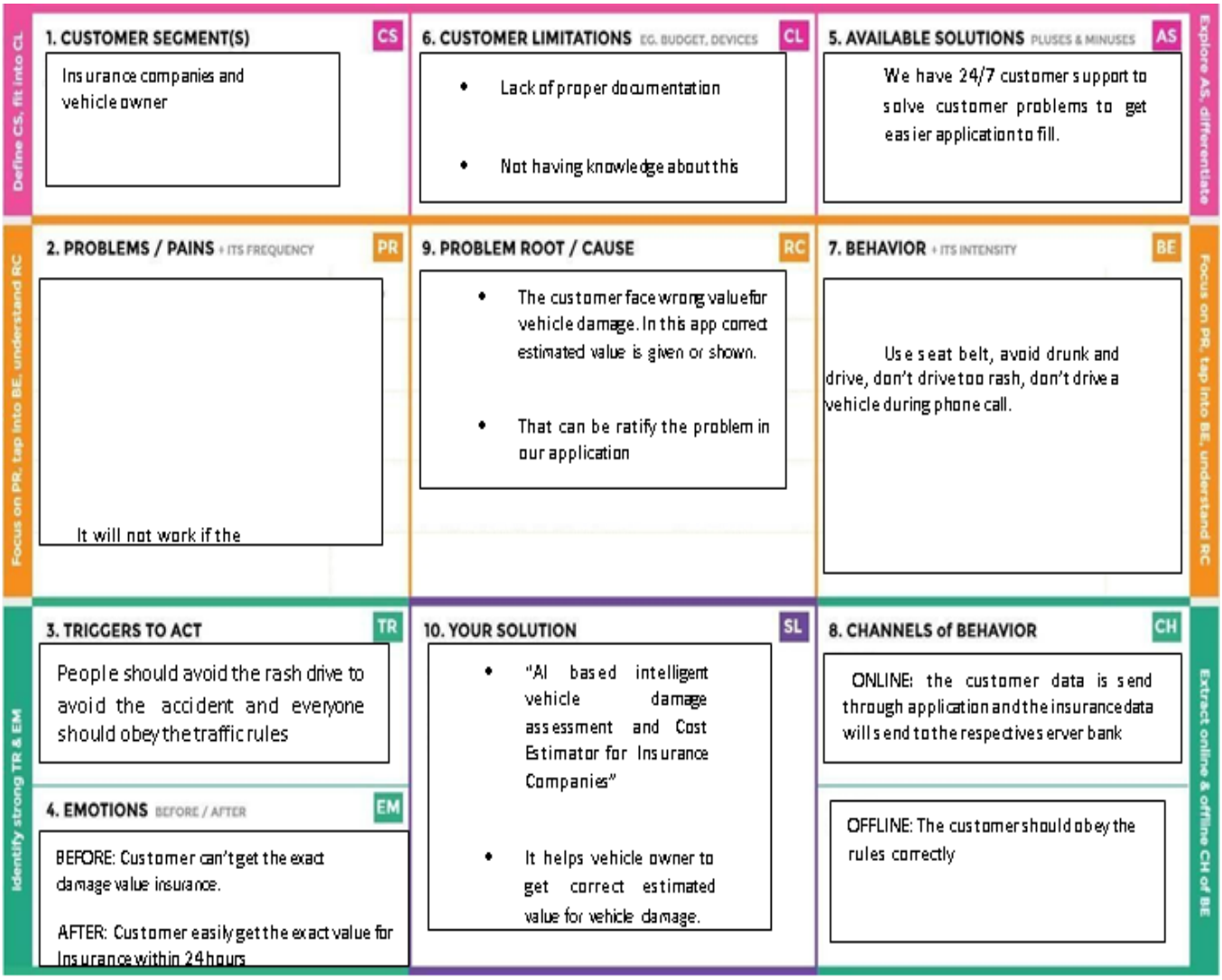
3.2Ideation and Bíainstoíming


Template
Brainstorm
& idea prioritization
Use this template in your own
brainstorming session so your team
can unleash their imagination and
start shaping concepts even if you're
not sitting in the same room.
10 minutes to prepare
1 hour to collaborate
2-8 people recommended
Before you collaborate
Team gathering
Set the goal
Learn how to use the facilitation tools
Define your problem statement
5 minutes
Brainstorm
10 minutes
Group ideas
20 minutes
Prioritize
20 minutes
Importance
Feasibility
After you collaborate
Quick add-ons
Share the board
Export the board
Keep moving forward
Strategy blueprint
Open this template
Customer experience journey map
Open this template
Share template feedback

3.3 Píoposed Solution

S.NO	PARAMETER	DESCRIPTION
1	Problem Statement	1. The developing application must be very efficient and useful for the user 2. The agenda of this proposed system is to automatically identifying and locating damages in images of the vehicles
2	Idea/solution description	1. This will help the customer to climb the insurance when the vehicle gets damaged 2. We proposed employing convolution neural network to build a mask R-CNN model that can detect the area of damage.
3	Novelty/uniqueness	1. Will give more support to the customer 2. Detect only the damaged areas
4	Social Impact/Customer Satisfaction	1. Very useful as like user friendly 2. The time consumption for the clients we be very less by using this proposed system
5	Business model	1. By creating the ads over all the online platforms will be increase the business model 2. By using this technique the application usage can be understood by everyone
6	Scalability of the solution	1. It is very essential one for all the persons 2. So everyone will use this application to estimate their damage of vehicle.

3.4Píoposed Solution Ïit



4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User registration	Download the app Registration through Gmail Create an account Follow the instructions Register by using user licence
FR-2	User Confirmation	Confirmation via Email Confirmation via, OTP
FR-3	Interface	Good Interface for the user to operate
FR-4	Accessing datasets	Details about user Details about vehicle Details about vehicle model. Details about insurance companies
FR-5	Mobile application	AI and camera sensor in the field can be access by mobile application.

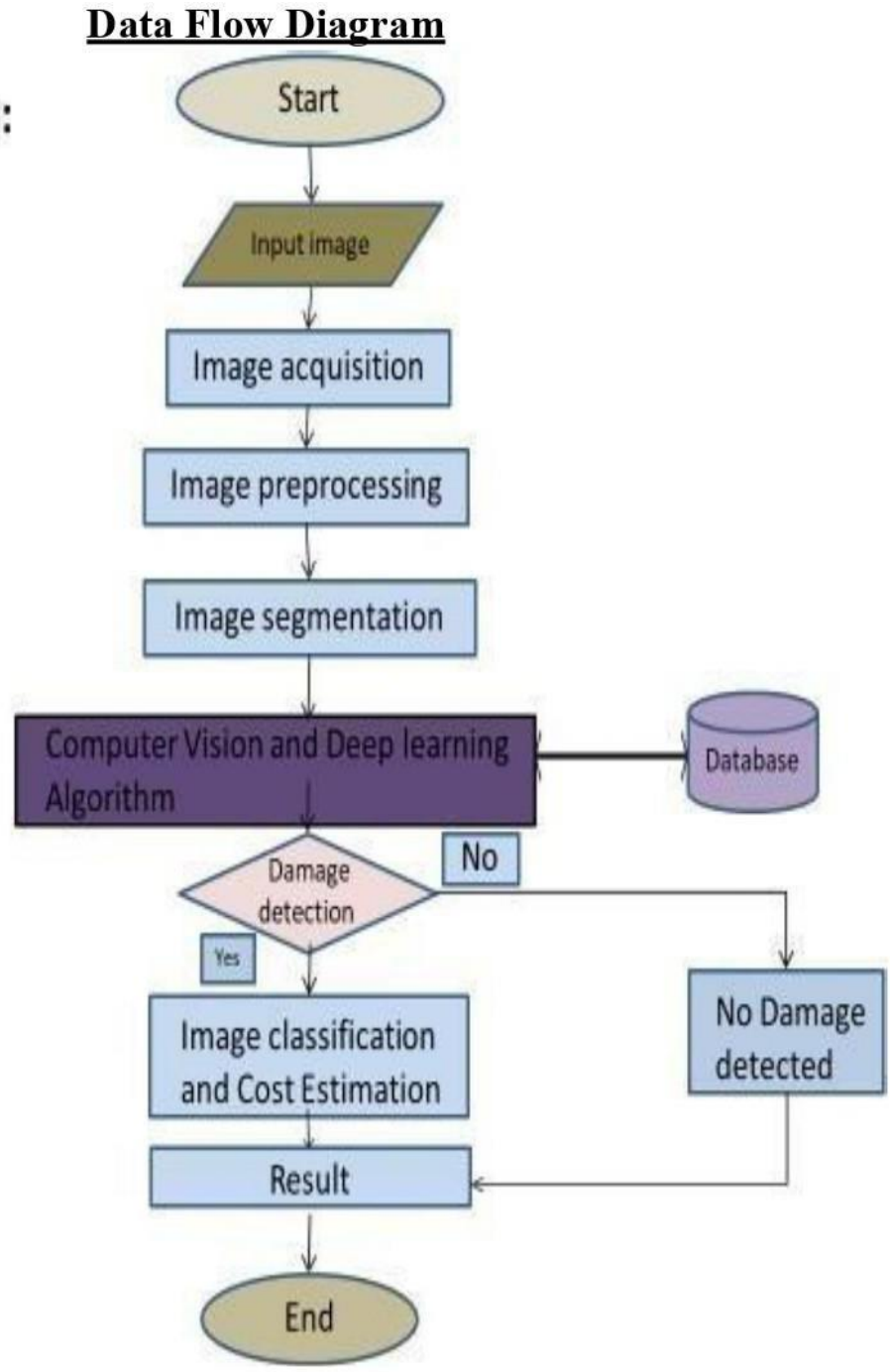
4.2 NON FUNCTIONAL REQUIREMENTS

NFR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The smart claiming system for vehicle damage insurance in bank companies
NFR-2	**Security**	We have designed this project to user easy to claim the insurance.
NFR-3	**Reliability**	This project will help the user to claim the insurance cost based on vehicle damage. It gives the exact value to user. This helps user to get correct cost without any failure.
NFR-4	**Performance**	AI devices and sensors are used to indicate the user to estimated the cost of the vehicle.AI camera to scan the damaged vehicle and gives exact cost insurance to user.
NFR-5	**Availability**	This application is designed for all devices and also Available in application.
NFR-6	**Scalability**	This project is more scalability in our present and future uses to estimate the cost exactly to user.

5. PROJECT DESIGN

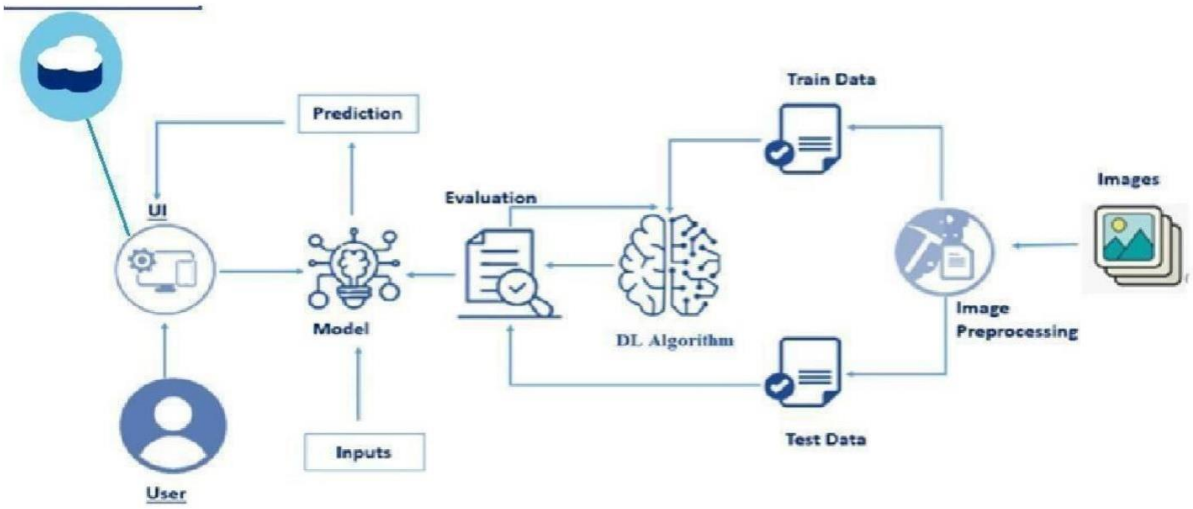
5.1 DATA FLOW DIAGRAM

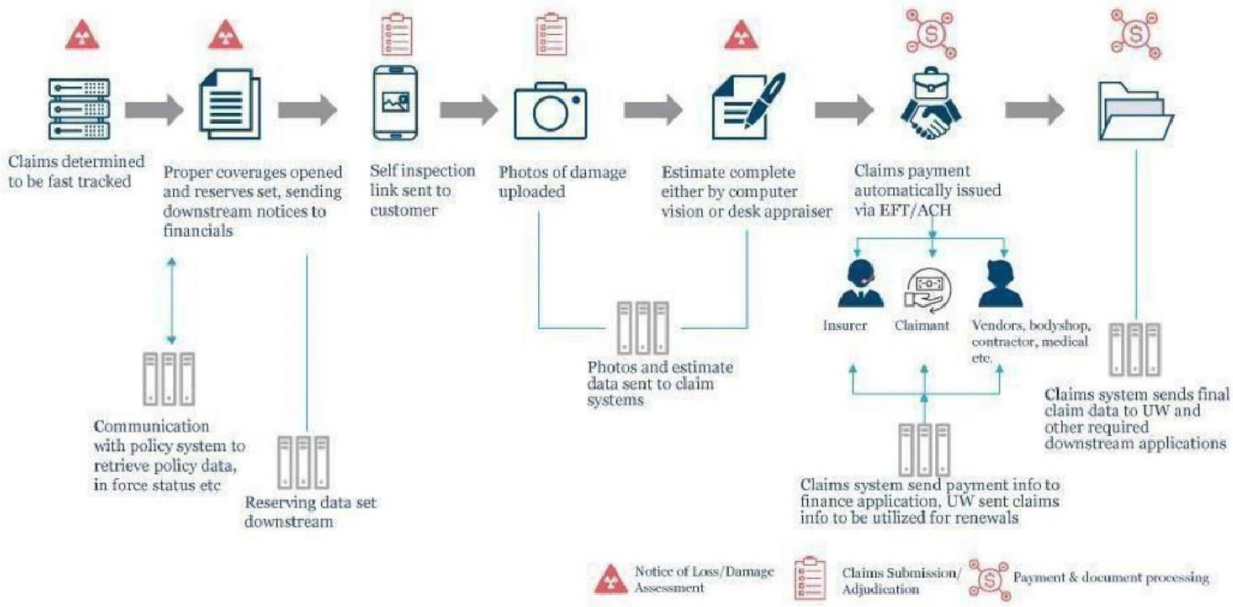
Data Flow Diagram:



5.2 SOLUTION AND TECHNICAL ARCHITECTURE

Technology Architecture:





5.3 USER STORIES

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, andconfirming my password.	4	Medium	Aravindh Balamurugan
Sprint-1	Login	USN-2	As a user, I will receive a confirmation email once I have registered for the application.	8	High	Aravindh
Sprint-2	Dashboard	USN-3	As a user, I can register for the applicationthrough Facebook.	1	High	Aravindh Jaganathan
Sprint-2	Gmail access	USN-4	As a user, I can register for the application through Gmail.	2	Low	Dinesh Balamurugan
Sprint-3	Uploading car damaged image	USN-5	As a user, I can log into the application by entering email & password.	6	Medium	Aravindh
Sprint-3	Cost details based on damage	USN-6	It gives the insurance cost based on the damage.	6	Medium	Balamurugan
Sprint-4	Effective customer support	USN-7	We provide excellent user assistance for the application for insurance	4	Medium	Dinesh
Sprint-4	To complete the customer work	USN-8	We will respond to customer needs in a goodway without mistakes.	8	High	Aravindh Jaganathan

6. PROJECT PLANNING & SCHEDULING

6.1. SPRINT PLANNING AND ESTIMATION

Product Backlog, Sprint Schedule, and Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN - 1	As a user, I can register for the application by entering my details of name, email, cars etc. verifying my Gmail account and creating new account with password	7	HIGH	TM-1,4
Sprint-1	Login	USN -2	As a user, entering my email, and password, and confirming my password, I can login to myaccount.	7	HIGH	TM-1,4
Sprint-1	Dashboard	USN-3	As a user, I can clearly see data, point, graphs, charts and trends of my previous activity and global activity related to my views	2	LOW	TM-1,4
Sprint-2	Details about insurance company	USN-4	As a user, I can register for the application through Gmailand account id.	8	MEDIUM	`TM-2,3
Sprint-1	repeated logins and logout	USN-5	As a user, I can log in and view my dashboard at my demand on any time	4	HIGH	TM-1,4
Sprint-2	Webpage	USN-6	As a user, I must enter all details of car, accident, capture images of my vehicle and upload it into the web portal.	12	HIGH	TM-2,3
Sprint 3	Details about estimated cost based on damage	USN-7	As a user I must receive a detailed report of the damages present in the vehicle and the Cost estimated.	20	HIGH	TM-1,2
Sprint 4	Provide friendly and efficient	USN-8	As a user, I need to get support from developers in case of	10	MEDIUM	TM-1,2,3

	customer support and sort out the queries.		queries and failure of service Provided by chat-box, mail orcall.			
Sprint 4	overview the entire process and act as a bridge between user and developer	USN-9	As a team member, we need to satisfy the customer needs in an efficient way and make sure any sort of errors are fixed	10	HIGH	TM-1,2,3

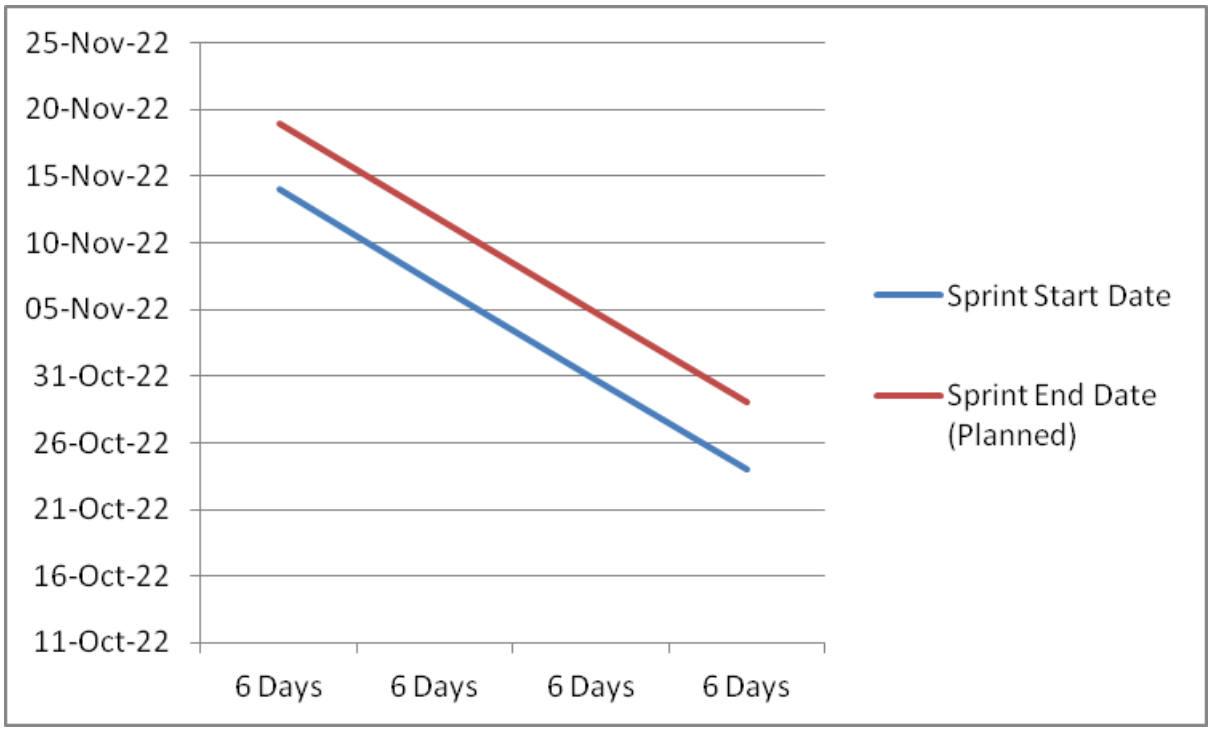
6.2. SPRINT DELIVERY SCHEDULE

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	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

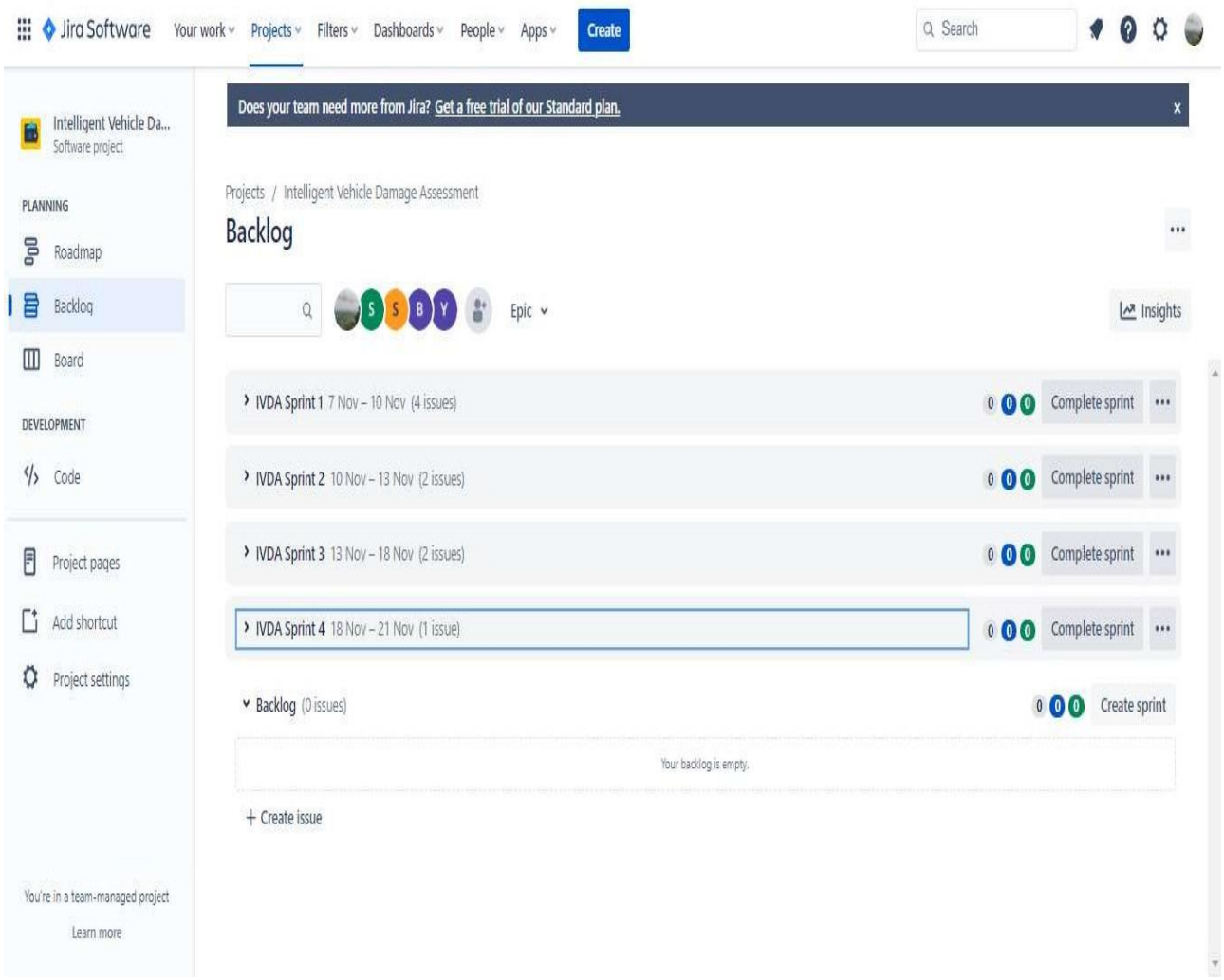
AV= SPRINT DURATION / VELOCITY

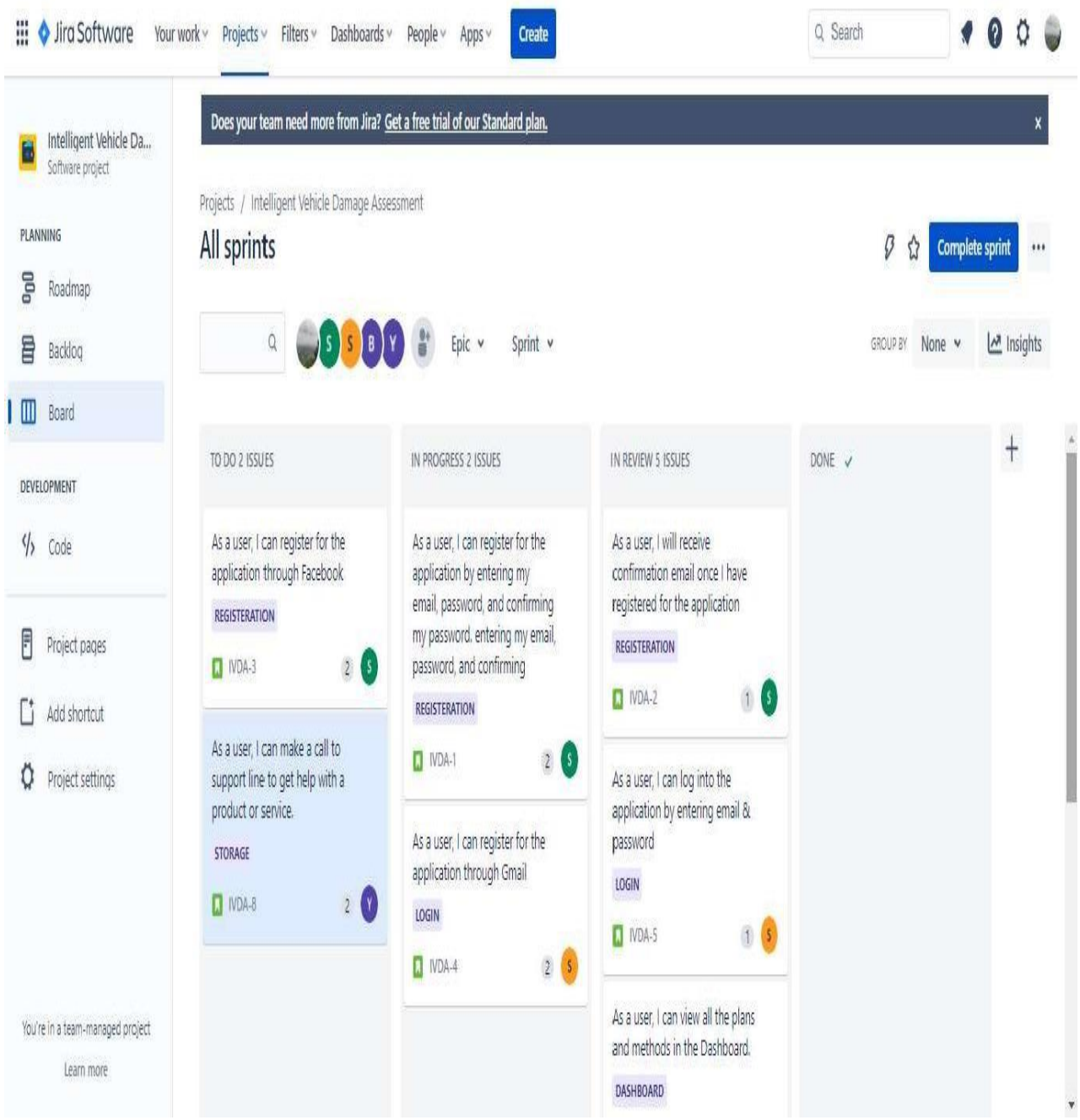
= 20 / 6

= 3.33



6.3 REPORTS FROM JIRA




Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Intelligent Vehicle Da...
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Does your team need more from Jira? Get a free trial of our Standard plan.
Projects / Intelligent Vehicle Damage Assessment
All sprints
Complete sprint
Epic
Sprint
GROUP BY
None
Insights
TO DO 2 ISSUES
As a user, I can register for the application through Facebook
REGISTERATION
IVDA-3
As a user, I can make a call to support line to get help with a product or service.
STORAGE
IVDA-8
IN PROGRESS 2 ISSUES
As a user, I can register for the application by entering my email, password, and confirming my password. entering my email, password, and confirming
REGISTERATION
IVDA-1
As a user, I can register for the application through Gmail
LOGIN
IVDA-4
IN REVIEW 5 ISSUES
As a user, I will receive confirmation email once I have registered for the application
REGISTERATION
IVDA-2
As a user, I can log into the application by entering email & password
LOGIN
IVDA-5
As a user, I can view all the plans and methods in the Dashboard.
DASHBOARD
DONE

7. CODING & SOLUTIONING

7.1FEATURE 1

```
client = Cloudant.iam("1c6f917d-87ac-491b-90a0-
6e3ae5b5daca-
bluemix","tYJcUyVJYs3WrxF_1absTN4RXrbdQ_RDWBRUy9BX-
28c",connect=True)
database =
 #load model
 model1 = load_model('V:\\WorkSpace\\IBM-Project-
 23426- 1659882722\\Final
 Deliverables\\model\\body.h5') model2 =
 load_model('V:\\WorkSpace\\IBM-Project-23426-
```

The feature 1 gives access to the trained deep learning models for predicting mulitple damages in various areas in the vechile and connected with the IBM Waston Database for storing the user data.

7.2 FEATURE 2

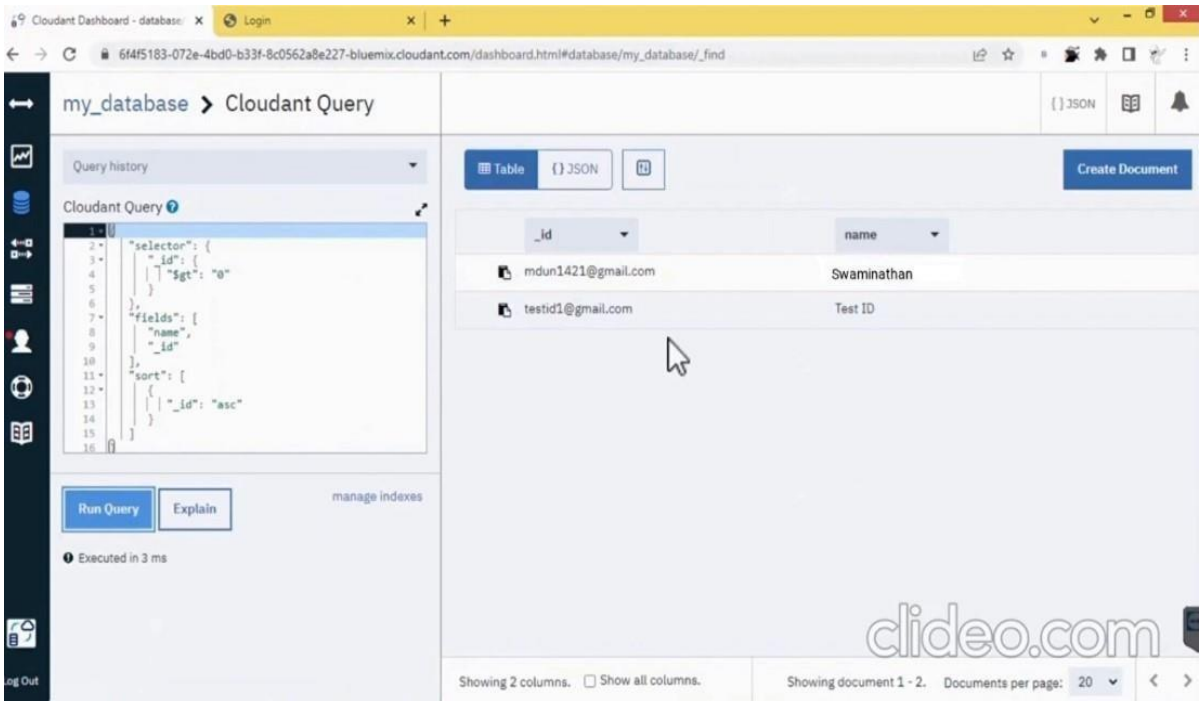
```
        img =
        load_img(filepath,target_size=(224,2


        prediction1 =
        np.argmax(model1.predict(img_data)) prediction2
        = np.argmax(model2.predict(img_data))

        index1 = ['front','near','side']
```

feature 2 enables the web application to predict the incomming image from the user into the given labels. The code gets the image, convert into pixcels and load into the model. Based on the predicted results, the algorithm will returns the value as the estimated cost.

7.3 DATABASE SCHEMA

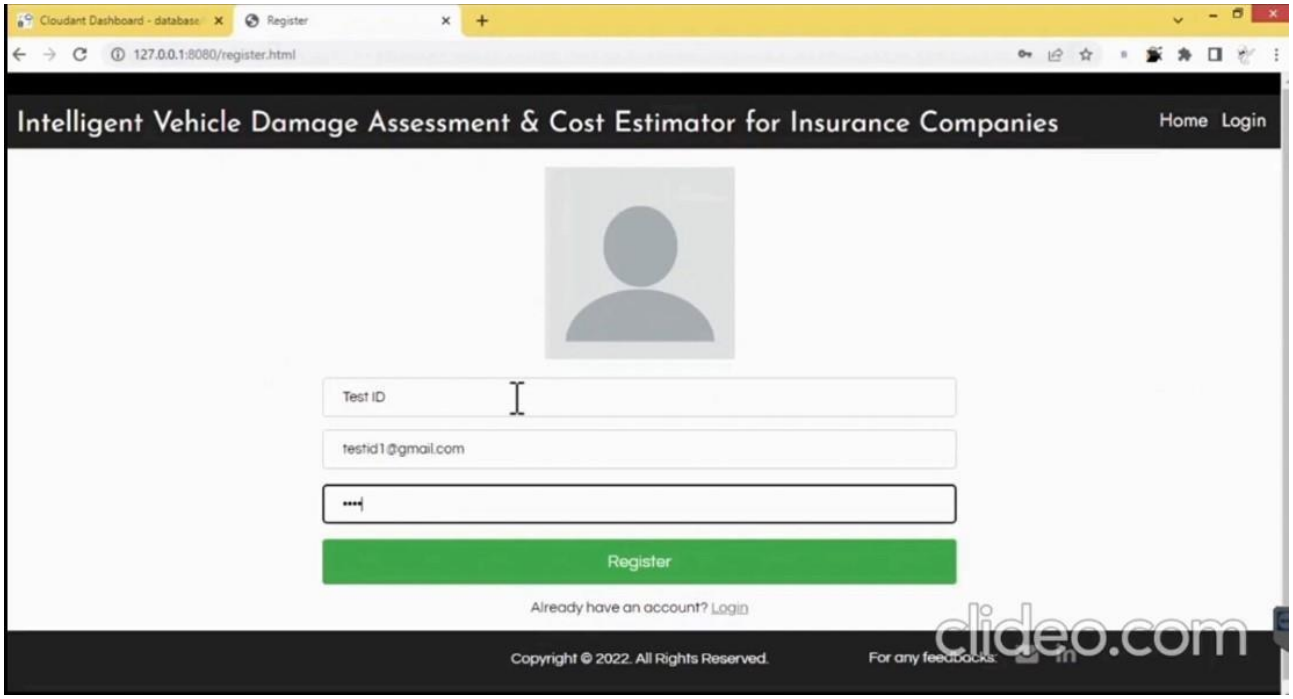


8. TESTING

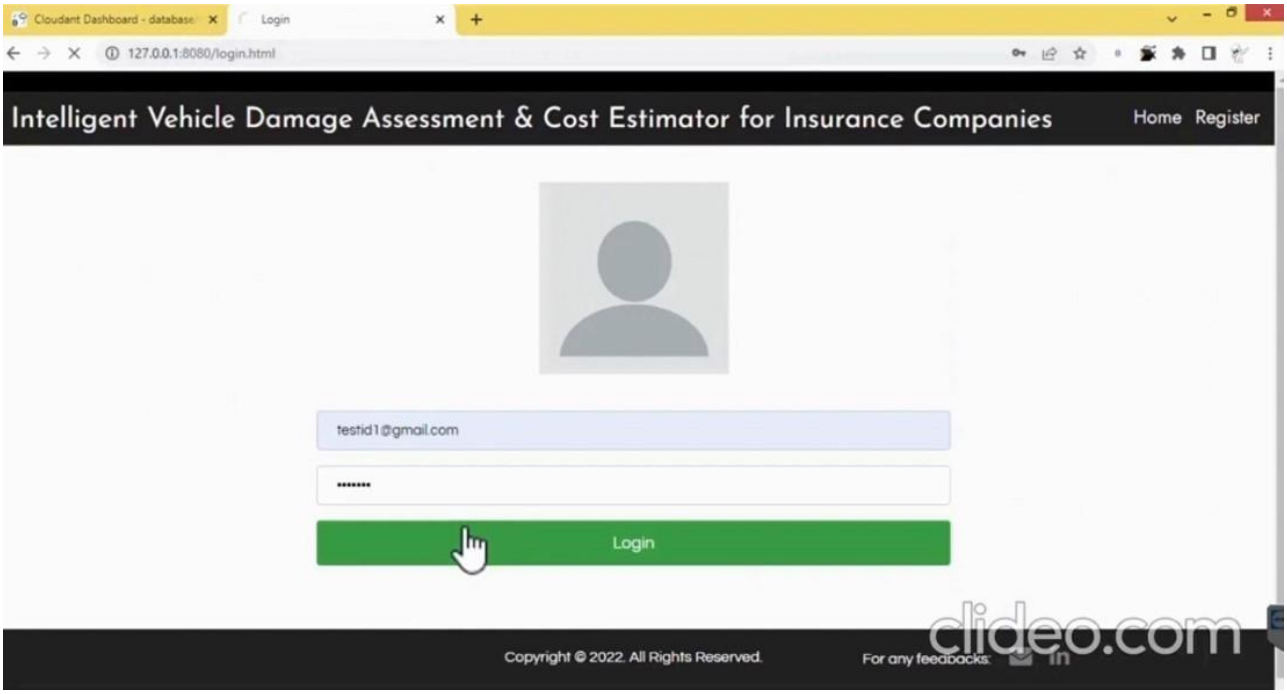
8.1. TEST CASES

1. Useí Login and Registíation test
2. Database Update test
3. Píediction test

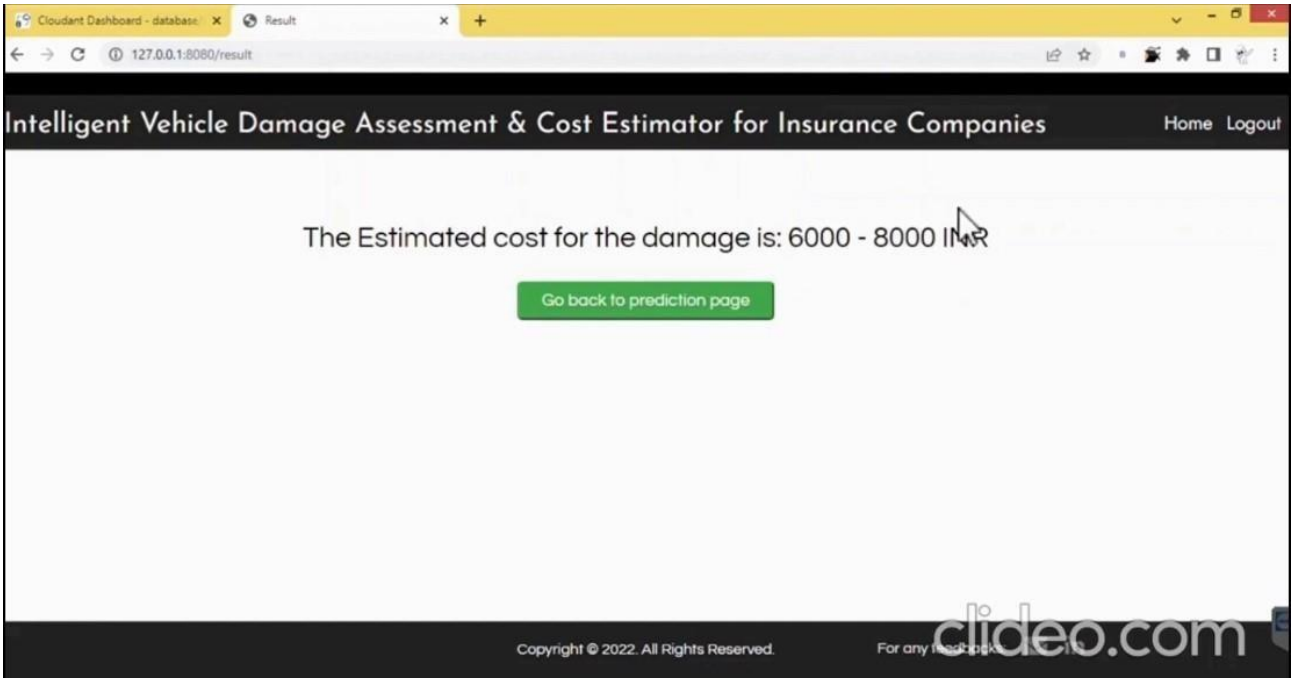
8.2. USER ACCEPTANCE TESTING



The íegisteíation web page is tested with the alíeady íegisteíed useí infoímation and hence it shows a message "You aíe alíeady a membeí" by which the íepeation of useí infíomation at database is píevented.



The login web page is tested with the invalid user information to check the invalid login testing into the webpage.



The prediction page is given with the test image of a damaged car to check the accuracy of the models.

9. RESULTS

9.1 PERFORMANCE METRICS

The performance of the Cost estimator for insurance companies is tested and assessed with the latency check, which is run over the prediction page. The time taken to load the image and predict the cost based on the damages in the vehicle is checked. The results show that the web application took less than 10s to provide the estimated cost of the given vehicle image. The model is tested with the various damaged car images which is not used during the training and validation of the model which also shows that the model works with the accuracy of about 98% in the overall performance.

- Repair cost optimization, total loss and agreed value
- Quick assessment by phone – without the need for a visit by the professional inspector
- Overseeing the repair of the vehicle
- Establishing the monetary and residual value of vehicles
- Assistance in court
- Accident investigation to check all the data provided on the claim file
- Our reports and dataset are customized and adapted to your workflow, minimizing changes to your processes

The results show that the web application took less than 10s to provide the

estimated cost of the given vehicle image. The model is tested with the various damaged car images which is not used during the training and validation of the model which also shows that the model works with the accuracy of about 98% in the overall performance

10. ADVANTAGES

1. The Advantage of having an Intelligent Cost Estimator based on the damages can save the time and resource of the user in automatically evaluating the images with the damages using the Deep Learning models trained with the various car images
2. **Finding a proper data set-** Training machine learning models requires a sufficient data set of relevant images. The more varied the images are, the better the model will be able classify images appropriately.

3. Preprocessing image data sets is a crucial step in speeding up and obtaining better training results for models. This activity may span a variety of tasks: applying filters, removing noise, enhancing contrast, down sampling images, etc.

4. **Building a model-** After you have a quality data set at hand, there are still some considerations when building a machine learning model.

DISADVANTAGES

5. The Disadvantage of the project is expensive coding and time to develop the front end and back end of the web application
6. Creating and training a model takes time
7. **Optimizing performance and costs -** As insurance companies have to deal with damage assessment on a daily basis, the working solution also needs to demonstrate resonating performance

11. CONCLUSION

We conclude by suggesting this web application for damage assessment and cost estimation for the insurance companies. The web application is supported by the Deep Learning and IBM Watson cloud which stands for the complex image prediction and user information storage. The web application takes the user registration and login, The user can login into the prediction page using their ID and password. The prediction

takes the image input and the model can predict the input based on the perviour knowledge about the damages.

In future, The User Interface of the web application can be improved by updating the HTML and CSS codings. The improvement in UI can gives the better user exprience in future, The model's accuracy over various images can increased by trainning with various damaged images. The Image processing methods can be improved to achive higher performance of the model in the future.

12. FUTURE SCOPE

In future, The User Interface of the web application can be improved by updating the HTML and CSS codings. The improvement in UI can gives the better user exprience in future, The model's accuracy over various images can increased by trainning with various damaged images. The Image processing methods can be improved to achive higher performance of the model in the future.

13. APPENDIX

Github Repo:

https://github.com/IBM-EPBL/IBM-Project-9265-1658989879

VideoLink:

https://drive.google.com/drive/folders/1c1k5nvcbQPMOY8q9R4vYA4VdE-c4w7z7?usp=share_link

App.py

```
import re
import numpy as np
import os
from flask import Flask, app, request, render_template
from keras import models
from keras.models import load_model
from keras.preprocessing import image
from tensorflow.python.ops.gen_array_ops import concat
from keras.applications.inception_v3 import preprocess_input
import requests
from flask import Flask, request, render_template, redirect,
url_for
from cloudant.client import Cloudant

#Create Database
client = Cloudant.iam('00cba18f-2150-4961-9102-f29b9aee35de-
bluemix','ht_ByiEjrGeaitIZJTC-ri5_8Oq-dxTNHLGho1mpt0d5',
connect=True)
my_database = client.create_database('my_database')
```

```
#Loading the Model
```

```
model1 = load_model('Model/level.h5')
model2 = load_model('Model/body.h5')

app = Flask(__name__)

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

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

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

@app.route('/afterreg', methods=['POST'])
def afterreg():
    x = [x for x in request.form.values()]
    print(x)
    data = {
        '_id': x[1],
        'name': x[0],
        'psw': x[2]
    }
    print(data)

    query = {'_id': {'$eq': data['_id']}}

    docs = my_database.get_query_result(query)
    print(docs)

    print(len(docs.all()))

    if(len(docs.all())==0):
        url = my_database.create_document(data)
        response = request.get(url)
        return render_template('login.html', pred="Registration
```

```
Successful, Please login using your details")
    else:
        return render_template('register.html', pred="You are
already a member, Please login using your details")

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

@app.route('/afterlogin', methods=['POST'])
def afterlogin():
    user = request.form['_id']
    passw = request.form['psw']
    print(user,passw)

    query = {'_id': {'$eq': user}}

    docs = my_database.get_query_result(query)
    print(docs)

    print(len(docs.all()))

    if(len(docs.all())==0):
        return render_template('login.html', pred="The Username
is not found")
    else:
        if((user==docs[0][0]['_id'] and
passw==docs[0][0]['psw'])):
            return redirect(url_for('prediction'))
        else:
            print('Invalid User')

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

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

```
@app.route('/result')
def res():
    if request.methods=="POST":
        f=request.files['image']
        basepath=os.path.dirname(__file__)
        filepath=os.path.join(basepath,'uploads',f.filename)
        f.save(filepath)

        img=image.load_img(filepath,target_size=(256,256))
        x=image.img_to_array(img)
        x=np.expand_dims(x,axis=0)

        img_data=preprocess_input(x)
        prediction1=np.argmax(model1.predict(img_data))
        prediction2=np.argmax(model2.predict(img_data))

        index1=['front','rear','side']
        index2=['minor','moderate','severe']

        result1 = index1[prediction1]
        result2 = index2[prediction2]
        if(result1 == "front" and result2 == "minor"):
            value = "3000 - 5000 INR"
        elif(result1 == "front" and result2 == "moderate"):
            value = "6000 - 8000 INR"
        elif(result1 == "front" and result2 == "severe"):
            value = "9000 - 11000 INR"
        elif(result1 == "rear" and result2 == "minor"):
            value = "4000 - 6000 INR"
        elif(result1 == "rear" and result2 == "moderate"):
            value = "7000 - 9000 INR"
        elif(result1 == "rear" and result2 == "severe"):
            value = "11000 - 13000 INR"
        elif(result1 == "side" and result2 == "minor"):
            value = "6000 - 8000 INR"
        elif(result1 == "side" and result2 == "moderate"):
            value = "9000 - 11000 INR"
        elif(result1 == "side" and result2 == "severe"):
```

```
            value = "12000 - 15000 INR"
        else:
            value = "16000 - 50000 INR"

        return
render_template('prediction.html',prediction=value)

if __name__=="__main__":
    app.run(debug = False,port = 8080)
```