Project Report Format

1. INTRODUCTION

1.1 PROJECT OVERVIEW：

The Vehicles counts are incresed rapidly now a days. Because most of the people were use vehicles for their personal or business purpose. Accidents may happen because of driving the vehicles at high speeds or does not follow the traffic rules. When the accident occurs, people may file a report for claim the insurance to repair thier vehicles. Most of the insurance companies doesn't provide correct insurance amount or the claim procedure takes more time. Instead of taking that much of time, a system may perform it in a minute by giving a damaged vehicle image.The system may analysis the image to detect the damagesa in the vehicle.

1.2 PURPOSE：

Most of the people may use vehicles for their day to day life. The accident may happen when the people are driving the vehicle at high speed. When an accident occurs, people may claim insurance easily by using this system. The people only upload their damaged vehicle image and the system may analysis the image then provide a correct insurance amout for the vehicle.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM：

Insurance fraud is one of the most prevalent kinds of fraud. In particular, the cost of automotive insurance fraud is significant for property insurance companies and has a longterm influence on insurance businesses' pricing tactics. And Car insurance fraud detection has become required in order to reduce insurance prices. Although predictive models for detecting insurance fraud are widely used in practise, there are few published research on the use of machine learning algorithms to identify insurance fraud, most likely due to a lack of available data. Evaluate 13 machine learning approaches in this paper using real-world data. Predicting insurance fraud has become a big difficulty due to the uneven datasets in this domain. Because our data consists primarily of "non-fraud claims" with a minor number of "fraud claims." As a result, classification models predict fraud poorly; thus, the current study seeks to propose an approach that improves machine learning algorithms' results by using resampling techniques, such as Random over Sampler, Random under Sampler, and hybrid methods, to address the issue of unbalanced data.

2.2 REFERENCES :

Vaibhav Agarwal, Utsav Khandelwal, Shivam Kumar, Raja Kumar, Shilpa 2022 IJCRT | Volume 10, Issue 4 April 2022 | ISSN: 2320-2882

R.E. van Ruitenbeek, S.Bhulai Machine Learning with ApplicationsVolume 9, 15 September 2022, 100332

Ruixing Ming, Using Machine Learning Models To Compare Various Resampling Methods In Predicting Insurance Fraud, 2021

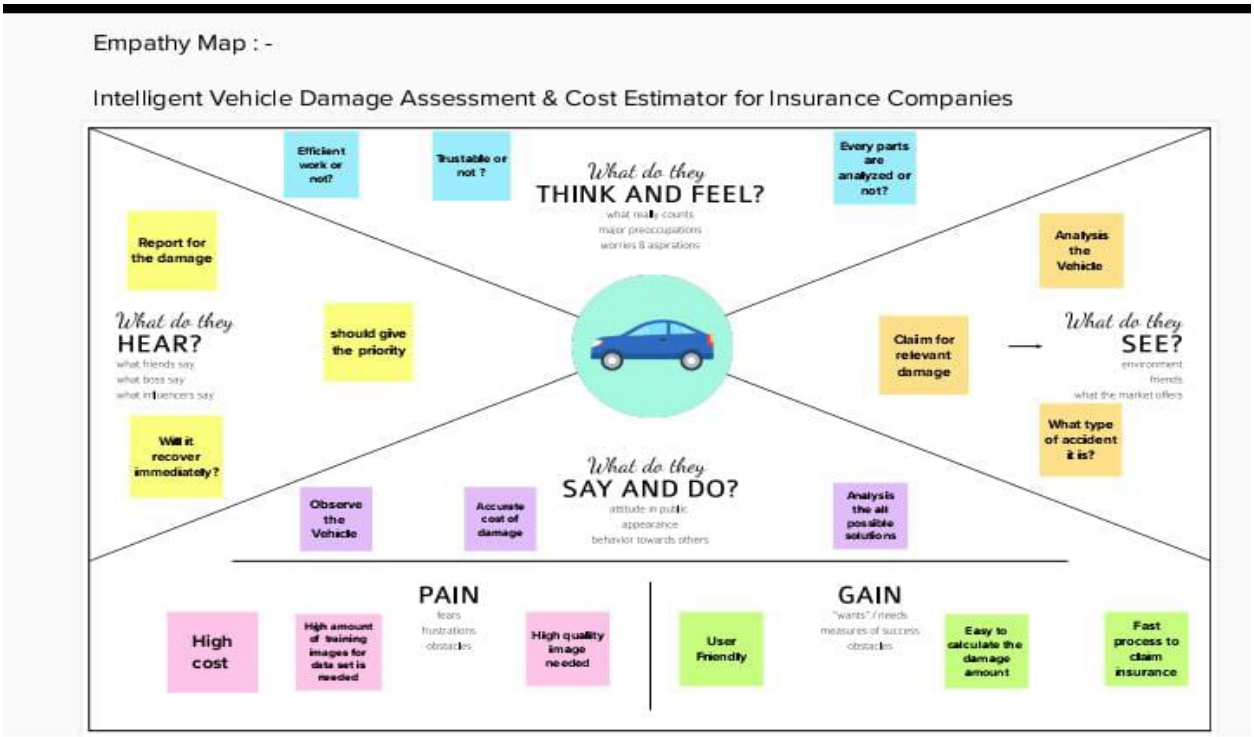
Siddhant Gole, Car Damage Assessment to Automate Insurance Claim, 2022

2.3 PROBLEM STATEMENT DEFINITION :

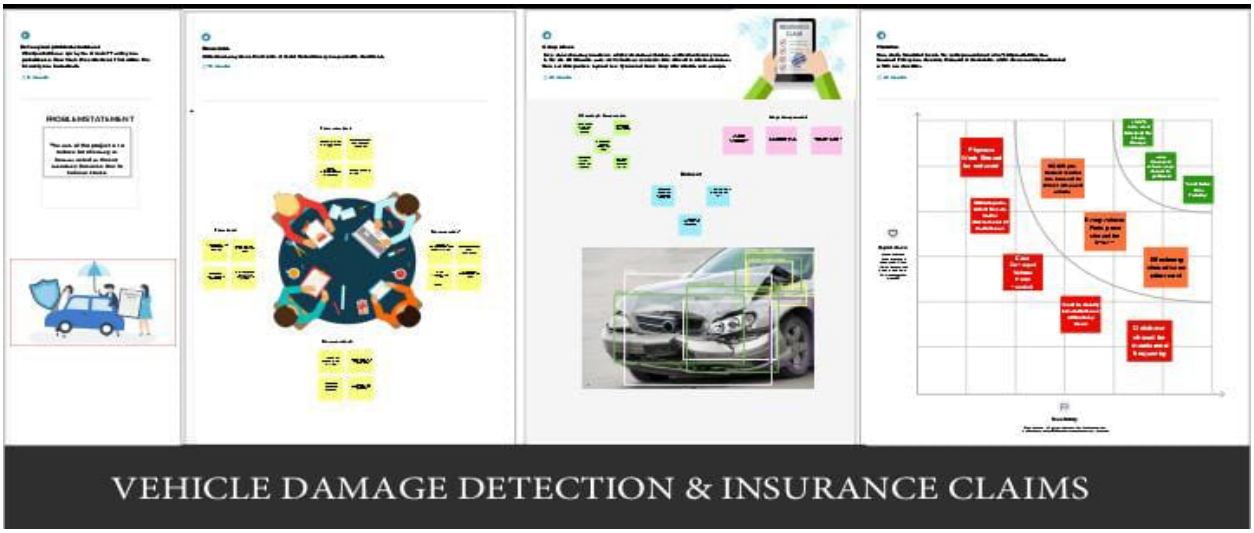
Nowadays, a lot of money is being wasted in the car insurance business due to leakage claims. Claims leakage Underwriting leakage is characterized as the discrepancy between the actual payment of claims made and the sum that should have been paid if all of the industry's leading practices were applied. Visual examination and testing have been used to may these results. However, they impose delays in the processing of claims. So we decide to overcome these issue without any delay and efficiently this is our proposed system.The discrepancy between the company's actual spending and what they should have really spent is known as claim leakage. Ineffective claim processing, erroneous payments, human error such as a lack of quality control or poor customer service or even claim fraud may be to blame for this. Auditing closed claim files is the only way to find claim leakage.

3. IDEATION & PROPOSED SOLUTION

3.1EMPATHYMAPCANVAS :



3.2IDEATION&BRAINSTORMING

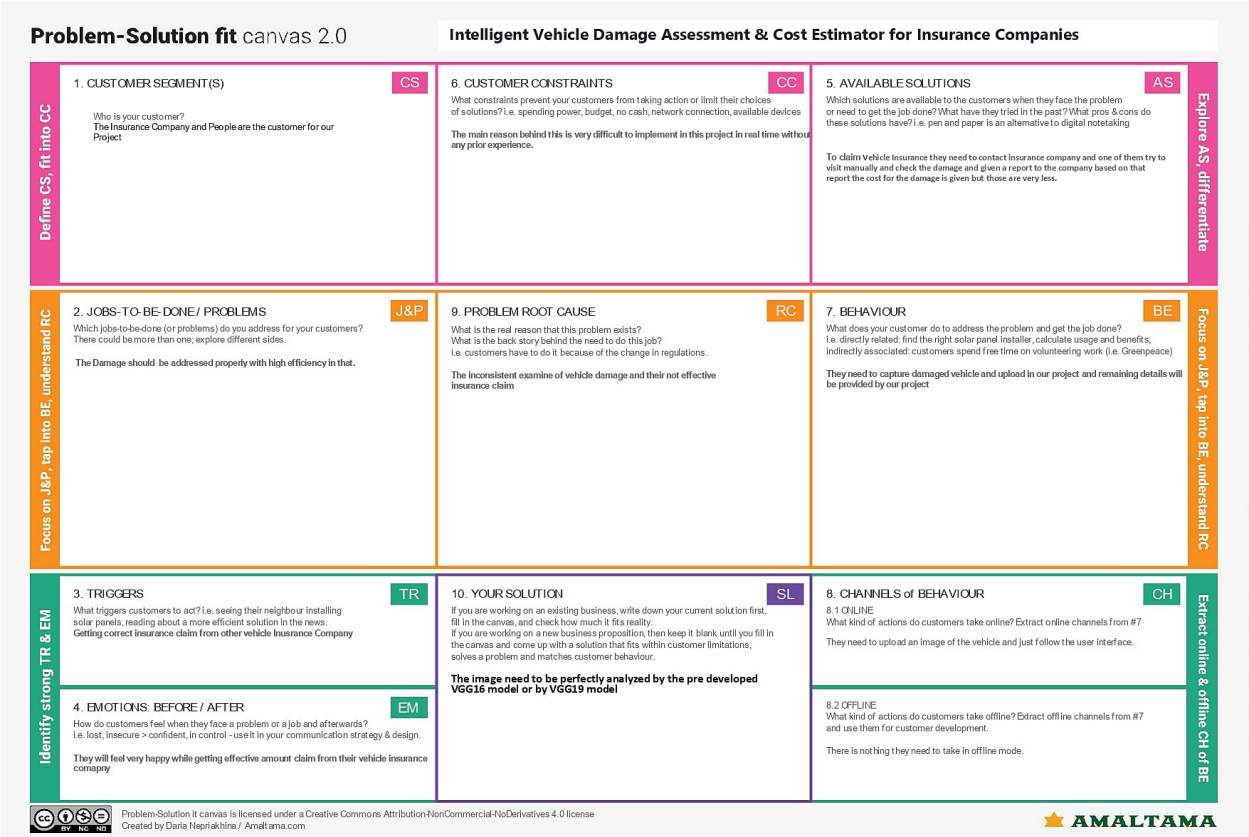


3.3 PROPOSED SOLUTION

Date	30 September 2022
Team ID	PNT2022TMID32157
Project Name	Intelligent Vehicle Damage Assessment & Cost Estimator for Insurance Companies
Maximum Marks	2 Marks

S.No.	Parameter	Description
1	Problem Statement (problem to be solved)	To develop a VGG16 Model, that is used to detect the damaged area in the car. This is used in a insurance companies to easy and faster way to claim the insurance. The amount will be detected by user uploading a damaged image of the car and model.
2	Idea / Solution Description	To accomplish this, firstly create Train and Test Folders. Secondly, image processing in which Import the image data generator library and apply image data generator functionality to Trainset and test set. The third step is Model Building in which Import the model building Libraries, Adding Flatten layers then Adjoin Output Layer further Creating Model Object and Configure the Learning Process & next Train, Save, Test The Model. Step four is Cloud and DB in which Register & Login to IBM Cloud subsequently Create Service Instance and Credentials then Launch Cloud and DB thereupon Create Database. The last step is Application Building in which Building HTML Pages and Build Python Code accordingly. Finally Run the Application.
3	Novelty / Uniqueness	Image processing Detected a car in AI based.
	Social Impact / Customer satisfaction	There is no need to give full amount to the policy holder. The amount is based on the damage.
	Business Model (Revenue Model)	Subscription and advertising model.
	Scalability of the solution	It may provide, the client to avoid giving the total insurance amount to the policyholder for a small damage in a vehicle.

3.4 PROBLEM SOLUTION FIT



4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	No Registration needed.
FR-2	Image	Image of the Damaged Vehicle is needed to predict the damage cost
FR-3	PC / Desktop or Mobile	Device needed to run the application

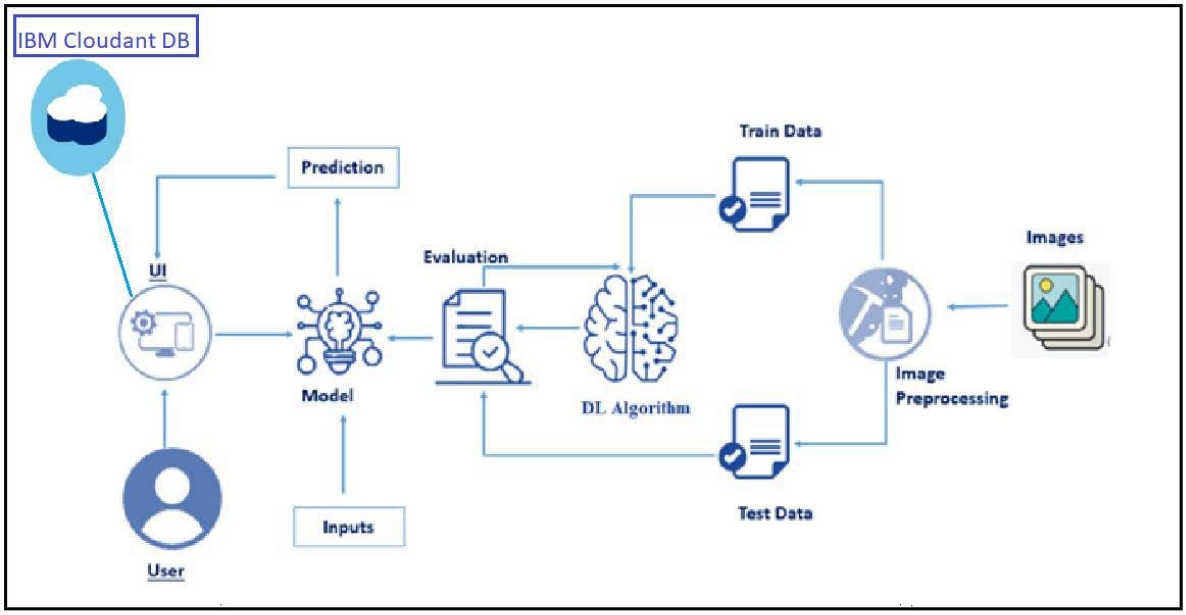
4.2 NON FUNCTIONAL REQUIREMENTS

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Used to predict damage of the vehicle
NFR-2	**Security**	No security needed while using this application
NFR-3	**Reliability**	Reliable on anytime anywhere with the environment
NFR-4	**Performance**	High performance to predict damage which is difficult to predict by manual
NFR-5	**Availability**	Available for customers on both web and mobile
NFR-6	**Scalability**	Scalable with predicting outer damage which is known but the interior is not possible at this time

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS :



5.2 SOLUTION ARCHITECTURE

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

• Find the best tech solution to solve existing business problems.

• Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.

• Define features, development phases, and solution requirements.

• Provide specifications according to which the solution is defined, managed, and delivered.

Solution Architecture Diagram:

5.3 USER STORIES :

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Dashboard	USN-1	As a user, I can use this user-friendly application without any registration and login	I can access this dashboard	High	Sprint-2
	Predict page	USN-2	As a user, I can upload the image of the damaged vehicle	I can access this predict page	Medium	Sprint-3
	Solution	USN-3	As a user, I can get the solution with the trained model	I can access this on predict page	Medium	Sprint-3
Customer (Web user)	Dashboard	USN-1	As a user, I can use this user-friendly web application without any registration and login	I can access this dashboard	High	Sprint-2
	Predict page	USN-2	As a user, I can upload the image of the damaged vehicle	I can access this predict page	Medium	Sprint-3
	Solution	USN-3	As a user, I can get the solution with the trained model	I can access this on predict page	Medium	Sprint-3

6. PROJECT PLANNING & SCHEDULING

Date	18 October 2022
Team ID	PNT2022TMID32157
Project Name	Project - **Intelligent Vehicle Damage Assessment & Cost Estimator for Insurance Companies**
Maximum Marks	8 Marks

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Pre-Processing	USN-1	As a user, I can upload any kind of Car images that need to be process	2	High	LingaPrabu Abinanthan
Sprint-1		USN-2	As a user, I will upload any resolution images	1	Low	Surya Ramkumar
Sprint-2	Model Building	USN-3	As a user, I will know the damage condition of the car	2	High	Surya Abinanthan
Sprint-2		USN-4	As a user, I need to know whether damaged in front or backside	2	Medium	LingaPrabu Ramkumar
Sprint-3	UI	USN-5	As a user, I can log into the application by entering email & password	5	High	LingaPrabu Abinanthan
		USN-6	As a user, I will need a Home page which contain basic details	2	Low	Abinanthan Ramkumar
		USN-7	As a user, I need a simple page for registration	5	Medium	Surya Abinanthan
Sprint-4		USN-8	As a user, I will need a page for upload vehicle image to analyse the damage cost	8	High	Ramkumar Surya

	Cloud Deployment	USN-9	As a user, I can access the app all over the world	13	High	LingaPrabu Surya

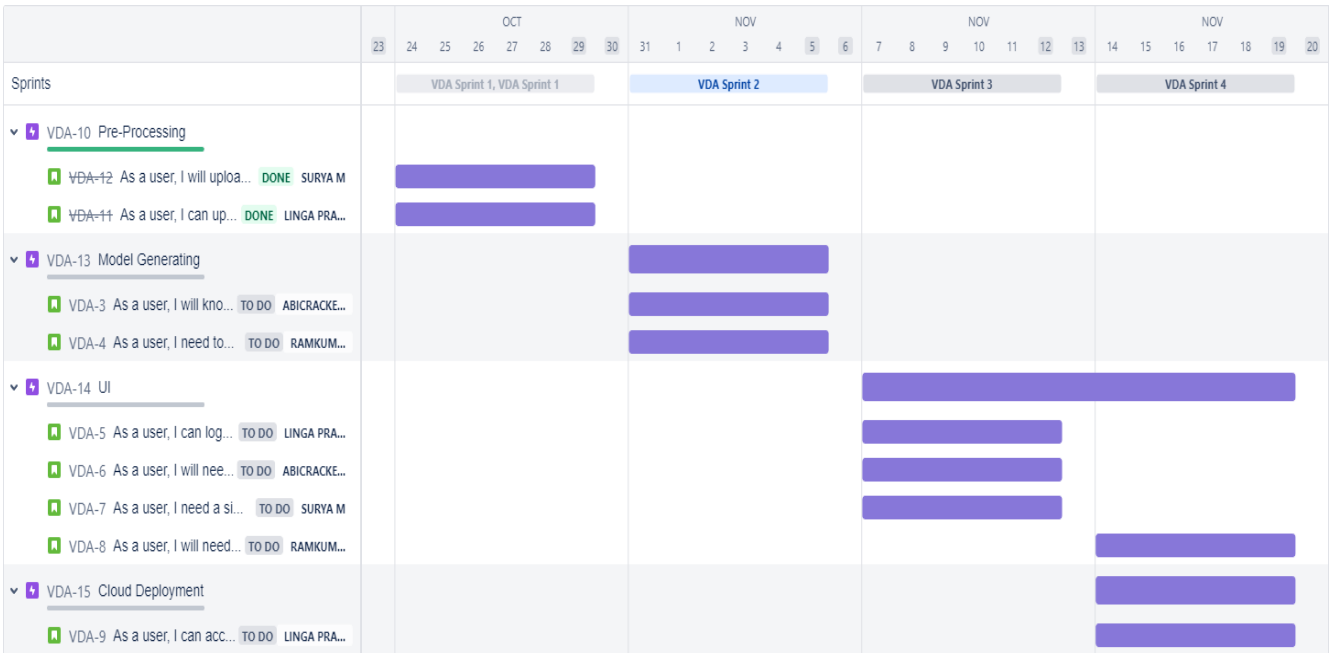
6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	3	6 Days	24 Oct 2022	29 Oct 2022	3	29 Oct 2022
Sprint-2	4	6 Days	31 Oct 2022	05 Nov 2022	4	05 Nov 2022
Sprint-3	12	6 Days	07 Nov 2022	12 Nov 2022	12	12 Nov 2022
Sprint-4	21	6 Days	14 Nov 2022	19 Nov 2022	21	19 Nov 2022

Velocity:

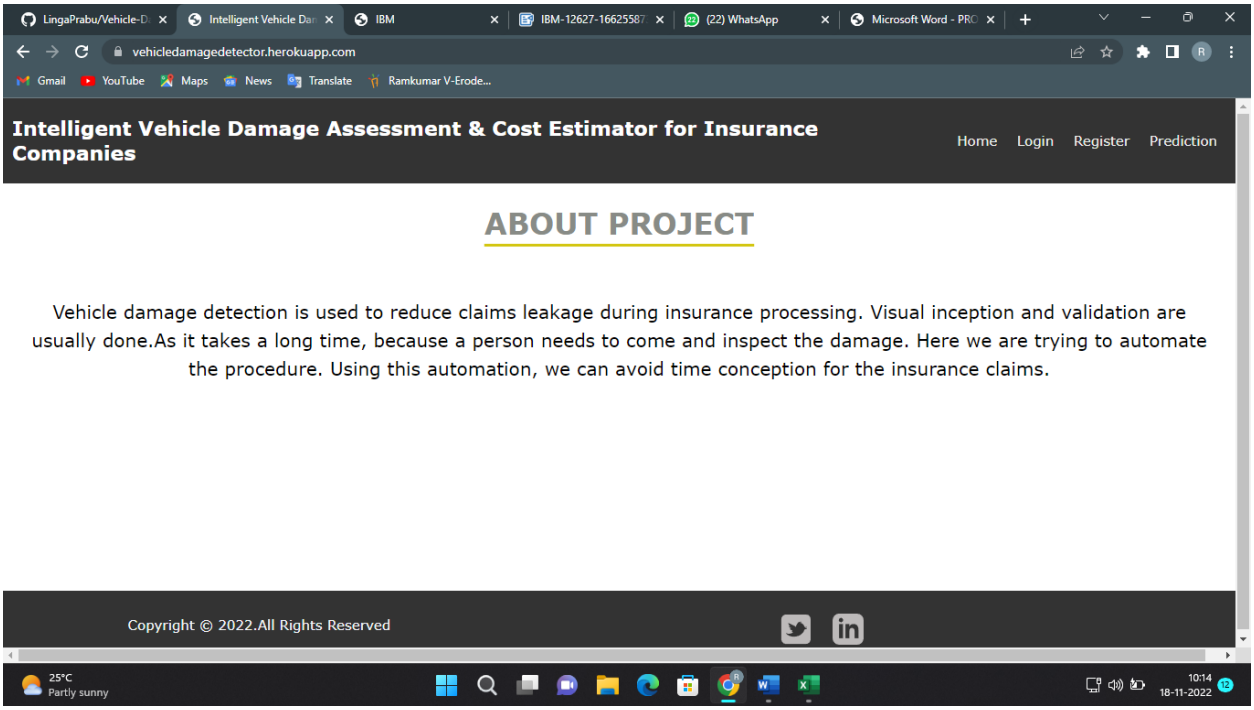
$$\text{Average Velocity} = \frac{\text{Sprint Duration}}{\text{Velocity}} = 20/8 = 2.5$$

6.3 REPORTS FROM JIRA :

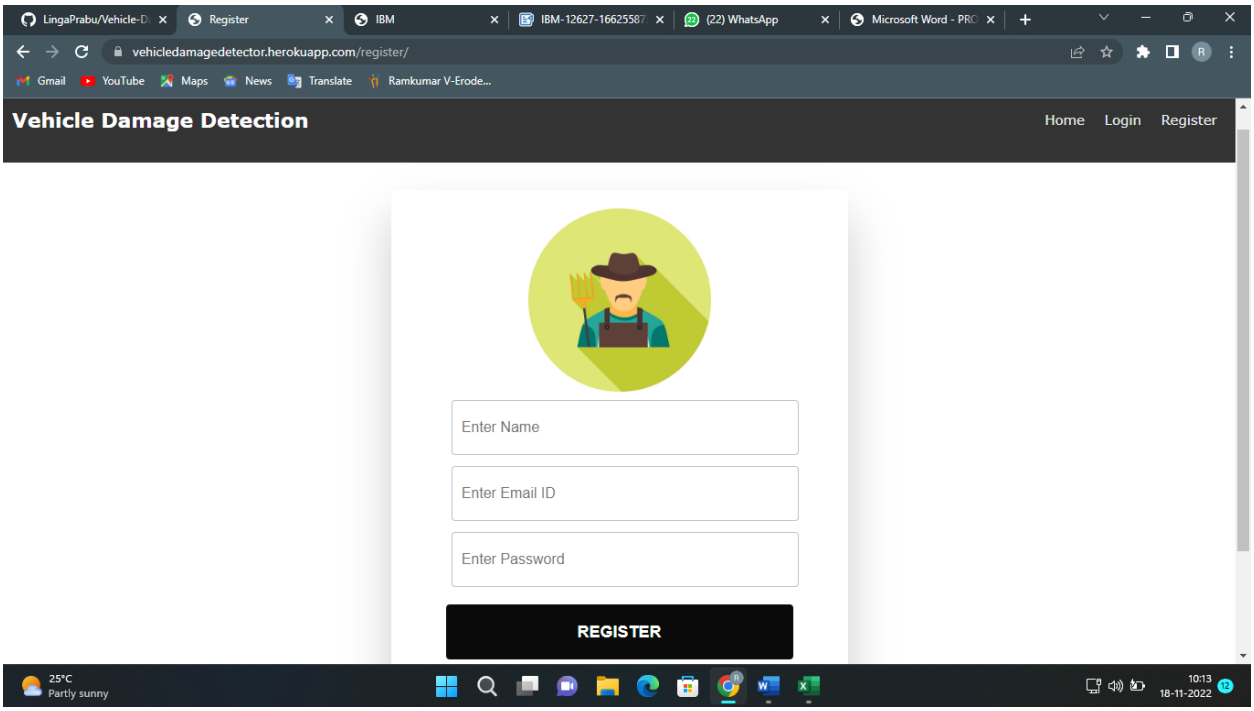


7. CODING & SOLUTIONING

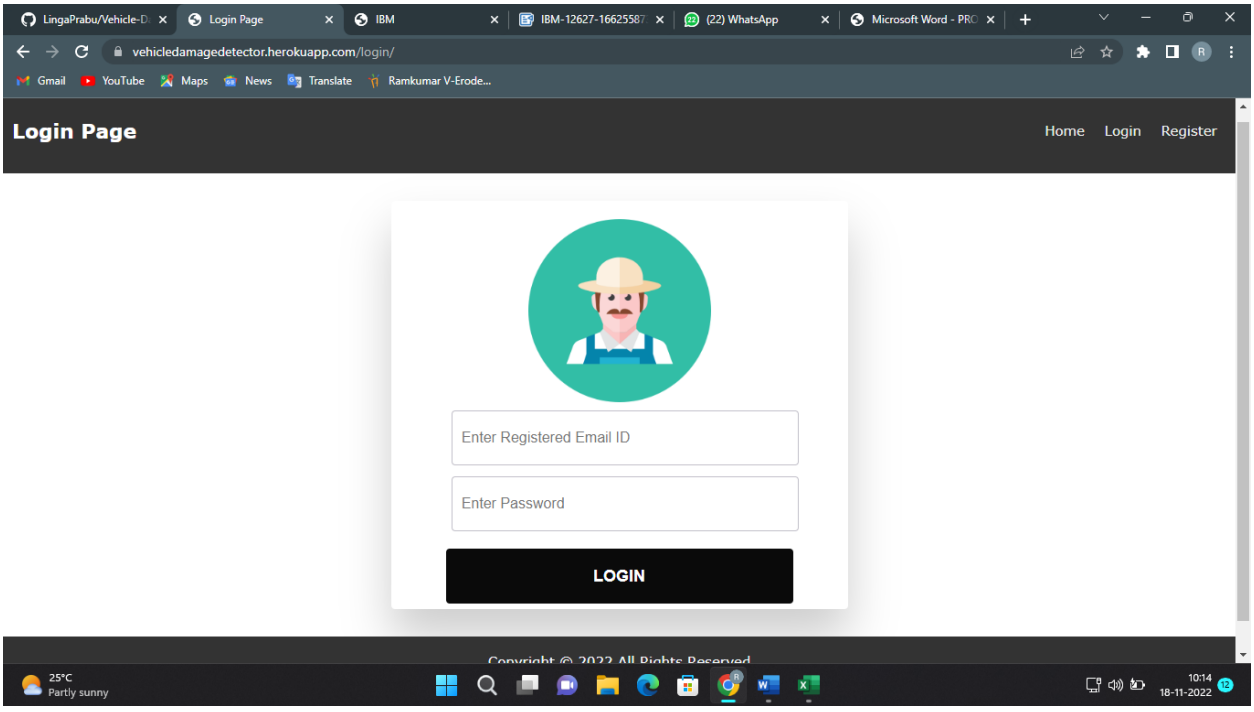
7.1 FEATURE 1



7.2 FEATURE 2


Register
IBM
IBM-12627-16625587
(22) WhatsApp
Microsoft Word - PRO
vehicledamagedetector.herokuapp.com/register/
Gmail
YouTube
Maps
News
Translate
Ramkumar V-Erode...
Vehicle Damage Detection
Home
Login
Register
Enter Name
Enter Email ID
Enter Password
REGISTER
25°C
Partly sunny
10:13
18-11-2022

7.3 FEATURE 3


Login Page
IBM
IBM-12627-16625587
(22) WhatsApp
Microsoft Word - PRO
vehicledamagedetector.herokuapp.com/login/
Gmail
YouTube
Maps
News
Translate
Ramkumar V-Erode...
Login Page
Home
Login
Register
Enter Registered Email ID
Enter Password
LOGIN
25°C
Partly sunny
10:14
18-11-2022

8. TESTING

8.1 TEST CASES

Test case ID	Feature Type	Component	Test Scenario
Test Case - O1	Functional	web Page	Verify user is able to see the page popup when user they enter to web page
Test Case - O2	UI	Home Page	Verify the UI elements in Hompage
Test Case - O3	Functional	Home page	Verify user is able to See the deatils about the webpage
Test Case - O4	Functional	Introduction	Verify user is able to details about uses of the Vehicle Damage Assessment
Test Case - O5	Functional	Register	Verify user is able to navigate and Register
Test Case - O6	Functional	Login	Verify user is able to navigate and Login

8.2 USER ACCEPTANCE TESTING

Date	15 November 2022
Team ID	PNT2022TMID32157
Project Name	Project - **Intelligent Vehicle Damage Assessment & Cost Estimator for Insurance Companies**
Maximum Marks	4 Marks

1.Purpose of document

The purpose of this document is to briefly explain the test coverage and open issuesof the [**Intelligent Vehicle Damage Assessment**] project at the time of the release to

User Acceptance Testing (UAT).

2.Defect Analysis

This report showsthe 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	6	2	0	1	9
Duplicate	2	0	3	0	5
External	3	2	0	0	5
Fixed	10	3	2	16	31
Not Reproduced	0	0	0	0	0
Skipped	0	0	1	1	2
Won't Fix	0	0	0	0	0
Totals	21	7	6	18	52

3.Test Case Analysis

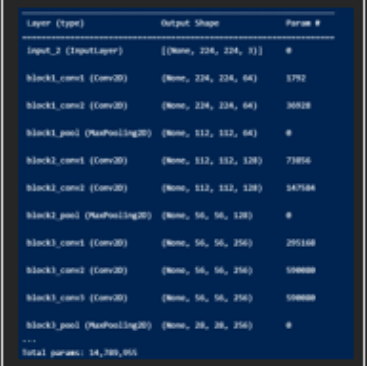
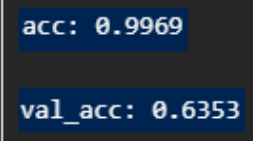
This report shows the number of test cases that have passed, failed,and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	10	1	0	10
Security	0	0	0	0
Outsource Shipping	6	0	0	6
Exception Reporting	6	0	0	6
Final Report Output	5	0	0	5

Version Control	1	0	0	1

9. RESULTS

9.1 PERFORMANCE METRICS

S.No.	Parameter	Values	Screenshot
1.	Model Summary	**Model Value - 989**	
2.	Accuracy	Training Accuracy – 0.9686 **Validation Accuracy - 0.6353**	

10. ADVANTAGES & DISADVANTAGE

ADVANTAGE

- Easy way to claim insurance.
- Give the accurate result of the damaged vehicle
- Easy to calculate the damage amount

DISADVANTAGE

- High quality image needed
- High amount of training images for data set is needed

11. CONCLUSION

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

12. FUTURE SCOPE

Increase the accuracy of the model by training for higher epochs and preventing overfitting issue Carrying out multiple functions in one package. A reduction of model training time is a challenge. An advanced model to classify the damage of the exact components and the extend of the damage.Detecting damages to vehicle panels such as dents, deformations in panels and also estimating the degree of damage in terms of repair costs. Making use of the reflections and specular highlights in the process of recovering the 3D pose.Cross-validation between annotators and studies of the effect of different annotation granularities can be carried out.

13. APPENDIX

SOURCE CODE

index.html

```
<!DOCTYPE html>
<html lang="en">
 <head>
   <meta charset="UTF-8" />
   <meta http-equiv="X-UA-Compatible" content="IE=edge" />
   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
   <link rel="stylesheet" href="../static/style.css" />
   <title>Intelligent Vehicle Damage Assessment & Cost Estimator for Insurance Companies</title>
 </head>
 <body>
   <nav class="navbar">
     <div class="logo">Intelligent Vehicle Damage Assessment & Cost Estimator for Insurance Companies</div>
     <ul class="nav-links">
       <input type="checkbox" id="checkbox_toggle" />
       <label for="checkbox_toggle"
```

```
class="hamburger">&#9776;</label>
        <div class="menu">
          <li><a href="/">Home</a></li>
          <li><a href="/login">Login</a></li>
          <li><a href="/register">Register</a></li>
          <li><a href="/prediction">Prediction</a></li>
        </div>
      </ul>
    </nav>
    <h1>ABOUT PROJECT</h1>
    <p>Vehicle damage detection is used to reduce claims leakage during insurance processing.
     Visual inception and validation are usually done.As it takes a long time, because a person needs to come and inspect the damage.
     Here we are trying to automate the procedure. Using this automation, we can avoid time conception for the insurance claims.
    </p>
    <footer>
     <div class="footer-content">
       <p>Copyright &copy; 2022.All Rights Reserved</p>
       <ul class="social"> <li><a href="#"><img width="40px" height="40px" src="../static/images/twitter.svg"></a></li>
         <li><a href="#"><img width="34px" height="34px" src="../static/images/linkedin.svg"></a></li>
         </ul>
     </div>
    </footer>
 </body>
</html>
```

login.html

```
<!DOCTYPE html>
<html lang="en" >
<head>
  <meta charset="UTF-8">
  <title>Login Page</title>
<link rel="stylesheet" href="../static/login.css">
```

```
7
8  </head>
9  <body>
10   <nav class="navbar">
11     <div class="logo">Login Page</div>
12     <ul class="nav-links">
13       <input type="checkbox" id="checkbox_toggle" />
14       <label for="checkbox_toggle"
   class="hamburger">&#9776;</label>
15       <div class="menu">
16         <li><a href="/">Home</a></li>
17         <li><a href="/login">Login</a></li>
18         <li><a href="/register">Register</a></li>
19       </div>
20     </ul>
21   </nav>
22 <div id="login-form-wrap">
23   <img width="200vw" height="200vw"
   src="../static/images/Login.png" alt="login">
24   <p>{{errors}}</p>
25   <form id="login-form" action="/afterlogin/"
   enctype="multipart/form-data" method="POST">
26     <p>
27     <input type="email" id="email" name="_id" placeholder="Enter
   Registered Email ID" required>
28     </p>
29     <p>
30       <input type="password" id="password" name="psw"
   placeholder="Enter Password" required>
31     </p>
32     <p>
33     <input type="submit" id="Login" value="Login">
34     </p>
35   </form>
36 </div>
37 <footer>
38   <div class="footer-content">
39     <p>Copyright &copy; 2022.All Rights Reserved</p>
40   </div>
41  </footer>
```

```
42
43 </body>
44 </html>
```

logout.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6     <meta name="viewport" content="width=device-width, initial-
   scale=1.0" />
7     <link rel="stylesheet" href="../static/style.css" />
8     <title>Vehicle Damage Detection</title>
9   </head>
10  <body>
11    <nav class="navbar">
12      <div class="logo">Vehicle Damage Detection</div>
13      <ul class="nav-links">
14        <input type="checkbox" id="checkbox_toggle" />
15        <label for="checkbox_toggle"
   class="hamburger">&#9776;</label>
16        <div class="menu">
17          <li><a href="/">Home</a></li>
18          <li><a href="/login">Login</a></li>
19          <li><a href="/register">Register</a></li>
20        </div>
21      </ul>
22    </nav>
23    <div class="Logout">
24    <h2>Successfully Logged Out!</h2>
25    <h3>Login for more information</h3>
26    <a href="/login">Login</a>
27    </div>
28    <footer>
29     <div class="footer-content">
30       <p>Copyright &copy; 2022.All Rights Reserved</p>
31       <ul class="social"> <li><a href="#"><img width="40px"
   height="40px" src="../static/images/twitter.svg"></a></li>
```

```
        <li><a href="#"><img width="34px" height="34px"
  src="../static/images/linkedin.svg"></a></li>
        </ul>
    </div>
   </footer>
 </body>
</html>
```

Prediction.html

```
<!DOCTYPE html>
<html lang="en" >
<head>
  <meta charset="UTF-8">
  <title>Prediction</title>
<link rel="stylesheet" href="../static/login.css">
<style>
  label {
  display:block;
  width:100%;
  margin-top: 5%;
  height:55px;
  line-height:50px;
  text-align:center;
  background:#1172c2;
  color:#fff;
  font-size:15px;
  font-family:"Open Sans",sans-serif;
  text-transform:Uppercase;
  font-weight:600;
  border-radius:5px;
  cursor:pointer;
}
</style>
<script>
  function showPreview(event){
  if(event.target.files.length > 0){
    var src = URL.createObjectURL(event.target.files[0]);
    var preview = document.getElementById("file-ip-1-preview");
    preview.src = src;
```

```
    preview.style.display = "block";
  }
}
</script>
</head>
<body>
  <nav class="navbar">
    <div class="logo">Vehicle Damage Detection</div>
    <ul class="nav-links">
      <input type="checkbox" id="checkbox_toggle" />
      <label for="checkbox_toggle"
class="hamburger">&#9776;</label>
      <div class="menu">
        <li><a href="/">Home</a></li>
        <li><a href="/logout">Logout</a></li>
      </div>
    </ul>
  </nav>
<div class="prediction" id="prediction-form-wrap">
  <form action="/result" method="POST" enctype="multipart/form-
data" id="prediction-form">
    <p>
      <label for="file">Upload Image</label>
    <input type="file" style="display: none;" accept="image/*"
id="file" name="file" onchange="showPreview(event);" required>
    </p>
    <div class="preview">
      <img id="file-ip-1-preview">
    </div>

    <p>
    <input type="submit" id="submit" value="Submit">
    </p>
  </form>
</div>
<h1>Your Car Damaged ( {{damage}} ) in ({{area}})<br> The
Estimated Cost For The Damage Is : {{prediction}}</h1>
<footer>
  <div class="footer-content">
    <p>Copyright &copy; 2022.All Rights Reserved</p>
```

```
  </div>
 </footer>

</body>
</html>
```

Register.html

```
<!DOCTYPE html>
<html lang="en" >
<head>
  <meta charset="UTF-8">
  <title>Register</title>
<link rel="stylesheet" href="../static/login.css">

</head>
<body>
  <nav class="navbar">
    <div class="logo">Vehicle Damage Detection</div>
    <ul class="nav-links">
      <input type="checkbox" id="checkbox_toggle" />
      <label for="checkbox_toggle" class="hamburger">&#9776;</label>
      <div class="menu">
        <li><a href="/">Home</a></li>
        <li><a href="/login">Login</a></li>
        <li><a href="/register">Register</a></li>
      </div>
    </ul>
  </nav>
<div id="register-form-wrap">
  <img width="200vw" height="200vw" src="../static/images/register.jpg" alt="register">
  <p>{{errors}}</p>
  <form id="register-form" action="/afterreg" enctype="multipart/form-data" method="POST">
    <p>
    <input type="text" id="name" name="name" placeholder="Enter Name" required>
    </p>
```

```
    <p>
    <input type="email" id="email" name="_id" placeholder="Enter Email ID" required>
    </p>
    <p>
      <input type="password" id="password" minlength="6" name="psw" placeholder="Enter Password" required>
    </p>
    <p>
    <input type="submit" id="Register" value="Register">
    </p>
  </form>
  <div id="already">
    <p>Already have an account? <a href="/login">Login</a><p>
  </div><!--already-->
</div>
<footer>
  <div class="footer-content">
    <p>Copyright &copy; 2022.All Rights Reserved</p>
  </div>
 </footer>

</body>
</html>
```

Login.css

```
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
 }
 body {
  font-family: Verdana;
 }
.navbar {
  display: flex;
  align-items: center;
  justify-content: space-between;
  padding: 10px;
```

```
 height: 15vh;
 background-color: #333333;
 color: #fff;
}
#file-ip-1-preview{
    width:100%;
    margin-top: 5%;
    display:none;
    margin-bottom:30px;
}
.nav-links a {
 color: #edefec;
}
a {
 text-decoration: none;
}
li {
 list-style: none;
}
/* LOGO */
.logo {
 font-size: 22px;
 font-weight: 600;
}
/* NAVBAR MENU */
.menu {
 display: flex;
 gap: .1em;
 font-size: 15px;
}
.menu li:hover {
 background-color: #4c9e9e;
 border-radius: 5px;
 transition: 0.3s ease;
}
.menu li {
 padding: 3px 10px;
}
input[type=checkbox]{
 display: none;
```

```
54  }
55  /*HAMBURGER MENU*/
56  .hamburger {
57   display: none;
58   font-size: 24px;
59   user-select: none;
60  }
61  /* APPLYING MEDIA QUERIES */
62  @media (max-width: 768px) {
63  .menu {
64   display:none;
65   position: absolute;
66   background-color:#333333;
67   right: 0;
68   left: 0;
69   text-align: center;
70   padding: 16px 0;
71  }
72  .menu li:hover {
73   display: inline-block;
74   background-color:#4c9e9e;
75   transition: 0.3s ease;
76  }
77  .menu li + li {
78   margin-top: 12px;
79  }
80  input[type=checkbox]:checked ~ .menu{
81   display: block;
82  }
83  .hamburger {
84   display: block;
85  }
86  }
87
88
89 p {
90   text-align: center;
91 }
92
93 #already a,
```

```
#already a:link,
#already a:visited,
#already a:active {
  color: #3ca9e2;
  -webkit-transition: all 0.2s ease;
  transition: all 0.2s ease;
}
#already a:focus,#already a:hover,
#already a:link:focus,
#already a:link:hover,
#already a:visited:focus,
#already a:visited:hover,
#already a:active:focus,
#already a:active:hover {
   color: #329dd5;
   -webkit-transition: all 0.2s ease;
   transition: all 0.2s ease;
}

#register-form-wrap,#login-form-wrap,#prediction-form-wrap {
   background-color: #fff;
   width:  500px;
   min-width: 400px;
   margin: 30px auto;
   text-align: center;
   padding: 20px 0 0 0;
   border-radius: 4px;
   box-shadow: 0px 30px 50px 0px rgba(0, 0, 0, 0.2);
}
#prediction-form-wrap{
   margin-top: 10vh;
   margin-bottom: 10vh;
   width: 300px;
}
h1{
   color: #888b86;
   text-decoration: underline;
   text-decoration-color: #d3c611;
   text-align: center;
   line-height: 10vh;
```

```
  margin-top: 1vh;
  margin-bottom: 21.5vh;
  text-underline-offset: 10px;
}
footer{
  position: relative;
  background-color: #333333;
  color: #ffff;
  width: 100%;
  bottom: 0;
  left:0;
  right:0;
  height: 10vh;
  padding-top:1vh;
}
.footer-content p{
  font-size:15px;
  color: #f5f7fa;
  margin-top: 2vh;
}
.footer-content ul{
  list-style: none;
  display: flex;
  align-items: center;
  justify-content: center;
  margin: 1rem 0 3rem 0;
}
.footer-content li{
  margin: 0 10px;
}

#register-form,#login-form,#prediction-form {
  padding: 0 60px;
}

input {
  display: block;
  box-sizing: border-box;
  width: 100%;
  outline: none;
```

```
  margin-top: 2vh;
  margin-bottom: 1vh;
  height: 60px;
  line-height: 60px;
  border-radius: 4px;
}

input[type="text"],
input[type="password"],
input[type="email"] {
  width: 100%;
  padding: 0 0 0 10px;
  margin: 1vh;
  color: #8a8b8e;
  border: 1px solid #c2c0ca;
  font-style: normal;
  font-size: 16px;
  -webkit-appearance: none;
     -moz-appearance: none;
          appearance: none;
  position: relative;
  display: inline-block;
  background: none;
}
input[type="text"]:focus,
input[type="password"]:focus,
input[type="email"]:focus {
  border-color: #3ca9e2;
}
input[type="text"]:focus:invalid,
input[type="password"]:focus:invalid,
input[type="email"]:focus:invalid {
  color: #cc1e2b;
  border-color: #cc1e2b;
}
input[type="text"]:valid ~ .validation,
input[type="password"]:valid ~ .validation,
input[type="email"]:valid ~ .validation {
  display: block;
  border-color: #0C0;
```

```
214 }
215 input[type="text"]:valid ~ .validation span,
216 input[type="password"]:valid ~ .validation span,
217 input[type="email"]:valid ~ .validation span {
218   background: #0C0;
219   position: absolute;
220   border-radius: 6px;
221 }
222 input[type="text"]:valid ~ .validation span:first-child,
223 input[type="password"]:valid ~ .validation span:first-child,
224 input[type="email"]:valid ~ .validation span:first-child {
225   top: 30px;
226   left: 14px;
227   width: 20px;
228   height: 3px;
229   -webkit-transform: rotate(-45deg);
230           transform: rotate(-45deg);
231 }
232 input[type="text"]:valid ~ .validation span:last-child,
233 input[type="email"]:valid ~ .validation span:last-child,
234 input[type="password"]:valid ~ .validation span:last-child {
235   top: 35px;
236   left: 8px;
237   width: 11px;
238   height: 3px;
239   -webkit-transform: rotate(45deg);
240           transform: rotate(45deg);
241 }
242
243 .validation {
244   display: none;
245   position: absolute;
246   content: " ";
247   height: 60px;
248   width: 30px;
249   right: 15px;
250   top: 0px;
251 }
252
253 input[type="submit"] {
```

```
  border: none;
  display: block;
  background-color: #0A0A0A;
  color: #fff;
  font-weight: bold;
  text-transform: uppercase;
  cursor: pointer;
  -webkit-transition: all 0.2s ease;
  transition: all 0.2s ease;
  font-size: 18px;
  position: relative;
  display: inline-block;
  cursor: pointer;
  text-align: center;
}
input[type="submit"]:hover {
  background-color: #329dd5;
  -webkit-transition: all 0.2s ease;
  transition: all 0.2s ease;
}

#already {
  background-color: #eeedf1;
  color: #8a8b8e;
  font-size: 14px;
  margin-top: 2vh;
  width: 100%;
  padding: 10px 0;
  border-radius: 0 0 4px 4px;
}
```

Style.css

```
* {
 margin: 0;
 padding: 0;
 box-sizing: border-box;
}
body {
 font-family: Verdana;
```

```
 }
 a {
  text-decoration: none;
}
li {
 list-style: none;
}
h1{
    color: #888b86;
    text-decoration: underline;
    text-decoration-color: #d3c611;
    text-align: center;
    margin-top: 4vh;
    text-underline-offset: 10px;
}
p{
    margin: 10vh 5vh 2vh 5vh;
    text-align:center;
    font-size: 20px;
    line-height: 5vh;
}
footer{
    position: absolute;
    background-color: #333333;
    color: #ffff;
    width: 100vw;
    bottom: 0;
    left:0;
    right:0;
    height: 10vh;
    padding-top:1vh;
}
.footer-content p{
    float: left;
    font-size:15px;
    margin:0 0 0 10vw;
    color: #f5f7fa;
}
.footer-content ul{
    list-style: none;
```

```
    display: flex;
    align-items: center;
    justify-content: center;
    margin: 1rem 0 3rem 0;
}
.footer-content li{
    margin: 0 10px;
}

.navbar {
 display: flex;
 align-items: center;
 justify-content: space-between;
 padding: 10px;
 height: 15vh;
 background-color: #333333;
 color: #fff;
}
.nav-links a {
 color: #edefec;
}
/* LOGO */
.logo {
 font-size: 22px;
 font-weight: 600;
}
/* NAVBAR MENU */
.menu {
 display: flex;
 gap: .1em;
 font-size: 15px;
}
.menu li:hover {
 background-color: #4c9e9e;
 border-radius: 5px;
 transition: 0.3s ease;
}
.menu li {
 padding: 3px 10px;
}
```

```
input[type=checkbox]{
 display: none;
}
/*HAMBURGER MENU*/
.hamburger {
 display: none;
 font-size: 24px;
 user-select: none;
}
/* APPLYING MEDIA QUERIES */
@media (max-width: 768px) {
.menu {
  display:none;
  position: absolute;
  background-color:#333333;
  right: 0;
  left: 0;
  text-align: center;
  padding: 16px 0;
 }
 .menu li:hover {
  display: inline-block;
  background-color:#4c9e9e;
  transition: 0.3s ease;
 }
 .menu li + li {
  margin-top: 12px;
 }
 input[type=checkbox]:checked ~ .menu{
  display: block;
 }
 .hamburger {
  display: block;
 }
 }
 .Logout h2,h3{
     text-align: center;
     margin-top: 5vh;
 }
 .Logout h3{
```

```
128     color: green;
129 }
130 .Logout a{
131         background: #0A0A0A;
132         background-image: -webkit-linear-gradient(top, #0A0A0A,
    #2D3036);
133         background-image: -moz-linear-gradient(top, #0A0A0A,
    #2D3036);
134         background-image: -ms-linear-gradient(top, #0A0A0A,
    #2D3036);
135         background-image: -o-linear-gradient(top, #0A0A0A,
    #2D3036);
136         background-image: -webkit-gradient(to bottom, #0A0A0A,
    #2D3036);
137         -webkit-border-radius: 20px;
138         -moz-border-radius: 20px;
139         border-radius: 20px;
140         color: #FFFFFF;
141         font-family: Times New Roman;
142         font-size: 39px;
143         margin:5vh 35vw;
144         font-weight: 100;
145         padding: 10px;
146         -webkit-box-shadow: 1px 1px 20px 0 #000000;
147         -moz-box-shadow: 1px 1px 20px 0 #000000;
148         box-shadow: 1px 1px 20px 0 #000000;
149         text-decoration: none;
150         display: block;
151         cursor: pointer;
152         text-align: center;
153      }
```

App.py

```
1  import re
2  import numpy as np
3  import os
4  from flask import Flask,
   app,request,render_template,redirect,url_for,session
5  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
from cloudant.client import Cloudant

model1=load_model('./Model/body.h5')
model2 = load_model('./Model/level.h5')
app=Flask(__name__)
client = Cloudant.iam("c059a4ef-8c13-4d65-bc33-da592b41838a-
bluemix", "511lW2p85eY0cZqJRC5srXUULERV69Ru9z9UxM4lwngE",
connect=True)
my_database=client.create_database('my_database')
app.secret_key = 'Nothing'

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

@app.route('/register/', methods=["GET","POST"])
def register():
    return render_template('register.html')

@app.route('/afterreg/', methods=["GET","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)
```

```
43      print(docs)
44      print(len(docs.all()))
45      if(len(docs.all())==0):
46          url=my_database.create_document(data)
47          return render_template('register.html',
    errors="Registration Successful, Please Login")
48      else:
49          return render_template('register.html', errors="Your
    Account already exist, Please Login Using that")
50
51  @app.route('/login/',methods=["GET","POST"])
52  def login():
53      if('user' in session):
54          return render_template('prediction.html')
55      else:
56          return render_template('login.html')
57
58
59  @app.route('/afterlogin/', methods=["GET", "POST"])
60  def afterlogin():
61      user=request.form['_id']
62      passw=request.form['psw']
63      print(user,passw)
64      query={'_id':{'$eq':user}}
65      docs=my_database.get_query_result(query)
66      print(docs)
67      print(len(docs.all()))
68      if(len(docs.all())==0):
69          return render_template('login.html', errors="Username not
    found")
70      else:
71          if((user==docs[0][0]['_id'] and
    passw==docs[0][0]['psw'])):
72              session['user'] = user
73              return redirect(url_for('prediction'))
74          else:
75              return render_template('login.html', errors="Wrong
    Credentials.")
76
77  @app.route('/logout/')
```

```
def logout():
    session.pop('user')
    return render_template('logout.html')

@app.route('/prediction/')
def prediction():
    if('user' in session):
        return render_template('prediction.html')
    else:
        return render_template('login.html')

@app.route('/result/',methods=["GET","POST"])
def res():
    if request.method=="POST":
        f=request.files['file']
        basepath=os.path.dirname(__file__)
        filepath=os.path.join(basepath,"hello.jpg")
        f.save(filepath)

        img=image.load_img(filepath,target_size=(224,224))
        x=image.img_to_array(img)
        x=np.expand_dims(x,axis=0)
         img_data=preprocess_input(x)
         prediction1=np.argmax(model1.predict(img_data))
         prediction2=np.argmax(model2.predict(img_data))
         index1=['Front','Rear','Side']
         index2=['Minor','Moderate','Severe']
         result1=index1[prediction1]
         result2=index2[prediction2]
         if(result1=="Front" and result2=="Minor"):
             value="3000 - 5000 INR"
         elif(result1 == "Front" and result2 == "Moderate"):
             value = "6000 - 8000 INR"
         elif(result1 == "Front" and result2 == "Severe"):
             value = "9000 - 11000 INR"
         elif(result1 == "Rear" and result2 == "Minor"):
             value = "4000 - 6000 INR"
         elif(result1 == "Rear" and result2 == "Moderate"):
             value = "7000 - 9000 INR"
         elif(result1 == "Rear" and result2 == "Severe"):
```

```
            value = "11000 - 13000 INR"
        elif(result1 == "Side" and result2 == "Minor"):
            value = "6000 - 8000 INR"
        elif(result1 == "Side" and result2 == "Moderate"):
            value = "9000 - 11000 INR"
        elif(result1 == "Side" and result2 == "Severe"):
            value = "12000 - 15000 INR"
        else:
            value="16000 - 50000 INR"
        return
   render_template('prediction.html',prediction=value,damage=result2
   ,area=result1)


if __name__ == "__main__":
    app.run()
```

GITHUB & PROJECT DEMO LINK

GITHUB LINK:https://github.com/IBM-EPBL/IBM-Project-12627-1659455519

DEMO VIDEO LINK : https://drive.google.com/file/d/1drgty-pPPMCTYYGO29dd6rdqQc8d6S0I/view?usp=sharing