INTELLIGENT VEHICLE DAMAGE ASSESSMENT AND COST ESTIMATION FOR INSURANCE COMPANIES

DOMAIN NAME: ARTIFICIAL INELLIGENCE

TEAM ID: PNT2022TMID43010

TEAM LEADER: NANDHINI R

TEAM MEMBERS: SANJEEV KUMAR M
GOWTHAM P
PREMKUMAR N

TABLE OF CONTENTS

1. INTRODUCTION

PROJECT OVERVIEW

A broad understanding of image data Know how to pre-process/clean the data using different data preprocessing techniques. Know how to Know fundamental concepts and techniques of VGG16.Gain build a web application using the Flask framework.

The user interacts with the UI (User Interface) to choose the image. The chosen image is analyzed by the model which is integrated with the flask application. VGG16 Model analyzes the image, then the prediction is showcased on the Flask UI.

AI in automotive insurance holds significant potential to quickly estimate vehicle damages. Soon with the advancement in AI algorithms, assessment done manually would be a thing of the past. Traditionally the damage assessment was carried out by multiple parties which were time-consuming, highly prone to human error, leading to inaccurate cost estimations.

PURPOSE

The aim of this project is **to build a VGG16 model that can detect the area of damage on a car**. The rationale for such a model is that it can be used by insurance companies for faster processing of claims if users can upload pics and the model can assess damage be it dent scratch from and estimates the cost of damage.

2. LITERATURE SURVEY

EXISTING PROBLEM

Car insurers need to perform many daily operations, including validation, inspection, data processing, management, and storing of huge volumes of data generated by different parties. Moreover, the variety of cars increases as well as the number of insurance claims, and car rental services have to adjust their calculations accordingly.

The insurance sector has to stick to strict regulations which sometimes cause delays in obtaining insurance for its customers. McKinsey estimates that AI investments could potentially cost insurers as much as $1.3 trillion annually. However, the losses caused by fraud and inaccurate assessment overreach this sum considerably. The process of analysis of insurance claims is often delayed because the inspection involves human intervention. AI-powered technology allows for automatic car repair detection and auto-detection monitoring with the possibility of manual intervention.

REFERENCE

[1]. A.Neela Madheswari, J.haripriya, G.Kiruthika, R.M.Meyammai Mahendra Engineering college, India, exterior vehicular damage detection using deep learning, department of computer science and engineering

[2]. Girish N, Mohammed Aqeel Arshad, car damage detection using machine learning. International journal of advances research in computer and communication engineering, vol. 10, issue 8, August 2021 DOI 10.17148/IJARCCE.2021.10808.

[3]. Phyu Mar Kyu,car damage detection and classification, faculty of information technology king Mongkut's institute of technology ladkrabang Bangkok,Thailand62606003@kmitl.ac.in.

[4]. S. Jayawardena, Image based automatic vehicle damage detection. PhD thesis, College of Engineering and Computer Science (CECS), 12 2013.

[5]. S. Gontscharov, H Baumgartel, A.Kneifel, and K.-L. Krieger, Algorithm development for minor damage identification in vehicle bodies using adaptive sensor data processing," Procedia Technology, vol. 15, pp. 586 {594, 2014. 2nd International Conference on System-Integrated Intelligence: Challenges for Product and Production Engineering.

[6]. Y.-J. Cha, J. Chen, and O. B''uy''uk''ozt''urk, Output-only computer visionbased damage detection using phase-based optical flow and unscented kalman _lters," Engineering Structures, vol. 132, pp. 300, {313, 2017.

PROBLEM STATEMENT DEFINITION

Nowadays, a lot of money is being wasted in the car insurance business due to leakage claims. Claims leakage Underwriting leakage is characterized as the discrepancy between the actual payment of claims made and the sum that should have been paid if all of the industry's leading practices were applied. Visual examination and testing have been used to may these results. However, they impose delays in the processing of claims.

3. IDEATION & PROPOSED SOLUTION

EMPATHY MAP CANVAS

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community.

The empathy map represents a principal user and helps teams better understand their motivations, concerns, and user experience. Empathy mapping is a simple yet effective workshop that can be conducted with a variety of different users in mind, anywhere from stakeholders, individual use cases, or entireteamsofpeople.

The 4 Attributes of Empathy:

1. Perspective taking.
2. Staying out of judgment.
3. Recognizing emotion in another person.
4. Communicating the understanding of another person's emotions.

IDEATION & BRAINSTORMING

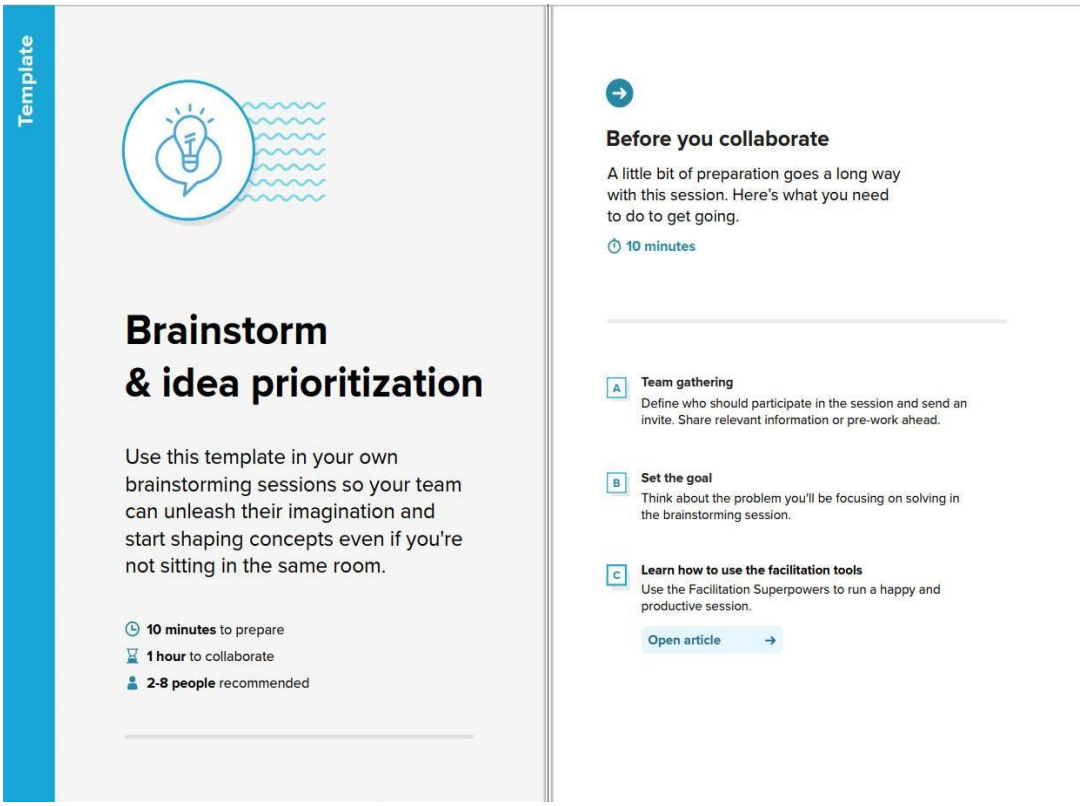
Brainstorming is a group problem-solving method that involves the spontaneous contribution of creative ideas and solutions. This technique requires intensive, freewheeling discussion in which every member of the group is encouraged to think aloud and suggest as many ideas as possible based on their diverse knowledge**.**

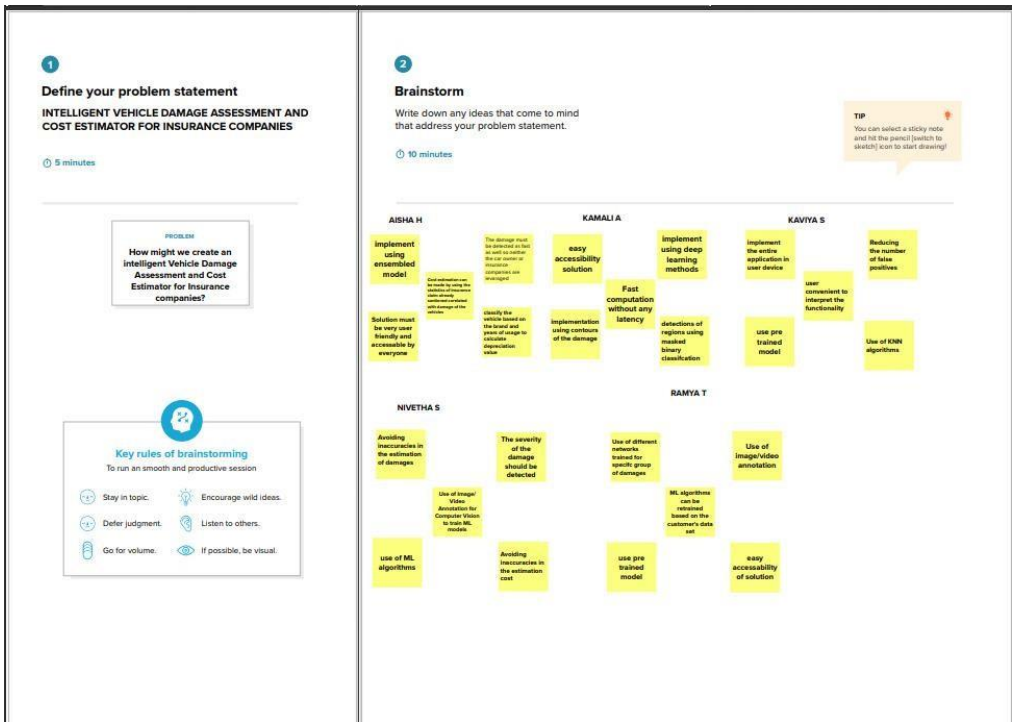
4 Types of Brainstorming:

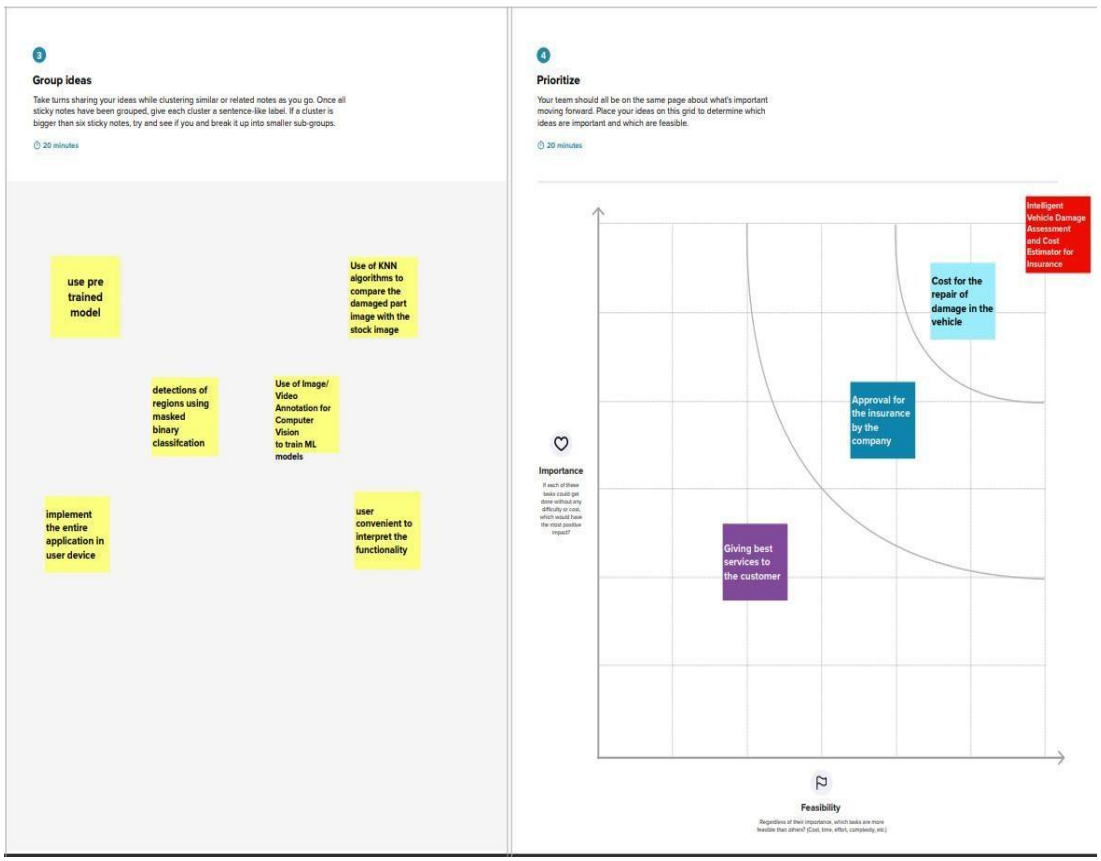
1. Reverse Brainstorming. A creative problem-solving technique in which the problem is turned around and considered from a different point of view to spur new and different solutions.
2. Stop-and-Go Brainstorming.
3. Phillips 66 Brainstorming.
4. Brainwriting.

Benefits of Brainstorming:

1. Provides a quick and easy class activity. Brainstorming sessions can be effectively used in the classroom.
2. Contributes to classroom collective power.
3. Creates a student-centered activity.
4. Supports learning in a relaxed environment.
5. Strengthens problem-based learning.
6. Encourages creative thought.




1
Define your problem statement
INTELLIGENT VEHICLE DAMAGE ASSESSMENT AND COST ESTIMATOR FOR INSURANCE COMPANIES
5 minutes
PROBLEM
How might we create an intelligent Vehicle Damage Assessment and Cost Estimator for Insurance companies?
Key rules of brainstorming
To run an smooth and productive session
Stay in topic.
Defer judgment.
Go for volume.
Encourage wild ideas.
Listen to others.
If possible, be visual.
2
Brainstorm
Write down any ideas that come to mind that address your problem statement.
10 minutes
TIP
AISHA H
KAMALI A
KAVIYA S
implement using ensembled model
easy accessibility solution
implement using deep learning methods
implement the entire application in user device
Reducing the number of false positives
Fast computation without any latency
user convenient to interpret the functionality
Solution must be very user friendly and accessible by everyone
classify the vehicle based on the brand and years of usage to calculate depreciation value
Implementation using contours of the damage
detections of regions using masked binary classification
use pre trained model
Use of KNN algorithms
RAMYA T
NIVETHA S
Avoiding inaccuracies in the estimation of damages
The severity of the damage should be detected
Use of different networks trained for specific group of damages
Use of image/video annotation
Use of Image/ Video Annotation for Computer Vision to train ML models
ML algorithms can be retrained based on the customer's data set
use of ML algorithms
Avoiding inaccuracies in the estimation cost
use pre trained model
easy accessability of solution


3
Group ideas
Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes
use pre trained model
Use of KNN algorithms to compare the damaged part image with the stock image
detections of regions using masked binary classifcation
Use of Image/ Video Annotation for Computer Vision to train ML models
implement the entire application in user device
user convenient to interpret the functionality
4
Prioritize
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.
20 minutes
Intelligent Vehicle Damage Assessment and Cost Estimator for Insurance
Cost for the repair of damage in the vehicle
Approval for the insurance by the company
Giving best services to the customer
Importance
Feasibility

PROPOSED SOLUTION

Problem Statement (Problem to be solved): Nowadays lot of money is being wasted in the car insurance business due to leakage claims. Claims leakage Underwriting leakage is characterized as the discrepancy between the actual payment of claims made and the sum that should have been paid if all of the industry's leading practices were applied. Visual examination and testing have been used to may these results. However, they impose delays in the processing of claims. There is no easy way of accessing and knowing about the part of the vehicle getting damaged. Often the processing of such a damaged part of the vehicle carrying the area of damaged part is cumbersome. New methods have to be proposed in order to make it faster and efficient. Processing of Insurance for the cars needs to be assessed in a quicker way so that claims can be provided to the damaged parts.

Idea / Solution description: Automobile Industry is one of the major industry in a Country. This proposed system is Intelligent vehicle damage assessment and cost estimator for insurance companies using computer vision in artificial intelligence. The model will predict the location of the damage as in front, side or rear, and the severity of such a damage as in minor, moderate or severe and estimate the cost of damage of both car and bike.

Novelty / Uniqueness: Deep learning method used to fixed the problem and then Working on with VGG16 pretrained model by adding extra layers to increase the accuracy while implementing the project.

Social Impact / Customer Satisfaction: The model developed will be used to fix the damage caused to the Vehicle quickly so that the vehicle can be modified to the old look and also for faster processing of cost of the damage to claim insurance quickly. This project can be used to save time for calculating the area and level of the damage quickly such that the insurance claim can be made efficiently.

Business Model (Revenue Model): This can also be used to help car companies as well. Can collaborate with insurance companies. Can collaborate with car companies.

Scalability of the Solution: AI guided Application provides 24/7 service to clear all customer queries and guide them through all the processes. In future, it can be scaled as per the requirements of the insurance or car company to include answers to queries related to the cost based on the inputs provided

PROBLEM SOLUTION FIT

There is no systematic approach to receive a rapid answer from an insurance company. A week of waiting is required. The proposed solution should enable consumers to contact with the insurance provider and receive payments both online and offline. After uploading the damaged image and determining the extent of the damage, the user may obtain insurance only if the company approves the damaged image and the condition is more than 80%.

4. REQUIREMENT ANALYSIS

FUNCTIONAL REQUIREMENT

Framework Creation:

This approach provides a way for evaluating vehicle damage that insurance companies may utilize when processing claims. This module offered a framework for submitting a vehicle's damaged parts and requesting insurance from an organization. The dataset needed to train the Damage Detection and it has prepared by an admin. In order to make the images useful for training, they were manually annotated; damages were categorized into 7 distinct types such as Door Dent, Bumper Dent, Body Scratch, Broken Windshield, Broken Glass, Broken Lights and Smash By modifying its settings and loading the learned dataset, the model was set up to train on user data.

Object Detection:

Employ a specially trained CNN model utilizing transfer learning on to identify the object. This model takes different forms of damage into account validation sets such as Bumper Dent, Bumper Scratch, Door Dent, Door Scratch, Glass Shattered, Head Lamp, Tail Lamp, Undamaged, etc. The classification of car damage severity is as follows: Minor Damage which typically involves slight damage to the vehicle that does not impede the vehicle to cause severe injuries. It includes the headlight scratches, dents and digs in the hood or windshield, from gravel or debris, scratches in the paint. Moderate Damage which deals with any kind of damage that impairs the functionality of the vehicle in any way is moderate damage. It involves large dents in hood, fender or door of a car. Even if the airbags are deployed during collision, then it comes under moderate damage. Severe Damage – Structural damages such as bent or twisted frames, broken/bent axels, and missing pieces of the vehicles and in some cases even the destruction of airbags. These types of damages are a big threat to the human life.

Damage Detection:

To locate damaged areas in a picture and create a bounding box around each object found, object localization is used which combines object localisation and classification to

provide a bounding box and a class for each item for object detection. Use CNN to generate a convolutional features map from an image to forecast the class and bounding box of an item. If the car is undamaged then it simply detects it and if it's a damaged one, then there are further localizations made models. The model shows accuracy on the validation set. To automate such a system, the easiest method would be to build a Convolution Neural Network model capable of accepting images from the user and determining the location and severity of the damage. The model is required to pass through multiple checks would first ensure that given image is that of a car and then to ensure that it is in fact damaged. These are the gate checks before the analysis begins. Once all the gate checks have been validated, the damage check will commence. The model will predict the location of the damage as in front, side or rear, and the severity of such damage as in minor, moderate or severe.

Claim Insurance:

The procedure of claiming insurance is done by persons who are in need. For access to the company's insurance, the user must register and authenticate. After that, users may access their insurance information and submit an insurance claim request. The request for an insurance claim can be viewed and approved by the insurance company. Once the damaged image has been uploaded and the degree of the damage has been determined, the user may receive insurance only if the firm accepts the damaged image and the condition is greater than 80%.

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 Interface	• User friendly and simple website

FR-4	Collect the datasets	• Collect the data from the user side and their vehicle side information. • Collect the data from about Insurance companies plans.
FR-5	Final Results	•Model should be trained with high accuracy. • Results obtained from the model should be displayed to The user with easy interpretability.

NON-FUNCTIONAL REQUIREMENTS:

Non-functional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	• Intelligent model used to assessment the damage in the vehicle and estimate the cost to be provided by the insurance company.
NFR-2	Security	• The credibility of the user and the confidentiality of user details about their vehicle must be maintained.
NFR-3	Reliability	• This scheme can achieve good accuracy in damage

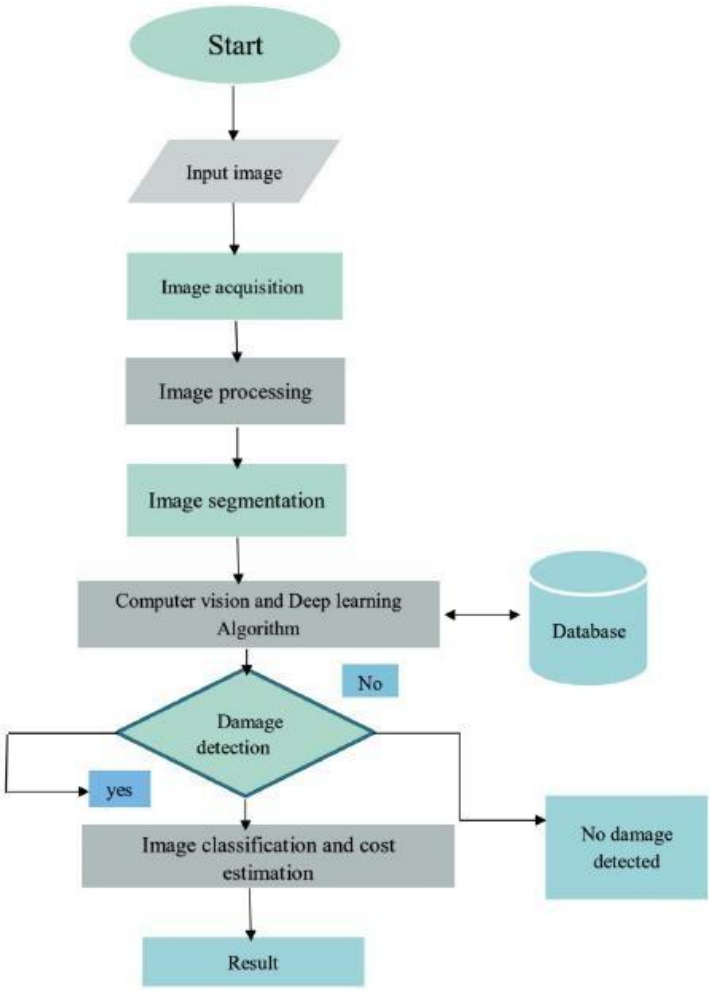
		estimation and cost estimation, thus providing accurate and unbiased insurance coverage to the user.
NFR-4	Performance	• Real-time images are to be captured and uploaded to the website, where the proposed model performs damage assessment and gives the insurance cost accordingly.
NFR-5	Availability	• The website should be compatible with web browsers on both mobile phones and computers.

5. PROJECT DESIGN

DATA FLOW DIAGRAMS

A data flow diagram (DFD) is a graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement. They are often elements of a formal methodology such as Structured Systems Analysis and Design Method (SSADM).

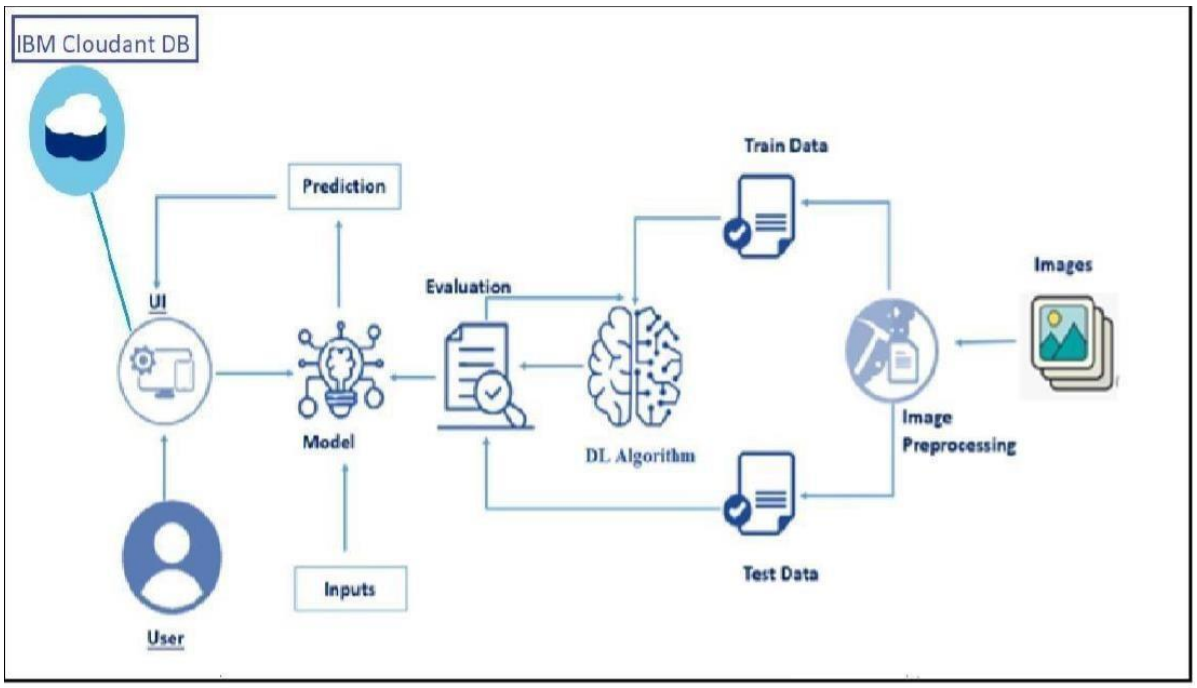
It helps us to understand the functioning and the limits of a system. It is a graphical representation which is very easy to understand as it helps visualize contents. Data Flow Diagram represent detailed and well explained diagram of system components



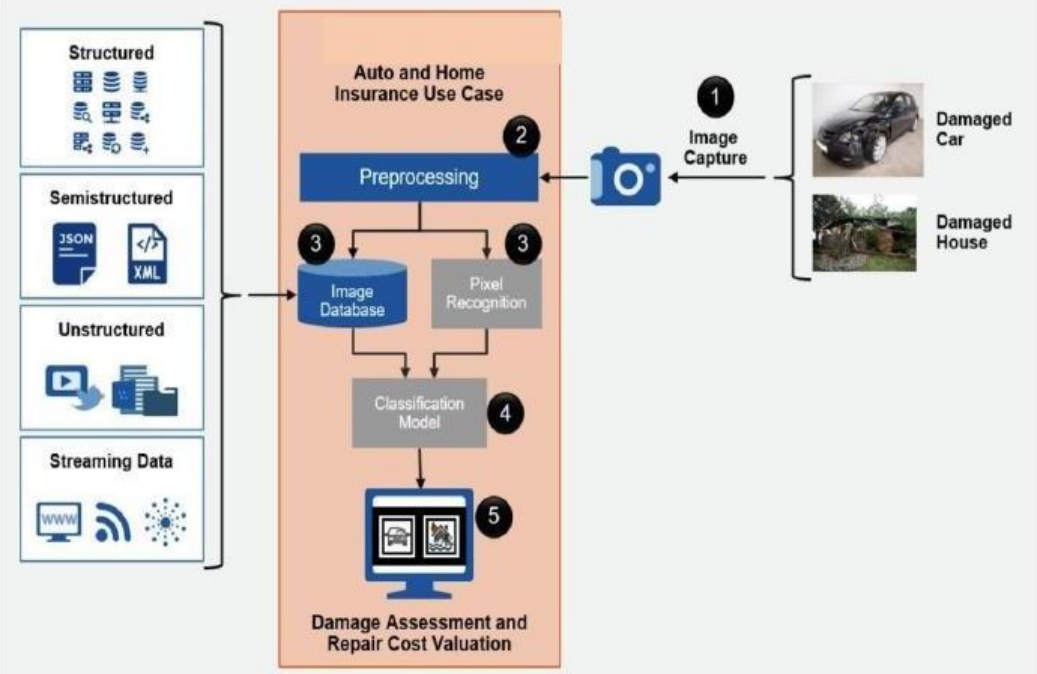
SOLUTION & TECHNICAL ARCHITECTURE

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called architecture description languages (ADLs).

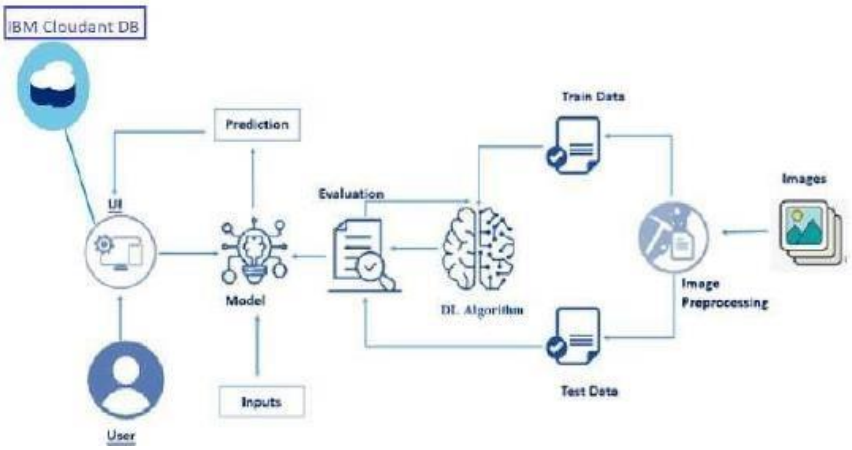
SOLUTION ARCHITECTURE :



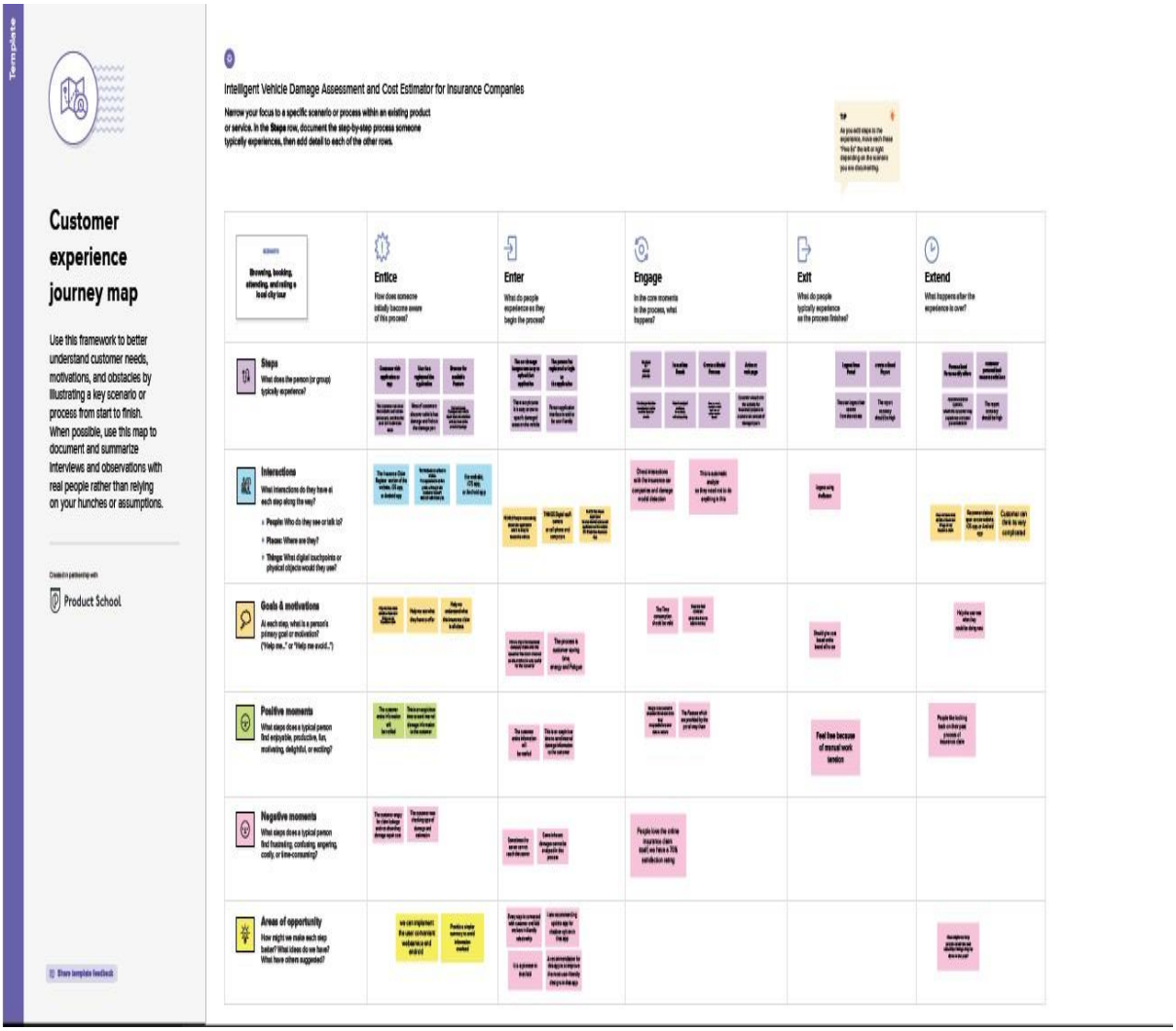
OUTLINE ARCHITECTURE:



TECHNICAL ARCHITECTURE:



USER STORIES:



6.PROJECT PLANNING & SCHEDULING

SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint - 1	Registration	USN - 1	As an owner of a particular vehicle, I can log into the application by entering email & password.	**2**	High	AISHA H KAVIYA S KAMALI A NIVETHA S RAMYA T
Sprint - 1	User Confirmation	USN - 2	As an owner of a particular vehicle, I will receive confirmation email once I have registered for the application.	**1**	Medium	AISHA H KAVIYA S KAMALI A NIVETHA S RAMYA T
Sprint - 1	Login	USN - 3	As an owner of a particular vehicle, I can log into the application by entering email & password.	**2**	High	AISHA H KAVIYA S KAMALI A NIVETHA S RAMYA T
Sprint - 2	Data Collection	USN - 1	Download the dataset used in	**2**	High	AISHA H KAVIYA S

			intelligent vehicle damage assessment & cost estimator for insurance companies.			KAMALI A NIVETHA S RAMYA T
Sprint - 2	Image Pre Processing	USN - 1	Improve the image data that suppresses unwilling distortions or enhances some image features important for further processing, although performing some geometric transformations of images like rotation,scaling, etc.	**2**	High	AISHA H KAVIYA S KAMALI A NIVETHA S RAMYA T
Sprint - 3	Model Building	USN - 1	Define the model architecture and adding CNN layer and testing ,saving the model.	**2**	High	AISHA H KAVIYA S KAMALI A NIVETHA S RAMYA T
Sprint - 3	Cloud DB	USN - 1	Below are steps that need to follow for creating and using cloudant service. • Register & login to IBM	**2**	High	AISHA H KAVIYA S KAMALI A NIVETHA S RAMYA T

			cloud • Create service instance • Creating service credentials • Launch cloudant DB • Create database			
Sprint-4	Application Building	USN - 1	Building a web application that is integrated into the model we built. A UI is provided to the user where he has uploaded the image. Based on the saved model, the uploaded image will be analyzed and prediction is showcased on the UI.	**2**	High	AISHA H KAVIYA S KAMALI A NIVETHA S RAMYA T
Sprint-4	Train The Model On IBM	USN - 1	Build Deep learning model and computer vision Using the IBM cloud.	**2**	High	AISHA H KAVIYA S KAMALI A NIVETHA S RAMYA T

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	4 Days	5nov 2022	09nov 2022	20	10nov 2022
Sprint-2	20	4 Days	6nov 2022	10nov 2022	20	11nov 2022
Sprint-3	20	5 Days	7nov 2022	11nov 2022	20	12nov 2022
Sprint-4	20	9 Days	11nov 2022	18nov 2022	20	19nov 2022

7. CODING & SOLUTIONING

7.1 FEATURE 1

```
import datetime
from flask import jsonify
from flask import Flask, render_template, request
from cloudant.client import Cloudant

client = Cloudant.iam('d9b401b4-6c0f-4740-9da7-4376a6dc8fdf-bluemix',
'TsO1xlMzArJKQ1vqNv08hxxraWpbSt9lOWNxtAHvYGv8',
                      connect=True)
my_database = client.create_database('my_database')
app = Flask(_name_)
app.config.from_object(__name__)
app.config['SECRET_KEY'] = '"083458892a3c1ab6f18660a9cfeae6f5c'


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


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


@app.route("/addamount")
@app.route("/register")
def NewUser():
    return render_template('register.html')
```

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


@app.route("/newuse", methods=['GET', 'POST'])
def newuse():
    if request.method == 'POST':
        x = [x for x in request.form.values()]
        print(x)
        data = {
            '_id': x[1],
            'name': x[0],
            'psw': x[2]
        }
        print(data)
        query = {'_id': {'Seq': 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)
            return render_template('login.html', data="Register, please login using your details")
        else:
            return render_template('register.html', data="You are already a member, please login
using your details")


@app.route("/userlog", methods=['GET', 'POST'])
def userlog():
    if request.method == 'POST':
```

```
        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('goback.html', pred="The username is not found.")
        else:
            if user == docs[0][0]['_id'] and passw == docs[0][0]['psw']:
                return render_template("index.html")
            else:
                return render_template('goback.html', data="user name and password incorrect")


@app.route("/predict", methods=['GET', 'POST'])
def predict():
    if request.method == 'POST':
        file = request.files['fileupload']
        DateTimeMilliSeconds =
datetime.datetime.now().strftime("%Y%m%d_%H%M%S_%f")
        image_file_path =
r'media/images/DamageImage_{}.jpg'.format(DateTimeMilliSeconds)
        file.save(image_file_path)

        import tensorflow as tf
        import numpy as np
        import warnings

        warnings.filterwarnings('ignore')
```

```
    test_image = tf.keras.preprocessing.image.load_img(image_file_path, target_size=(200,
200))

    # test_image = image.img_to_array(test_image)
    test_image = np.expand_dims(test_image, axis=0)
    # DAMAGE_COST MODEL
    classifierLoad = tf.keras.models.load_model(r'model/body.h5')
    result = classifierLoad.predict(test_image)
    result1 = ''
    if result[0][0] == 1:
        result1 = "front"
    elif result[0][1] == 1:
        result1 = "rear"
    elif result[0][2] == 1:
        result1 = "side"
    print('[INFO!!]', result1)

    ########################################
    # file = request.files['fileupload1']
    # DateTimeMilliSeconds =
datetime.datetime.now().strftime("%Y%m%d_%H%M%S_%f")
    # image_file_path =
r'media/images/DamageType_{}.jpg'.format(DateTimeMilliSeconds)
    # file.save(image_file_path)
    # test_image = tf.keras.preprocessing.image.load_img(
    #     r'C:\Users\Macro\Downloads\Car damage\level\validation\03-severe\0017.JPEG',
target_size=(200, 200))
    #
    # test_image = np.expand_dims(test_image, axis=0)
    ################################
    # Damage_type Model
    classifierLoad = tf.keras.models.load_model(r'model/level.h5')
    result = classifierLoad.predict(test_image)
```

```
    result2 = ''
    if result[0][0] == 1:
        result2 = "minor"
    elif result[0][1] == 1:
        result2 = "moderate"
    elif result[0][2] == 1:
        result2 = "severe"
    print('[INFO!!]', result2)
    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"
    print('[INFO!!] Damage Cost Range: ', value)
    # Please comment this return and uncomment the 'render_template' in 147 line
    return jsonify({'Damage Cost Range': value, 'Damage_angle': result1, 'Damage_type':
result2})
    # return render_template('userhome.html', prediction=value)
```

```
if__name__== '_main_':
    app.run(debug=True, use_reloader=True)
```

8. TESTING

TEST CASES

A test case has components that describe input, action and an expected response, in order to determine if a feature of an application is working correctly. A test case is a set of instructions on "HOW" to validate a particular test objective/target, which when followed will tell us if the expected behavior of the system is satisfied or not.

Characteristics of a good test case:

1. Accurate: Exacts the purpose.
2. Economical: No unnecessary steps or words.
3. Traceable: Capable of being traced to requirements.
4. Repeatable: Can be used to perform the test over and over.
5. Reusable: Can be reused if necessary.

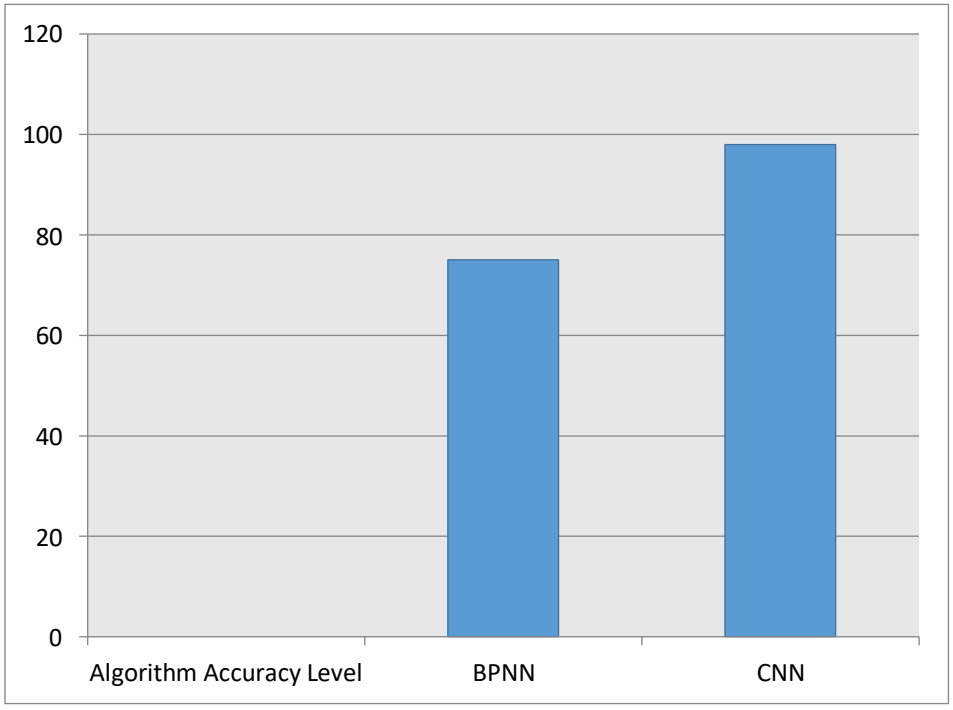
S.NO	**Scenario**	**Input**	**Excepted output**	**Actual output**
1	User login	User name and password	Login	Login success
2	Login success	Upload damaged vehicle image as a input	Detecting object and analyze for claim insurance	Details are stored in a database.

USER ACCEPTANCE TESTING

This sort of testing is carried out by users, clients, or other authorised bodies to identify the requirements and operational procedures of an application or piece of software. The most crucial stage of testing is acceptance testing since it determines whether or not the customer will accept the application or programme. It could entail the application's U.I., performance, usability, and usefulness. It is also referred to as end-user testing, operational acceptance testing, and user acceptance testing (UAT).

9. RESULTS

9.1 PERFORMANCE METRICS



10. ADVANTAGES & DISADVANTAGES

ADVANTAGE

1. Digitalized claim process makes easy to use
2. Give the accurate result of the damaged vehicle
3. Helps the insurance company to analyze the damaged vehicle and also payment process.

DISADVANTAGE

1. It will take more time to claim the insurance in manual process
2. Because of incorrect claims, the company behaves badly and doesn't make payments currently.
3. Poor customer support.

11. CONCLUSION

In this research proposal, a neural network-based solution for automobile detection will be used to address the issues of automotive damage analysis and position and severity prediction. This project does several tasks in one bundle. The method will unquestionably assist the insurance firms in conducting far more thorough and systematic analyses of the vehicle damage. Simply sending the system a photograph of the vehicle, it will evaluate it and determine whether there is damage of any type, where it is located, and how severe it is.

12. FUTURE SCOPE

In future work, need to use several regularization methods with a big dataset in our next work. Anticipate the cost of a car damaged component more accurately and reliably if we have higher quality datasets that include the attributes of a car (make, model, and year of production), location data, kind of damaged part, and repair cost. This study makes it possible to work together on picture recognition projects in the future, with a focus on the auto insurance industry. The study was able to accurately validate the presence of damage, its location, and its degree while eliminating human bias. These can be further enhanced by adding the in the fly data augmentation approaches.

13. APPENDIX

SOURCE CODE:

```
import datetime
from flask import jsonify
from flask import Flask, render_template, request
from cloudant.client import Cloudant

client = Cloudant.iam('d9b401b4-6c0f-4740-9da7-4376a6dc8fdf-bluemix',
'TsO1xlMzArJKQ1vqNv08hxxraWpbSt9lOWNxtAHvYGv8',
                      connect=True)
my_database = client.create_database('my_database')
app = Flask(_name_)
app.config.from_object(__name__)
app.config['SECRET_KEY'] = "'083458892a3c1ab6f18660a9cfeae6f5c'



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

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


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


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


@app.route("/newuse", methods=['GET', 'POST'])
def newuse():
    if request.method == 'POST':
        x = [x for x in request.form.values()]
        print(x)
        data = {
            '_id': x[1],
            'name': x[0],
            'psw': x[2]
        }
        print(data)
        query = {'_id': {'Seq': data['_id']}}
        docs = my_database.get_query_result(query)
        print(docs)
        print(len(docs.all()))
```

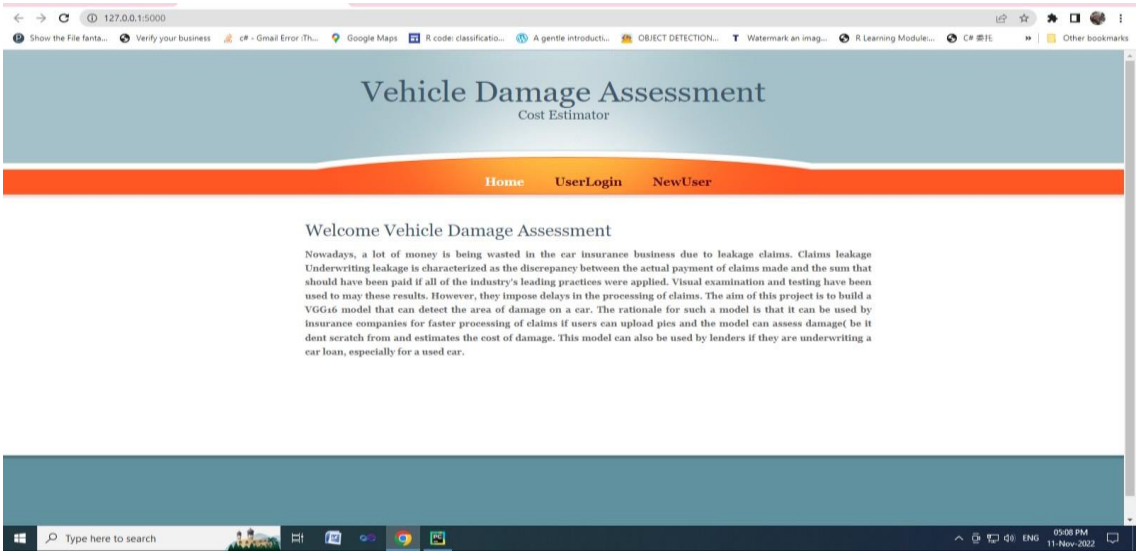
```
        if (len(docs.all()) == 0):
            url = my_database.create_document(data)
            return render_template('login.html', data="Register, please login using your details")
        else:
            return render_template('register.html', data="You are already a member, please login
using your details")


@app.route("/userlog", methods=['GET', 'POST'])
def userlog():
    if request.method == 'POST':
        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('goback.html', pred="The username is not found.")
        else:
            if user == docs[0][0]['_id'] and passw == docs[0][0]['psw']:
                return render_template("index.html")
            else:
                return render_template('goback.html', data="user name and password incorrect")


@app.route("/predict", methods=['GET', 'POST'])
def predict():
    if request.method == 'POST':
        file = request.files['fileupload']
        DateTimeMilliSeconds =
datetime.datetime.now().strftime("%Y%m%d_%H%M%S_%f")
```

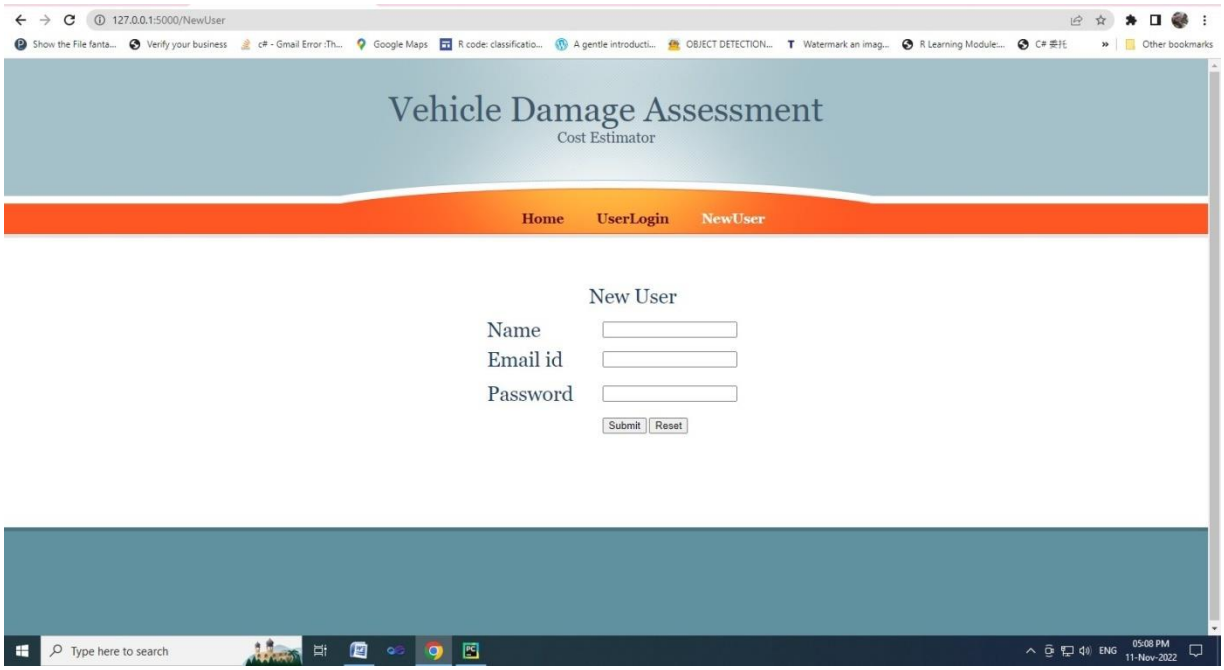
```
        image_file_path =
r'media/images/DamageImage_{}.jpg'.format(DateTimeMilliSeconds)
        file.save(image_file_path)

        import tensorflow as tf
        import numpy as np
        import warnings

        warnings.filterwarnings('ignore')

        test_image = tf.keras.preprocessing.image.load_img(image_file_path, target_size=(200,
200))

        # test_image = image.img_to_array(test_image)
        test_image = np.expand_dims(test_image, axis=0)
        # DAMAGE_COST MODEL
        classifierLoad = tf.keras.models.load_model(r'model/body.h5')
        result = classifierLoad.predict(test_image)
        result1 = ''
        if result[0][0] == 1:
            result1 = "front"
        elif result[0][1] == 1:
            result1 = "rear"
        elif result[0][2] == 1:
            result1 = "side"
        print('[INFO!!]', result1)

        ##########################################
        # file = request.files['fileupload1']
        # DateTimeMilliSeconds =
datetime.datetime.now().strftime("%Y%m%d_%H%M%S_%f")
        # image_file_path =
r'media/images/DamageType_{}.jpg'.format(DateTimeMilliSeconds)
```

```
    # file.save(image_file_path)
    # test_image = tf.keras.preprocessing.image.load_img(
    #     r'C:\Users\Macro\Downloads\Car damage\level\validation\03-severe\0017.JPEG',
target_size=(200, 200))
    #
    # test_image = np.expand_dims(test_image, axis=0)
    ##################################
    # Damage_type Model
    classifierLoad = tf.keras.models.load_model(r'model/level.h5')
    result = classifierLoad.predict(test_image)
    result2 = ''
    if result[0][0] == 1:
        result2 = "minor"
    elif result[0][1] == 1:
        result2 = "moderate"
    elif result[0][2] == 1:
        result2 = "severe"
    print('[INFO!!]', result2)
    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":
```

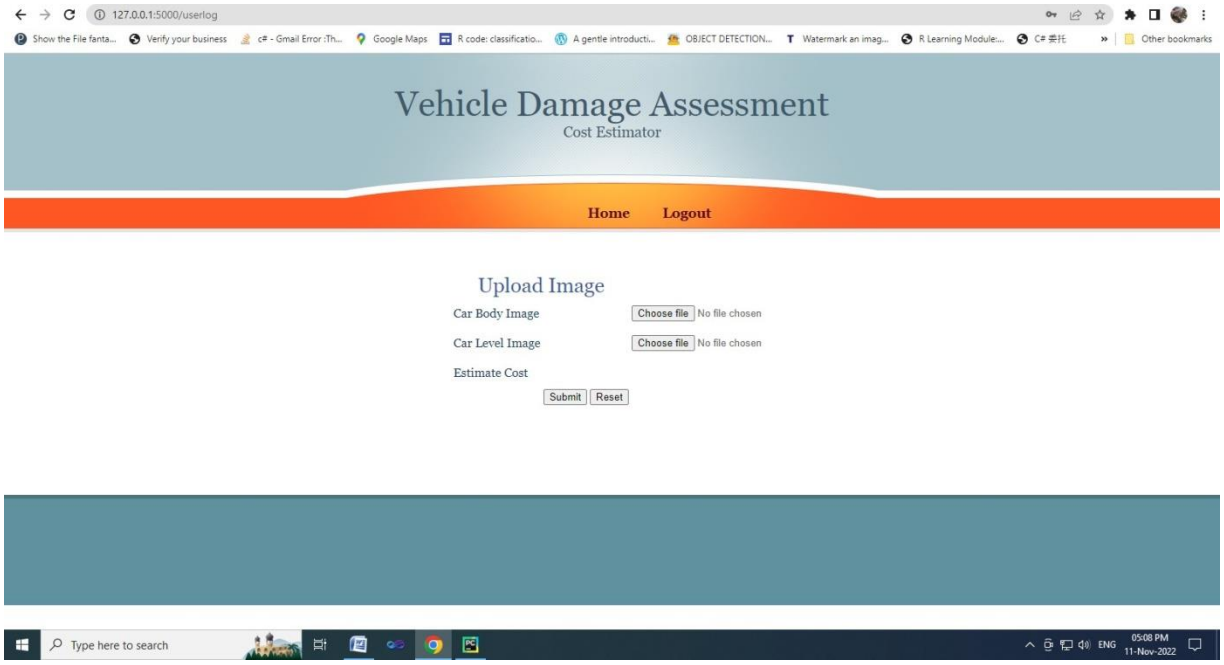
```
        value = "9000 - 11000 INR"
    elif result1 == "side" and result2 == "severe":
        value = "12000 - 15000 INR"
    else:
        value = "16000 - 50000 INR"
    print('[INFO!!] Damage Cost Range: ', value)
    # Please comment this return and uncomment the 'render_template' in 147 line
    return jsonify({'Damage Cost Range': value, 'Damage_angle': result1, 'Damage_type': result2})
    # return render_template('userhome.html', prediction=value)


if__name__== '_main_':
    app.run(debug=True, use_reloader=True)
```

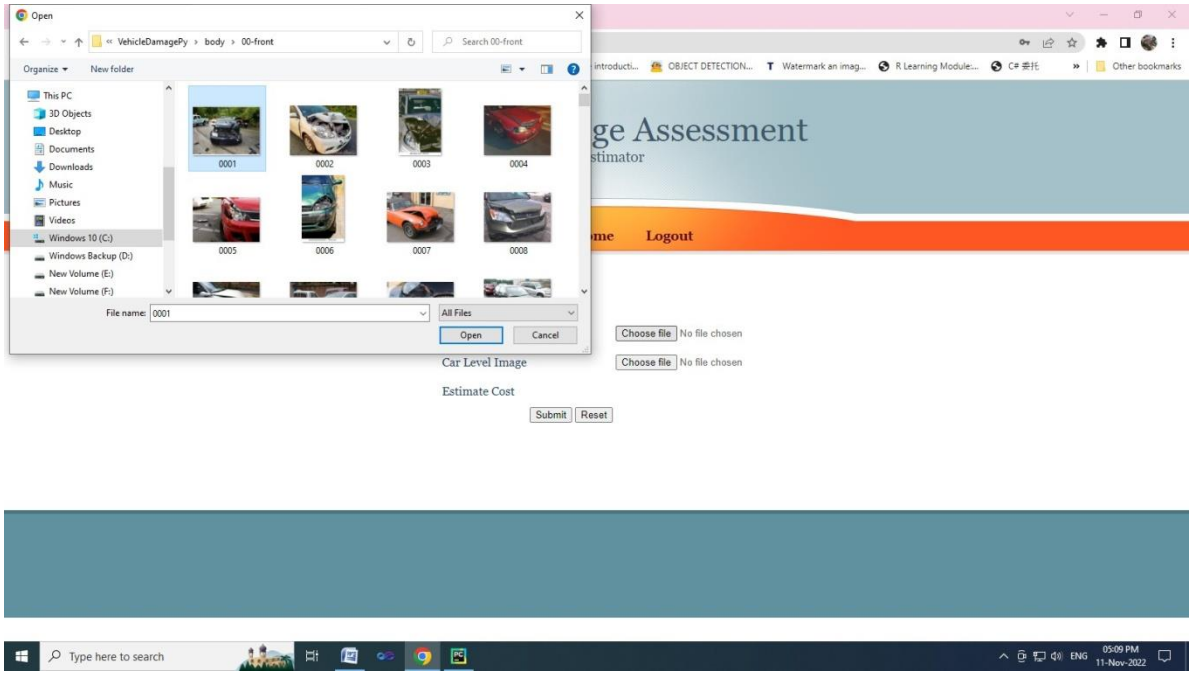
OUTPUT:

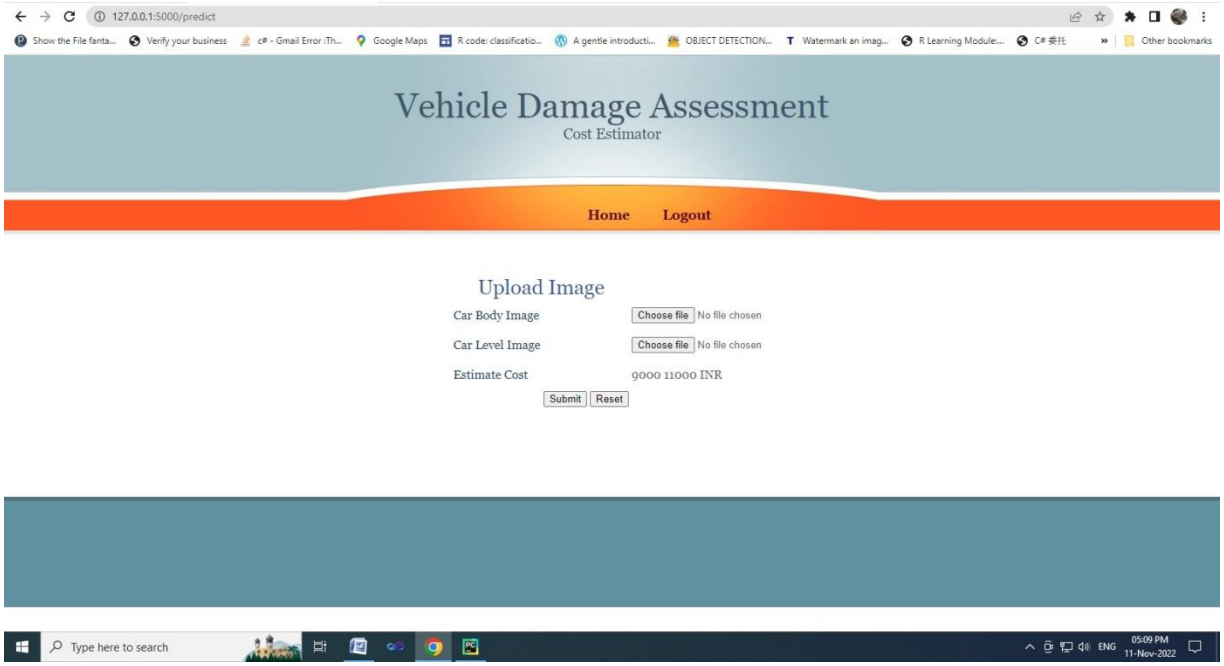



127.0.0.1:5000/NewUser
Vehicle Damage Assessment
Cost Estimator
Home
UserLogin
NewUser
New User
Name
Email id
Password
Submit
Reset
Type here to search
ENG
05:08 PM
11-Nov-2022


Vehicle Damage Assessment
Cost Estimator
Home
UserLogin
NewUser
UserLogin
Name
aisha
Email id
aishahabib2101@gmail
Password
Submit
Reset
Pause
Stop
Recorded in FREE version of Icecream Screen Recorder


127.0.0.1:5000/userlog
Vehicle Damage Assessment
Cost Estimator
Home
Logout
Upload Image
Car Body Image
Choose file
No file chosen
Car Level Image
Choose file
No file chosen
Estimate Cost
Submit
Reset
Type here to search
ENG
05:08 PM
11-Nov-2022


Open
VehicleDamagePy
body
00-front
Search 00-front
Organize
New folder
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Windows 10 (C:)
Windows Backup (D:)
New Volume (E:)
New Volume (F:)
0001
0002
0003
0004
0005
0006
0007
0008
File name: 0001
All Files
Open
Cancel
Logout
Choose file
No file chosen
Car Level Image
Choose file
No file chosen
Estimate Cost
Submit
Reset
Type here to search
ENG
05:09 PM
11-Nov-2022



GITHUB & PROJECT DEMO LINK:
https://youtu.be/Xk8s1IMpenI