TEAM ID : PNT2022TMID44616

PROJECT NAME : INTELLIGENT VEHICLE DAMAGE ASSESSMENT AND COST ESTIMATOR FOR INSURANCE COMPANIES

1.INTRODUCTION

1.1 Project Overview

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.

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. This model can also be used by lenders if they are underwriting a car loan, especially for a used car.

1.2 Purpose

At present, under the guidance of the new generation of information technology, the

rapid accumulation of data, the continuous improvement of computing power, the continuous

optimization of algorithm models, and the rapid rise of multi-scene applications have made profound changes in the development environment of artificial intelligence. In this paper, based on the demand of automobile insurance claims and intelligent transportation, combined with abundant basic data and advanced machine vision algorithm, an intelligent damage determination system of 'Artificial Intelligence + Vehicle Insurance' is constructed. This paper first introduces the functions of the intelligent damage assessment system. Secondly, it discusses the realization path of each functional module in detail, and finally puts forward thevision for the future.

2.LITERATURE SURVEY

2.1 Existing Problem

In today's world, Vehicles are increasing heavily. Because of increasing the vehicles, accidents are very common because the peoples are driving a car very fastly on the road. The people claim the money for repair the car through vehicle insurance when the accident happens. Because of incorrect claims, the company behaves badly and doesn't make payments currently. This

happens due to claims leakage, the claims leakage refers to the difference between the amounts secured by the company to the amount that company should have secured based on the claims. Still the damage to the car is examined clearly and it will take more time to claim the process according to the company policy. Although the company does one's best to speed up the claiming process delay. Differentiate the proposed system that is maybe speed up the car damage that can be check in process. Just by sending the image containing a damaged car and can system performs car damage detection in a minute rather than hours if it is inspected visually. The system can utilizes machine learning approach as well as computer vision to decide the damage analysis, location of the damage as well as severity of the damage.

2.2 References

THE FUNCTIONS OF INTELLIGENT DAMAGE ASSESSMENT SYSTEM

Intelligent damage determination system can be used to determine the appearance damage of vehicles in small cases. The system completes the whole process of survey and damage determination through four functions. They are:

(1)Accident investigation: Photographs of target vehicles and multiple trio vehicles were taken and uploaded, intelligent recognition, information input, intelligent recognition and event finalization are completed in accident investigation.

(2)Intelligent image damage assessment: image damage assessment is achieved by intelligent component recognition and intelligent damage recognition.

(3)Damage result output: Damage results including maintenance scheme recommendation and maintenance price recommendation are automatically given according to damage recognition results.

(4)Vehicle insurance anti-fraud: In the process of fixing the damage, the anti-fraud screening of vehicle insurance is completed by means of image fraud recognition and logical detection.

2.3 Problem Statement Definition

THE REALIZATION PATH OF INTELLIGENT DAMAGE ASSESSMENT SYSTEM

Accident Investigation : Accident investigation module includes the photography of certificates and vehicle photos, the intelligent recognition of certificate photos and the intelligent stereotyping work based on the basic information data of vehicle accessories.

Take Photos: The photographs taken in the accident investigation of intelligent damage determination system include driving license (front and side pages), driving license (front and side pages), person-car photograph, vehicle corner photograph and vehicle damage photograph. In order to apply the photograph of vehicle damage to the image damage based on artificial intelligence image recognition algorithm, some shooting requirements are put forward.

Intelligent ID Recognition: For the photos of the uploaded driving license (front and side pages), driving license (front and side pages) and other documents, the intelligent damage determination system embedded OCR recognition technology. The VIN code, license plate number, engine number, driver's name and other information of the uploaded driving license and driver's license can be intelligently recognized and filled in.

Intelligent Stereotyping and Fixing: The advantages of intelligent loss determination system are also reflected in its abundant basic information data. Through VIN code, the basic information database of vehicles and accessories can be automatically linked to realize the output of specific vehicle information such as brand, vehicle system, vehicle type, and OE code of parts corresponding to vehicle type, so as to realize one-to-one correspondence between vehicles and accessories.

Intelligent Image Damage Assessment : The core of intelligent damage fixing products is to determine which kind of damage happened to the exterior parts of the vehicle by image. The system has been

experimented many times in the development of intelligent image damage algorithm. Finally, it divides the problem into three parts: the recognition of appearance parts by image, the recognition of damage parts by image, and the determination of damage parts by relative position relationship.

Vehicle Appearance Component Recognition Algorithms : According to the statistics of vulnerable parts in vehicle accidents, thirty-one vehicle exterior parts have been identified in this product. Each part is divided into front and back parts, regardless of left and right parts. Aiming at the recognition of 31 vehicle appearance parts (regardless of left or right), the recognition algorithm for panoramic or local vehicles is realized, in the complex environment of rain and snow, too strong light or dark, by using the self-built data set of vehicle appearance parts and the depth learning target detection algorithm.

Damage Recognition Algorithms for Vehicle Appearance Components: This product is aimed at six types of vehicle appearance damage, and also applies the deep learning target detection method. Through the self-built damage data set, it can recognize highlight pictures, low contrast pictures and multi-category mixed damage. The damage recognition algorithm AP50 is 87.6%.

An Algorithm for Locating Components and Damages: By calculating the intersection relationship between the polygon identified by the algorithm of vehicle appearance components and the rectangular position identified by the algorithm of appearance damage, the appearance parts where the damage occurs are finally determined. At present, based on vehicle appearance component recognition algorithm, vehicle appearance damage recognition algorithm and image position determination algorithm, the comprehensive accuracy of image damage determination algorithm reaches 87.3%.

Output of Loss Assessment Result : The output of fixed-loss results can not be separated from maintenance rules and repair logic. Among them, the maintenance rules are based on the experience of fixing damage and testing the appearance of components in the specific material damage needs to be maintained. The repair logic needs to formulate the damage inclusion relation logic, for example, if there are two damages in the same component, the maintenance scheme should adopt the scheme with higher maintenance level.

CONCLUSION

In the future, we will continue to explore the innovation of insurance technology of 'AI + Vehicle Insurance'. We hope that we can use the power of intelligent damage determination system. On the one hand, the owner can take photos by one click to achieve rapid loss determination, price estimation and immediate compensation. On the other hand, it assists insurance companies to achieve rapid and accurate pricing in the process of fixing losses and claims. Finally, by combining the rapid compensation of accident vehicles to relieve traffic pressure, to avoid more serious personal and property losses caused by secondary accidents .

3.IDEATION&PROPOSED SOLUTION

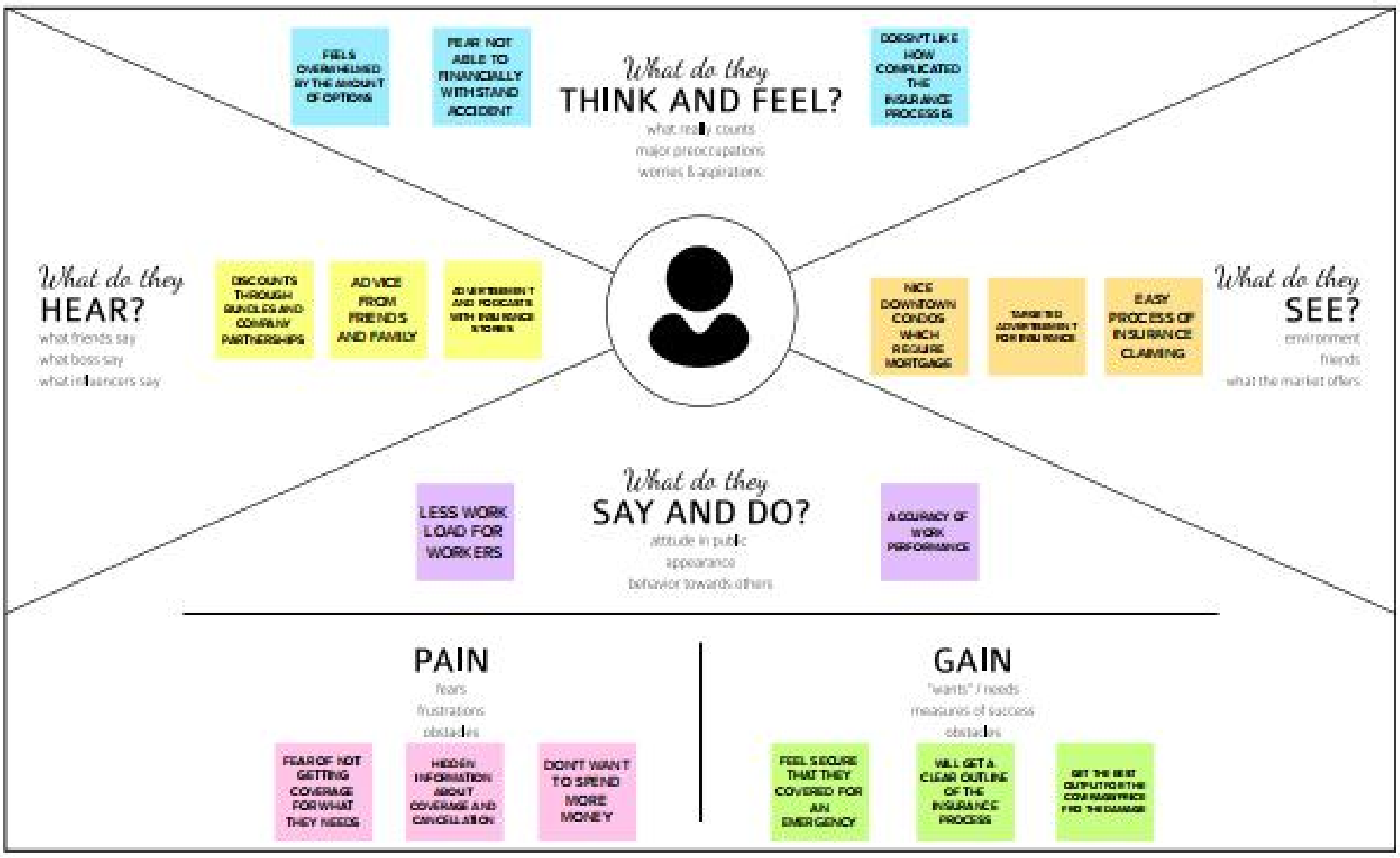
3.1 Empathy Map Canvas

Empathy Map Canvas

Gain insight and understanding on solving customer problems.

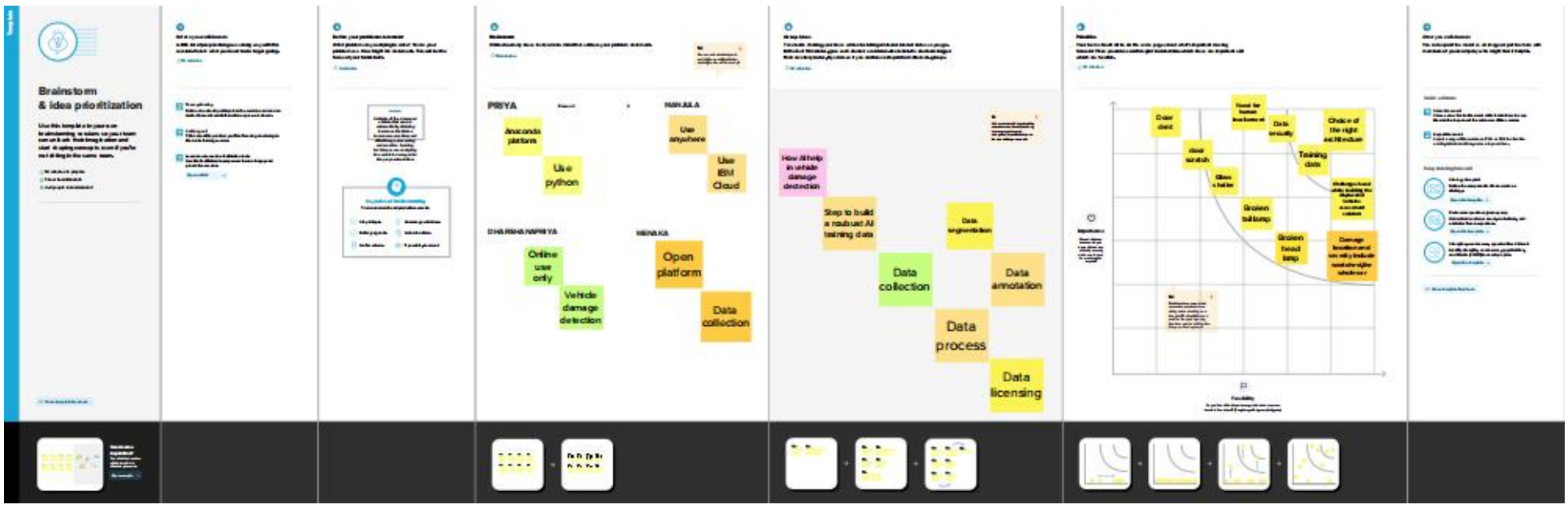
1

Intelligent Vehicle Damage Assessment and Cost Estimator for Insurance Companies



Share your feedback

3.2 Ideation & Brainstorming



3.3 Proposed Solution

Project Design Phase-I
Proposed Solution Template

Date	26 September 2022
Team ID	PNT2022TMID44616
Project Name	Project – Intelligent Vehicle Damage Assessment And Cost Estimator for Insurance Companies
Maximum Marks	2 Marks

Proposed Solution Template:

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

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The goal of this dissertation is to develop a system capable of automatically identifying and locating damages in images of cars.
2.	Idea / Solution description	Computer vision and, in particular,image classification are fields of study where major break throughs were achieved in the last few years.
3.	Novelty / Uniqueness	The work presented here merely lays ground for further, more detailed work in this field.The usage of some alternative methods ,possibly using these models only as feature extractors, might prove worthy.
4.	Social Impact / Customer Satisfaction	The problem of identifying damage in image of cars is far from trivial.Cars are complex objects with important intra- class differences but also suble onces.
5.	Business Model (Revenue Model)	Previous research on insurer cost of equlity focuses on single – period asset pricing models.
6.	Scalability of the Solution	Both architectures were also tried in tried in the damage location task. The task consists of classifying the images according to the approximate location of the damage exhited.

3.4 Problem Solution Fit

Problem-Solution fit canvas 2.0

Purpose / Vision

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) — CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

The Traditional insurance segmentation process relies only one basic demographic information such as age,gender,and life stage.
coming up with innovative insurance products marketing strategies and personalized experiences in only possible with effective customer segment.
The biggest difference rator in 2022 and beyond is going to be the customer-centric approach to segmentation.

6. CUSTOMER CONSTRAINTS — CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

The direct-to-consumer insurance sales model has the potential to reshape the insurance industry in its entirety-from the types of products sold , to the way they are bundled ;from how those products are distributed, to the role that agents and brokers play in the sales. but to every thing ,there is a limit .
According to Slaylon search, the first answer is legal.D2C is relatively new model in a highly regulated industry, and as such,its likely to run into some regulatory roadblocks "that make this transition more complex and intimidating".
The plan for the customer of the future.
They integrate interactions across all channels , including digital ones.
They offer board insurance service.
They have invested in flexible structures.

5. AVAILABLE SOLUTIONS — AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

The decision to stay with your insurance company and choose a different car insurance provide should rest on many things.
Advantages of life insurance as an investment ,whenever you visit a financial advisor for finandal planning you can see that most of them suggest you go for life insurance .
They encourage you to invest in life insurance so that you and your loved ones are not only protected but also a considerable amount of returns can be obtains from the policy.
This is the major disadvantage of life insurance policy. The higher the age the higher would the premium to be paid in the life insurance .
In india many insurance companies offer different types of life insurance plans.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS — J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

After a car accident ,you could receive a call from the other drive's insurance company.
This may happen regardless of who was at fault for the accident.
even if the collision was not your fault, they might try to contact you.
They want accurate information from their agent available with just a single phone call consumer also expect the information they gather from insurance agent to be tailored to their situation .
they want empathy, advice, and meaningful information.

9. PROBLEM ROOT CAUSE — RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

An increasingly common technique used to understand the "why" of loss events in order to prevent their reoccurrence of a ll claim and insurance fraud may not be a realistic goals,the RCA technique will often yield clues useful in modifying the exposure or risk and possibly reduce similar loss events in the mid to long term.
That is a realistic objective and one that has perfect application to fraud.
Easy-to-use-and-understanding reference explaining the various funding options for your organization's risks
Explains reinsurance, alternative markets,and tax and accounting implications of various risks financing mechanisms.
Excellent resource for state specific regulations.
Learn what it takes to establish a successful captive insurance company-one that sets the standards the test of time .
provides step-by-step instructions that would benfit novices and seasoned veterans alike.
This is THE reference package for any risk or insurance professional who works in specialty lines.

7. BEHAVIOUR — BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits;
indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

December 17,2008 -trying to resolve issues with managed care and other health insurance companies can be a frustrating and time-consuming experience.
For example,the insurer may pay you a lower in -network rate instead of the higher out-of-network rate even though you are an out-of-network provider.
You may have difficulty determining why the company is not authorizing certain services for your patients.
1.anticipate problems and create a place to keep records
2.Clarify the company's basis for its action
3.Gather relevant information
4.Start cordially, then escalate
5.Create a paper trail.
6.keep your communications concise.
7.Affirm what you need for the company to do
8.Engage people and entities they may give you leverage
9.Be ready to go higher up the chain of command
10.Be persistent

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS — TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

Seeing their neighbour installing solar planels ,reading about a more efficient solution in the news.one court held that injury triggers coverage,including injury occuring with inhalation exposure,while the injurious substance is"in existence" within the injured person and at manifestation of the illness or disease.

4. EMOTIONS: BEFORE / AFTER — EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

customer inquiries in the insurance industry are often issue-driven ,where clients reach out to initiate a plan,when they seek information regarding a scheme,when they find any change in policy,and ultimately when make a claim.

10. YOUR SOLUTION — SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

Challenges
Fraud Detection
Customer Insight Management
Claims Management
Personalized Risk Pricing
360-Degree Customer Profiling and segmentation
Smart product Innovation
Connected payment Capabilities
Multichannel Optimization

8. CHANNELS of BEHAVIOUR — CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

Online term insurance is much cheaper..as there is no intermediary or agents involved, the overall cost comes out much less as the insurer saves on the commission

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

Offline term insurance plans are sold by agents working for the company.
As the agents take care of your application ,they in return charge a commission making the overall cost of the term insurance higher.

Extract online & offline CH of BE

AMALTAMA

4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

4.2 Non Functional Requirement

Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	17 October 2022
Team ID	PNT2022TMID44616
Project Name	Project - Intelligent Vehicle Damage Assessment and Cost Estimator for Insurance Companies
Maximum Marks	4 Marks

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through phone number
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User dashboard	Single Sample Prediction Multiple Sample Prediction View User History

Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR1	**Usability**	To predict the cost for the exact damaged parts of the vehicle.
NFR2	**Security**	It is secure to claim the insurance from the company with efficiency.
NFR3	**Reliability**	It can detect the damage from all parts of the vehicle.
NFR4	**Performance**	Detect the damages of any kind of vehicle, It may be minor or major damage.
NFR5	**Availability**	It is accessible for both insurance company and vehicle owner to estimate the cost of damage.
NFR6	**Scalability**	To measure the accurate cost for the damage of vehicle.

5. PROJECT DESIGN

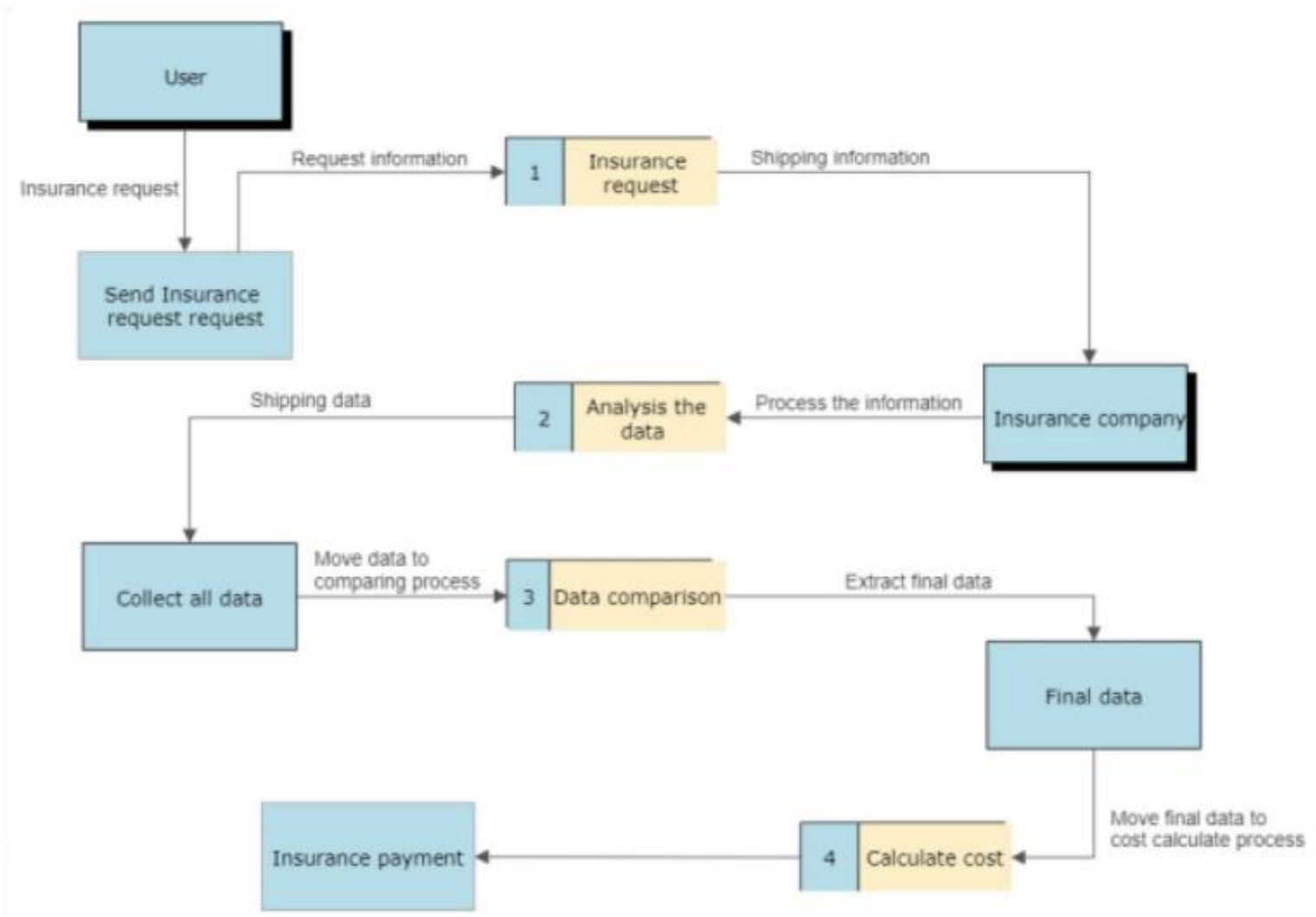
5.1 Data Flow Diagrams

Project Design Phase- II
Data Flow Diagram & User Stories

Date	17 October 2022
TeamID	PNT2022TMID44616
ProjectName	Project-Intelligent Vehicle Damage Assessment & Cost Estimator For Insurance Companies
MaximumMarks	4Marks

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows with in a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



User stories

Use the below template to list all the user stories for the product.

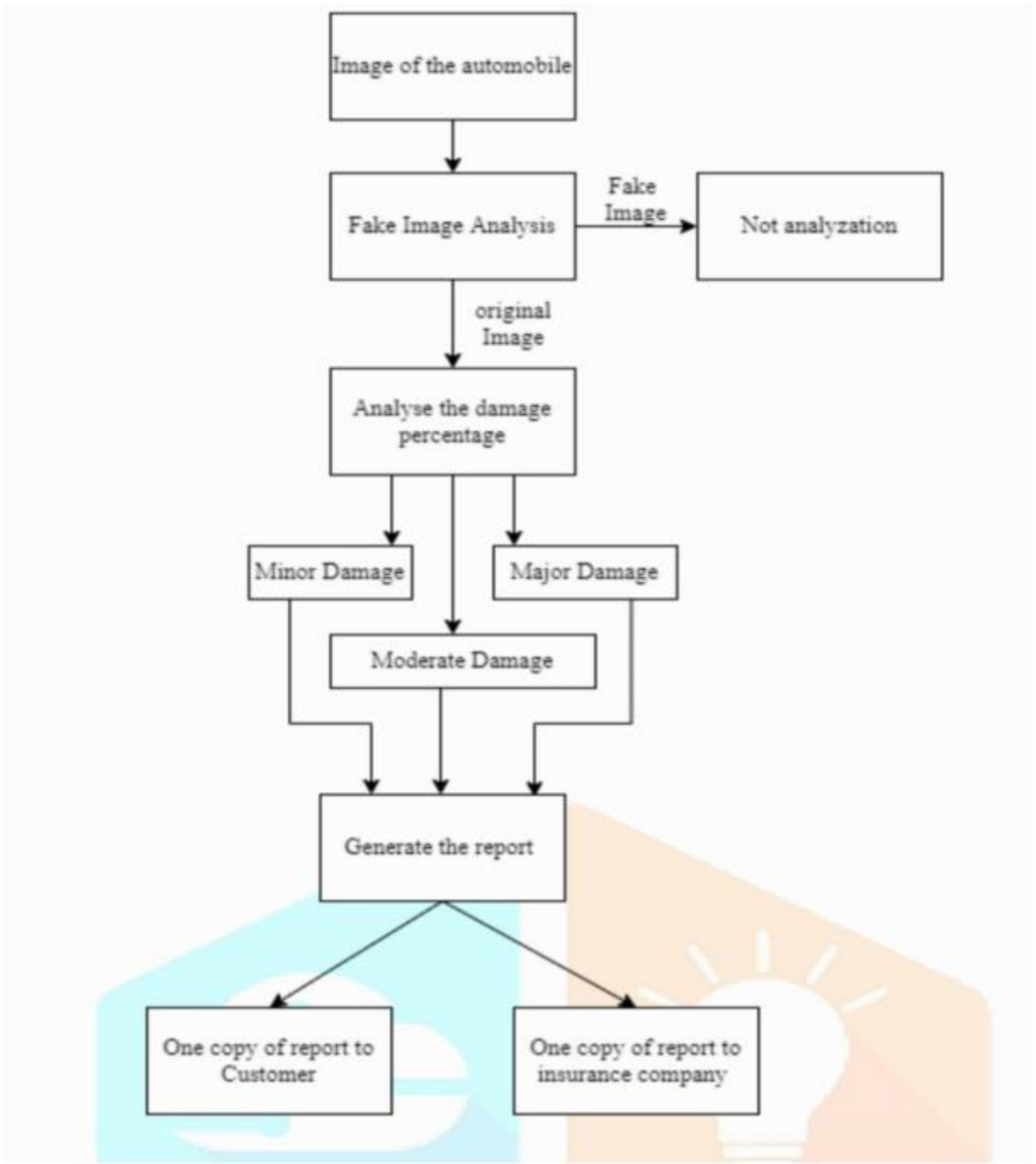
User Type	Functional Requirement (Epic)	User Story Number	User Story/ Task	Acceptanc e criteria	Priority	Release
Customer (Mobile user)	Registration	US N-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		US N-2	As a user, I will receive confirmation email once I have Registered for the application	I can receive confirmatio n email & click confirm	High	Sprint-1
		US N-3	As a user, I can register for the application through G mail	I can receive confirmatio n Gmail & click confirm	Medium	Sprint-1
	Login	US N-4	As a user, I can login to the application by entering email & password		High	Sprint-1
	Dashboard	US N-5	As a user, I can view all the plans and methods in dashboard		High	Sprint-1
Customer (Web user)	Insurance claim	US N-6	As a user, I can register for claim my insurance	I can receive confirmation Email & claim my Insurance	High	Sprint-2
Customer Care Executive	Q/A services	US N-7	As a user, I can make a call to Support line to get help with a product or service.	Phone call, messages and Email	High	Sprint-3
Administrator	Insurance	US N-8	As a user, I can claim my insurance After getting confirmation from the administrator.	I can accept the insurance After verified the documents	High	Sprint-3

5.2 Solution Architecture

ProjectDesignPhase-I
SolutionArchitecture

Date	17 ctober 2022
TeamID	PNT2022TMID44616
ProjectName	Project – Intelligent vechicle damage and cost estimator insurance companies
MaximumMarks	4Marks

SolutionArchitecture:



Project Design Phase-II

Technology Stack (Architecture & Stack)

Date	17 October 2022
Team I D	PNT2022TMID44616
Project Name	Intelligent Vehicle Damage Assessment and Cost Estimator for Insurance Companies
Maximum Marks	4 Marks

Technical Architecture:

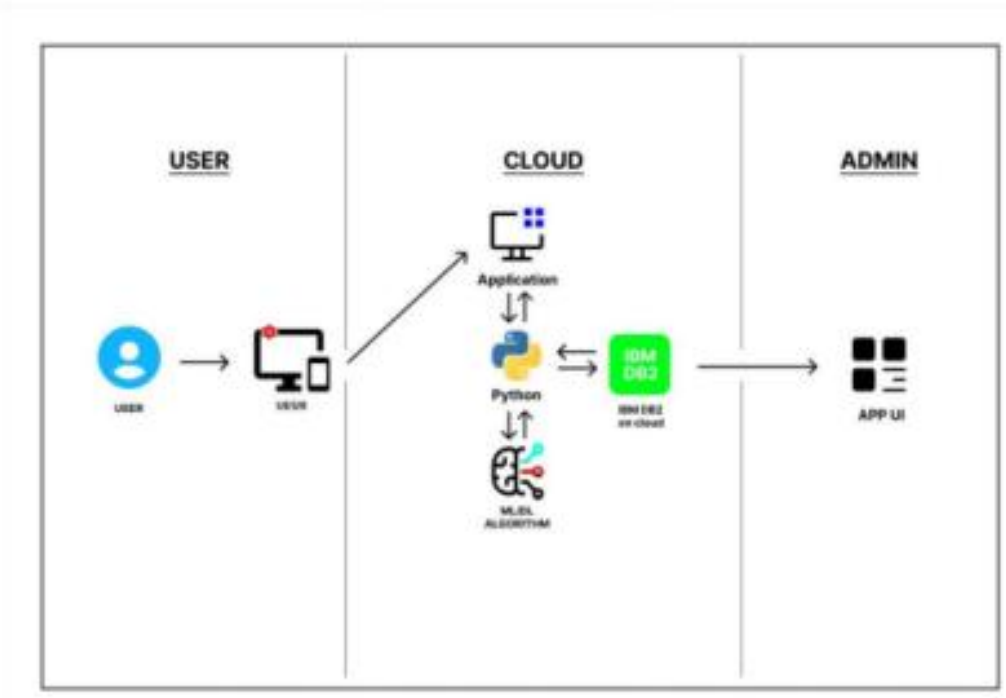


Table-1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Using Web UI, MobileApp, Chatbot etc.	HT M L, CSS, JavaScript / Angular Js / React Js.
2.	Application Logic- 1	Application mainly used for predicting cost.	Java / Python
3.	Application Logic- 2	It is used for detecting damaged parts.	I B M Watson STT service
4.	Application Logic- 3	The Customer claims Insurance from the companies.	I B M Watson Assistant
5.	Database	Data Type, Configurations, Data set are used in the data base.	MySQ L.
6.	Cloud Database	The Vehicle dataisstored in Cloud data base for retrieval uses.	I B M D B2, I B M Cloudant
7.	File Storage	In application, it contains all data types in file storage.	I B M Block Storage or Other Storage Service or Local File system.
8.	External A P I	To perform a designed function built around sharing data and executing pre- defined processes.	I B M Weather A P I.
9.	Machine Learning Model	The purpose of machine learning, make decisions only based on the given input.	Object Recognition Model.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open- Source Frameworks	Python open- source frame works used	Python
2.	Security Implementations	It is secure to caim the insurance from the company with efficiency.	A I
3.	Scalable Architecture	To measure the accurate cost for the damage of a vehicle.	Python
4.	Availability	It is accessible for both insurance companies and vehicle owners to estimate the cost of damage.	A I
5.	Performance	Detecting the damage of any kind of vehicle, It may be minor or major damage.	Python, C N N.

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story/ Task	Acceptanc e criteria	Priority	Release
Customer (Mobile user)	Registration	US N-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		US N-2	As a user, I will receive confirmation email once I have Registered for the application	I can receive confirmatio n email & click confirm	High	Sprint-1
		US N-3	As a user, I can register for the application through G mail	I can receive confirmatio n Gmail & click confirm	Medium	Sprint-1
	Login	US N-4	As a user, I can login to the application by entering email & password		High	Sprint-1
	Dashboard	US N-5	As a user, I can view all the plans and methods in dashboard		High	Sprint-1
Customer (Web user)	Insurance claim	US N-6	As a user, I can register for claim my insurance	I can receive confirmation Email & claim my Insurance	High	Sprint-2
Customer Care Executive	Q/A services	US N-7	As a user, I can make a call to Support line to get help with a product or service.	Phone call, messages and Email	High	Sprint-3
Administrator	Insurance	US N-8	As a user, I can claim my insurance After getting confirmation from the administrator.	I can accept the insurance After verified the documents	High	Sprint-3

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Project Planning Phase
(Product Backlog, Sprint Planning, Stories, Story points)

Date	30 October 2022
Team ID	PNT2022TMID44616
Project Name	Intelligent Vehicle Damage Assessment and Cost Estimation for Insurance Companies.

Title	Description	Date
Literature Survey and Information Gathering	Gathering Information by referring the technical papers, research publications etc	10 September 2022
Prepare Empathy Map	To capture user pain and gains Prepare List of Problem Statement	17 September 2022
Ideation	Prioritise a top 3 ideas based on feasibility and Importance	18 September 2022
Proposed Solution	Solution include novelty, feasibility, business model, social impact and scalability of solution	1 October 2022
Problem Solution Fit	Solution fit document	1 October 2022
Solution Architecture	Solution Architecture	1 October 2022
Customer Journey	To Understand User Interactions and experiences with application	8 October 2022
Functional Requirement	Prepare functional Requirement	9 October 2022
Data flow Diagrams	Data flow diagram	11 October 2022
Technology Architecture	Technology Architecture diagram	15 October 2022

Milestone & sprint delivery plan	Activity what we done &further plans	21 October 2022
Project Development- Delivery of sprint 1,2,3 &4	Develop and submit the developed code by testing it	24 October 2022 – 19 November 2022

6.2 Sprint Delivery Schedule

Project Planning Phase
(Product Backlog, Sprint Planning, Stories, Story points)

Date	31 October 2022
Team ID	PNT2022TMID44616
Project Name	AI-Intelligent Vehicle Damage Assessment and Cost Estimator for Insurance Companies .

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 resister for the application by entering my email,password, and confirming my password.	2	High	Priya.G
Sprint-1	Registration	USN-2	As a user, I will receive confirmation email once I have Registered for the Application	1	High	Priya G
Sprint-1	Registration	USN-3	As a user ,I can register for the application Gmail.	2	Low	Priya G
Sprint-1	Login	USN-4	As a user ,I can Login to the application by entering email & password .	1	Medium	Priya G
Sprint-2	Dashboard	USN-5	As a user ,I can view all the plans and methods in the Dashboard.	1	High	Manjula V
Sprint-3	Storage	USN-1	As a user, I can Register for claim my insurance.	2	High	Priyadharsini S
Sprint-3		USN-2	As a user, I can make a call to support line to get help with a product or service	2	High	Priyadharsini S
Sprint-4		USN-3	As a user, I can claim my insurance After getting from the administrator.	1	Medium	Menaka A

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

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022		
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022		
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		

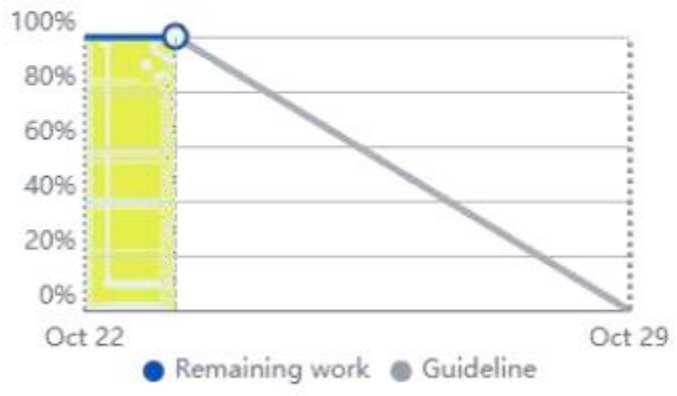
Velocity:
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

6.3 Reports From JIRA

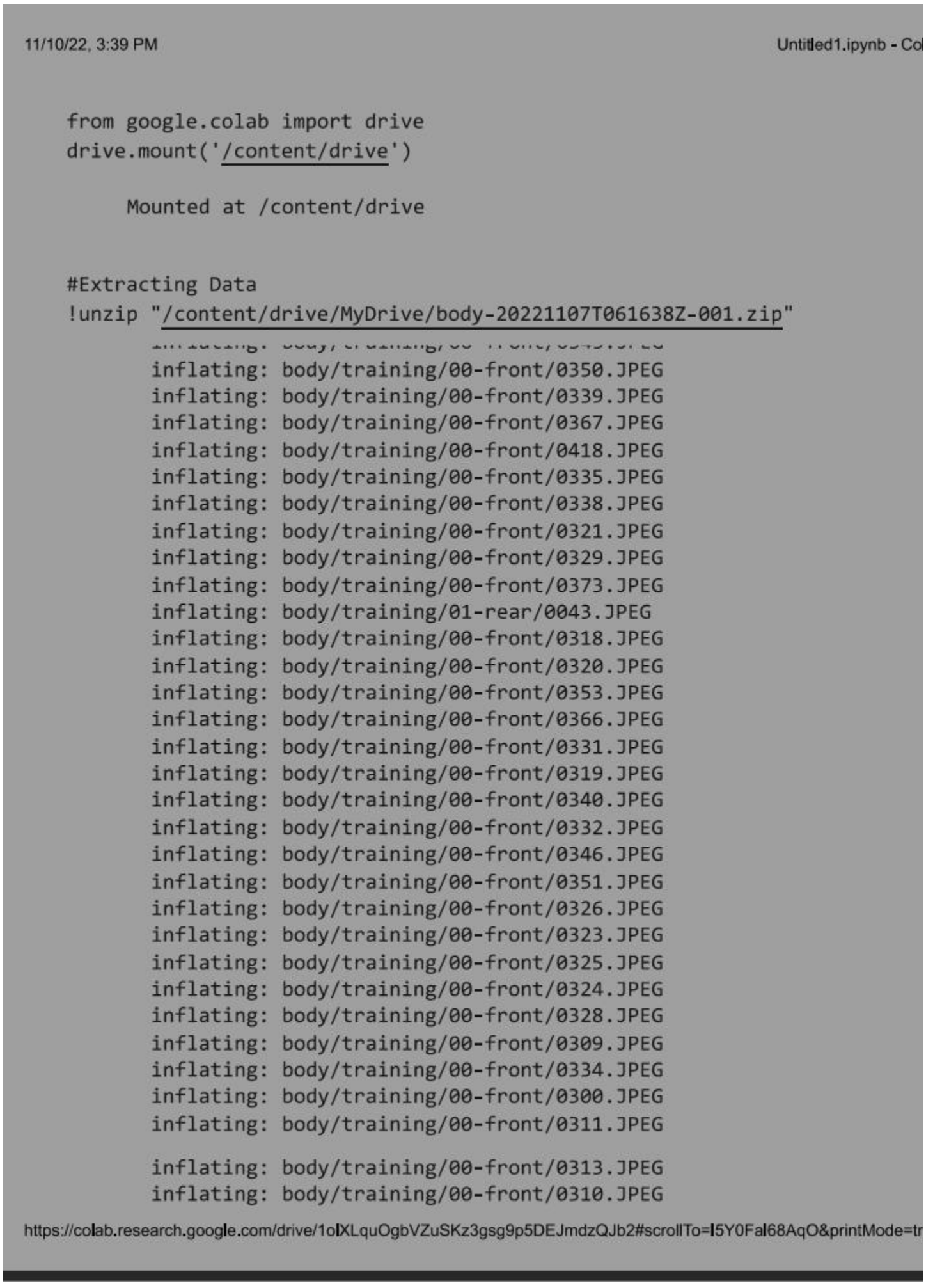
Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



7 . CODING & SOLUTIONING

7 .1 Feature 1



```
inflating: body/training/00-front/0303.JPEG
inflating: body/training/00-front/0302.JPEG
inflating: body/training/00-front/0330.JPEG
inflating: body/training/00-front/0312.JPEG
inflating: body/training/00-front/0304.JPEG
inflating: body/training/00-front/0301.JPEG
inflating: body/training/00-front/0419.JPEG
inflating: body/training/00-front/0342.JPEG
inflating: body/training/00-front/0333.JPEG
inflating: body/training/00-front/0314.JPEG
inflating: body/training/00-front/0307.JPEG
inflating: body/training/00-front/0327.JPEG
inflating: body/training/00-front/0315.JPEG
inflating: body/training/00-front/0297.JPEG
inflating: body/training/00-front/0316.JPEG
inflating: body/training/00-front/0298.JPEG
inflating: body/training/00-front/0299.JPEG
inflating: body/training/00-front/0295.JPEG
inflating: body/training/00-front/0322.JPEG
inflating: body/training/00-front/0343.JPEG
inflating: body/training/00-front/0317.JPEG
inflating: body/training/00-front/0305.JPEG
inflating: body/training/00-front/0306.JPEG
inflating: body/training/00-front/0308.JPEG
inflating: body/training/00-front/0296.JPEG
inflating: body/training/01-rear/0049.JPEG
```

```
#Import req. Lib.
from tensorflow.keras.preprocessing.image import ImageDataGenerator


#Augmentation On Training Variable
train_datagen = ImageDataGenerator(rescale= 1./255,
                                   shear_range=0.1,
                                   zoom_range=0.2,
                                   horizontal_flip =True)


#Augmentation On Testing Variable
test_datagen = ImageDataGenerator(rescale= 1./255)
```

```
#Augmentation On Training Variable
training_test = train_datagen.flow_from_directory('/content/body/training',
target_size=(224,224),
batch_size=10,
class_mode='categorical',)
```

```
Found 979 images belonging to 3 classes.
```

```
#Augmentation On Training Variable
test_set = test_datagen.flow_from_directory('/content/body/validation',
target_size=(224,224),
batch_size=100,
class_mode='categorical')
```

```
Found 171 images belonging to 3 classes.
```

```
#Extracting Data
!unzip "/content/drive/MyDrive/level-20221107T084432Z-001.zip"
```

```
inflating: level/training/03-severe/0014.JPEG
inflating: level/training/03-severe/0286.JPEG
inflating: level/training/03-severe/0288.JPEG
inflating: level/training/03-severe/0015.JPEG
inflating: level/training/03-severe/0313.jpeg
inflating: level/training/03-severe/0021.JPEG
inflating: level/training/02-moderate/0022.JPEG
inflating: level/training/02-moderate/0029.JPEG
inflating: level/training/02-moderate/0019.JPEG
inflating: level/training/03-severe/0013.JPEG
inflating: level/training/03-severe/0294.JPEG
inflating: level/training/03-severe/0007.JPEG
inflating: level/training/03-severe/0377.JPEG
inflating: level/training/03-severe/0287.JPEG
inflating: level/training/03-severe/0291.JPEG
inflating: level/training/03-severe/0011.JPEG
inflating: level/training/02-moderate/0028.JPEG
inflating: level/training/03-severe/0008.JPEG
inflating: level/training/03-severe/0217.jpeg
```

```
inflating: level/training/02-mcderate/0020.JPEG
inflating: level/training/02-mcderate/0026.JPEG
inflating: level/training/03-severe/0017.JPEG
inflating: level/training/03-severe/0012.JPEG
inflating: level/training/02-mcderate/0027.JPEG
inflating: level/training/03-severe/0005.JPEG
inflating: level/training/03-severe/0010.JPEG
inflating: level/training/02-mcderate/0021.JPEG
inflating: level/training/03-severe/0016.JPEG
inflating: level/training/02-mcderate/0015.JPEG
inflating: level/training/03-severe/0002.JPEG
inflating: level/training/02-mcderate/0008.JPEG
inflating: level/training/02-mcderate/0017.JPEG
inflating: level/training/02-mcderate/0023.JPEG
inflating: level/training/03-severe/0009.JPEG
inflating: level/training/03-severe/0023.jpeg
inflating: level/training/03-severe/0003.JPEG
inflating: level/training/02-mcderate/0003.JPEG
inflating: level/training/02-mcderate/0018.JPEG
inflating: level/training/02-mcderate/0010.JPEG
inflating: level/training/02-mcderate/0014.JPEG
inflating: level/training/02-mcderate/0012.JPEG
inflating: level/training/02-mcderate/0005.JPEG
inflating: level/training/02-mcderate/0013.JPEG
inflating: level/training/02-mcderate/0009.JPEG
inflating: level/training/02-mcderate/0036.JPEG
inflating: level/training/03-severe/0006.JPEG
inflating: level/training/02-mcderate/0007.JPEG
inflating: level/training/02-mcderate/0004.JPEG
inflating: level/training/02-mcderate/0016.JPEG
inflating: level/training/02-mcderate/0001.JPEG
inflating: level/training/03-severe/0004.JPEG
inflating: level/training/02-mcderate/0002.JPEG
inflating: level/training/02-mcderate/0042.JPEG
inflating: level/training/02-mcderate/0011.JPEG
inflating: level/training/02-mcderate/0006.JPEG
inflating: level/training/02-mcderate/0058.JPEG
```

```
#Import req. Lib.
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
#Augmentation On Training Variable
train_datagen = ImageDataGenerator(rescale= 1./255,
                                   shear_range=0.1,
                                   zoom_range=0.1,
                                   horizontal_flip =True)



#Augmentation On Testing Variable
test_datagen = ImageDataGenerator(rescale= 1./255)


#Augmentation On Training Variable
training_set = train_datagen.flow_from_directory('/content/level/training',
target_size=(224,224),
batch_size=10,
class_mode='categorical')

     Found 979 images belonging to 3 classes.


#Augmentation On Training Variable
test_set = test_datagen.flow_from_directory('/content/level/validation',
target_size=(224,224),
batch_size=10,
class_mode='categorical')

     Found 171 images belonging to 3 classes.
```

Colab paid products - Cancel contracts here

✓ 0s completed at 3:38 PM

7 .2 Feature 2

FOR BODY DAMAGE

IMAGE PRE PROCESSING

1. Import The ImageDataGenerator Library

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

:2. Configure ImageDataGenerator Class

Image Data Augmentation

```
train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.1,
                                   zoom_range = 0.1,
                                   horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1./255)
```

3. Apply ImageDataGenerator Functionality To Trainset And Testset

```
training_set =
train_datagen.flow_from_directory('/content/drive/MyDrive/IBM -
PROJECT/Data set/body-20221023T072112Z-001/body/training',
                                                  target_size = (224,
224),
                                                  batch_size = 10,
                                                  class_mode =
'categorical')
test_set =
test_datagen.flow_from_directory('/content/drive/MyDrive/IBM -
PROJECT/Data set/body-20221023T072112Z-001/body/validation',
                                                target_size = (224, 224),
                                                batch_size = 10,
                                                class_mode =
'categorical')
```

```
Found 979 images belonging to 3 classes.
Found 171 images belonging to 3 classes.
```

MODEL BUILDING

1. Importing The Model Building Libraries

```
import tensorflow as tf
from tensorflow.keras.layers import Input, Lambda, Dense, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.applications.vgg19 import VGG19
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import
ImageDataGenerator,load_img
```

```
from tensorflow.keras.models import Sequential
import numpy as np
from glob import glob
```

2. Loading The Model

```
IMAGE_SIZE = [224, 224]

train_path = '/content/drive/MyDrive/IBM - PROJECT/Data set/body-
20221023T072112Z-001/body/training'
valid_path = '/content/drive/MyDrive/IBM - PROJECT/Data set/body-
20221023T072112Z-001/body/validation'

vgg16 = VGG16(input_shape=IMAGE_SIZE + [3], weights='imagenet',
include_top=False)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-
applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58889256/58889256 [==============================] - 0s 0us/step
```

3. Adding Flatten Layer

```
for layer in vgg16.layers:
    layer.trainable = False

folders = glob('/content/drive/MyDrive/IBM - PROJECT/Data set/body-
20221023T072112Z-001/body/training/*')

folders
```

```
['/content/drive/MyDrive/IBM - PROJECT/Data set/body-20221023T072112Z-
001/body/training/02-side',
 '/content/drive/MyDrive/IBM - PROJECT/Data set/body-20221023T072112Z-
001/body/training/01-rear',
 '/content/drive/MyDrive/IBM - PROJECT/Data set/body-20221023T072112Z-
001/body/training/00-front']
```

```
x = Flatten()(vgg16.output)

len(folders)
```

```
3
```

4. Adding Output Layer

```
prediction = Dense(len(folders), activation='softmax')(x)
```

5. Creating A Model Object

```
model = Model(inputs=vgg16.input, outputs=prediction)

model.summary()
```

```
Model: "model"
_________________________________________________________________
```

```
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 224, 224, 3)]     0
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
flatten (Flatten)            (None, 25088)             0
dense (Dense)                (None, 3)                 75267
=================================================================
Total params: 14,789,955
Trainable params: 75,267
Non-trainable params: 14,714,688
_________________________________________________________________
```

6. Configure The Learning Process

```
model.compile(
  loss='categorical_crossentropy',
  optimizer='adam',
  metrics=['accuracy']
)
```

7. Train The Model

```
r = model.fit_generator(
  training_set,
  validation_data=test_set,
  epochs=25,
  steps_per_epoch=len(training_set),
  validation_steps=len(test_set)
)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6:
UserWarning: `Model.fit_generator` is deprecated and will be removed
in a future version. Please use `Model.fit`, which supports
generators.


Epoch 1/25
98/98 [==============================] - 560s 6s/step - loss: 1.2275 -
accuracy: 0.5383 - val_loss: 0.8698 - val_accuracy: 0.6608
Epoch 2/25
98/98 [==============================] - 584s 6s/step - loss: 0.7810 -
accuracy: 0.7007 - val_loss: 0.8931 - val_accuracy: 0.6491
Epoch 3/25
98/98 [==============================] - 538s 5s/step - loss: 0.4842 -
accuracy: 0.8264 - val_loss: 0.8348 - val_accuracy: 0.6842
Epoch 4/25
98/98 [==============================] - 537s 5s/step - loss: 0.3813 -
accuracy: 0.8560 - val_loss: 0.9010 - val_accuracy: 0.6901
Epoch 5/25
98/98 [==============================] - 537s 5s/step - loss: 0.2735 -
accuracy: 0.8999 - val_loss: 1.0660 - val_accuracy: 0.6901
Epoch 6/25
98/98 [==============================] - 538s 5s/step - loss: 0.2211 -
accuracy: 0.9295 - val_loss: 1.0073 - val_accuracy: 0.7076
Epoch 7/25
98/98 [==============================] - 536s 5s/step - loss: 0.2163 -
accuracy: 0.9224 - val_loss: 0.9560 - val_accuracy: 0.7251
Epoch 8/25
98/98 [==============================] - 538s 6s/step - loss: 0.1728 -
accuracy: 0.9397 - val_loss: 1.0719 - val_accuracy: 0.6491
Epoch 9/25
98/98 [==============================] - 540s 6s/step - loss: 0.1423 -
accuracy: 0.9581 - val_loss: 1.0706 - val_accuracy: 0.6901
```

```
Epoch 10/25
98/98 [==============================] - 539s 6s/step - loss: 0.1118 -
accuracy: 0.9704 - val_loss: 1.1651 - val_accuracy: 0.6842
Epoch 11/25
98/98 [==============================] - 538s 5s/step - loss: 0.0808 -
accuracy: 0.9785 - val_loss: 1.1212 - val_accuracy: 0.7076
Epoch 12/25
98/98 [==============================] - 549s 6s/step - loss: 0.0751 -
accuracy: 0.9857 - val_loss: 1.1451 - val_accuracy: 0.6842
Epoch 13/25
98/98 [==============================] - 555s 6s/step - loss: 0.0730 -
accuracy: 0.9816 - val_loss: 1.0812 - val_accuracy: 0.6842
Epoch 14/25
98/98 [==============================] - 535s 5s/step - loss: 0.1074 -
accuracy: 0.9734 - val_loss: 1.2204 - val_accuracy: 0.6842
Epoch 15/25
98/98 [==============================] - 539s 6s/step - loss: 0.0598 -
accuracy: 0.9888 - val_loss: 1.6480 - val_accuracy: 0.6316
Epoch 16/25
98/98 [==============================] - 543s 6s/step - loss: 0.0810 -
accuracy: 0.9806 - val_loss: 1.2050 - val_accuracy: 0.6901
Epoch 17/25
98/98 [==============================] - 541s 6s/step - loss: 0.1196 -
accuracy: 0.9632 - val_loss: 1.3478 - val_accuracy: 0.6374
Epoch 18/25
98/98 [==============================] - 543s 6s/step - loss: 0.0915 -
accuracy: 0.9755 - val_loss: 1.2961 - val_accuracy: 0.7018
Epoch 19/25
98/98 [==============================] - 544s 6s/step - loss: 0.0687 -
accuracy: 0.9806 - val_loss: 1.2175 - val_accuracy: 0.6842
Epoch 20/25
98/98 [==============================] - 546s 6s/step - loss: 0.0492 -
accuracy: 0.9918 - val_loss: 1.3791 - val_accuracy: 0.6784
Epoch 21/25
98/98 [==============================] - 543s 6s/step - loss: 0.0674 -
accuracy: 0.9847 - val_loss: 1.5585 - val_accuracy: 0.6433
Epoch 22/25
98/98 [==============================] - 537s 5s/step - loss: 0.0740 -
accuracy: 0.9775 - val_loss: 1.7693 - val_accuracy: 0.6550
Epoch 23/25
98/98 [==============================] - 538s 6s/step - loss: 0.0822 -
accuracy: 0.9765 - val_loss: 1.9127 - val_accuracy: 0.6374
Epoch 24/25
98/98 [==============================] - 541s 6s/step - loss: 0.1048 -
accuracy: 0.9653 - val_loss: 1.5448 - val_accuracy: 0.6316
Epoch 25/25
98/98 [==============================] - 544s 6s/step - loss: 0.1373 -
accuracy: 0.9551 - val_loss: 1.4574 - val_accuracy: 0.6842
```

8. Save The Model

```
from tensorflow.keras.models import load_model

model.save('/content/drive/MyDrive/Intelligent Vehicle Damage
Assessment & Cost Estimator For Insurance Companies/Model/body.h5')
```

9. Test The Model

```
from tensorflow.keras.models import load_model
import cv2
from skimage.transform import resize

model = load_model('/content/drive/MyDrive/Intelligent Vehicle Damage
Assessment & Cost Estimator For Insurance Companies/Model/body.h5')

def detect(frame):
  img = cv2.resize(frame,(224,224))
  img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)

  if(np.max(img)>1):
    img = img/255.0
  img = np.array([img])
  prediction = model.predict(img)
  label = ["front","rear","side"]
  preds = label[np.argmax(prediction)]
  return preds

import numpy as np

data = "/content/drive/MyDrive/IBM - PROJECT/Data set/body-
20221023T072112Z-001/body/training/00-front/0008.jpeg"
image = cv2.imread(data)
print(detect(image))

1/1 [==============================] - 0s 498ms/step
front
```

FOR LEVEL DAMAGE

IMAGE PRE PROCESSING

1. Import The ImageDataGenerator Library

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

1. Configure ImageDataGenerator Class

```
train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.1,
                                   zoom_range = 0.1,
                                   horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1./255)
```

1. Apply ImageDataGenerator Functionality To Trainset And Testset

```
training_set =
train_datagen.flow_from_directory('/content/drive/MyDrive/IBM -
PROJECT/Data set/level-20221023T072121Z-001/level/training',
                                                  target_size = (224,
224),
                                                  batch_size = 10,
                                                  class_mode =
'categorical')
test_set =
test_datagen.flow_from_directory('/content/drive/MyDrive/IBM -
PROJECT/Data set/level-20221023T072121Z-001/level/validation',
                                                target_size = (224, 224),
                                                batch_size = 10,
                                                class_mode =
'categorical')
```

```
Found 979 images belonging to 3 classes.
Found 171 images belonging to 3 classes.
```

MODEL BUILDING

1. Importing The Model Building Libraries

```
import tensorflow as tf
from tensorflow.keras.layers import Input, Lambda, Dense, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.applications.vgg19 import VGG19
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import
ImageDataGenerator,load_img
from tensorflow.keras.models import Sequential
import numpy as np
from glob import glob
```

2. Loading The Model

```
IMAGE_SIZE = [224, 224]

train_path = '/content/drive/MyDrive/IBM - PROJECT/Data set/level-
20221023T072121Z-001/level/training'
valid_path = '/content/drive/MyDrive/IBM - PROJECT/Data set/level-
20221023T072121Z-001/level/validation'

vgg16 = VGG16(input_shape=IMAGE_SIZE + [3], weights='imagenet',
include_top=False)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-
applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58889256/58889256 [==============================] - 0s 0us/step
```

3. Adding Flatten Layer

```
for layer in vgg16.layers:
    layer.trainable = False

folders = glob('/content/drive/MyDrive/IBM - PROJECT/Data set/level-
20221023T072121Z-001/level/training/*')

folders
```

```
['/content/drive/MyDrive/IBM - PROJECT/Data set/level-
20221023T072121Z-001/level/training/03-severe',
 '/content/drive/MyDrive/IBM - PROJECT/Data set/level-
20221023T072121Z-001/level/training/02-moderate',
 '/content/drive/MyDrive/IBM - PROJECT/Data set/level-
20221023T072121Z-001/level/training/01-minor']
```

```
x = Flatten()(vgg16.output)

len(folders)
```

```
3
```

4. Adding Output Layer

```
prediction = Dense(len(folders), activation='softmax')(x)
```

5. Creating A Model Object

```
model = Model(inputs=vgg16.input, outputs=prediction)

model.summary()
```

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080
```

```
block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080
block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0
block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160
block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808
block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808
block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0
block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808
block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808
block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808
block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0
flatten (Flatten)           (None, 25088)             0
dense (Dense)               (None, 3)                 75267
=================================================================
Total params: 14,789,955
Trainable params: 75,267
Non-trainable params: 14,714,688
_________________________________________________________________
```

6. Configure The Learning Process

```
model.compile(
  loss='categorical_crossentropy',
  optimizer='adam',
  metrics=['accuracy']
)
```

7. Train The Model

```
r = model.fit_generator(
  training_set,
  validation_data=test_set,
  epochs=25,
  steps_per_epoch=len(training_set),
  validation_steps=len(test_set)
)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6:
UserWarning: `Model.fit_generator` is deprecated and will be removed
in a future version. Please use `Model.fit`, which supports
```

```
generators.


Epoch 1/25
98/98 [==============================] - 606s 6s/step - loss: 1.1697 -
accuracy: 0.5608 - val_loss: 0.9855 - val_accuracy: 0.6140
Epoch 2/25
98/98 [==============================] - 596s 6s/step - loss: 0.7030 -
accuracy: 0.7099 - val_loss: 0.9670 - val_accuracy: 0.6199
Epoch 3/25
98/98 [==============================] - 594s 6s/step - loss: 0.4431 -
accuracy: 0.8202 - val_loss: 1.0758 - val_accuracy: 0.5965
Epoch 4/25
98/98 [==============================] - 592s 6s/step - loss: 0.3887 -
accuracy: 0.8570 - val_loss: 1.0519 - val_accuracy: 0.6257
Epoch 5/25
98/98 [==============================] - 592s 6s/step - loss: 0.3058 -
accuracy: 0.8856 - val_loss: 1.5903 - val_accuracy: 0.6140
Epoch 6/25
98/98 [==============================] - 596s 6s/step - loss: 0.2978 -
accuracy: 0.9019 - val_loss: 1.1763 - val_accuracy: 0.6140
Epoch 7/25
98/98 [==============================] - 598s 6s/step - loss: 0.2060 -
accuracy: 0.9295 - val_loss: 1.2846 - val_accuracy: 0.6082
Epoch 8/25
98/98 [==============================] - 596s 6s/step - loss: 0.1685 -
accuracy: 0.9387 - val_loss: 1.1337 - val_accuracy: 0.6023
Epoch 9/25
98/98 [==============================] - 595s 6s/step - loss: 0.1926 -
accuracy: 0.9305 - val_loss: 1.1559 - val_accuracy: 0.6725
Epoch 10/25
98/98 [==============================] - 594s 6s/step - loss: 0.1206 -
accuracy: 0.9653 - val_loss: 1.2013 - val_accuracy: 0.6433
Epoch 11/25
98/98 [==============================] - 595s 6s/step - loss: 0.1151 -
accuracy: 0.9663 - val_loss: 1.2582 - val_accuracy: 0.6023
Epoch 12/25
98/98 [==============================] - 595s 6s/step - loss: 0.0615 -
accuracy: 0.9857 - val_loss: 1.1696 - val_accuracy: 0.6608
Epoch 13/25
98/98 [==============================] - 597s 6s/step - loss: 0.0659 -
accuracy: 0.9837 - val_loss: 1.1735 - val_accuracy: 0.6374
Epoch 14/25
98/98 [==============================] - 597s 6s/step - loss: 0.0417 -
accuracy: 0.9939 - val_loss: 1.1479 - val_accuracy: 0.6433
Epoch 15/25
98/98 [==============================] - 597s 6s/step - loss: 0.0504 -
accuracy: 0.9898 - val_loss: 1.5237 - val_accuracy: 0.5673
Epoch 16/25
98/98 [==============================] - 596s 6s/step - loss: 0.0437 -
```

```
accuracy: 0.9888 - val_loss: 1.4307 - val_accuracy: 0.6140
Epoch 17/25
98/98 [==============================] - 602s 6s/step - loss: 0.0428 -
accuracy: 0.9877 - val_loss: 1.2403 - val_accuracy: 0.6433
Epoch 18/25
98/98 [==============================] - 605s 6s/step - loss: 0.0359 -
accuracy: 0.9949 - val_loss: 1.3156 - val_accuracy: 0.6433
Epoch 19/25
98/98 [==============================] - 598s 6s/step - loss: 0.0289 -
accuracy: 0.9959 - val_loss: 1.4142 - val_accuracy: 0.6140
Epoch 20/25
98/98 [==============================] - 594s 6s/step - loss: 0.0256 -
accuracy: 0.9980 - val_loss: 1.3567 - val_accuracy: 0.6316
Epoch 21/25
98/98 [==============================] - 598s 6s/step - loss: 0.0248 -
accuracy: 0.9990 - val_loss: 1.3492 - val_accuracy: 0.6257
Epoch 22/25
98/98 [==============================] - 596s 6s/step - loss: 0.0222 -
accuracy: 1.0000 - val_loss: 1.3326 - val_accuracy: 0.6491
Epoch 23/25
98/98 [==============================] - 597s 6s/step - loss: 0.0137 -
accuracy: 0.9990 - val_loss: 1.4157 - val_accuracy: 0.6199
Epoch 24/25
98/98 [==============================] - 595s 6s/step - loss: 0.0398 -
accuracy: 0.9888 - val_loss: 1.4562 - val_accuracy: 0.6257
Epoch 25/25
98/98 [==============================] - 597s 6s/step - loss: 0.0292 -
accuracy: 0.9939 - val_loss: 1.5857 - val_accuracy: 0.5965
```

8. Save The Model

```
from tensorflow.keras.models import load_model

model.save('/content/drive/MyDrive/Intelligent Vehicle Damage
Assessment & Cost Estimator For Insurance Companies/Model/level.h5')
```

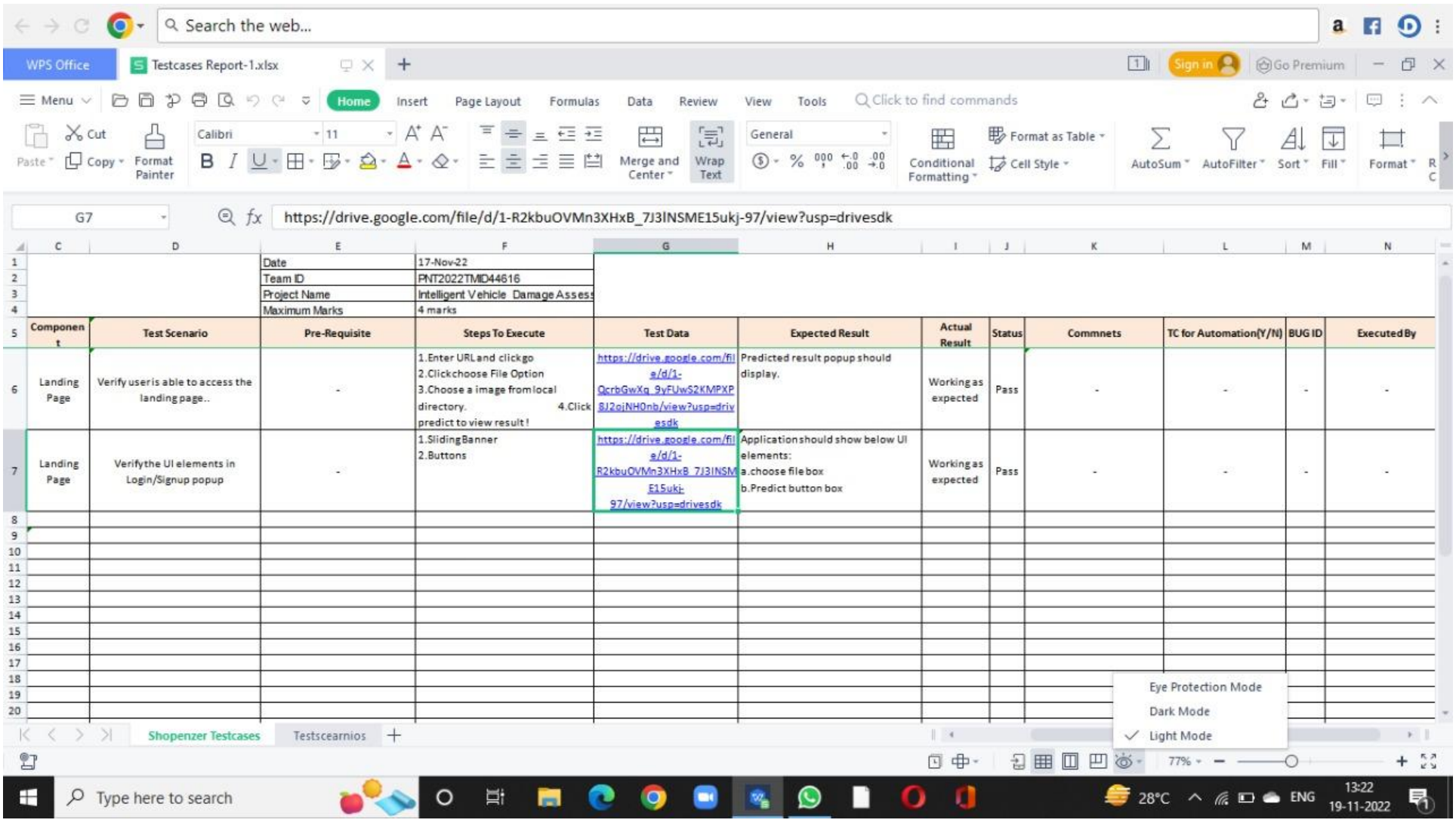
9. Test The Model

```
from tensorflow.keras.models import load_model
import cv2
from skimage.transform import resize


model = load_model('/content/drive/MyDrive/Intelligent Vehicle Damage
Assessment & Cost Estimator For Insurance Companies/Model/level.h5')

def detect(frame):
  img = cv2.resize(frame,(224,224))
  img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)

  if(np.max(img)>1):
    img = img/255.0
```

```
  img = np.array([img])
  prediction = model.predict(img)
  label = ["minor","moderate","severe"]
  preds = label[np.argmax(prediction)]
  return preds

import numpy as np

data = "/content/drive/MyDrive/IBM - PROJECT/Data set/level-
20221023T072121Z-001/level/validation/01-minor/0008.jpeg"
image = cv2.imread(data)
print(detect(image))

1/1 [==============================] - 1s 674ms/step
minor
```

8 . TESTING

8.1 Test Cases



Search the web...

WPS Office | Testcases Report-1.xlsx | Sign in | Go Premium

Menu | Home | Insert | Page Layout | Formulas | Data | Review | View | Tools | Click to find commands

Cut | Copy | Format Painter | Paste | Calibri | 11 | Merge and Center | Wrap Text | General | Format as Table | Conditional Formatting | Cell Style | AutoSum | AutoFilter | Sort | Fill | Format

G7 | fx | https://drive.google.com/file/d/1-R2kbuOVMn3XHxB_7J3lNSME15ukj-97/view?usp=drivesdk

Date	17-Nov-22		
Team ID	PNT2022TMID44616		
Project Name	Intelligent Vehicle Damage Assess		
Maximum Marks	4 marks		

Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
Landing Page	Verify user is able to access the landing page..	-	1.Enter URL and click go 2.Click choose File Option 3.Choose a image from local directory. 4.Click predict to view result !	https://drive.google.com/file/d/1-QcrbGwXq_9yFUwS2KMPXP8J2ojNH0nb/view?usp=drivesdk	Predicted result popup should display.	Working as expected	Pass	-	-	-	-
Landing Page	Verify the UI elements in Login/Signup popup	-	1.Sliding Banner 2.Buttons	https://drive.google.com/file/d/1-R2kbuOVMn3XHxB_7J3lNSME15ukj-97/view?usp=drivesdk	Application should show below UI elements: a.choose file box b.Predict button box	Working as expected	Pass	-	-	-	-

Eye Protection Mode
Dark Mode
Light Mode

Shopenzer Testcases | Testscearnios

77%

Type here to search | 28°C | ENG | 13:22 | 19-11-2022

8.2 User Acceptance Testing

Acceptance Testing
UAT Execution & Report Submission

Date	17 November 2022
Team ID	PNT2022TMID44616
Project Name	Intelligent Vehicle Damage Assessment And Cost Estimator for Insurance Companies

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	4	3	3	0	11
Duplicate	3	0	2	0	5
External	1	4	0	4	9
Fixed	2	1	3	0	6
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	10	8	8	4	31

3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	10	0	0	10
Client Application	20	0	0	20
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	4	0	0	4
Final Report Output	5	0	0	5
Version Control	1	0	0	1

9 . RESULTS

```
from cloudant.client import Cloudant


client = Cloudant.iam('6f4f5183-072e-4bd0-b33f-8c0562a8e227-bluemix',
'IcNwIUOYQsMwH32_e2m3xg93E-Af0KVHeiAwVijUAWWC', connect=True)

my_database=client['my_database']


import numpy as np

import os

from flask import Flask, app,request,render_template,redirect,url_for,session

from tensorflow.keras import models

from tensorflow.keras.models import load_model

from tensorflow.keras.preprocessing import image

from tensorflow.python.ops.gen_array_ops import concat

from tensorflow.keras.applications.inception_v3 import preprocess_input

import requests


os.add_dll_directory


model1=load_model(r'D:\Usman\IBM Project\Model\body.h5')

model2=load_model(r'D:\Usman\IBM Project\Model\level.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)
        return render_template("register.html",pred="Registration Successful, please login with  your
details")
    else:
        return render_template("register.html",pred="You are already a member, please login using
your registered 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)
```

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

10 . ADVANTAGES & DISADVANTAGE

- DEATH BENEFIT IN LIFE INSURANCE POLICY

  In case any unexpected thing happened to the insured, which results in the loss of income for their family, the insurance company provides compensation in the form of the death benefit. The appointed nominee receives the full sum assured plus the bonus accrued over a period.

- VALUABLE RETURN ON YOUR INVESTMENT

  Several financial advisors in India suggest that everyone must invest in a life insurance policy not only to provide your family with the financial protection when you are not around but also from the perspective of gaining valuable returns from the investment. Many life insurance schemes in India offer a decent recent in the form of bonus that no other investment tools offe*r*

- AVAILABILITY OF LOAN TO THE LIFE INSURANCE POLICYHOLDERS

  In the event of any emergency where you need money desperately, you can take advantage of your life insurance policy and take a loan against it. Today, almost all the major insurance companies in India provide loan facilities to the policyholders. You can borrow a certain percentage of the cash value of the policy, or the sum assured depending on the policy provisions. Make sure that you check with the loan policies with the insurer before you subscribe for the policy.

- GUARANTEED INCOME IN LIFE INSURANCE

Your family feel secured because you bring in regular income to cater to their needs. The income you earn aids in paying the loan (if any), rent, daily bills, child education and other household expenses. Certain life insurance policies provide regular pay-outs, which can compensate for the loss of income due to the death of the family's earning member.

Life Insurance Can be expensive for old-aged people

Buying a life insurance policy would seem to be the most logical thing to do when you are young and why not? The premium for young buyers is quite affordable. The premium amount for a life insurance policy is determined by your personal medical condition, family's medical condition and your age.

But if you are over 40 or if you are nursing an illness or if you have a history of bad medical condition in the family, the insurance company will consider you as a risky buyer and so to mitigate the risk they charge a higher premium. So, if you are old or carrying a chronic ailment, a life insurance policy can be helpful for your loved one, but it would an added burden on your expenses.

- ### The returns on life insurance are not significant

  Certain life insurance policies like a whole life insurance policy provide the dual benefit of investment-cum-protection. The cash-value component of the whole life policy is a great way to save money for your future needs like retirement and providing coverage for the family in the event of your demise.

  However, you must know that the returns offered on the investment are much lower than other investment tools. You can invest your hard-earned money in a term insurance plan and invest the additional cash in other investment tools and increase your chances of earning higher returns.

- ### Insurers may not pay the benefit

  There have been many instances wherein the insurance companies have denied paying the sum assured or the death benefit to the policyholder or the nominee. A lot of times, the insurance company uses various tricks to evade paying the benefits even after the maturity of the policy. They would cite many hidden charges or clauses to reduce the pay-out. It is, therefore, important to carefully understand the finer details of the policy and choose a company that has a positive pay-out rate. Further, it would be best advised to consult your financial advisor about the pros and cons of the policy before entering a contract.

- ### Complex Insurance policies

  In India, many insurance policy providers offer a wide range of life insurance policies to suit the different needs of the customers. While the vast choices give you the liberty to choose the best, it can also create confusion in the minds of the policy buyers, especially the ones who have no prior experience of buying an insurance policy.

11 . CONCLUSION

In the future, we will continue to explore the innovation of insurance technology of 'AI + Vehicle Insurance'. We hope that we can use the power of intelligent damage determination system. On the one hand, the owner can take photos by one click to achieve rapid loss determination, price estimation and immediate compensation. On the other hand, it assists insurance companies to achieve rapid and accurate pricing in the process of fixing losses and claims. Finally, by combining the rapid compensation of accident vehicles to relieve traffic pressure, to avoid more serious personal and property losses caused by secondary accidents.

12 . FUTURE SCOPE

AI’s underlying technologies are already being deployed in our businesses, homes, and vehicles, as well as on our person. The disruption from COVID-19 changed the timelines for the adoption of AI by significantly accelerating digitization for insurers. Virtually overnight, organizations had to adjust to accommodate remote workforces, expand their digital capabilities to support distribution, and upgrade their online channels. While most organizations likely didn't invest heavily in AI during the pandemic, the increased emphasis on digital technologies and a greater willingness to embrace change will put them in a better position to incorporate AI into their operations.

Four core technology trends, tightly coupled with (and sometimes enabled by) AI, will reshape the insurance industry over the next decade.

Explosion of data from connected devices

In industrial settings, equipment with sensors have been omnipresent for some time, but the coming years will see a huge increase in the number of connected consumer devices. The penetration of existing devices (such as cars, fitness trackers, home assistants, smartphones, and smart watches) will continue to increase rapidly, joined by new, growing categories such as clothing, eyewear, home appliances, medical devices, and shoes. Experts estimate there will be up to one trillion connected devices by 2025.2 The resulting avalanche of new data created by these

devices will allow carriers to understand their clients more deeply, resulting in new product categories, more personalized pricing, and increasingly real-time service delivery.

Experts estimate there will be up to one trillion connected devices by 2025.

Increased prevalence of physical robotics

The field of robotics has seen many exciting achievements recently, and this innovation will continue to change how humans interact with the world around them. Additive manufacturing, also known as 3-D printing, will radically reshape manufacturing and the commercial insurance products of the future. By 2025, 3-D-printed buildings will be common, and carriers will need to assess how this development changes risk assessments. In addition, programmable, autonomous drones; autonomous farming equipment; and enhanced surgical robots will all be commercially viable in the next decade. By 2030, a much larger proportion of standard vehicles will have autonomous features, such as self-driving capabilities. Carriers will need to understand how the increasing presence of robotics in everyday life and across industries will shift risk pools, change customer expectations, and enable new products and channels.

Open-source and data ecosystems

As data becomes ubiquitous, open-source protocols will emerge to ensure data can be shared and used across industries. Various public and private entities will come together to create ecosystems in order to share data for multiple use cases under a common regulatory and cybersecurity framework. For example, wearable data could be ported directly to insurance carriers, and connected-home and auto data could be made available through Amazon, Apple, Google, and a variety of consumer device manufacturers.

Advances in cognitive technologies

Convolutional neural networks and other deep learning technologies currently used primarily for image, voice, and unstructured text processing will evolve to be applied in a wide variety of applications. These cognitive technologies, which are loosely based on the human brain's ability to learn through decomposition and inference, will become the standard approach for processing the incredibly large and complex data streams that will be generated by "active" insurance products tied to an individual's behavior and activities. With the increased commercialization of these types of technologies, carriers will have access to models that are constantly learning and adapting to the world around them—enabling new product categories and engagement techniques while responding to shifts in underlying risks or behaviors in real time.

13 . APPENDIX

Source Code

```
print(docs)


print(len(docs.all()))


if (len(docs.all())==0):
    return render_template("login.html",pred="The username or password is incorrect. Please login
with correct details.")
else:
    if((user==docs[0][0]['_id']and passw==docs[0][0]['psw'])):
        return redirect(url_for('prediction'))
    else:
        return render_template("login.html",pred="The username is not found or the details you've
entered is incorrect.")


@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',methods=["GET","POST"])
def result():
  if request.method=="POST":
    f=request.files['file']
    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]
print(result1)
print(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"
return render_template("result.html", prediction="The Estimated cost for the damage is:
"+value)
```

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

GitHub Link

IBM-Project-45518-1660730670

Project Demo Link

https://drive.google.com/drive/folders/1mBJaYUMoSSN-MttsagJ02JdFuoxd4hhV