Project On

Intelligent Vechile Damage Assessment & Cost Estimator for Insurance Companies

powered By IBM India

Submitted By
Vimal W
Sachin A
Barath S
Suriya Prakash R

Project ID: PNT2022TMIDO7174

of
Departnment of Electronics and Communication
Jerusalem College of Engineering
Velachery main road, Narayanapuram, Chennai - 600100

College Mentor: **Dr. Gnanasivam (Dean - Industrial Relations)**
Industrial Mentor: **Swathi**

1. INTRODUCTION

1.1. Project Overview

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

1.2. Purpose

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

2. LITERATURE SURVEY

2.1. Existing Problem

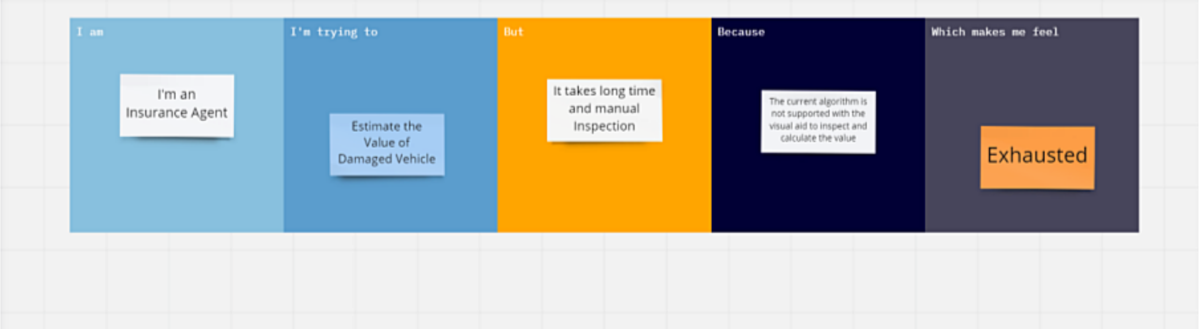
The problem is defined as the optimzed way to estimate insurance cost based on the mulitple damages in the various areas in the vechile for the insurance companies. As the existing methods for estimating the cost takes lot of time and energy in the way of inspecting the vechile.

2.2. References

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.

S.No	Paper Name	Year	Author Name
1.	Research on Intelligent Vehicle Damage Assessment System Based on Computer Vision	2020	Zhu Qianqian, Guo Weiming, Shen Ying and Zhao Zihao
2.	Car Damage Assessment Based on VGG Models	2021	Phyu Mar Kyu, Kuntpong Woraratpanya
3.	Assessing Car Damage with Convolutional Neural Networks	2021	Harit Bandi

2.3. Problem Statement Defination



I am

I'm an Insurance Agent

I'm trying to

Estimate the Value of Damaged Vehicle

But

It takes long time and manual Inspection

Because

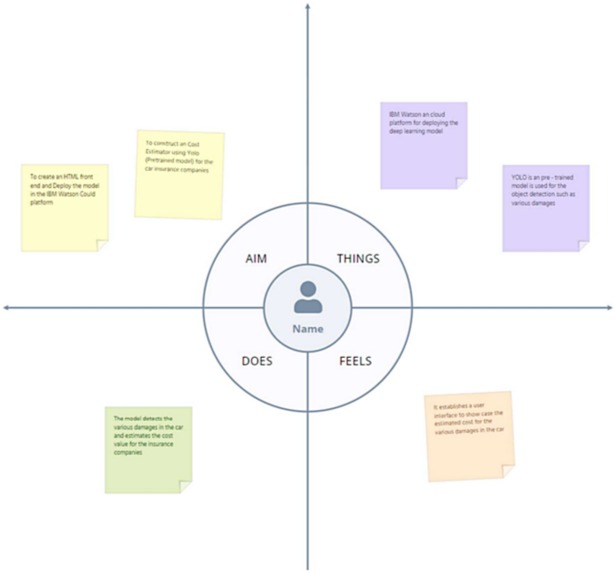
The current algorithm is not supported with the visual aid to inspect and calculate the value

Which makes me feel

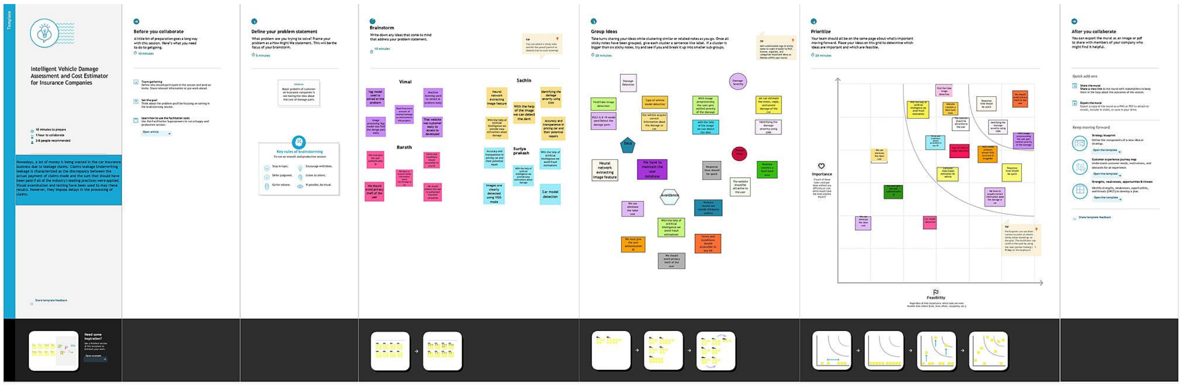
Exhausted

3. IDEATION & PROPOSED SOLUTION

3.1. Empathy Map


To create an HTML front end and Deploy the model in the IBM Watson Could platform
To construct an Cost Estimator using Yolo (Pretrained model) for the car insurance companies
IBM Watson an cloud platform for deploying the deep learning model
YOLO is an pre - trained model is used for the object detection such as various damages
AIM
THINGS
Name
DOES
FEELS
The model detects the various damages in the car and estimates the cost value for the insurance companies
It establishes a user interface to show case the estimated cost for the various damages in the car

3.2. Ideation and Brainstorming


Template
Vimal
Barath
Sachin
Surya prakash
Vgg model used to solved at the problem
Machine learning used to solved at problem easily
Image processing Vgg model was find the dange part easily
That website was customer easily to access to developed
We should avoid privacy theft of the user
Neural network extracting image feature
Identifying the damage severity using CNN
With the help of the image we can detect the dent
Accuracy and transparence in pricing car and their potential repairs
Images are clearly detected using VGG mode
Car model detection
Damage Detection
Damage Severity
Find Fake image detection
Type of vehicle model detection
VGG16 & 19 model used Detect the damage parts
the vehicle acquire correct information about the damage or car
With image preprocessing the user gets notified severity of the damage
with the help of the image we can detect the dent
we can estimate the minor, major, and severe damage of the car
Identifying the damage severity using CNN
Data
We have to maintain the user database
Response time should be quick
Website consists a feed back note
The website should be attractive to the user
Avoidance
We can eliminate the labor cost
With the help of Artificial Intelligence we avoid fraud estimations
Website should not contain thirdparty cookies
We must give the user authentication ID
Terms and Conditions should accessible to any OS
We should avoid privacy theft of the user
Importance
Feasibility

Intelligent Vehicle Damage Assessment and Cost Estimator for Insurance Companies

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

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.

Share template feedback

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM
Major problem of customer on insurance companies is not having the idea about the cost of damage parts.

Key rules of brainstorming
To run an smooth and productive session

Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

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

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

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

Share the mural
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

Export the mural
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

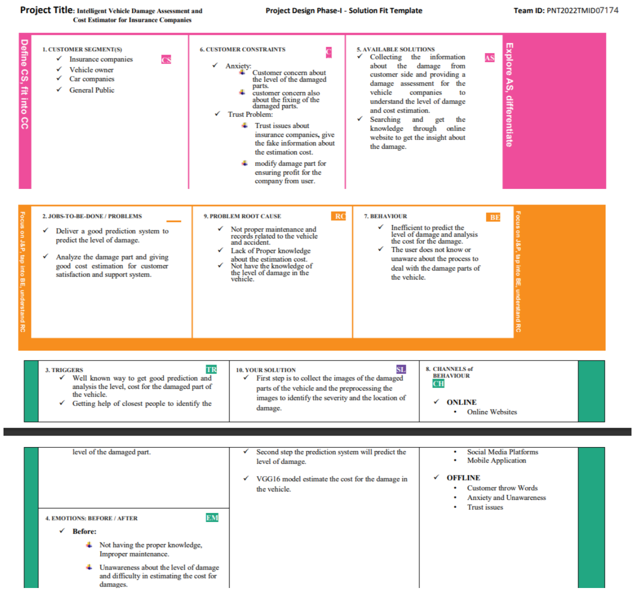
Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

3.3. Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Insurance firms frequently suffer losses because they are unable to accurately estimate the cost of damaged automobiles and they are unable to calculate the cost of damaged cars precisely, insurance companies regularly incur losses.
2.	Idea / Solution description	We are creating an AI model to sense and detect the precise amount of automobiles damage.
3.	Novelty / Uniqueness	Automated calculator for the cost of filing an insurance claim.
4.	Social Impact / Customer Satisfaction	Determining the extent of vehicle damage and offering insurance accordingly.
5.	Business Model (Revenue Model)	Underwriting and investment income are the main sources of income for insurers. Financial investments, including listed shares, government bonds, commercial real estate, and corporate bonds, make up the majority of insurance firms' assets. By estimating the level of car damage using our AI model and providing insurance in accordingly, they are able to save more money and invest it in their own businesses.
6.	Scalability of the Solution	Our artificial intelligence (AI) has the capacity to operate at the scale, speed, and complexity required for the aim. Our model's accuracy will improve with more testing and training using real-time data.

3.4. Proposed Solution Fit



Project Title: Intelligent Vehicle Damage Assessment and Cost Estimator for Insurance Companies

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID07174

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

- ✓ Insurance companies
- ✓ Vehicle owner
- ✓ Car companies
- ✓ General Public

6. CUSTOMER CONSTRAINTS C

- ✓ Anxiety:
  - Customer concern about the level of the damaged parts.
  - customer concern also about the fixing of the damaged parts.
- ✓ Trust Problem:
  - Trust issues about insurance companies, give the fake information about the estimation cost.
  - modify damage part for ensuring profit for the company from user.

5. AVAILABLE SOLUTIONS AS

- ✓ Collecting the information about the damage from customer side and providing a damage assessment for the vehicle companies to understand the level of damage and cost estimation.
- ✓ Searching and get the knowledge through online website to get the insight about the damage.

Explore AS, differentiate

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

2. JOBS-TO-BE-DONE / PROBLEMS

- ✓ Deliver a good prediction system to predict the level of damage.
- ✓ Analyze the damage part and giving good cost estimation for customer satisfaction and support system.

9. PROBLEM ROOT CAUSE RC

- ✓ Not proper maintenance and records related to the vehicle and accident.
- ✓ Lack of Proper knowledge about the estimation cost.
- ✓ Not have the knowledge of the level of damage in the vehicle.

7. BEHAVIOUR BE

- ✓ Inefficient to predict the level of damage and analysis the cost for the damage.
- ✓ The user does not know or unaware about the process to deal with the damage parts of the vehicle.

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

3. TRIGGERS TR

- ✓ Well known way to get good prediction and analysis the level, cost for the damaged part of the vehicle.
- ✓ Getting help of closest people to identify the level of the damaged part.

4. EMOTIONS: BEFORE / AFTER EM

- ✓ **Before:**
  - Not having the proper knowledge, Improper maintenance.
  - Unawareness about the level of damage and difficulty in estimating the cost for damages.

10. YOUR SOLUTION SL

- ✓ First step is to collect the images of the damaged parts of the vehicle and the preprocessing the images to identify the severity and the location of damage.
- ✓ Second step the prediction system will predict the level of damage.
- ✓ VGG16 model estimate the cost for the damage in the vehicle.

8. CHANNELS of BEHAVIOUR CH

- ✓ **ONLINE**
  - Online Websites
  - Social Media Platforms
  - Mobile Application
- ✓ **OFFLINE**
  - Customer throw Words
  - Anxiety and Unawareness
  - Trust issues

4. REQUIREMENT ANALYSIS

4.1. Functional Requirement

The functional Requirements of this projects invloves the better understanding of Deep Learning, Image Pre-processing, Application designing using HTML & CSS and IBM Waston Cloud. TensorFlow provides the deep learning architecture for learning the feature maps, Image generator module is used in generating the mulitple images based on the shape and agumentation mentioned in the passing parameters. IBM Watson provides the services such as Database, deployment etc.

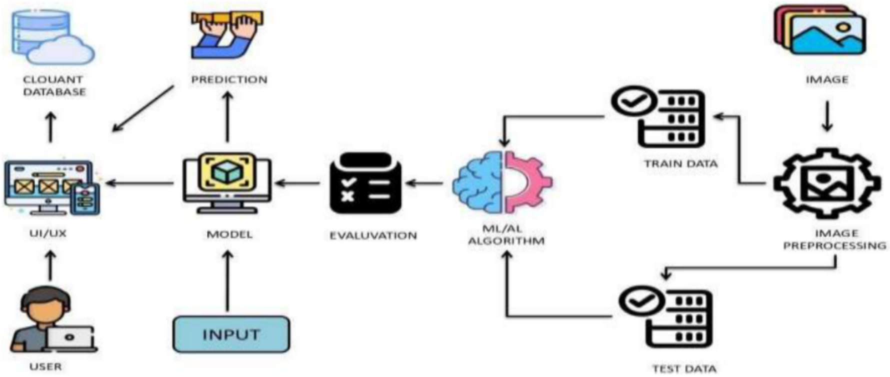
4.2. Non - Functional Requirements

The Non - Functional Requirements of this project are,

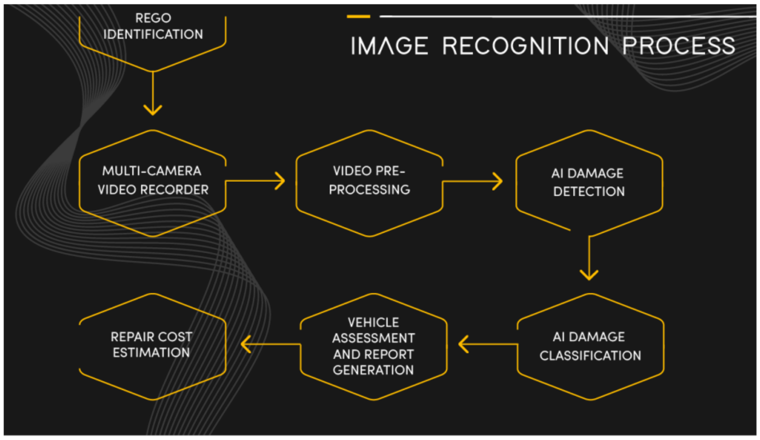
1. Highly accurate Image Predictive model
2. better user responsive web application
3. Cloud database for storing the informations

5. PROJECT DESIGN

5.1. DataFlow Diagram



5.2. Solution and Technical Architecture



5.3. User Stories

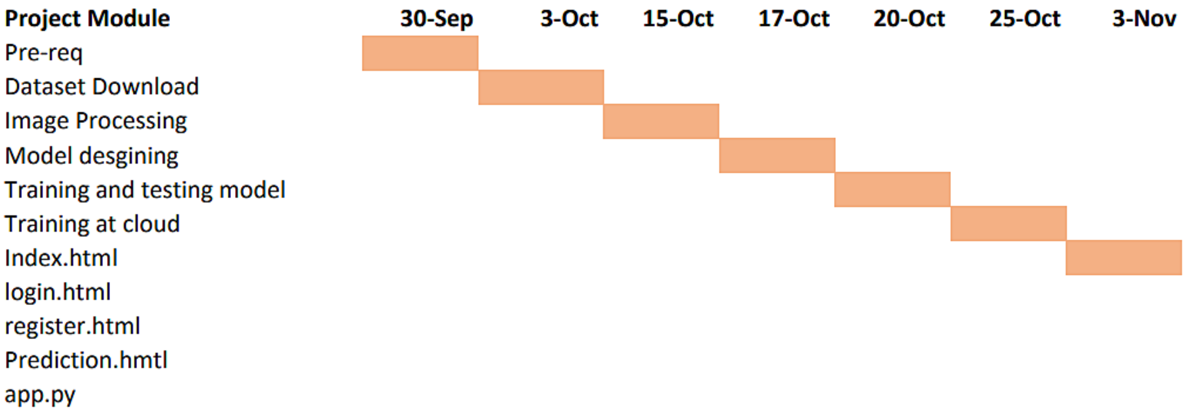
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard by entering valid credentials	High	Sprint-1
Customer Details	Login	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
Customer Uses	Dashboard	USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-4
Customer Options	Details about insurance companies	USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Facebook Gmail	Medium	Sprint-1
Customer usage	Login	USN-5	As a user, I can log into the application by entering email & password	I can log in and view my dashboard at my demand on any time	High	Sprint-1
Customer needs to do	Dashboard	USN-6	As a user I must capture images of my vehicle and upload it into the web portal	I can capture the entire vehicle and upload	High	Sprint-2
Customer (Web user)	Details about estimated cost based on damage	USN-7	As a user I must receive a detailed report of the damages present in the vehicle and the cost estimated	I can get the estimated insurance cost	High	Sprint-3
Customer Care Executive	Details about Estimated cost Based on damage	USN-8	As a user, I need to get support from developers in case of queries and failure of service provided	I can have smooth user experiences and all the issues raised is sorted	Medium	Sprint-4
Administrator	Details about Estimated cost Based on damage	USN-9	We need to satisfy the customer needs in an efficient way and make sure any sort of errors are fixed	I can finish the work without any problems	High	Sprint-4

6. PROJECT PLANNING

6.1. Sprint Planning and Estimation

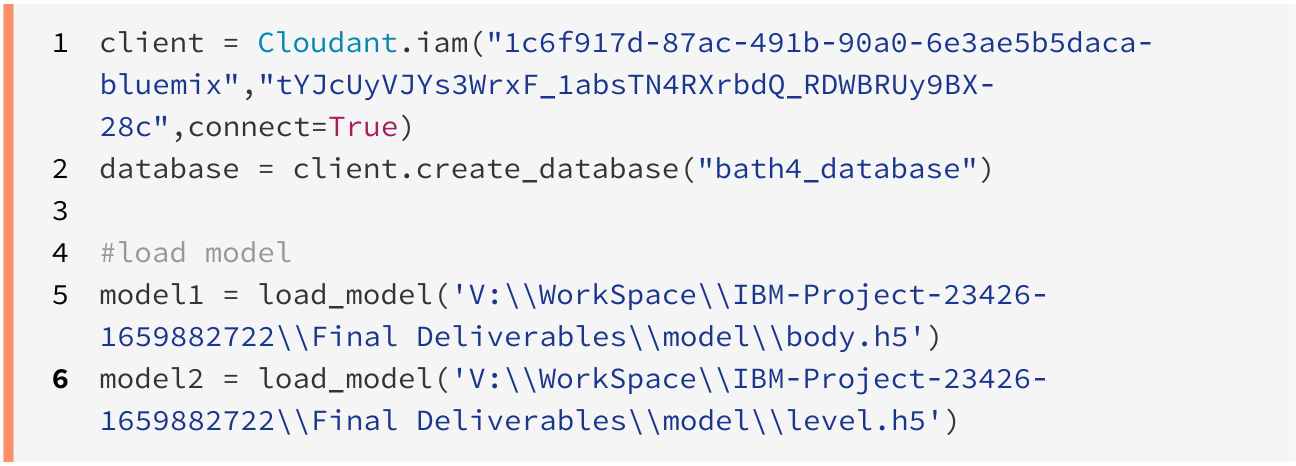
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Prediction model	USN-1	How can I estimate the cost for vehicle assessment	2	High	Vimal
Sprint-1	Creating the Database in the cloud	USN-2	How can I store the information	1	High	Vimal
Sprint-2	Index.html	USN-3	As user, how can I access the prediction model in the fancy way	2	Low	Sachin
Sprint-3	Login.html	USN-4	As a user, I can log into the application by entering email & password	2	Medium	Suriya
Sprint-3	Register.html	USN-5	As a user, I can register for the application through Gmail	1	Medium	Suriya
Sprint-4	Prediction.html	USN-6	As the user, I can upload the image and make the estimation	1	High	Barath
Sprint-5	App.py	USN-7	How to connect the front end with the back end	1	High	Vimal

6.2. Sprint Delivery Schedule



7. CODING & SOLUTIONING

7.1. Feature 1

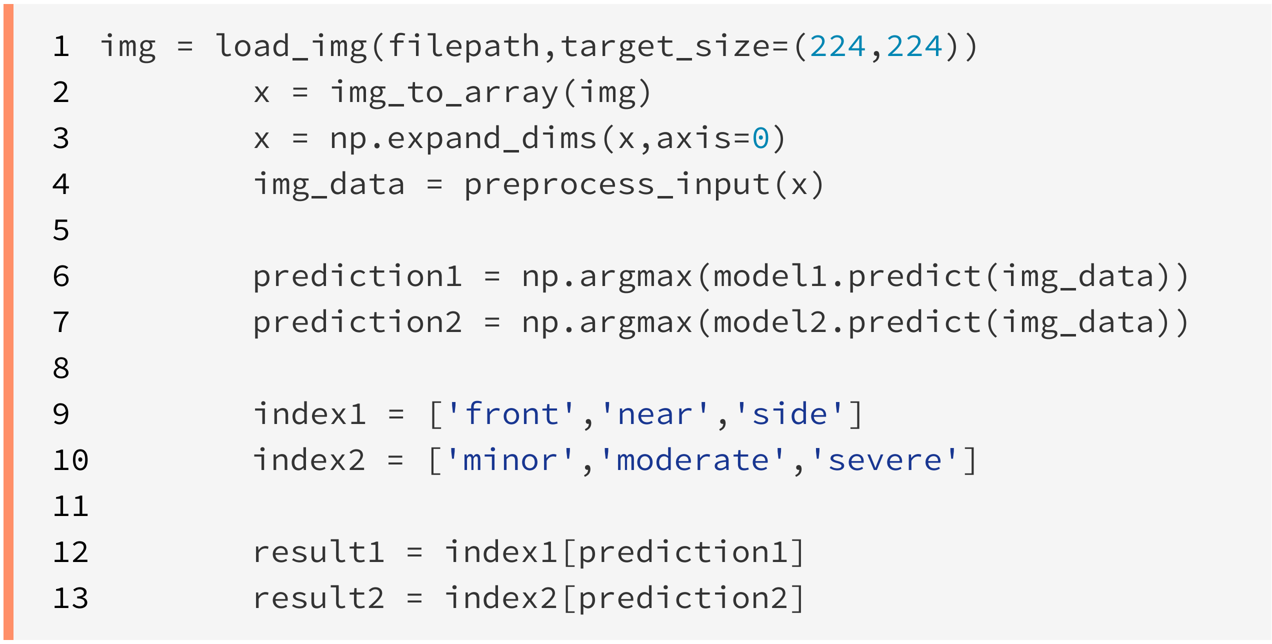


```
client = Cloudant.iam("1c6f917d-87ac-491b-90a0-6e3ae5b5daca-bluemix","tYJcUyVJYs3WrxF_1absTN4RXrbdQ_RDWBRUy9BX-28c",connect=True)
database = client.create_database("bath4_database")

#load model
model1 = load_model('V:\\WorkSpace\\IBM-Project-23426-1659882722\\Final Deliverables\\model\\body.h5')
model2 = load_model('V:\\WorkSpace\\IBM-Project-23426-1659882722\\Final Deliverables\\model\\level.h5')
```

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

7.2. Feature 2



```
img = load_img(filepath,target_size=(224,224))
        x = 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','near','side']
        index2 = ['minor','moderate','severe']

        result1 = index1[prediction1]
        result2 = index2[prediction2]
```

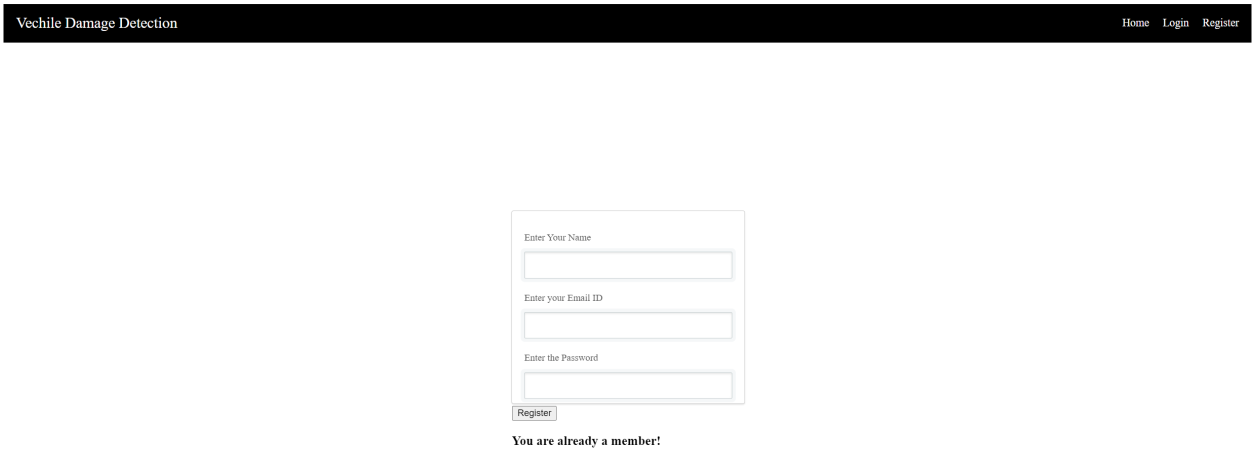
feature 2 enables the web application to predict the incomming image from the user into the given labels. The code gets the image, convert into pixcels and load into the model. Based on the predicted results, the algorithm will returns the value as the estimated cost.

8. TESTING

8.1. Test Cases

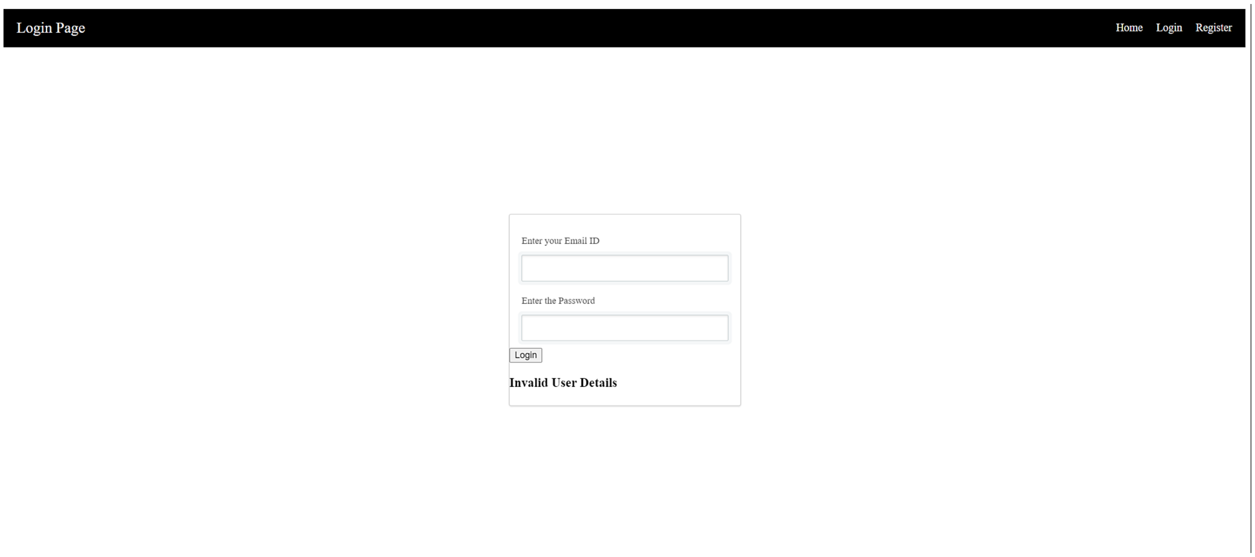
1. User Login and Registration test
2. Database Update test
3. Prediction test

8.2. User Acceptance Testing



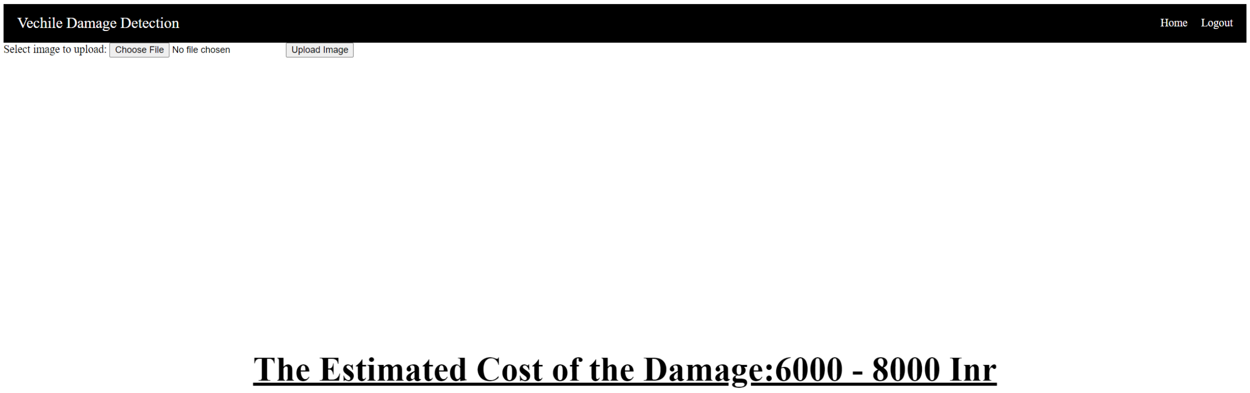
You are already a member!

The registeration web page is tested with the already registered user information and hence it shows a message "You are already a member" by which the repeation of user infromation at database is prevented.



Invalid User Details

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



The Estimated Cost of the Damage:6000 - 8000 Inr

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

9. PERFORMANCE

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

10. ADVANTAGES AND DISADVANTAGES

1. The Advantage of having an Intelligent Cost Estimator based on the damages can save the time and resource of the user in automatically evaluating the images with the damages using the Deep Learning models trained with the various car images.
2. The Disadvantage of the project is expensive coding and time to develop the front end and back end of the web application

11. CONCLUSION

We conclude by suggesting this web application for damage assessment and cost estimation for the insurance companies. The web application is supported by the Deep Learning and IBM waston cloud which stands for the complex image prediction and user information storage. The web application takes the user registration and login, The user can login into the prediction page using their ID and password. The prediction takes the image input and the model can predict the input based on the perviour knowledge about the damages.

12. FUTURE SCOPE

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

13. APPENDIX

Github Repo:
https://github.com/IBM-EPBL/IBM-Project-23426-1659882722

VideoLink:
https://drive.google.com/file/d/1wZ_BgP7G8zyOnZ5q5xBCwM7FwBfBapm1/view

App.py

```
from cloudant.client import Cloudant
import os
import tensorflow
from keras.utils import load_img, img_to_array
from werkzeug.utils import secure_filename
import numpy as np
from keras.models import load_model
from tensorflow.python.ops.gen_array_ops import concat
from keras.applications.inception_v3 import preprocess_input

#creating the Cloudant Database
client = Cloudant.iam("1c6f917d-87ac-491b-90a0-6e3ae5b5daca-bluemix","tYJcUyVJYs3WrxF_1absTN4RXrbdQ_RDWBRUy9BX-28c",connect=True)
database = client.create_database("bath4_database")

#load model
model1 = load_model('V:\\WorkSpace\\IBM-Project-23426-
1659882722\\Final Deliverables\\model\\body.h5')
model2 = load_model('V:\\WorkSpace\\IBM-Project-23426-1659882722\\Final Deliverables\\model\\level.h5')

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

app = Flask(__name__)

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

#login page setting

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

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

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

    docs = database.get_query_result(query)
    print(docs)
    print(len(docs.all()))

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


#Register page setting

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

@app.route('/afterRegister',methods=['POST'])
def afterregister():
    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 = database.get_query_result(query)

    if(len(docs.all())==0):
        url = database.create_document(data)
        return render_template('register.html', message="Registration is Successfully Completed")
    else:
        return render_template("register.html", message="You are already a member!")

#prediction

```

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

#logout page

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

#results

@app.route('/result', methods = ['GET', 'POST'])
def upload_file():
    if request.method == 'POST':
        f = request.files['_file']
        basepath = os.path.dirname(__name__)
        filepath = os.path.join(basepath,'uploads',f.filename)
        f.save(filepath)

        img = load_img(filepath,target_size=(224,224))
        x = 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','near','side']
        index2 = ['minor','moderate','severe']

        result1 = index1[prediction1]
        result2 = index2[prediction2]

        if(result1=="front" and result2=="minor"):
            value= "3000 - 5000 Inr"
        elif(result1=="front" and result2=="moderate"):
            value ="6000 - 8000 Inr"
        elif(result1=="front" and result2=="severe"):
            value="9000 - 11000 Inr"
        elif(result1=="near" and result2=="minor"):
            value="4000 to 6000 Inr"
        elif(result1=="near" and result2=="moderate"):
            value="7000 - 9000 Inr"
        elif(result1=="near" 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 - 11000Inr"
        elif(result1=="side" and result2=="severe"):
            value="12000 - 15000 Inr"
        else:
            value = "16000 - 50000 Inr"

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



if (__name__ == '__main__'):
    app.run(debug=True)
```