INTELLIGENT VEHICLE DAMAGE ASSESSMENT
AND COST ESTIMATION
FOR INSURANCE COMPANIES

A PROJECT REPORT

Submitted by

TEAM ID : PNT2022TMID44408

DHATCHANA MOORTHI AP	**731219104006**
MADHAN M	**731219104011**
SENTHILKUMAR V	**731219104022**
SUBASHCHANDRAN P	**731219104023**

PROFESSIONAL READINESS FOR INNOVATION,
EMPLOYABILITY AND ENTREPRENEURSHIP

J.K.K.MUNIRAJAH COLLEGE OF TECHNOLOGY
T.N.PALAYAM, GOBI-638 506

ANNA UNIVERSITY::CHENNAI 600025
DECEMBER 2022

BONAFIDE CERTIFICATE

Certified that this **IBM Nalaiya Thiran** project report on **"HX8001- INTELLIGENT VEHICLE DAMAGE ASSESSMENT AND COST ESTIMATION FOR INSURANCE COMPANIES"** is the bonafide work of **"AP.DHATCHANMOORTHI (731219104006), M.MADHAN (731219104011), V.SENTHILKUMAR (731219104022) And P.SUBASHCHANDRAN (731219104023)"**who carried out the project work Under my supervision.

SIGNATURE

Dr.N.SATHYABALAJI.M.E.Ph.D,M.I.S.T.E,

ASSOCIATE   PROFESSOR

HEAD OF THE DEPARTMENT

 Dept.of Information Technology

J.K.K.Munirajah College of Technology

T.N.Palayam

SIGNATURE

Mr.E .ANANTH.M.E

ASSISTANT PROFESSOR

MENTOR

Dept.of Computer Science & Engineering

J.K.K.Munirajah College of Technology

T.N.Palayam

ABSTRACT

Nowadays, the proliferation of automobile industries is directly related to the number of claims in insurance companies. Those companies are facing many simultaneous claims and solving claims leakage The project can be extended to green houses where manual supervision is far and few in between.

In Advanced Artificial Intelligence (AI), machine learning and deep learning algorithms can help to solve these kinds of problems for insurance industries. we apply deep learning-based algorithms, VGG16 and VGG19, for car damage detection and assessment in real world datasets .The algorithms detect the damaged part of a car, assess its location and severity .After analyzing and implementing our models, we can find out that the results of using transfer learning and regularization can work better than those of fine-tuning. Then we apply transfer learning in pre-trained VGG models and use some techniques to improve the accuracy of our system

ACKNOWLEDGEMENT

TABLE OF CONTENTS

14   SCREENSHOTS   81

PROJECT REPORT

1. INTRODUCTION

PROJECT OVERVIEW

Nowadays, the proliferation of automobile industries is directly related to the number of claims in insurance companies. Those companies are facing many simultaneous claims and solving claims leakage The project can be extended to green houses where manual supervision is far and few in between.

In Advanced Artificial Intelligence (AI), machine learning and deep learning algorithms can help to solve these kinds of problems for insurance industries

PURPOSE

Auto-detection of vehicle damage using photos taken at the scene of an accident can be very useful as it can drastically reduce the cost of processing insurance claims, as well as provide greater convenience to vehicle users.

2. LITERATURE SURVEY

TITLE	METHODOLOGY	AUTHOR/ PUBLICATION YEAR	ADVANTAGE	DISADVANTAGE
Applying image analysis to auto insurance Triage	Image analysis and pattern recognition are applied to automatically identify and characterize automobile damage.	Li Ying & Dorai Chitra, 2012	Because of the advancement of image analysis and pattern recognition technologies, the auto insurance industry could significantly benefit.	The drawback is that the automobile damaged can be analyzed only having white background otherwise it will be not able to give thedesired results.
Image based automatic vehicle damage detection	This approach requires 3D computer aided design (CAD) modes of the considered vehicle to identify how it would look if it were undamaged.	Srimal Jayewardene' ,2013	Automatically detecting the damage of the vehicle using photographs clicked at the accident site is extremely functional as it can greatly decrease the rate of processing insurance claims, and it will also providegreater	Vehicles have very reflective metallic bodies the photographs taken insuch an uncontrolled environment can be expected to have a certain amount of inter object reflection. Application of standard computer vision.

A Secure AI-driven Architecture for Automated Insurance Systems Fraud Detection and Risk Measurement	Block chain, data analysis, machine learning, AI for damageidentification.	M.Wassel,2019	Proposed classifiers ensure not only the best accuracy in detecting fraudulent claims but also can classify different types of fraud for insurance unlike the existing solutions	The major drawback of the proposed model is that it only identifies the physical visible damage and not of the internal or the interior damage.
Car damage detection and classification	CNN model is trained on Image Net dataset. After fine tuning the dataset, transfer learning with L2 regularization is Applied	Phyu Mar Kyu, Kuntpong Woraratpanya, 2020	Pre-trained VGG model not only detect damaged part of a car but also assess its location and severity.	Transfer learning and regularization can work better than those of finetuning.

EXISTING PROBLEM

In proposed system firstly, it collects the pictures of one's damaged automobile, later use these pictures to feed into our ML model that makes use of image processing to identify the details of the image, using Image processing it analyses the percentage of damage of the automobile. Next, it segregates the pictures based on 2 factors which are replace and repair.

REFERENCES

[1] LeCun, Y., Boser, B., Denver, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D. Back propagation applied to handwritten zip code recognition. Neural computation, 1989, pp. 541-551.

[2] Krizhevsky, A., Sutskever, I., Hinton, G. Imagenet classification with deep conventional neural networks. In NIPS, 2012, pp. 1097-1105.

[3] Zeiler, M. D., Fergus, R. Visualizing and understanding convolutional neural networks. In ECCV, 2014, pp. 818-833.

[4] LeCun, Y., Bengio, Y., Hinton, G. Deep learning. Nature, 2015(521), pp. 436-444.

[5] Simonyan, K., Zisserman, A. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015, pp. 1409.1556.

[6] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. Faster R-

CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015, pp. 91-99.

 [7] Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick. Mask R-CNN. In ICCV,2017,pp. 2980-2988.

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

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

  [10]. Phyu Mar Kyu ,car damage detection and classification, faculty of information technology king Mongkut's institute of technology ladkrabang Bangkok,Thailand 62606003@kmitl.ac.in

PROBLEM STATEMENT DEFINITION

Mr.Vimal Rajesh is a 50 years old man. He had a own Car(Duster) and he worked at basic salary for past 30 Years , In this 30 Years he Faced a problem in Choosing Car Damage and Insurance claim.

• Vimal Rajesh wants to know the better recommendation for insurance claiming.

• He has faced huge losses for a long time.

• This problem is usually faced by lot of Customers.

• Mr. Vimal Rajesh needs to know the result immediately for Insurance claim.

| Who does the problem affect?	Persons was claim the assessment amount
What are the boundaries of the problem?	People who vehicle and facing Issues of Insurance claiming
Where does the issue occur?	The issue occurs in Automobile industry interior part damage not fully estimated so issues occur in company side, particularly City side

3. IDEATION & PROPOSED SOLUTION

EMPATHY MAP CANVAS
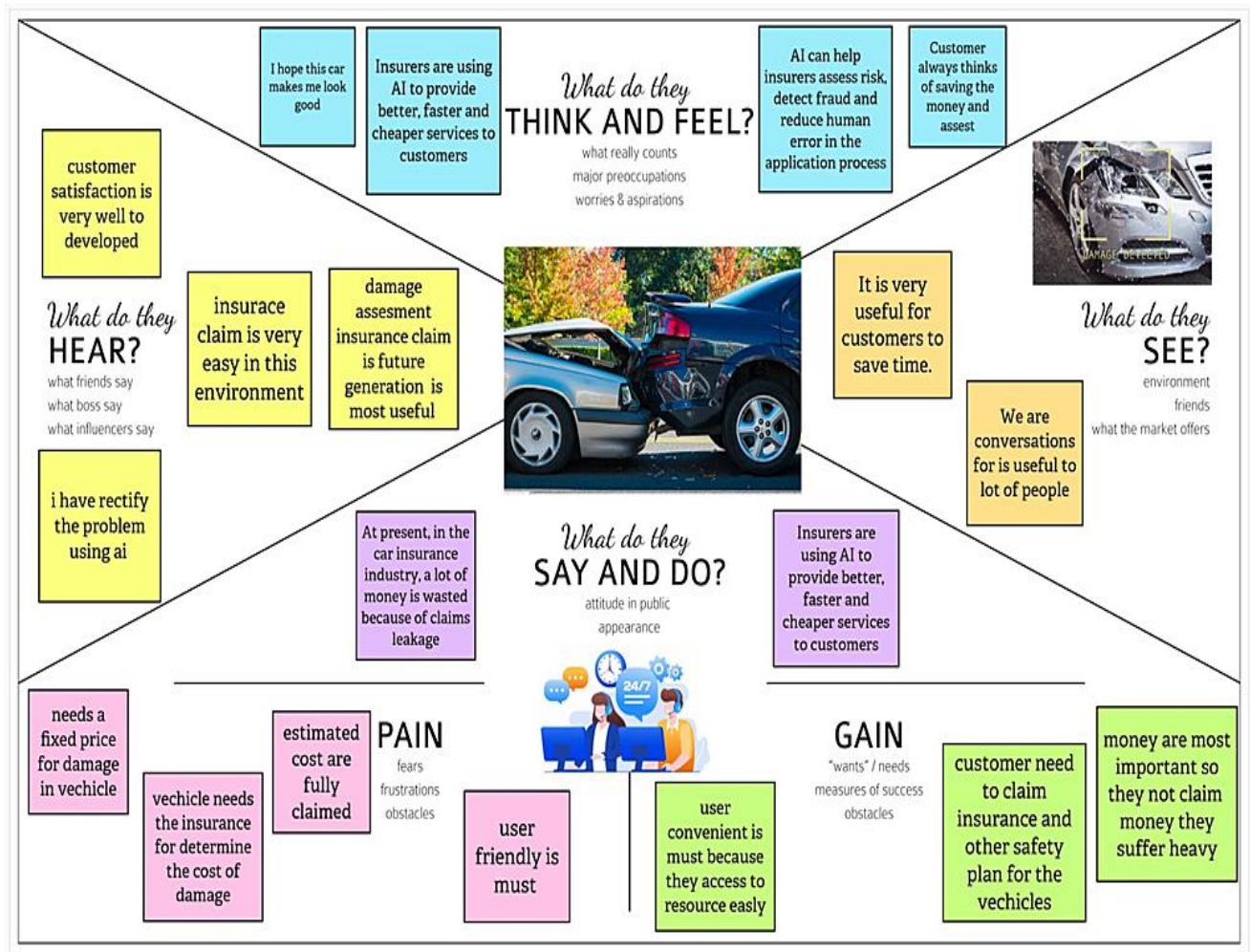
Figure 3.1.1 Empathy map

IDEATION & BRAINSTORMING
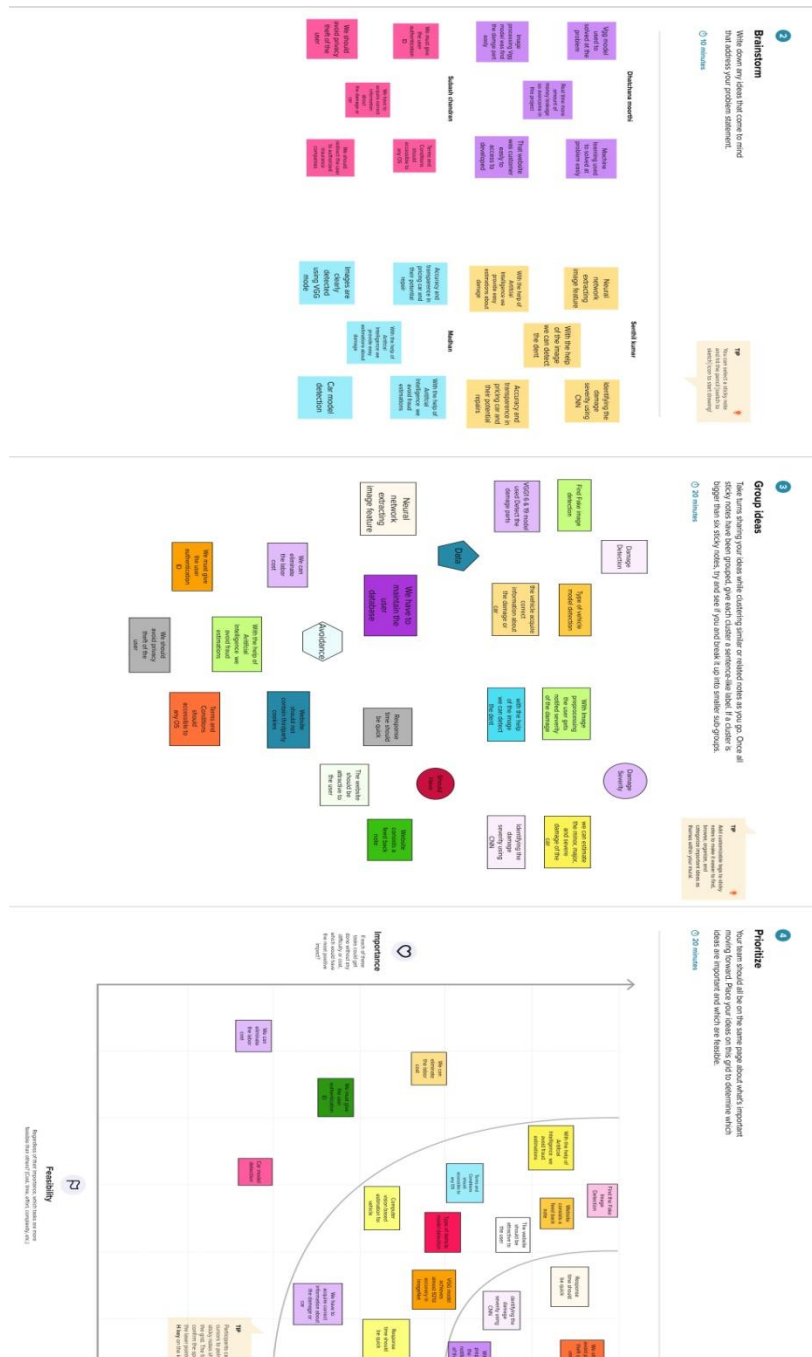
Figure 3.2.1 Brainstorm Diagram

PROPOSED SOLUTION

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

2	Idea / Solution description	Automobile Industry is one of the major industry in a Country. This proposed system is Intelligent vehicle damage assessment and cost estimator for insurance companies using computer vision in artificial intelligence. The model will predict the location of the damage as in front, side or rear, and the severity of such a damage as in minor, moderate or severe and estimate the cost of damage of both car and bike.
3	Novelty / Uniqueness	Deep learning method used to fixed the problem and then Working on with VGG16 pre[1]trained model by adding extra layers to increase the accuracy while implementing the project.
4	Social Impact / Customer Satisfaction	The model developed will be used to fix the damage caused to the Vehicle quickly so that the vehicle can be modified to the old look and also for faster processing of cost of the damage to claim insurance quickly. This project can be used to save time for calculating the area and level of the damage quickly such that the insurance claim can be made efficiently.

5	Business Model (Revenue Model)	This can also be used to help car companies as well. •Can collaborate with insurance companies. • Can collaborate with car companies
6	Scalability of the Solution	AI guided Application provides 24/7 service to clear all customer queries and guide them through all the processes. In future, it can be scaled as per the requirements of the insurance or car company to include answers to queries related to the cost based on the inputs Provided

PROBLEM SOLUTION FIT

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

Project design Phase - 1

Team ID: PNT2022TMID44408

Define CS, fit into CC

1. CUSTOMER SEGMENT — CS

- Common people
- Car companies
- Insurance companies
- Vehicle owner

6. CUSTOMER — CC

Trust Problem : The insurance companies Give the fake information about the estimation cost modify damage part of ensuring profit of company from user

Anxiety : Customer concern is the level of severity of the damaged parts and fixing the extent of the damaged parts is also the customer concern.

5. AVAILABLE SOLUTIONS — AS

By gathering information about the damage assessment from customer side and vehicle companies to understand the level of damage and cost needed to same

Searching the through the internet such as online website get the insight about the damage

Explore AS, differentiate

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

2. JOBS-TO-BE-DONE /a PROBLEMS — J&P

- Analyze the damage part and giving good cost estimation for customer satisfy and support system

- Provide the good prediction system to predict the level of damage and modify the damage parts

9. PROBLEM ROOT CAUSE — RC

- Unaware about the level of damage
- Proper information about the estimation cost
- Not proper maintenance and accident Something etc......

7. BEHAVIOUR — BE

The user does not know or is unaware how to deal with the damage part of vehicle

Inefficient to predict the level of damage and analysis to estimate the cost the damaged part this is The behavior

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

3. TRIGGERS — TR

- Seeking help to identify the level of the damaged part.

- To help people with good prediction and analysis of level and cost for the damaged part of the vehicle.

10. YOUR SOLUTION — SL

- Collecting damaged parts of vehicles and preprocessing them to identify the severity and the location of damage.

- Users will be able to identify the level of damage by the prediction system.

- Using VGG16 Model to train the damaged parts and trying to estimate the cost for the same.

8. CHANNELS of BEHAVIOR — CH

ONLINE
- Online Websites
- Social Media Platforms

OFFLINE
- Customer throw Words
- Anxiety and Unawareness

Identify strong TR & EM

4. EMOTIONS: BEFORE / AFTER — EM

- **Before:** Lack of proper knowledge ,Improper maintenance,Unawareness about the level of damage and difficulty in estimating the cost for damages.

- **After:** Good knowledge about the Vehicle parts,Efficient prediction system giving the level of damage and analyzing cost for the same.

4. REQUIREMENT ANALYSIS

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	Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Interface	User friendly and simple website
FR-4	Collect the datasets	Collect the data from the user side and their vehicle side information. Collect the data from about Insurance companies plans.
FR-5	Final Results	Model should be trained with high accuracy. Results obtained from the model should be displayed to The user with easy interpretability.

NON-FUNCTIONAL REQUIREMENTS

| NFR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Intelligent model used to assessment the damage in the vehicle and estimate the cost to be provided by the insurance company.
NFR-2	**Security**	The credibility of the user and the confidentiality of user details about their vehicle must be maintained.
NFR-3	**Reliability**	This scheme can achieve good accuracy in damage estimation and cost estimation, thus providing accurate and unbiased insurance coverage to the user.
NFR-4	**Performance**	Real-time images are to be captured and uploaded to the website, where the proposed model performs damage assessment and gives the insurance cost accordingly.
NFR-5	**Availability**	The website should be compatible with web browsers on both mobile phones and computers.
NFR-6	**Scalability**	The proposed solution will be scalable in the future due to efficient and rapid analysis and accurate cost Estimation

5. PROJECT DESIGN

DATA FLOW DIAGRAMS
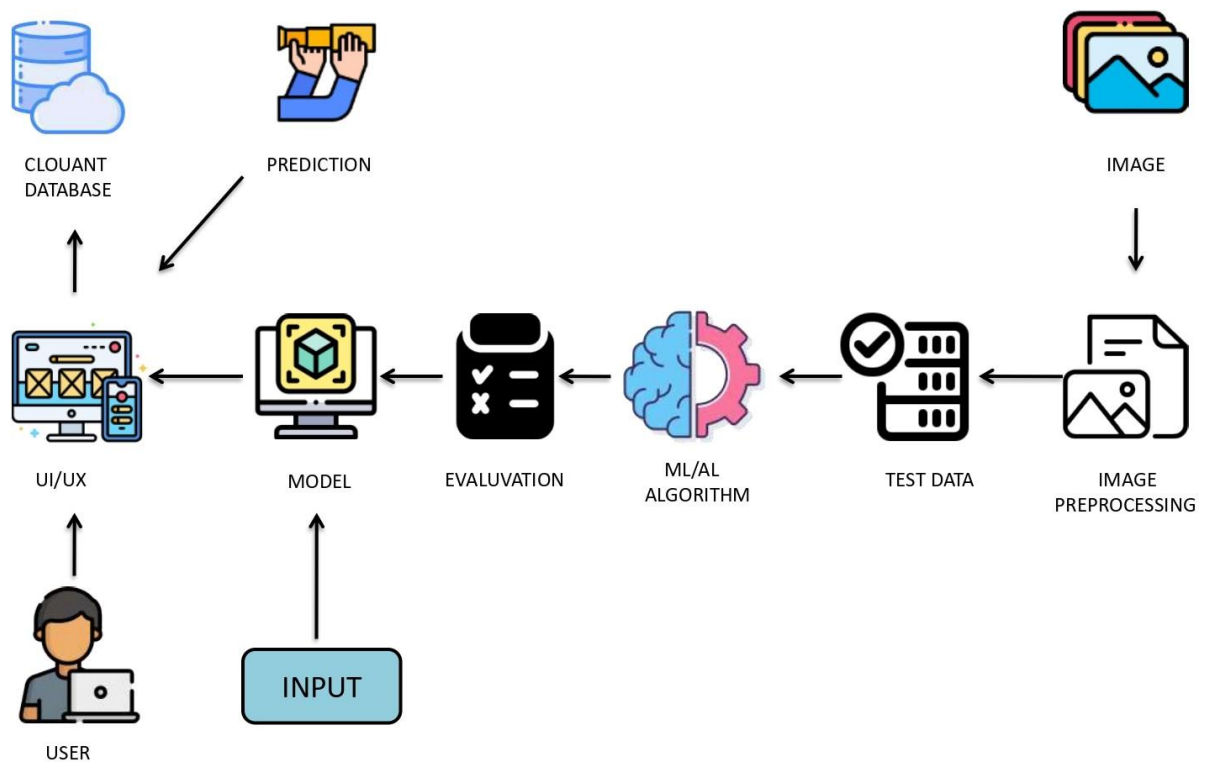
Figure 5.1.1 Data flow diagram
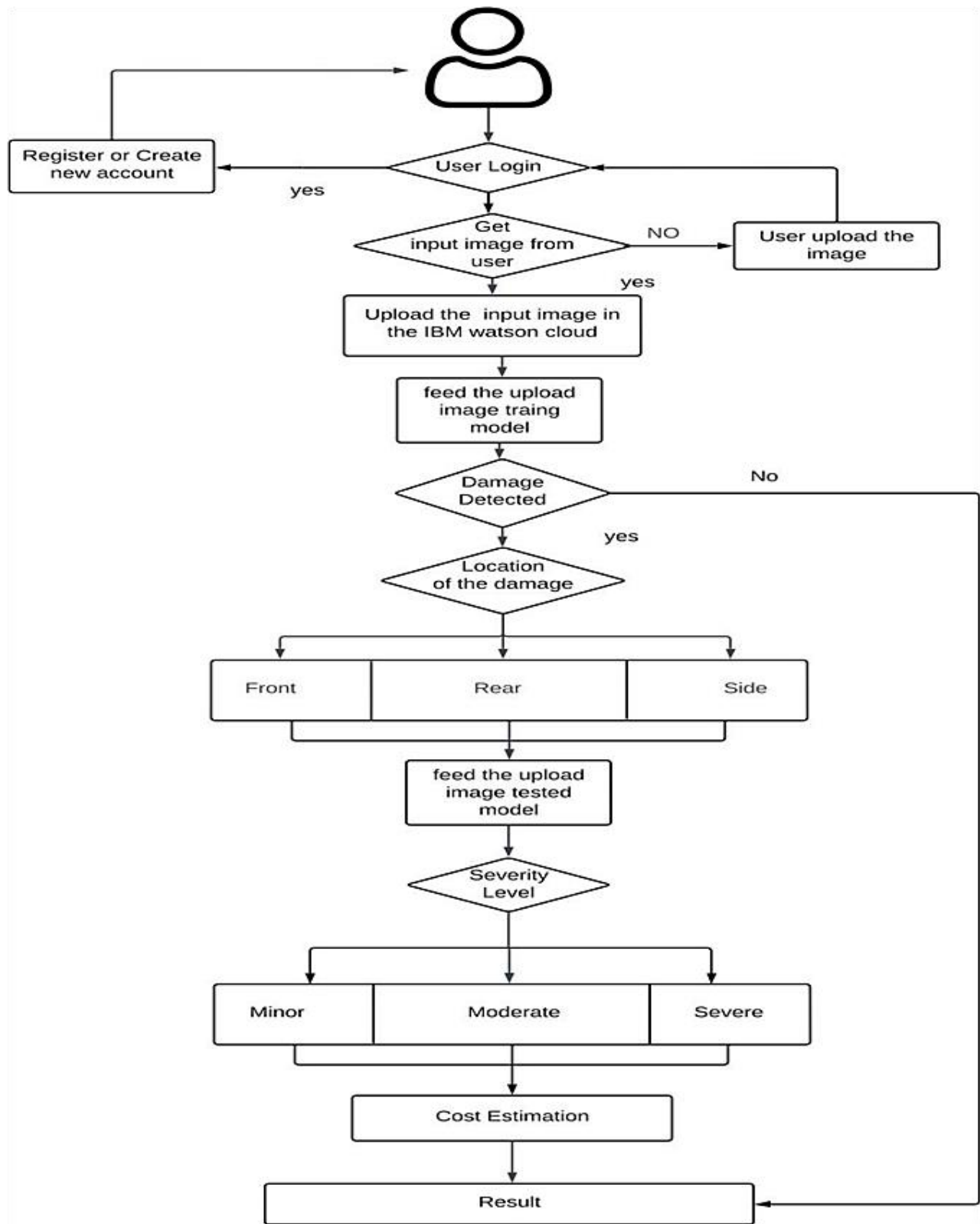
Figure 5.1.2 Data flow architecture diagram

SOLUTION & TECHNICAL ARCHITECTURE

Technical Architecture
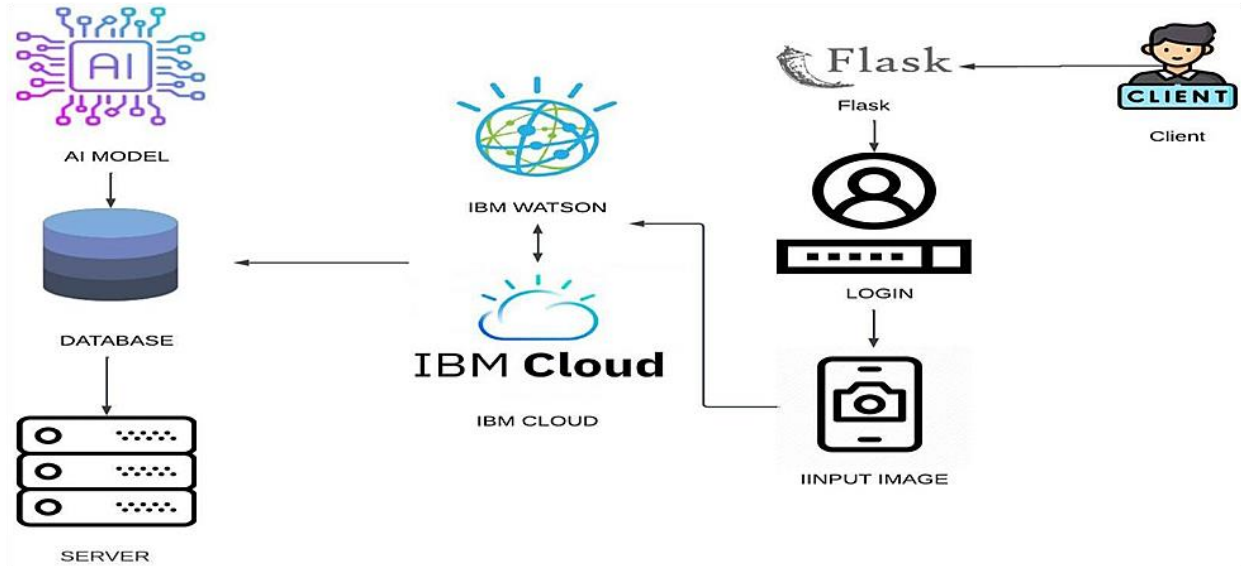
Figure 5.2.1 Technical Architecture

Table-1 : Components & Technologies

S.No	Component	Description	Technology
1	User Interface	The user interacts with the web application	HTML, CSS, Python
2	Application Logic-1	Gets the user input image	Python
3	Application Logic-2	Obtaining model output for damage prediction	IBM Watson , Python
4	Application Logic-3	Obtaining model output for cost	IBM Watson , Python

5	Database	Data Type – Details of images and user inputs is stored	MySQL, Js, IBM DB2
6	Cloud Database	A database service in the cloud	IBM DB2, IBM Cloud ant etc.
7	File Storage	User details and user input received The images of the vehicle are stored in the cloud	IBM Block Storage or Other Storage Service or Local File system
8	Machine Learning Model	The purpose of the AI model is to estimate cost of the damaged vehicle.	The purpose of the AI model is to estimate cost of the damaged vehicle.
9	Infrastructure (Server / Cloud)	Let's use the AI model on the cloud server Using Flask on a Web Page	Local, Cloud Foundry, Kubernetes, Python Flask etc.

Table-2: Application Characteristics

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	IBM Watson Opensource frameworks are used	Open source architecture technology IBM Watson
2.	Security Implementations	IBM Cloud	Certified WatsonAssistant Encrypted file systems are encrypted. Storage systems, key management systems.
3.	Scalable Architecture	Web Server - Static and dynamic website content Existing on the website will be updated accordingly User requestsand suggestions Application Server- Basic Upgrade Integrating websitefunctionality and updates The logic in the website can be done Database Server – Based on varying inputs The user supplied database will be modified Continuously	IBM Watson Assistant, Python,MySQL

4.	Availability	The AI misleadingly available to users anytime	IBM WatsonCloud assistance
5.	Performance	IBM Watson- Automated processes, deepThe learning model is trained using IBMWatson StudioFor better performance and faster access.	IBM WatsonAssistant

USER STORIES

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

UserType	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 mypassword.	I can access my account / dashboard by entering valid credentials	High	Sprint-1
Customer Details	Login	USN-2	As a user, I willreceive confirmation emailonce I have registered for the application	I canreceive confirmation email& click confirm	High	Sprint-1
Customer Uses	Dashboard	USN-3	As a user, I can register for the applicationthrough 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 applicationthrough Gmail	I can register & access the dashboard with FacebookGmail	Medium	Sprint-1
Customer usage	Login	USN-5	As a user, I can log into the application byentering email &password	I can log in and view my dashboard at my demandon any time	High	Sprint-1
Customer needsto 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 entirevehicle and upload	High	Sprint-2
Customer (Web user)	Details about estimated cost basedon damage	USN-7	As a user I must receive a detailed report ofthe damages present in the vehicle and thecost 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 andfailure of service provided	I can have smooth user experiences and all the issues raisedis sorted	Medium	Sprint-4
Administrator	Details about Estimated cost Based on damage	USN-9	We needto satisfy the customer needsin an efficient way and make sure any sort of errorsare fixed	I can finish the work without any problems	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

Title	Description	Date
Literature Survey and Information Gathering	Gathering Information by referring the technical papers, research publications etc	10 October 2022
Prepare Empathy Map	To capture user pain and gains Prepare List of Problem Statement	10 October 2022
Ideation	Prioritize a top 3 ideas based on feasibility and Importance	10 October 2022
Proposed Solution	Solution include novelty, feasibility, business model, social impact and scalability of solution	10 October r 2022

Problem Solution Fit	Solution fit document	11 October 2022
Solution Architecture	Solution Architecture	11 October 2022
Customer Journey	To Understand User Interactions and experiences with application	18 October 2022
Functional Requirement	Prepare functional Requirement	18 October 2022
Data flow Diagrams	Data flow diagram	18 October 2022
Technology stack	Technology Architecture Diagram	18 October 2022
Milestone & sprint delivery plan	Activity what we done &further plans	3 November 2022
Project Development-Delivery of sprint1,2,3&4	Develop and submit the developed code by testing It	3 November 2022 – 19November 2022

SPRINT PLANNING & ESTIMATION

Product Backlog, Sprint Schedule and Estimation

UserType	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 mypassword.	I can access my account / dashboard by entering valid credentials	High	Sprint-1
Customer Details	Login	USN-2	As a user, I willreceive confirmation emailonce I have registered for the application	I canreceive confirmation email& click confirm	High	Sprint-1
Customer Uses	Dashboard	USN-3	As a user, I can register for the applicationthrough 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 applicationthrough Gmail	I can register & access the dashboard with FacebookGmail	Medium	Sprint-1
Customer usage	Login	USN-5	As a user, I can log into the application byentering email &password	I can log in and view my dashboard at my demandon any time	High	Sprint-1
Customer needsto 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 entirevehicle and upload	High	Sprint-2
Customer (Web user)	Details about estimated cost basedon damage	USN-7	As a user I must receive a detailed report ofthe 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 andfailure of service provided	I can have smooth user experiences and all the issues raisedis sorted	Medium	Sprint-4
Administrator	Details about Estimated cost Based on Damage	USN-9	We needto satisfy the customer needsin an efficient way and make sure any sort of errorsare fixed	I can finish the work without any problems	High	Sprint-4

SPRINT DELIVERY SCHEDULE Project

Tracker, Velocity & Burn down Chart

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	3 Days	3 Nov 2022	6 Nov 2022	20	09/11/2022
Sprint-2	20	3 Days	6 Nov 2022	9 Nov 2022	20	19/11/2022
Sprint-3	20	3 Days	9 Nov 2022	12 Nov 2022	20	17/11/2022
Sprint-4	20	3 Days	12 Nov 2022	15 Nov 2022	20	20/11/2022

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).

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

Burn down 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 projectcontaining measurable progress over time.

7. CODING & SOLUTIONING

FEATURE 1

DASHBOARD

```html
<!DOCTYPE html>
<html lang="en">
<head>
   <title>DASHBOARD</title>
   <link rel="stylesheet" href="style.css">
      <meta charset="utf-8">
      <meta name="viewport" content="width=device-width, initial-scale=1"><link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
      <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
<style>*{
   margin: 0;
   padding: 0;}
.main{
```

```css
    width: 100%;

    background:linear-gradient(to top,
rgba(0,0,0,0.5)50%,rgba(0,0,0,0.5)50%), url("/static/images/2.jpg");

    background-position: center;

    background-size: cover;

    height: 100vh;

}

.navbar{

    width: 1200px;

    height: 75px;

    margin: auto;

}

.icon{

    width: 200px;

    float: left;

    height: 70px;

}

.logo{

    color: #ff7200;

    font-size: 35px;

    font-family: Arial;
```

```css
    padding-left: 20px;

    float: left;

    padding-top: 2px;

    margin-top: 5px

}

.menu{

    width: 700px;

    float: right;

    height: 70px;

}

ul{

    float: right;

    display: flex;

    justify-content: center;

    align-items: center;

}

ul li{

    list-style: none;

    margin-left: 62px;

    margin-top: 27px;
```

```css
    font-size: 14px;
}
ul li a{
    text-decoration: none;
    color: #fff;
    font-family: Arial;
    font-weight: bold;
    transition: 0.4s ease-in-out;
}
ul li a:hover{
    color: #ff7200;
}
.search{
    width: 330px;
    float: left;
    margin-left: 270px;
}
.srch{
    font-family: 'Times New Roman';
    width: 200px;
```

```css
    height:  40px;

    background: transparent;

    border: 1px solid #ff7200;

    margin-top: 13px;

    color: #fff;

    border-right: none;

    font-size: 16px;

    float:  left;

    padding: 10px;

    border-bottom-left-radius: 5px;

    border-top-left-radius: 5px;
}
.btn{

    width: 100px;

    height: 40px;

    background: #ff7200;

    border: 2px solid #ff7200;

    margin-top: 13px;

    color: #fff;

    font-size: 15px;
```

```
    border-bottom-right-radius: 5px;

    border-bottom-right-radius: 5px;

    transition: 0.2s ease;

    cursor: pointer;

}

.btn:hover{

    color: #000;

}


.btn:focus{

    outline: none;

}

.srch:focus{

    outline: none;

}

.content{

    width: 1200px;

    height: auto;

    margin: auto;

    color: #fff;
```

```css
    position: relative;

}

.content .par{

    padding-left: 20px;

    padding-bottom: 25px;

    font-family: Arial;

    letter-spacing: 1.2px;

    line-height: 30px;

}

.content h1{

    font-family: 'Times New Roman';

    font-size: 50px;

    padding-left: 20px;

    margin-top: 9%;

    letter-spacing: 2px;

}

.content .cn{

    width: 160px;

    height: 40px;

    background: #ff7200;
```

```css
    border: none;

    margin-bottom: 10px;

    margin-left: 20px;

    font-size: 18px;

    border-radius: 10px;

    cursor: pointer;

    transition: .4s ease;

}

.content .cn a{

    text-decoration: none;

    color: #000;

    transition: .3s ease;

}

.cn:hover{

    background-color: #fff;

}

.content span{

    color: #ff7200;

    font-size: 65px

}
```

```css
.form{

    width: 300px;

    height: 340px;

    background: linear-gradient(to top,
rgba(0,0,0,0.8)50%,rgba(0,0,0,0.8)50%);

    position: absolute;

    top: -20px;

    left: 870px;

    transform: translate(0%,-5%);

    border-radius: 10px;

    padding: 25px;

}
.form  h2{

    width: 220px;

    font-family: sans-serif;

    text-align: center;

    color: #ff7200;

    font-size: 22px;

    background-color: #fff;

    border-radius: 10px;

    margin: 2px;
```

```css
    padding: 8px;

}

.form input{

    width: 240px;

    height: 35px;

    background: transparent;

    border-bottom: 1px solid #ff7200;

    border-top: none;

    border-right: none;

    border-left: none;

    color: #fff;

    font-size: 15px;

    letter-spacing: 1px;

    margin-top: 30px;

    font-family: sans-serif;

}

.form input:focus{

    outline: none;

}

::placeholder{
```

```css
    color: #fff;

    font-family: Arial;

}

.btnn{

    width: 240px;

    height: 40px;

    background: #ff7200;

    border: none;

    margin-top: 30px;

    font-size: 18px;

    border-radius: 10px;

    cursor: pointer;

    color: #fff;

    transition: 0.4s ease;

}

.btnn:hover{

    background: #fff;

    color: #ff7200;

}

.btnn a{
```

```css
    text-decoration: none;

    color: #000;

    font-weight: bold;

}

.form .link{

    font-family: Arial, Helvetica, sans-serif;

    font-size: 17px;

    padding-top: 20px;

    text-align: center;

}

.form .link a{

    text-decoration: none;

    color: #ff7200;

}

.liw{

    padding-top: 15px;

    padding-bottom: 10px;

    text-align: center;

}

.icons a{
```

```css
    text-decoration: none;

    color: #fff;

}

.icons ion-icon{

    color: #fff;

    font-size: 30px;

    padding-left: 14px;

    padding-top: 5px;

    transition: 0.3s ease;

}

.icons ion-icon:hover{

    color: #ff7200;

}

.end {

            overflow: hidden;

            background-color: rgb(63, 63, 63);

            position: auto;

            bottom: 0;

            height: 55px;

            width: 100%;
```

```html
        }

</style>

</head>

<body = onload="flashMessage()">

  <script>

   function flashMessage(){

     if("{{flash_message}}" == "True"){

     alert("account created successfully")

     }

     if("{{flash_message}}" == "Fals"){

          alert("invalid credentials")

     }

     if("{{flash_message}}" == "Fal"){

      alert("Logged in successfully")

     }

    }

  </script>

  <body style:"background-image:static\images\2.jpg;">

  <div class="main">

    <div class="navbar">
```

```html
<div class="icon">

  <h6 class="logo"></h6>

</div>


<div class="menu">

  <ul>

    <li><a href="{{ url_for('dashboard') }}">HOME</a></li>

    <!--<li><a href="{{ url_for('login') }}">LOGIN</a></li>-->

    <!--<li><a href="{{ url_for('register') }}">REGISTER</a></li>-->

    <li><a href="{{ url_for('prediction') }}">PREDICTION</a></li>

    <li><a href="{{ url_for('logout') }}">LOGOUT</a></li>

    <li><a href="#section -1">ABOUT</a></li>

  </ul>

</div>

</div>

<div class="content">

<h1>Intelligent Vehicle <br><span>Damage Assessment &<br>Cost Estimator for </span><br>Insurance Companies</h1>

  <br><br><br>
```

```html
<button class="cn"><a href="{{ url_for('prediction')
}}">PREDICTION</a></button>

<!--form action="dashboard" method="POST">

<div class="form">

<input type="email" name="email" id="email"
placeholder="Enter Your Email Id">

<input type="password" name="password" id="password"
placeholder="Enter Your Password ">

<button class="btnn"><a href="">Login</a></button>

<p class="link">Don't have an account<br>

<a href="#">Sign up </a> here</a></p>

</div>

</form!-->

</div>

</div>

</div>

</div>

<br><br><br><center>

<div id= "section -1"> <h2 style:font-family:'Times New Roman',
Times, serif;"><center>ABOUT PROJECT</center></h2></div>

<br><br>

<p style:"font-size:50px;font-family: 'Times New Roman', Times,
```

serif;">Vehicle Damage Detection is used to reduce claims leakage during insurance processing.Vehicle Inspectin and validation are usually done.As it takes a long time,because a person need to come and Inspect Damage.Here we are trying to Automate the procedure .Using this Automation we can avoid time consumption for Insurance claim procedure</p>

 <a style:"color:white;

            padding: 22px;

            background-color: black;

            opacity: 85%;

            border-radius: 12px;

            margin-left: 25px;

            margin-bottom: -20px;"
href="https://www.facebook.com" class="fa fa-facebook">

        <a style:"color:white;

            padding: 20px;

            background-color: black;

            opacity: 85%;

            border-radius: 12px;

            margin-left: 25px;

            margin-bottom: -20px;" href="https://www.twitter.com"
class="fa fa-twitter">

```html
<a style:"color:white;

        padding: 20px;

        background-color: black;

        opacity: 85%;

        border-radius: 12px;

        margin-left: 25px;

        margin-bottom: -20px;" href="https://www.linkedin.com"
class="fa fa-linkedin"></a>

    <a style:"color:white;

        padding: 20px;

        background-color: black;

        opacity: 85%;

        border-radius: 12px;

        margin-left: 25px;

        margin-bottom: -20px;"
href="https://www.instagram.com" class="fa fa-instagram"></a>

  </center>

            </div><br><br><br>

    <div class="end">

      <p style="color:rgb(255, 246, 246);

          margin-top: 20px;
```

```
                    text-align: center;">

          <b> Copyright &#169;  2022. All Rights Reserved</b>

            </p>

    <script src="https://unpkg.com/ionicons@5.4.0/dist/ionicons.js">

</script>

</body>

    </html>
```

FEATURE 2

PREDICTION

```html
<!DOCTYPE html>

<html lang="en">

    <head>

        <title> PREDICTION </title>

        <meta charset="utf-8">

        <meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"> <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

        <style>

            body{

                margin: 0;

                padding: 0;

                font-family:Arial, Helvetica, sans-serif

            }

            nav{

                position:relative;

                top: 0;
```

```css
        left: 0;

        width: 100%;

        height: 70px;

        padding: 10px 100px;

        box-sizing:border-box;

    }
    nav .logo{

        padding: 15px;

        height: 30px;

        float: left;

        font-size: 25px;

        font-weight: bold;

        color: #fff;

    }
    nav ul {

        list-style:none;

        float: right;

        margin: 0;

        padding: 0;

        display: flex;

        font-weight: 600;

    }
```

```css
nav ul li a{

    float: right;

    display: block;

    color: #f2f2f2;

    text-align: center;

    padding: 15px;

    text-decoration: none;

    font-size: 17px;

}
nav ul li a:hover{

    background: rgb(200, 212, 200);

    border-radius: 6px;

    color: rgb(51, 48, 47);

}
nav ul li a.active{

    background: #e2472f;

    border-radius: 6px;


}
body {

 margin:0;

 padding:0;
```

```css
        font-family: sans-serif;

        background: linear-gradient(to top,

rgba(0,0,0,0.5)50%,rgba(0,0,0,0.5)50%),url("/static/images/2.jpg");

        background-position: center;
        background-size: cover;

         height: 100vh;

}

        .end {

            overflow: hidden;

            background-color: rgb(63, 63, 63);

            position: fixed;

            bottom: 0;

            height: 55px;

            width: 100%;

    }

    button{

                background-color:black;

                opacity: 78%;

                color: rgb(255, 255, 255);

                font-size: 20px;

                border-radius: 50px;

        }
                width: 150px;"
    input{
```

```
            background-color:black;

            opacity: 78%;

            color: white;

            font-size: 15px;

            width: 250px;

             }

    </style>

     </head>

     <body style="background image=/static/images/2.jpg;">

        <div class="wrap">

        <nav>

        <div class="logo">PREDICTION  PAGE</div>

        <ul>

           <li><a href="{{ url_for('dashboard') }}">Home</a></li>

           <li><a href="{{ url_for('logout') }}">Logout </a></li>

        </ul>

        </div>

        </nav><br><br><br><br><br><br><br><br><br><br>

      <div style="margin-top: -45px;" class="container">

       <center>

          <h2 style="font-family:'Franklin Gothic Medium', 'Arial Narrow',
```

```html
Arial, sans-serif; color: #ff7200;"><b>UPLOAD IMAGE TO
PREDICT</b></h2><br>

    <form action="{{url_for('prediction')}}" method="POST"
enctype="multipart/form-data">

      <div class="input">

              <input type="file" class="myFile" id="myFile"
name="myFile">

              </div>

      <br><br><br>

    <div class="button">

        <button input type="submit" >Submit</button></div>

   </form>

         <br>

    <!-- <script>

     function flashMessage(){

       if("{{ flash_message }}"=='True'){

         const im = document.createElement('img');

         im.src = "{{url_for('static', filename='imagedata/save.png')}}";

         im.height = "200px";

         im.width = '200px';

         im.alt = 'hello world'
```

```
        }

      }

    </script> -->

  <!-- <img src="{{url_for('static', filename='imagedata/save.png')}}"
alt=""

   height="200px"

     width="200px"   /> --><br><h2 style="font-family:Arial, Helvetica,
sans-serif; color: #ff7200;" > <b> The Estimated cost for the Damage is
= {{prediction_text}}</b></h2>

        </h3>

    </center>

</div>

</body>

</html>
```

8. TESTING

TEST CASES

Test case ID	Feature Type	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
						Date 17-Nov-22							
						Team ID PNT2022TMID44408							
						Project Name Intelligent Vehicle Damage Asse							
						Maximum Marks 4 marks							
Landing Page_TC_001	Functional	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 /drive/folders/1gWZh1E MuPVAsWV7N2Hkeb3 F-MPhk0ueX	Predicted result popup should display.	Working as expected	Pass	.	.	.	.
Landing Page_TC_002	UI	Landing Page	Verify the UI elements in Login/Signup popup	.	1.Sliding Banner 2.Buttons	https://drive.google.com /drive/folders/1gWZh1E MuPVAsWV7N2Hkeb3 F-MPhk0ueX	Application should show below UI elements: a.choose file box b.Predict button box	Working as expected	Pass	.	.	.	.

Drive Link:
https://drive.google.com/drive/folders/1gWZh1EMuPVAsWV7N2H keb3F-MPhk0ueX

S.NO.	PARAMETER	VALUES	SCREENSHOT
1.	Model Summary		
2.	Accuracy	Training Accuracy - 97.51% Validati on Accuracy - 70.42%	

USER ACCEPTANCE TESTING

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	Sub total
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	0	14	13	26	77

Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. RESULTS

PERFORMANCE METRICS

User will upload a car image or images and number plate of car, and we extract car registration number in text form using Googles Tesseract OCR. The performance metrics

	Performances of damage detection			Performances of damage location			Performance of damage severity		
Pre-trained VGG models	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
VGG16	**0.94**	**0.94**	**0.94**	**0.71**	0.69	0.69	**0.61**	0.55	0.53
VGG19	0.91	0.91	0.91	0.69	0.66	0.66	0.59	0.54	0.51

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES

✓ The advancement of image analysis and pattern recognition technologies, the auto insurance industry could significantly benefit.

✓ Our proposed system is that it was user friendly and highly efficient.

✓ The proposed system maintain privacy and also Working accuracy.

✓ It saves user time and energy.

✓ Images with various orientation of vehicle will also be analysed

DISADVANTAGES

- The Major Drawback Of The Proposed Model Is That It Only Identifies The Physical Visible Damage And Not Of The Internal Or The Interior Damage.

- Blurred image or image with poor lighting will not yield accurate results.

- Application file size is large, leading to long computation time.

11. CONCLUSION

- In this work of Damage analysis of a vehicle in general and insurance reclaim, a system has been designed using CNN and image classification which takes the input from a user as an image to test the severity of damage, which happens in a sequence of two steps.

- First being the image classification, here the input provided by the user is processed by the neural network to identify the car that is if the car is damaged or not.

- This software as a service (SaaS)will be a stepping stone to establish and server to client authenticated interaction to yield accurate results.Data security plays a vital role in the information era,aligning with CIA principle of cryptography user data is stored securely in cloud.

12. FUTURE SCOPE

◊ This further research is implementing the proposed algorithm with the existing public datasets. Also, various segmentation algorithms can be implemented to improve accuracy.

◊ A higher quality dataset which includes pivotal parameters like location information and repair costs, the research could go a step further in predicting the cost of damage repair based on the image.

13. APPENDIX

SOURCE CODE

```python
from flask import Flask, app, request, render_template
import os
import flask
import re
import flask_login
import base64
from PIL import Image
from io import BytesIO
import date  time
import cv2
import numpy as np
from tensorflow.keras.models import load_model
from cloudant.client import Cloudant
from cloudant.error import CloudantException
from cloudant.result import Result, ResultByKey
model1 = load_model('Model/level.h5')
model2 = load_model('Model/body.h5')
def detect(frame,model1,f):
```

```python
    img = cv2.resize(frame,(244,244))

    img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)

    if(np.max(img)>1):

        img=img/255.0

    img = np.array([img])

    prediction = model1.predict(img)

    if(f):

        label= ['front','rear','side']

    else:

        label =['minor','moderate','severe']

    preds = label[np.argmax(prediction)]

    return preds

client = Cloudant.iam('862d725c-4fb3-4619-bfcb-083c11c6a979-
bluemix','QM5pS9ePdxMpe6Lh-
8yIvNYoZ3SXtoIdQQKnyIRYlwFb',connect=True)

name = 'name'

email = 'a@b.c'

password = '123'

user_database = client.create_database('user_database')

user_image_database = client.create_database('user_image_database')

def image_database_updation(name,email,imagestr):
```

```python
    global user_image_database

    now = datetime.datetime.now()

    json_image_document={

        'name':name,

        'email':email,

        'image':imagestr,

        'datetime':now.strftime("%m/%d/%Y, %H:%M:%S")

    }

    new_image_document =
user_image_database.create_document(json_image_document)

    if(new_image_document.exists()):

        print('database updated')

    else:

        print('database couldn\'t be edited')

    return

def image_database_retrieval():

    global user_image_database

    image_result_retrieved =
Result(user_image_database.all_docs,include_docs=True)

    image_result ={}

    for i in image_result_retrieved:
```

```python
        if(i['doc']['email'] in image_result.keys()):

            # like current date> rx date('str')

            n = datetime.datetime.strptime(i['doc']['datetime'],'%m/%d/%Y,
%H:%M:%S')

            o =
datetime.datetime.strptime(image_result[i['doc']['email']]['date'],'%m/%d
/%Y, %H:%M:%S')

            if(n>o):

                image_result[i['doc']['email']] =
{'name':i['doc']['name'],'image':i['doc']['image'],'date':i['doc']['datetime']}

        else:

            image_result[i['doc']['email']] =
{'name':i['doc']['name'],'image':i['doc']['image'],'date':i['doc']['datetime']}

    return(image_result)

def database_updation(name,email,password):

    global user_database

    jsonDocument = {

        'name':name,

        'email':email,

        'password':password

    }

    newDocument = user_database.create_document(jsonDocument)
```

```python
    if(newDocument.exists()):

        print('database updated')

    else:

        print('database couldn\'t be edited')

    return

#database_updation(name,email,password)


def database_retrieval():

    global user_database

    result_retrieved = Result(user_database.all_docs,include_docs=True)

    #print(list(result_retrieved))

    result = {}

    for i in list(result_retrieved):

result[i['doc']['email']]={'name':i['doc']['name'],'password':i['doc']['password']}

    return result

#print(database_retrieval())

app = Flask(__name__)

app.secret_key = 'apple'

login_manager = flask_login.LoginManager()
```

```python
login_manager.init_app(app)

users = {'a@b.c': {'password': '123'}}

class User(flask_login.UserMixin):
    pass

@login_manager.user_loader
def user_loader(email):
    data = database_retrieval()
    if email not in data:
        return
    user = User()
    user.id = email
    user.name = data[email]['name']
    return user

@login_manager.request_loader
def request_loader(request):
    email = request.form.get('email')
    data = database_retrieval()
    if email not in data:
        return
    user = User()
```

```python
    user.id = email

    user.name = data[email]['name']

    return user

@app.route('/')

def index():

    if(flask_login.current_user.is_authenticated):

        return render_template('dashboard.html')

    else:

        return flask.redirect(flask.url_for('login'))

@app.route('/register',methods = ['GET','POST'])

def register():

    data = database_retrieval()

    if(flask.request.method == 'GET'):

        return render_template('register.html')

    email = flask.request.form['email']

    if(email in data):

        return render_template('register.html',flash_message='True')

    else:

database_updation(flask.request.form['name'],email,flask.request.form['p
assword'])
```

```python
        #users[email]={'password':flask.request.form['password']}
        user  = User()
        user.id = email
        user.name = flask.request.form['name']
        flask_login.login_user(user)
        return render_template('dashboard.html',flash_message='True')
@app.route('/login',methods =['GET','POST'])
def login():
    data = database_retrieval()
    if(flask.request.method == 'GET'):
            return render_template('login.html',flash_message='False')
    email = flask.request.form['email']
    if(email in data and
flask.request.form['password']==data[email]['password']):
        user  = User()
        user.id = email
        flask_login.login_user(user)
        return render_template('dashboard.html',flash_message='Fal')
    #flask.flash('invalid credentials !!!')
    return render_template('login.html',flash_message="True")
    #error = 'inavlid credentials')
```

```python
@app.route('/dashboard',methods = ['GET','POST'])
@flask_login.login_required
def dashboard():
    if(flask.request.method == 'GET'):
        return render_template('dashboard.html',flash_message='False')
    email = flask.request.form['email']
    if(email in users and
flask.request.form['password']==users[email]['password']):
        user  = User()
        user.id = email
        flask_login.login_user(user)
        return render_template('dashboard.html',flash_message="Fal")
    return render_template('dashboard.html',flash_message="Fals")


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


@app.route('/prediction',methods = ['GET','POST'])
```

```python
@flask_login.login_required

def prediction():

    from tensorflow.keras.models import load_model

    model1 = load_model('Model/level.h5')

    model2 = load_model('Model/body.h5')

    if(flask.request.method=='POST'):

        img = flask.request.files['myFile']

        try:

            os.remove('static\imagedata\save.png')

        except:

            pass

        imgstr = base64.b64encode(img.read()).decode('utf-8')


image_database_updation(flask_login.current_user.name,flask_login.current_user.id,imgstr)

        data = image_database_retrieval()

        print(flask_login.current_user.id)


#print(len(base64.b64decode(data[flask_login.current_user.id]['image'].strip())))

        image =
Image.open(BytesIO(base64.b64decode(data[flask_login.current_user.id
```

```python
        ]['image'])))

        img_retrived = np.array(image)

        '''img_retrived =
np.asarray(base64.b64decode(data[flask_login.current_user.id]['image']
))

        print(data[flask_login.current_user.id]['image'])

        print(img_retrived.shape)'''

        #img_retrived = np.resize(img_retrived,(244,244))

        img_retrive = Image.fromarray(img_retrived)

        img_retrive.save('static\imagedata\save.png')

        '''img_retrived = np.frombuffer(

            BytesIO(

                base64.b64decode(data[flask_login.current_user.id]['image'])

            ) )'''

        print('###############################')

        result1=detect(img_retrived,model1=model2,f=True)

        result2 = detect(img_retrived,model1=model1,f=False)

        value='' if(result1 == 'front' and result2 == 'minor'):

            value = '3000 - 5000 INR'

        elif(result1 == 'front' and result2 == 'moderate'):

            value = '6000 - 8000 INR'
```

```python
    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 = '900 - 11000 INR'

    elif(result1 == 'side' and result2 == 'severe'):

        value = '12000 - 15000 INR'

    else:

        value = '16000 - 50000 INR'

    print(result1,result2,value)

    print('################################')

    img_retrived = Image.fromarray(img_retrived)

    img_retrived.save('static\imagedata\save.png')
```

```
    print('image uploaded and retrieved')

    return
render_template('prediction.html',prediction_text='{}'.format(value),flas
h_message='False')


  return render_template('prediction.html',flash_message='True')

if__name__== '__main__':

  app.run(debug=True)
```

GITHUB & PROJECT DEMO LINK

Github Link _____
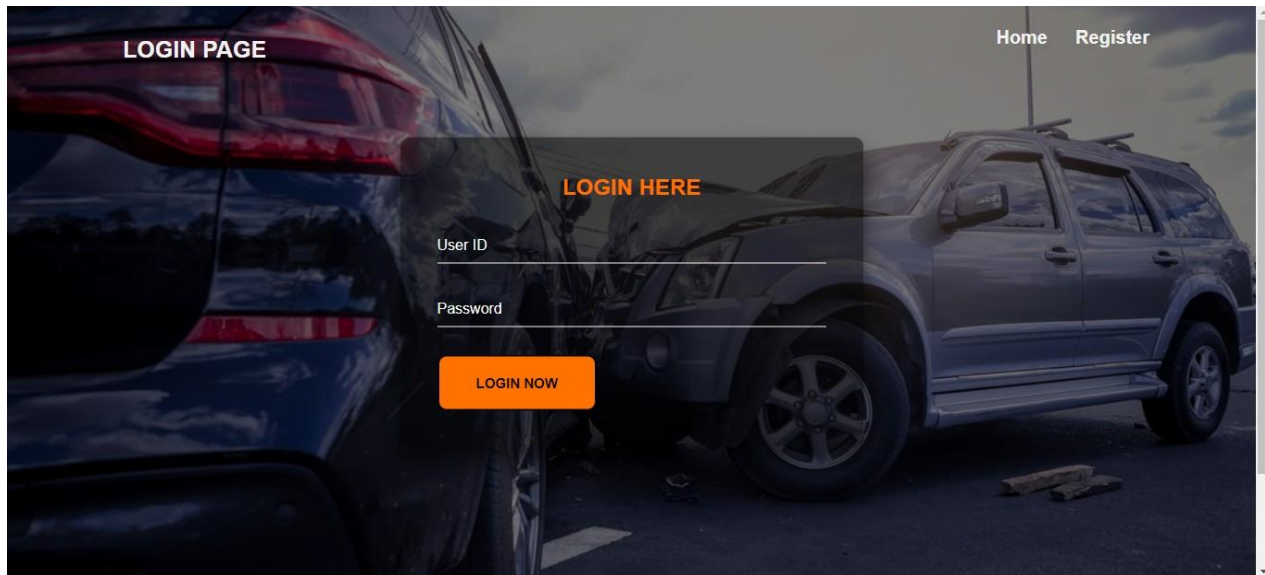
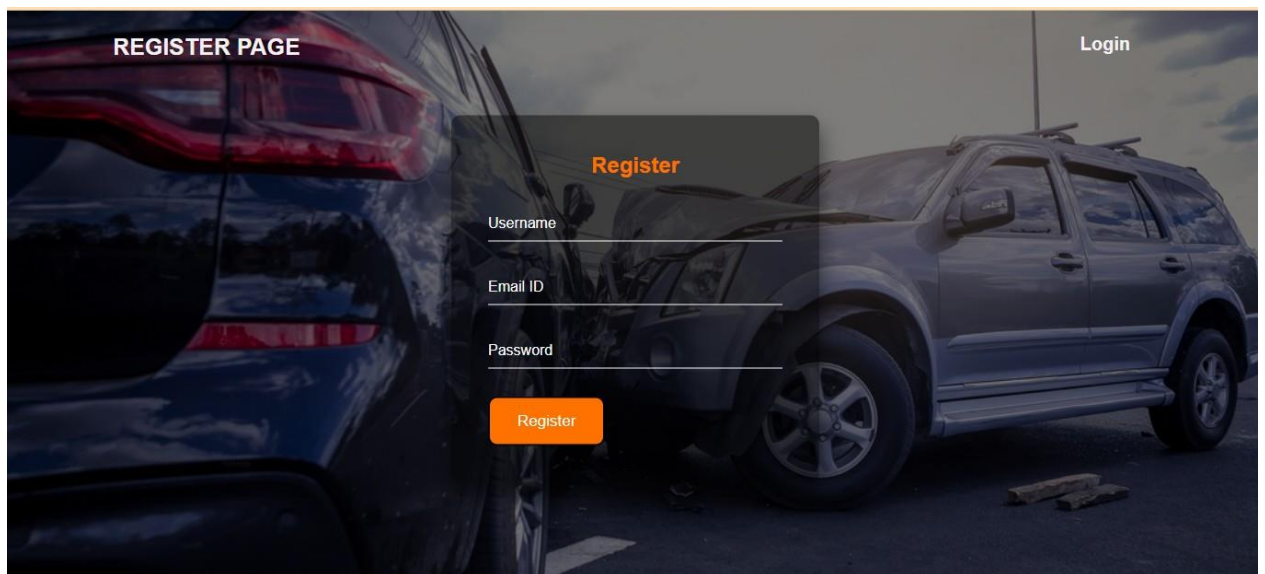**https://github.com/IBM-EPBL/IBM-Project-14442-1659585720**

Project demo link

https://drive.google.com/drive/folders/1Bj76pege0zOOQPFtNe52V mGUCuPbrOjQ

14. SCREENSHOTS

Login page

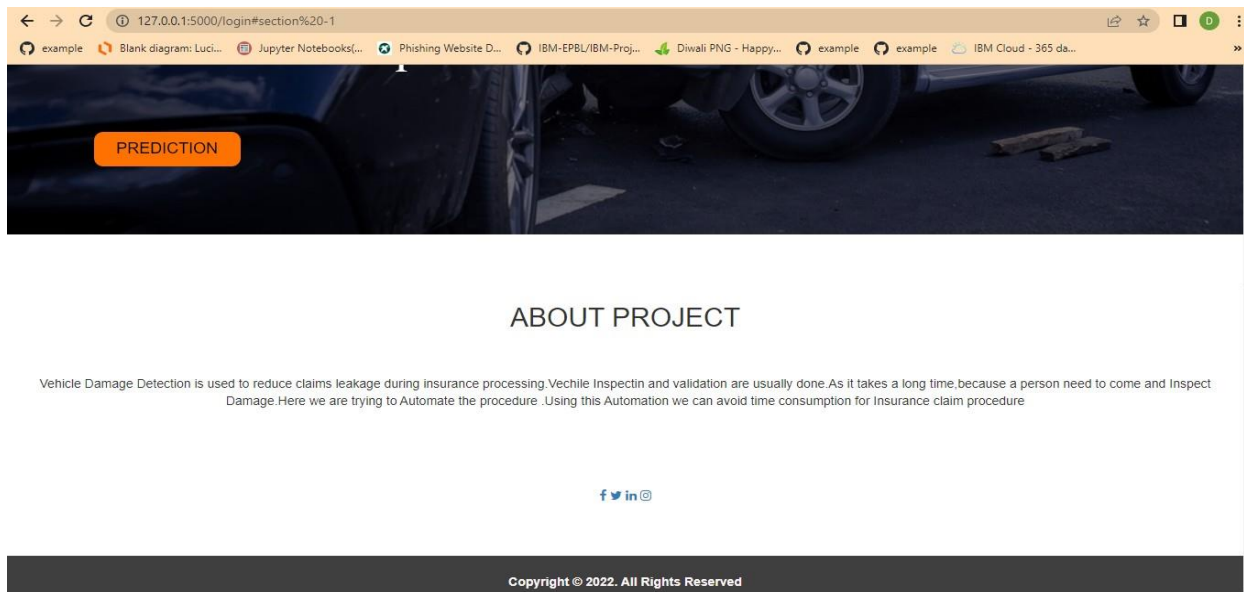**Figure 14.1** Login page

Register page

Figure 14.2 Register page

Dashboard
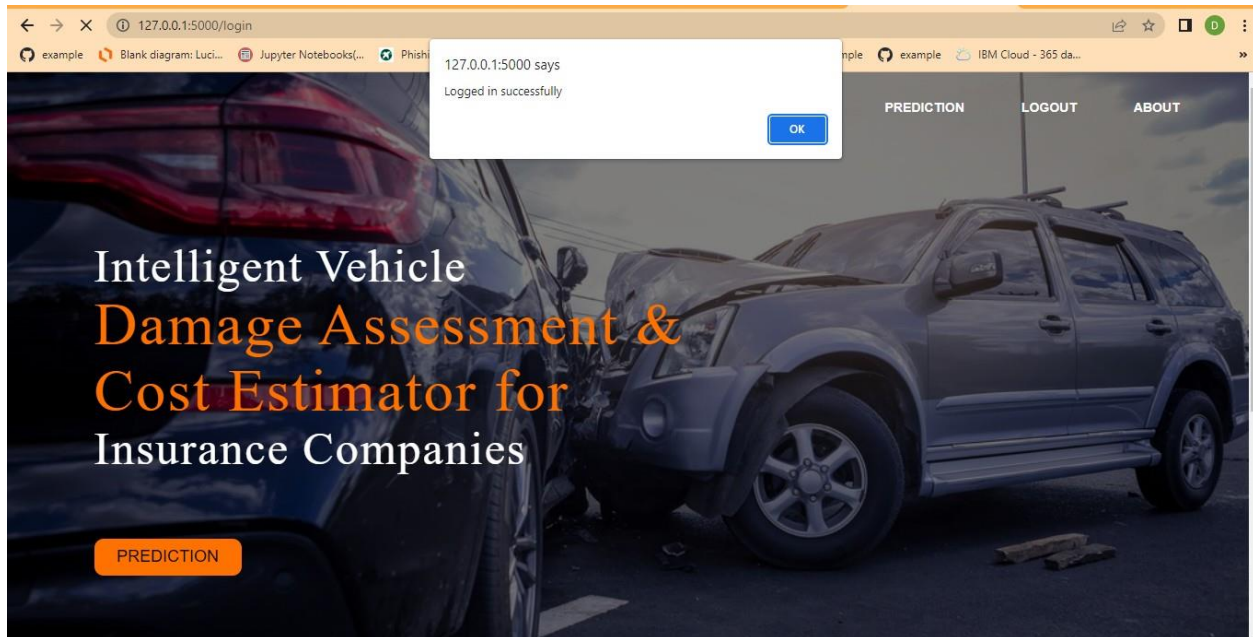
Figure 14.3 Dashboard

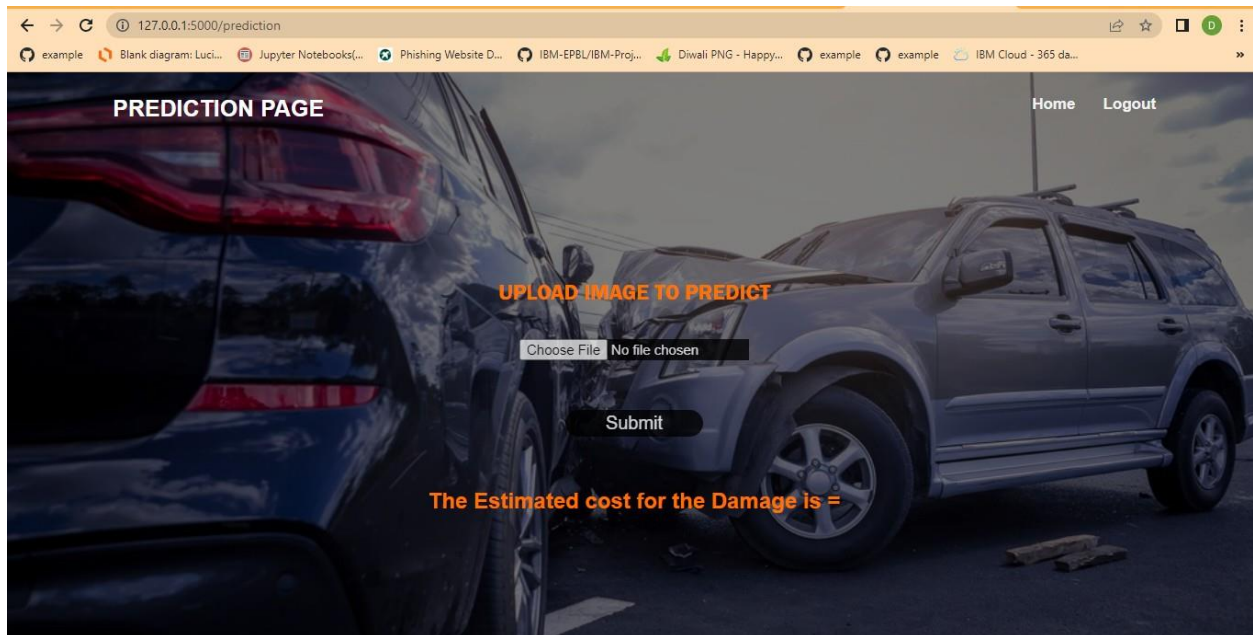# About page

Figure 14.4 About page

Prediction page

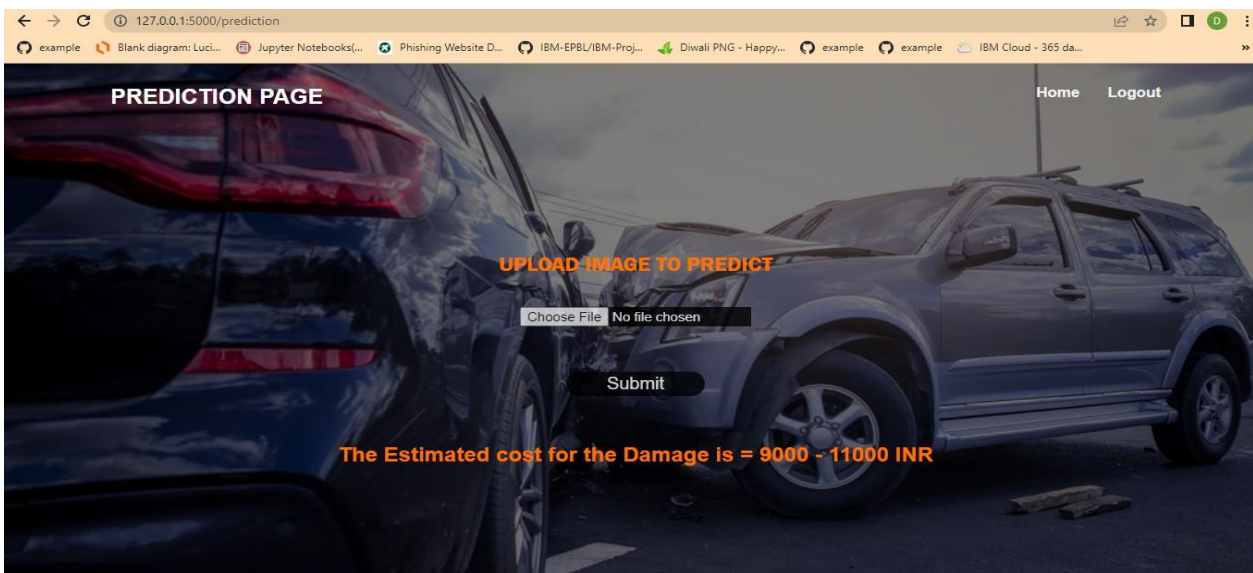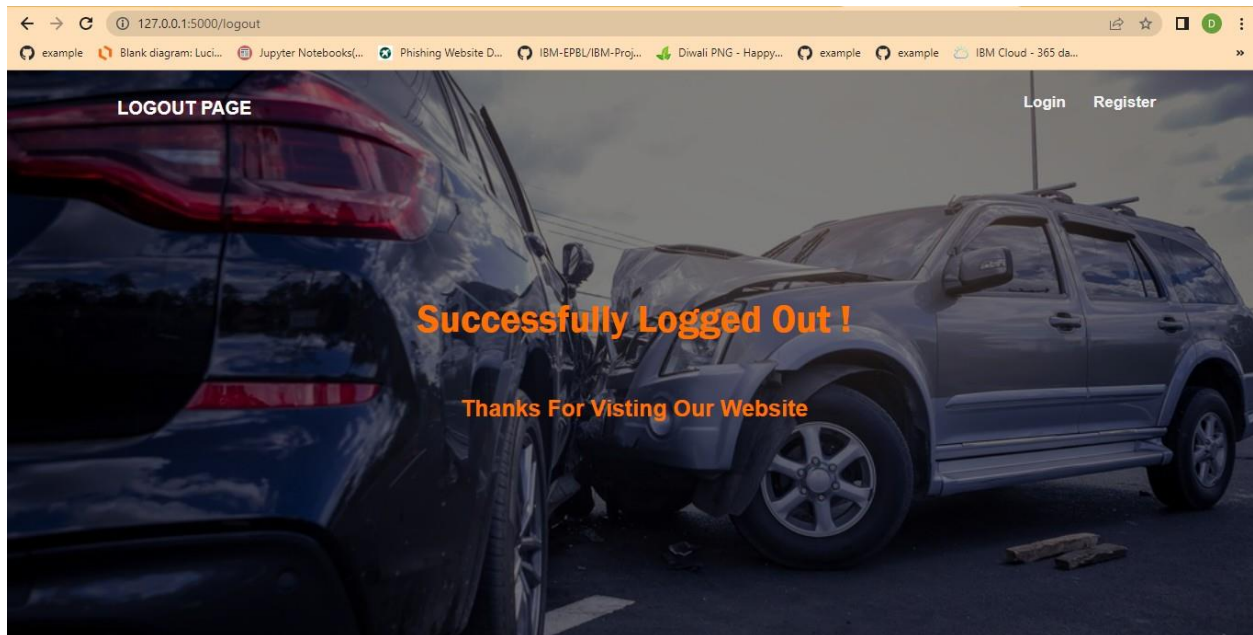**Figure 14.5** Prediction page

Estimated cost

Figure 14.6 Estimated cost

Log out page

84

Figure 14.7 Log out page