INTELLIGENT VEHICLE DAMAGE ASSESSMENT AND COST ESTIMATOR FOR INSURANCE COMPANIES

IBM PROJECT REPORT

SUBMITTED BY

PRINCIYA.V	- 911219104010
BOOMIKA.S	- 911219104004
VIJAYALAKSHMI	- 911219104016
NIVETHA.A	-911219104007

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

MAHATH AMMA INSTITUTE OF ENGINEERING AND TECHNOLOGY

. ARIYUR,PUDUKOTTAI

ANNA UNIVERSITY::CHENNAI 600 025

JUNE 2022

CERTIFICATE OF EVALUATION

COLLEGE NAME : MAHATH AMMA INSTITUTE OF ENGINEERING AND TECHNOLOGY

BRANCH : COMPUTER SCIENCE AND ENGINEERING

SEMESTER : VII

TITLE : INTELLIGENT VEHICLE DAMAGE ASSESSMENT AND COST ESTIMATOR FOR INSURANCE COMPANIES

TEAM ID : PNT2022TMID47627

STUDENT NAMES	**REGISTRATION**	**SUPERVISOR**
NUMBER		
PRINCIYA.V	- 911219104010	Mr. ALAGUSUNDARAM P
BOOMIKA.S	- 911219104004	
VIJAYALAKSHMI	- 911219104016	
NIVETHA.A	-911219104007	

The report of this project is submitted by the above students in partial fulfillment for the award of Bachelor of Engineering Degree, in Computer Science and Engineering of Anna University are evaluated and confirmed to the reports of the work done by the above students.

MENTOR **EVALUATOR**

1.INTRODUCTION

1.1 Project Overview

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

1.2 Purpose

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

2. LITERATURE SURVEY

2.1 Existing problem

2.1.1. TITLE: Convolutional Neural Networks for vehicle damage detection, 2021

AUTHOR NAME: R.E. Ruitenbeek Vehicle damage is becoming an increasing liability for shared mobility providers. The high number of driver handovers necessitates the use of an accurate and quick inspection system capable of detecting minor damage and categorising it. To address this, a damage detection model is created that locates vehicle damages and categorises them into twelve groups. To improve detection performance, multiple deep learning algorithms are used, and the effect of various transfer learning and training strategies is evaluated. The final model, which was trained on over 10,000 damage photos, can detect minor defects in a variety of environments, including water and dirt. A performance evaluation using domain experts reveals that the model performs comparably. Furthermore, the model is tested in a specially designed light street, demonstrating how strong reflections complicate detection performance.

2.1.2. TITLE: Deep Learning Based Car Damage Detection, Classification and Severity

AUTHOR NAME: Ritik Gandhi1, 2021 Because it is a manual procedure, resolving a claim in the accident insurance sector takes time, and there is a gap between the ideal and real settlement. We are using deep learning models to not only speed up the process, but also to deliver better customer service and boost insurance company profitability. In this paper, we use multiple pre trained models such as VGG 16, VGG 19, Resnet50, and DENSENET to choose the top performing models. We first use the Resnet50 model to determine whether or not the automobile is damaged, and if it is, we utilise the WPOD-net model to identify the licence plate. The YOLO model is used to detect the affected region. Finally, the damage severity is implemented using the DENSENET model. We discovered that transfer learning outperforms fine-tuning after applying multiple models. Furthermore, we present a framework that incorporates all of this into a single application, assisting in the automation of the insurance sector.

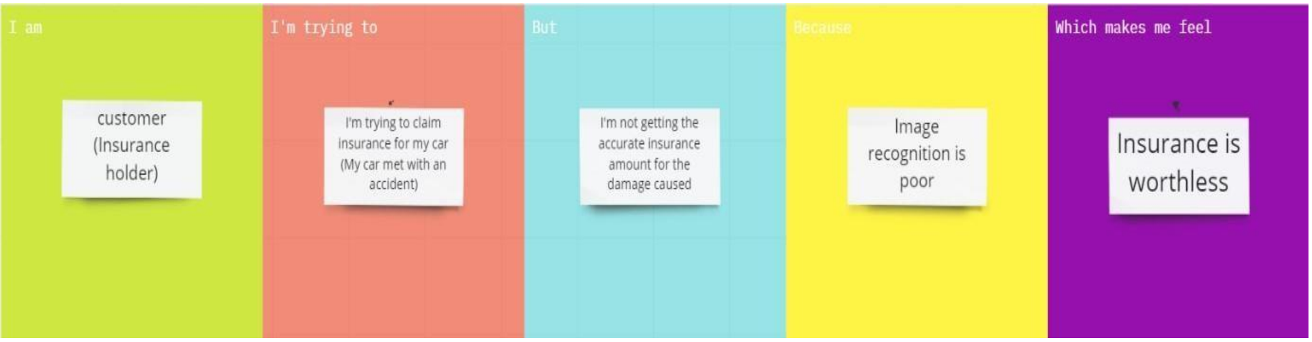
2.2 References

[1]. R.E. Ruitenbeek, Convolutional Neural Networks for vehicle damage detection, 2021

[2]. Ritik Gandhi1Deep Learning Based Car Damage Detection, Classification and Severity, 2021

2.3Problem Statement Definition

Nowadays, insurance companies use AI to analyze the image and to estimate the cost. Let us consider a customer who was met with an accident, and his car has been damaged and he tries to claim insurance from the insurance company in which he holds insurance for his car, He took picture of his car and posts it on the company’s site, the site analyses the picture and estimates the cost for the insurance (The amount to be claimed from the insurance company by the customer), Here is the problem, but the estimated amount is not accurate, the amount is too low or high, this may cause loss to either the customer or the insurance company **Example:**



I am

customer (Insurance holder)

I'm trying to

I'm trying to claim insurance for my car (My car met with an accident)

But

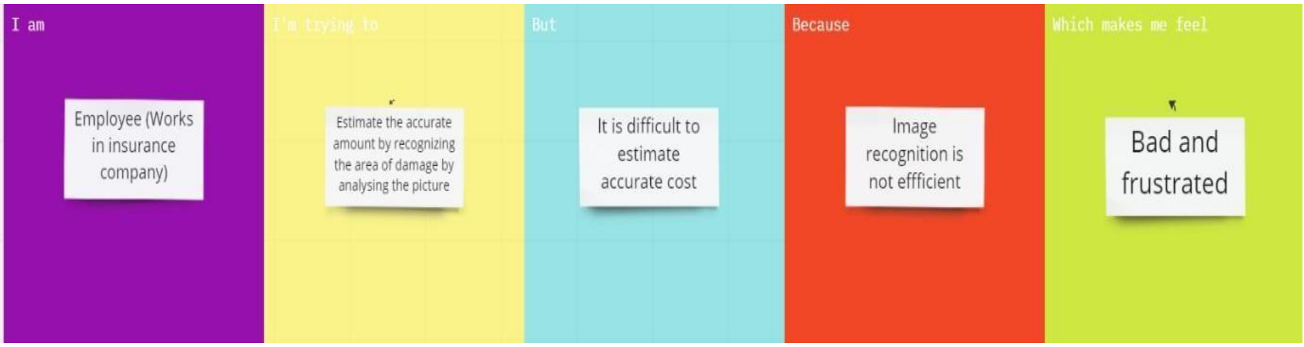
I'm not getting the accurate insurance amount for the damage caused

Because

Image recognition is poor

Which makes me feel

Insurance is worthless



I am

Employee (Works in insurance company)

I'm trying to

Estimate the accurate amount by recognizing the area of damage by analysing the picture

But

It is difficult to estimate accurate cost

Because

Image recognition is not effficient

Which makes me feel

Bad and frustrated

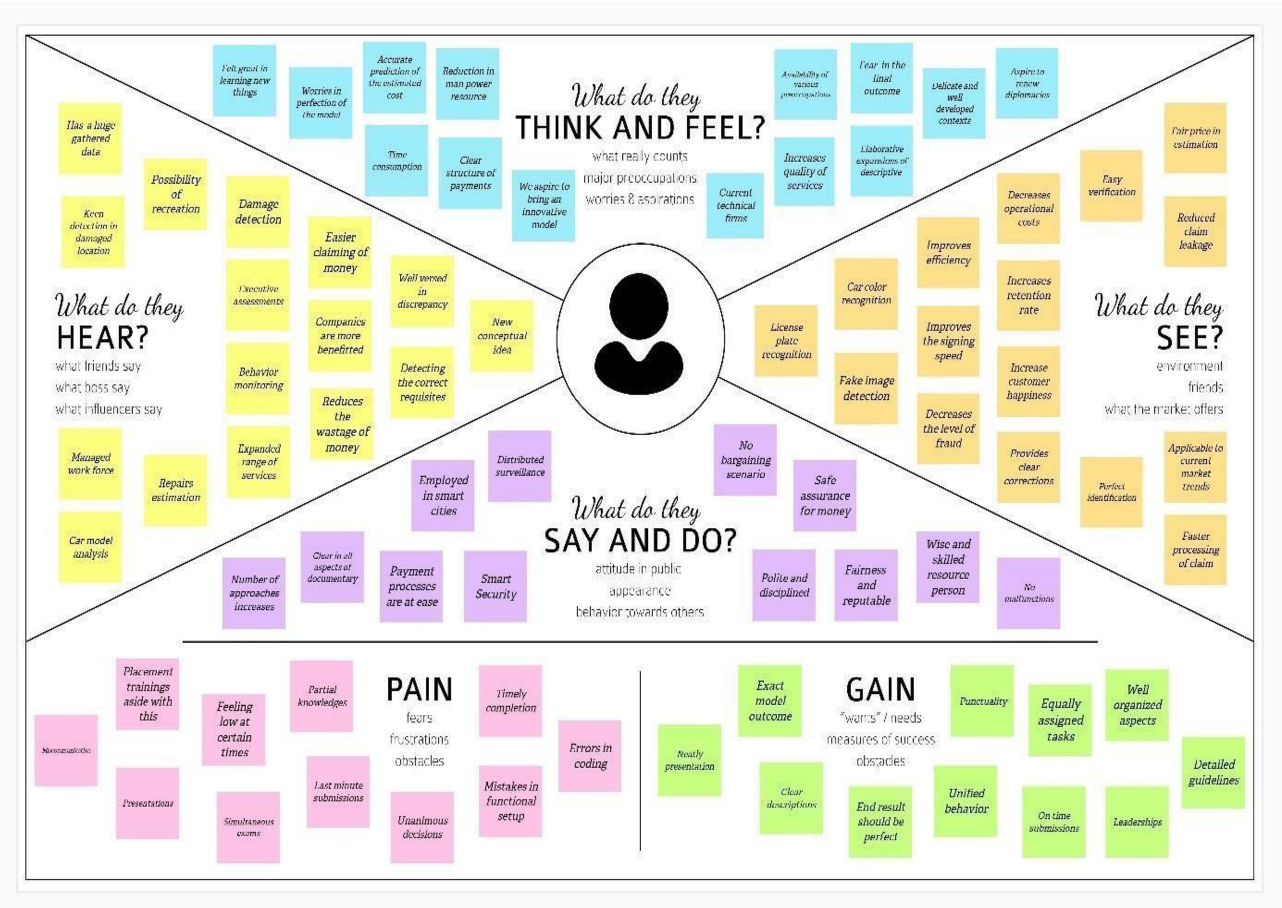
Problem Statement (PS)	I am	I’m trying to	But	Because	Which makes me feel
PS-1	Customer (Insurance holder)	To claim insurance for my car	I’m not getting the accurate amount for the damage caused	Image recognition is poor	Insurance is worthless
PS-2	Employee (insurance company)	Estimate the damage by analyzing the given image and	It’s difficult to estimate the cost with the picture	Image recognition is not efficient	Bad and frustrated

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

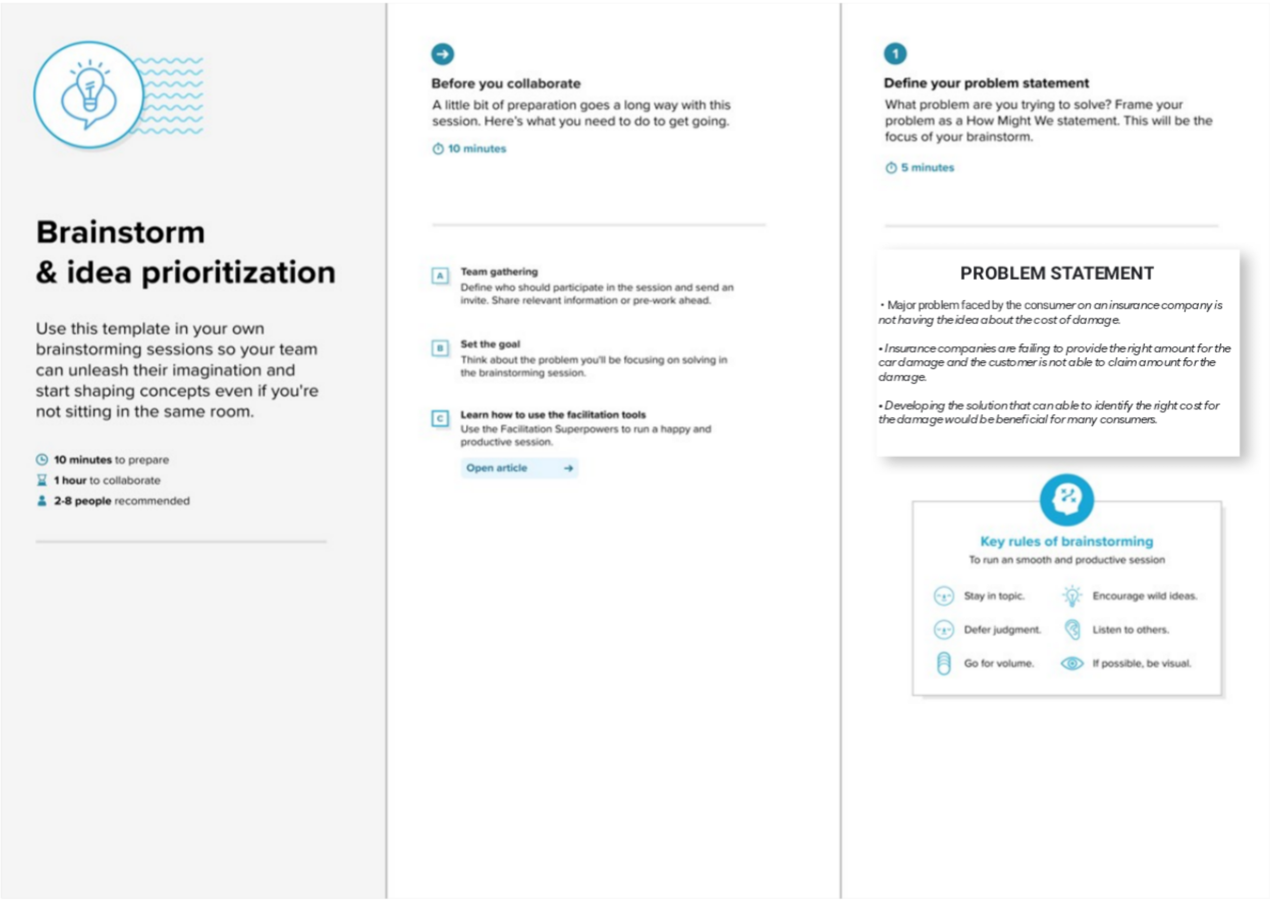
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

Example:



3.2 Ideation & Brainstorming

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

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

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

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

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

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

Open article →

1 **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 STATEMENT

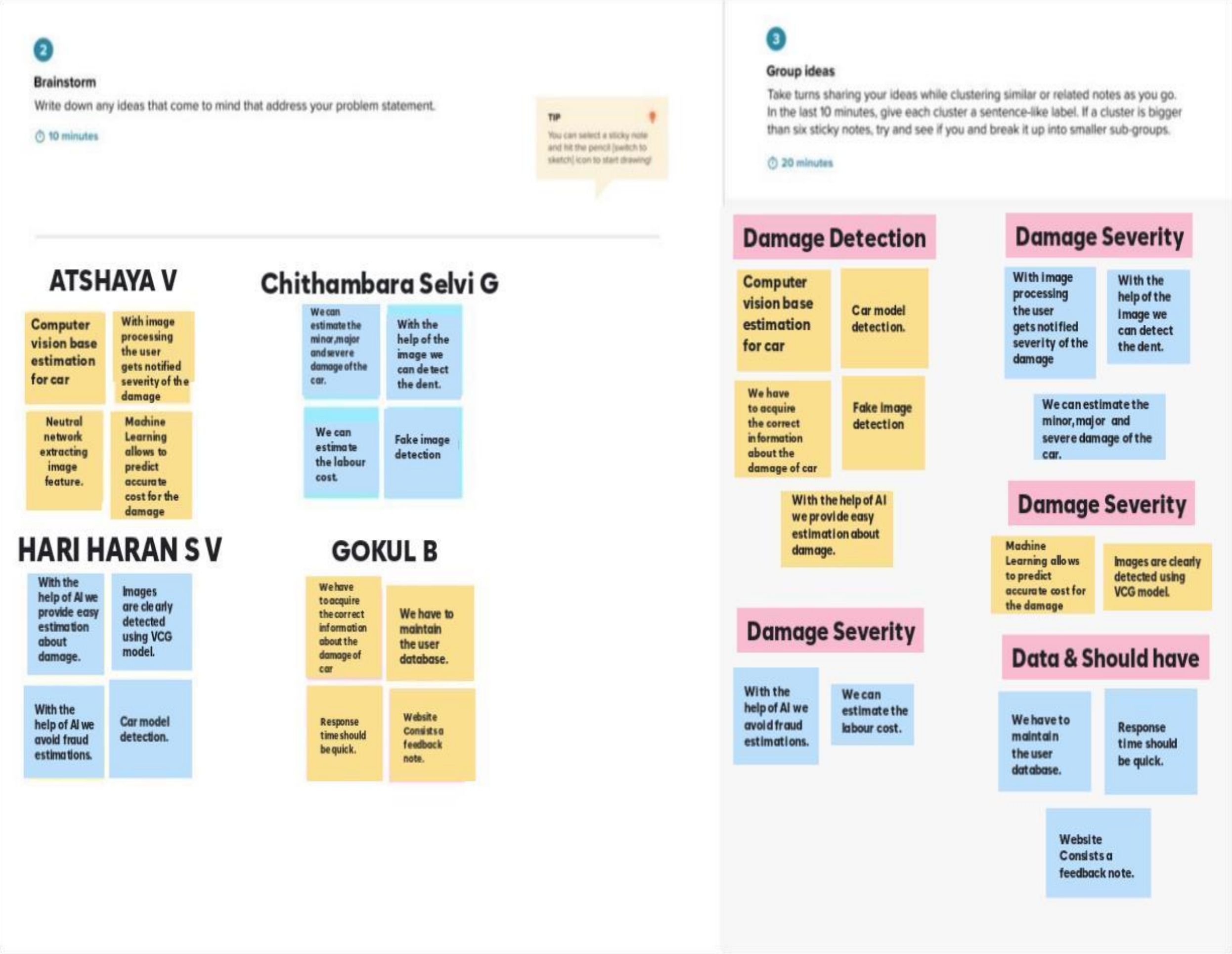
- Major problem faced by the consumer *on an insurance company is not having the idea about the cost of damage.*
- *Insurance companies are failing to provide the right amount for the car damage and the customer is not able to claim amount for the damage.*
- *Developing the solution that can able to identify the right cost for the damage would be beneficial for many consumers.*

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.

STEP-2: Brainstorm, Idea Listing and Grouping



2

Brainstorm

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

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

ATSHAYA V

Computer vision base estimation for car

With image processing the user gets notified severity of the damage

Neutral network extracting image feature.

Machine Learning allows to predict accurate cost for the damage

Chithambara Selvi G

We can estimate the minor,major and severe damage of the car.

With the help of the image we can detect the dent.

We can estimate the labour cost.

Fake image detection

HARI HARAN S V

With the help of AI we provide easy estimation about damage.

Images are clearly detected using VCG model.

With the help of AI we avoid fraud estimations.

Car model detection.

GOKUL B

We have to acquire the correct information about the damage of car

We have to maintain the user database.

Response time should be quick.

Website Consists a feedback note.

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, 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

Damage Detection

Computer vision base estimation for car

Car model detection.

We have to acquire the correct information about the damage of car

Fake image detection

With the help of AI we provide easy estimation about damage.

Damage Severity

With image processing 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.

Damage Severity

Machine Learning allows to predict accurate cost for the damage

Images are clearly detected using VCG model.

Damage Severity

With the help of AI we avoid fraud estimations.

We can estimate the labour cost.

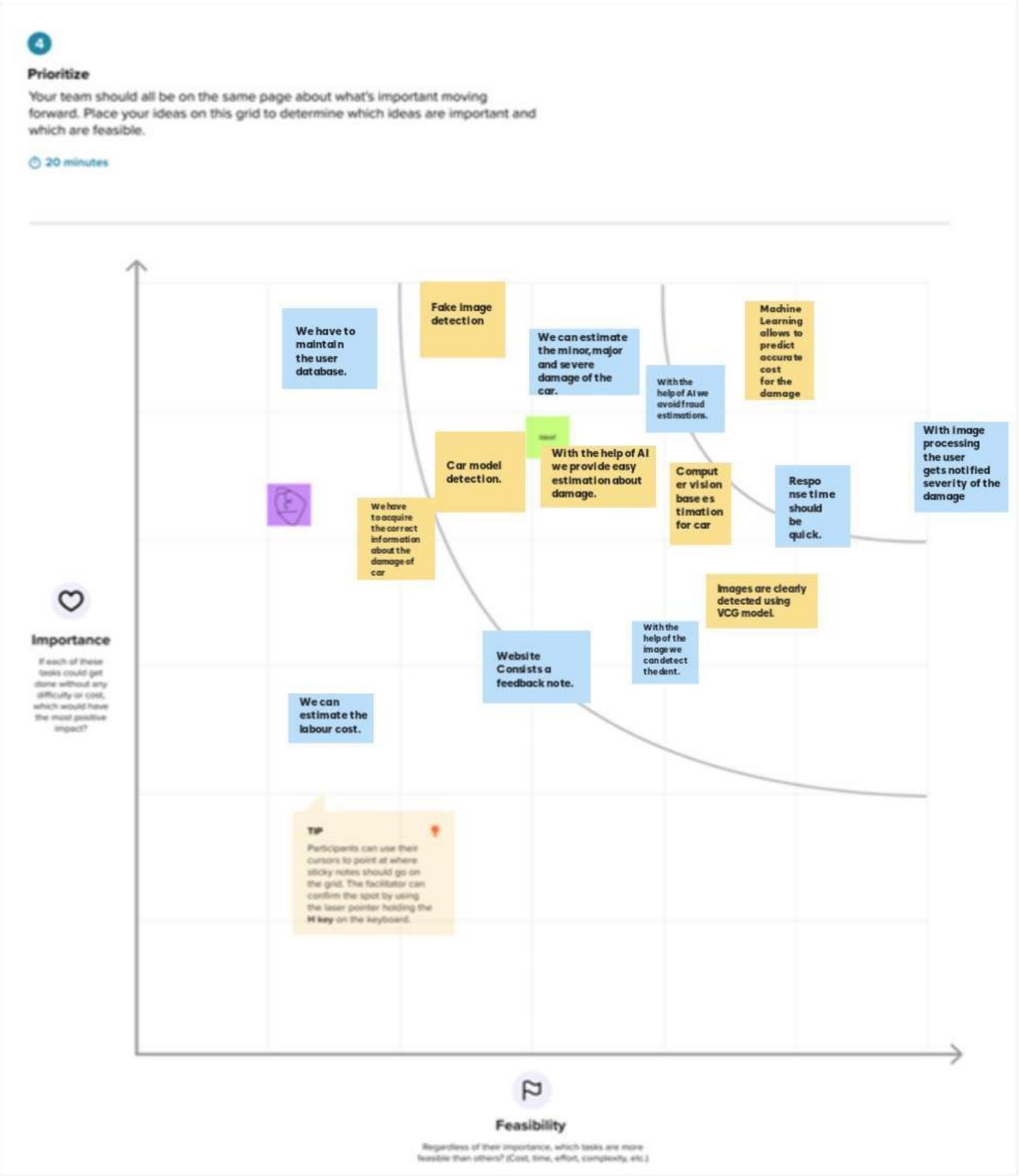
Data & Should have

We have to maintain the user database.

Response time should be quick.

Website Consists a feedback note.

Step-3: Idea Prioritization:



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

We have to maintain the user database.

Fake image detection

We can estimate the minor, major and severe damage of the car.

With the help of AI we avoid fraud estimations.

Machine Learning allows to predict accurate cost for the damage

Car model detection.

With the help of AI we provide easy estimation about damage.

Comput er vision base es timation for car

Respo nse time should be quick.

With image processing the user gets notified severity of the damage

We have to acquire the correct information about the damage of car

Images are clearly detected using VCG model.

With the help of the image we can detect the dent.

Website Consists a feedback note.

We can estimate the labour cost.

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

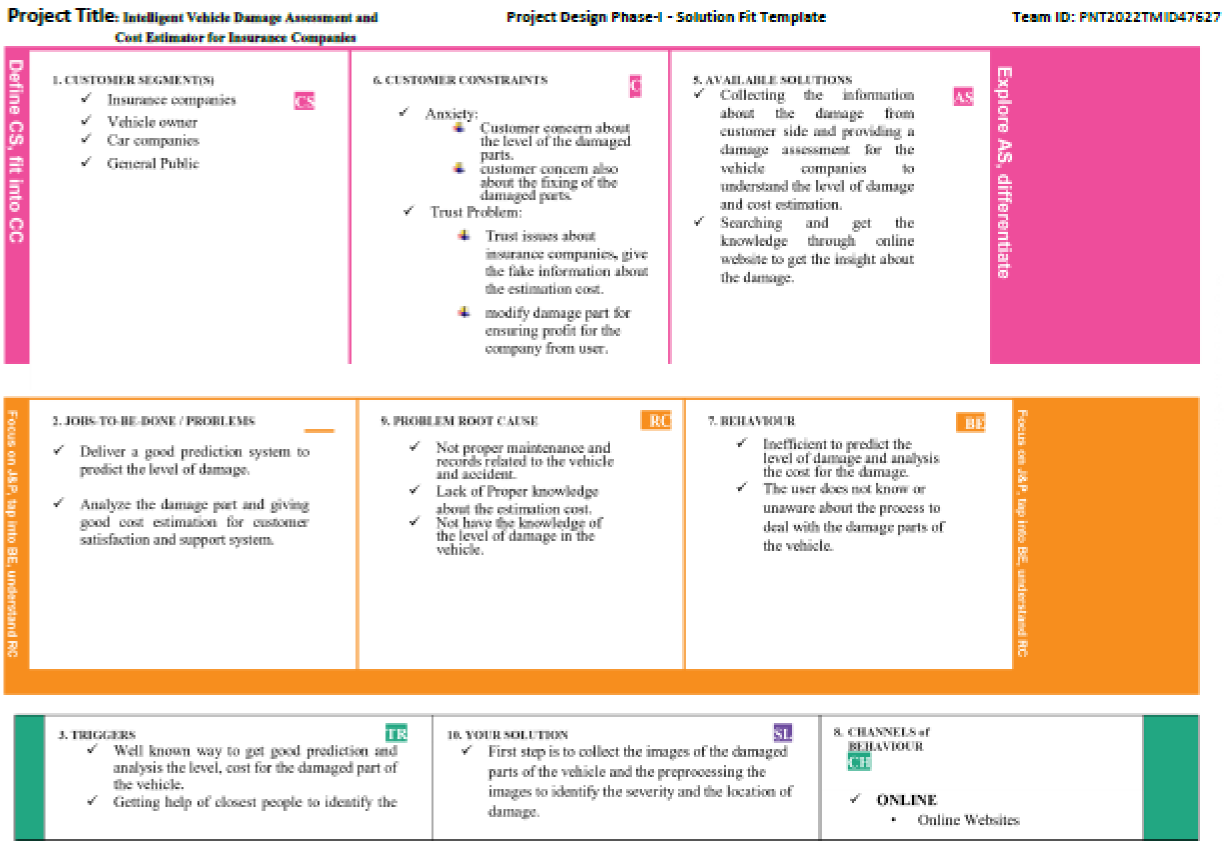
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3 Proposed Solution

S.NO.	PARAMETER	DESCRIPTION
1.	Problem Statement (Problem to be solved)	Nowadays, Insurance Companies faced the greatest problem which is the leakage of the insurance claim. Some of the customers claim an extra amount for the damage to the vehicle through fake bills for the claim. So The Insurance Companies lost most of the amount due to the leakage of the claim.
2.	Idea / Solution description	To solve this problem, we have to develop software that helps to insurance companies. The procedures we design involve creative initiative that will inspire the company has to believe in that software and also the customer..
3.	Novelty / Uniqueness	We applied deep learning-based algorithms, **Mask R-CNN**, for car damage detection and assessment in real world datasets. The algorithms detect the damaged part of a car and assess its location and then its severity. Initially, it discovers the effect of domain-specific **pre-trained CNN models,** which are trained by **datasets.** .
4.	Social Impact / Customer Satisfaction	On the website, customers have to take a photo of the damaged portion of the vehicle and send it to the company to claim the insurance. The process is quicker and they can easily access the website and post the picture on the company's website and estimate the cost for the damage .so we think it is easy for the customers.
5.	Business Model (Revenue Model)	It is more effective than others.It reduces the delay.This helps the customer to get the claim quickly.It has good accuracy.

6.	Scalability of the Solution	Mask R-CNN. Maximum object detection accuracy for training set is approximately 54% (using data augmentation and hyperparameter tuning).

3.4Problem Solution fit



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

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID47627

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

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.

8. CHANNELS of BEHAVIOUR CH

- ✓ **ONLINE**
  - Online Websites

4.REQUIREMENT ANALYSIS

4.1Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**User registeration**	Go to the website Register via email Set a password Reset the password
FR-2	**User Confirmation**	Confirmation via Email Confirmation via OTP
FR-3	**Interface**	Good Interface to user to operate
FR-4	**Accessing datasets**	Details about user Details about vehicle Details about insurance companies
FR-5	**Mobile application**	AI and camera sensors in the field can be accessed by mobile applications.

4.2 Non-Functional requirements

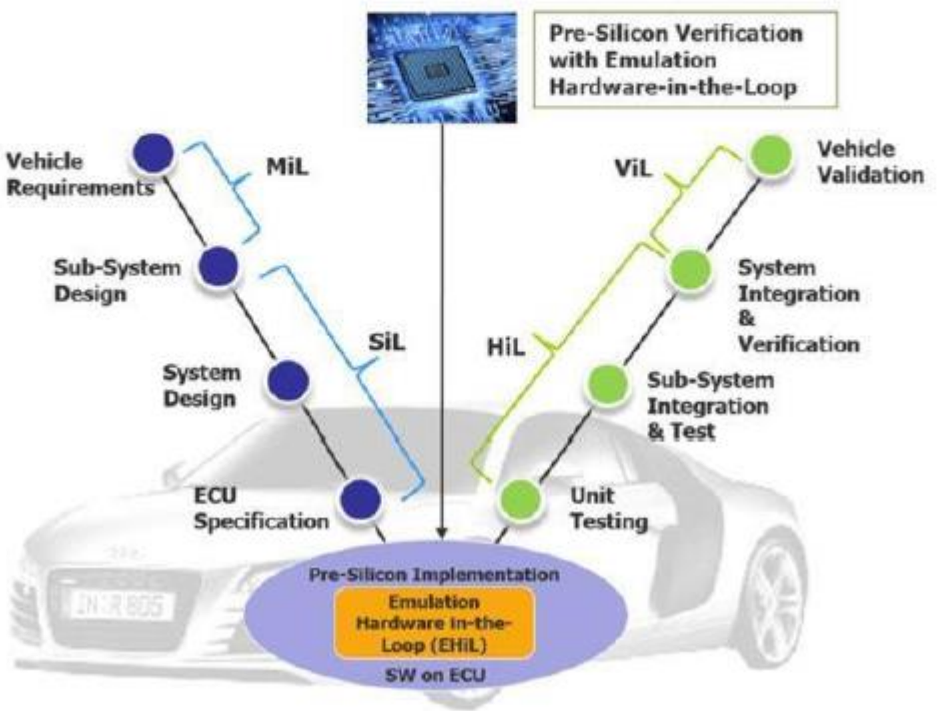
Following are the non-functional requirements of the proposed solution.

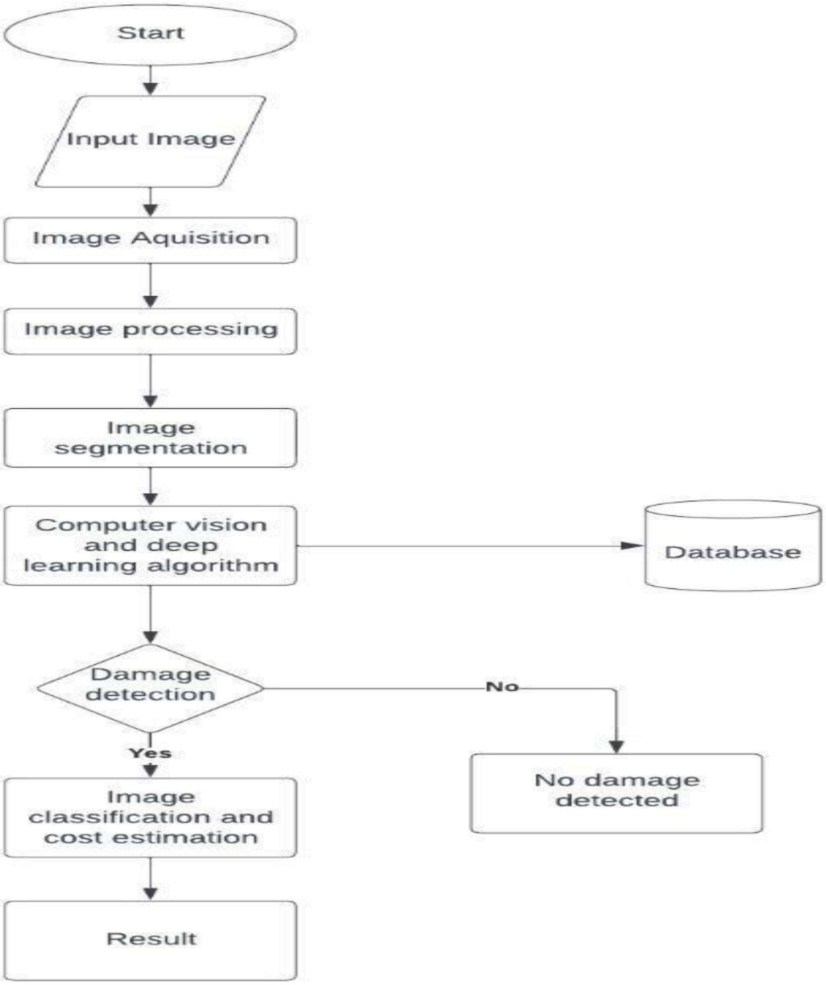
FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	The smart claiming system for vehicle damage insurance in insurance companies
NFR-2	**Security**	We have designed this project for users to claim insurance faster.
NFR-3	**Reliability**	This project will help the user to claim the insurance cost based on vehicle damage. It gives the exact value to the user. This helps the user to get the accurate cost without any failure.
NFR-4	**Performance**	AI devices and sensors are used to indicate to the user to estimate the cost of the vehicle.AI camera scans the damaged vehicle and gives exact cost insurance to the user.
NFR-5	**Availability**	This application can be accessed from any device browser.

NFR-6	**Scalability**	This project is more scalability in our present and future uses to estimate the cost accurately to the user.

5.PROJECT DESIGN

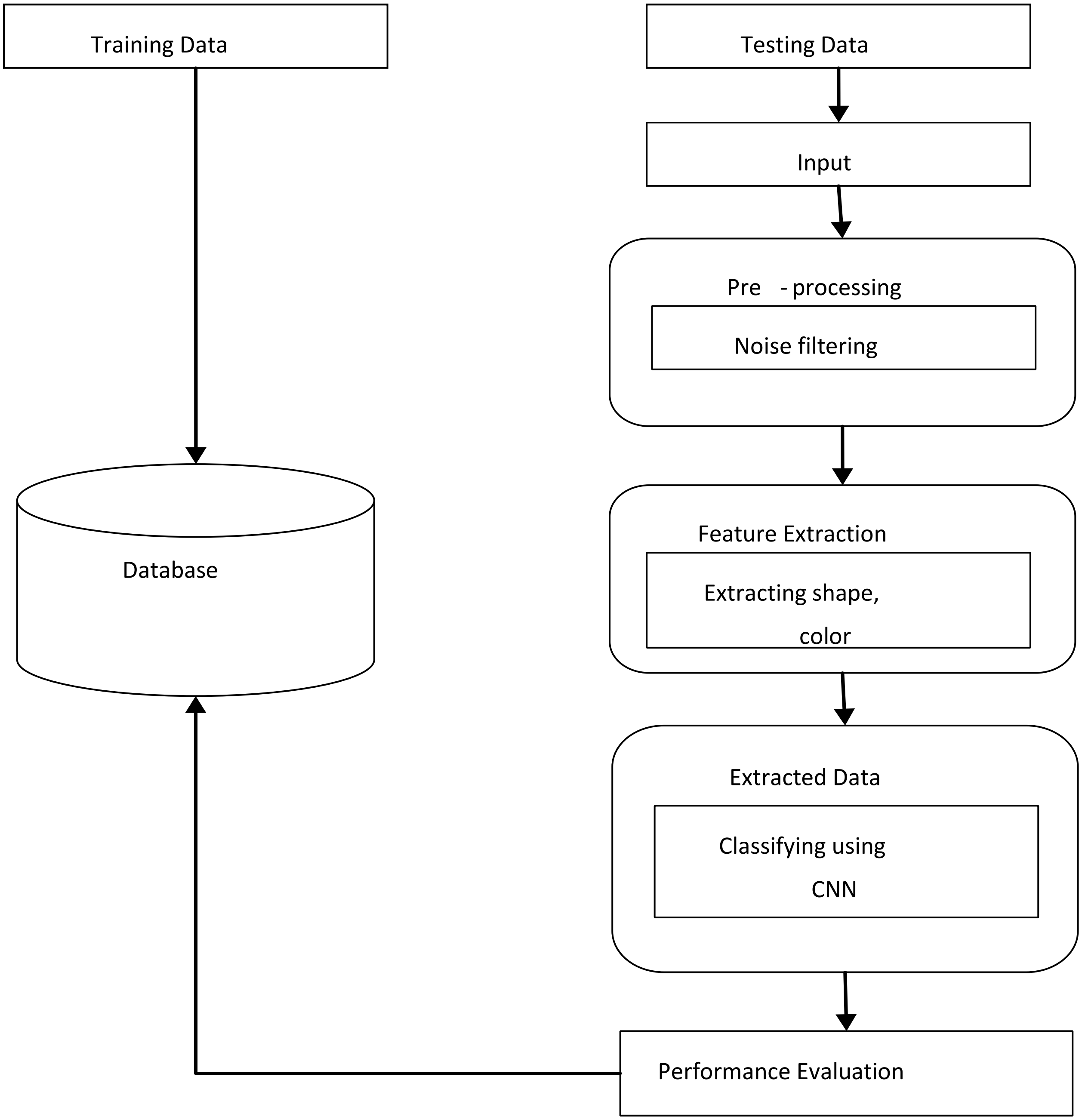
5.1Data Flow Diagrams


Pre-Silicon Verification with Emulation Hardware-in-the-Loop
Vehicle Requirements
MiL
Sub-System Design
SiL
System Design
ECU Specification
ViL
Vehicle Validation
System Integration & Verification
HiL
Sub-System Integration & Test
Unit Testing
Pre-Silicon Implementation
Emulation Hardware in-the-Loop (EHiL)
SW on ECU



5.1 Solution & Technical Architecture

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



5.2 User Stories

6 PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint 1	20	6 days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint 2	20	6 days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint 3	20	6 days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint 4	20	6 days	14 Nov 202	19 Nov 2022	20	19 Nov 2022

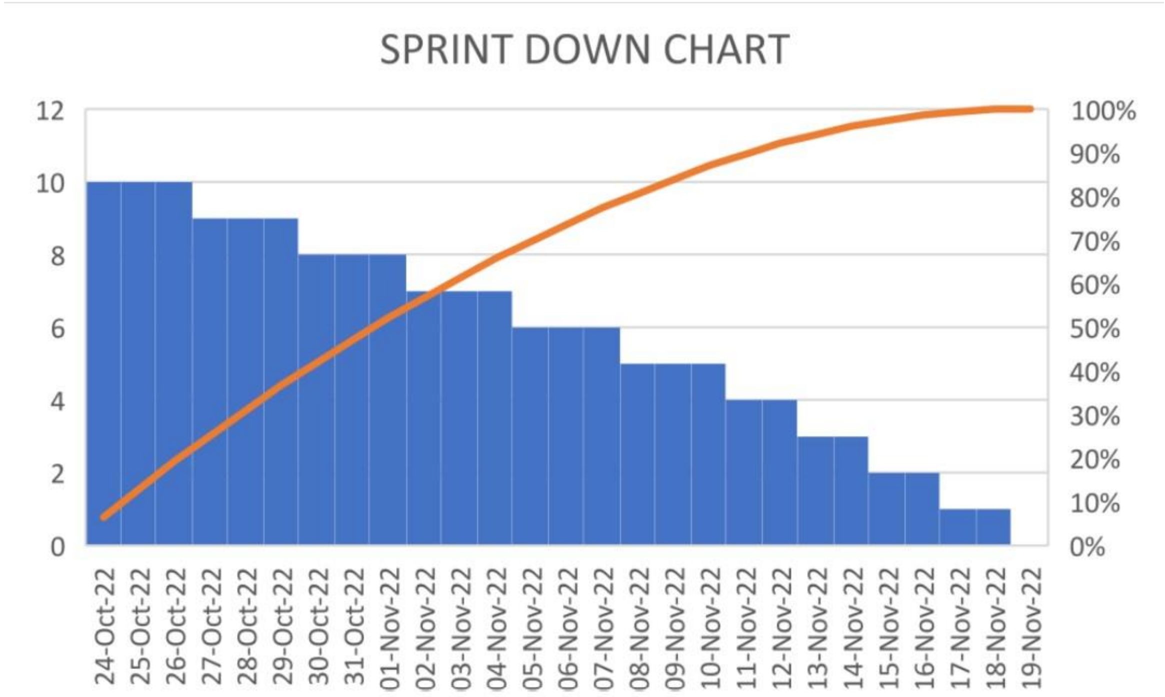
6.2 Sprint Delivery Schedule

Sprint	Duration		Sprint Start Date	Sprint End Date (Planned)	Sprint Release Date (Actual)
Sprint 1	6 days		24 Oct 2022	29 Oct 2022	29 Oct 2022
Sprint 2	6 days		31 Oct 2022	05 Nov 2022	05 Nov 2022
Sprint 3	6 days		07 Nov 2022	12 Nov 2022	12 Nov 2022
Sprint 4	6 days		14 Nov 202	19 Nov 2022	19 Nov 2022

6.3 Reports from JIRA

Velocity: 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 = Sprint duration/Velocity = 20/6 = 3 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 project containing measurable progress over time.

Sprint Down Chart:



7 CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

```
app = Flask(__name__)
run_with_ngrok(app)

@app.route('/register', methods = ['POST','GET'])
def signUp():
    username = request.args.get('username')
    carName = request.args.get('carName')
    regNo = request.args.get('regNo')
    regDate = request.args.get('regDate')
    password = request.args.get('password')

    sql_stmt = "INSERT INTO users VALUES(?,?,?,?,?,?,?)"
    stmt = ibm_db.prepare(conn, sql_stmt)
    uid = uuid.uuid4().hex[:8]
    ibm_db.bind_param(stmt, 1, uid)
    ibm_db.bind_param(stmt, 2, username)
    ibm_db.bind_param(stmt, 3, password)
    ibm_db.bind_param(stmt, 4, carName)
    ibm_db.bind_param(stmt, 5, regNo)
    ibm_db.bind_param(stmt, 6, regDate)
    ibm_db.bind_param(stmt, 7, "")

    try:
        resp = ibm_db.execute(stmt)
        if resp:
            try:
                resp = jsonify({"success": uid})
                resp.headers.add('Access-Control-AllowOrigin', '*')
                return resp
            except:
                resp = jsonify({"success": False})
                resp.headers.add('Access-Control-AllowOrigin', '*')
                return resp

    except:
        print(ibm_db.stmt_error)
```

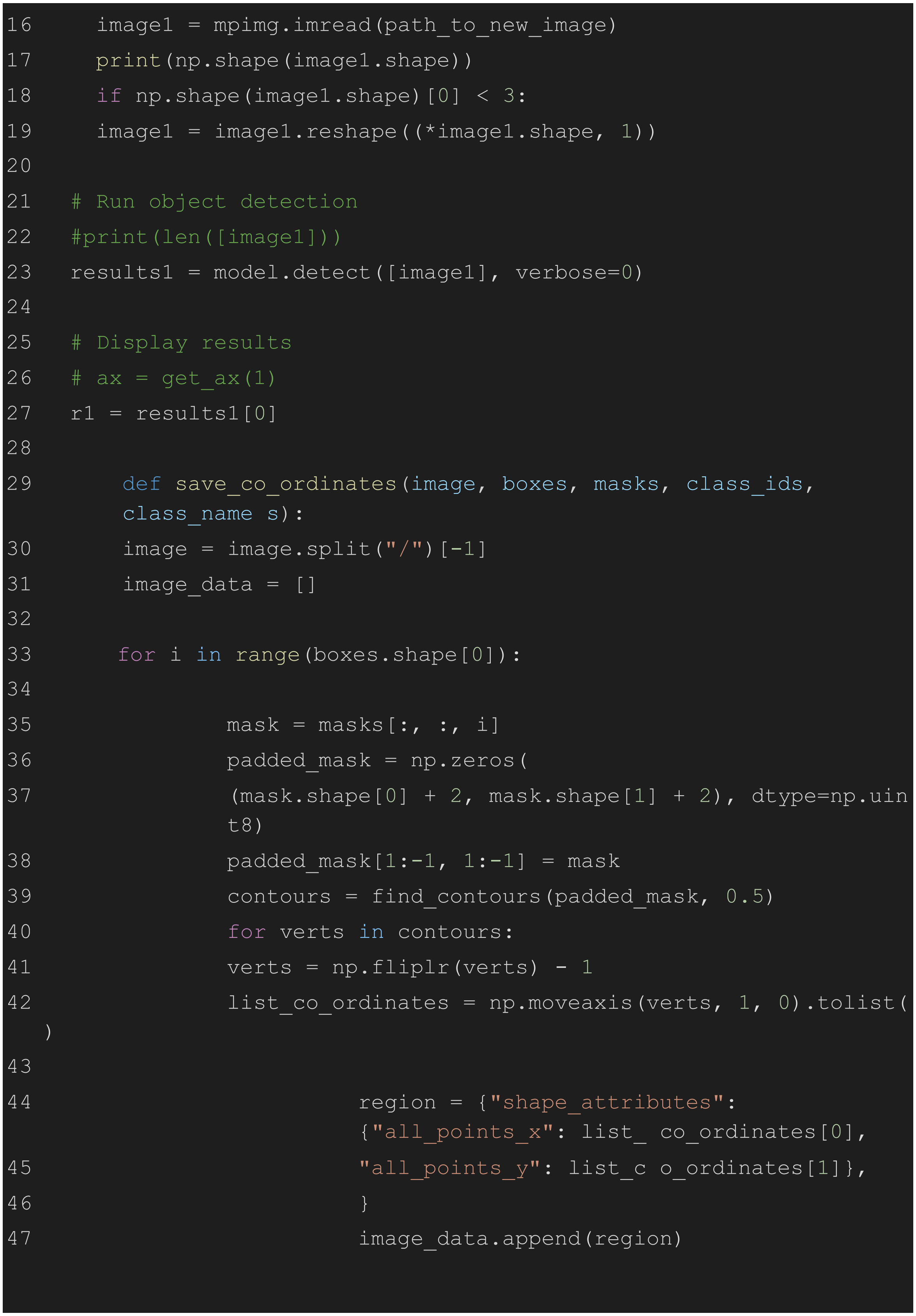
```python
def signIn():

    username = request.args.get('username')
    password = request.args.get('password')

sql_stmt = f"SELECT * FROM users WHERE
Username='{username}' A ND Password='{password}'"
stmt = ibm_db.prepare(conn, sql_stmt)

    print(stmt)

    try:
    resp = ibm_db.execute(stmt)
row = ibm_db.fetch_both(stmt)
    if row ==
False:
    resp = jsonify({"uid":False})
    resp.headers.add('Access-Control-Allow-Origin', '*')
return resp
    else:
    listData = {
    "uid": row['UID'],
    "username": row['USERNAME'],
    "carname": row['CARNAME'],
    "regno":
row['REGNO'],
    "regdate": row['REGDATE'],
    "claim": row['CLAIMAMOUNT']
    }
    resp = jsonify(listData)
    resp.headers.add('Access-Control-Allow-Origin', '*')
    return resp

    except:
        print(ibm_db.stmt_error)

@app.route('/postImage', methods = ['POST'])
def setImg():

    index = 1
    # print(request.form)
    insertValues = []
    claimAmount = 0

if not os.path.exists(ROOT_DIR + '/server/public/' + data['uid
']):
os.makedirs(ROOT_DIR + '/server/public/' + data['uid'])
    for x in vals:
    fileName = data['uid']+'_'+str(index)
            with open(ROOT_DIR + '/server/public/' + data['uid'] +
            '/'  + fileName + ".jpg", "wb") as f:
            f.write(base64.decodebytes(str.encode(x)))

        index += 1

     onlyfiles = [f for f in listdir(ROOT_DIR +
'/server/public/'+d ata['uid']+'/') if isfile(join(ROOT_DIR +
'/server/public/'+data[' uid']+'/', f))]

     print(onlyfiles)

     # prediction(data['uid'],onlyfiles[0])
     # prediction('das231564a','das231564a_1.jpg')

    # time.sleep(20)

    for x in onlyfiles:
    listData = {
    "filename": str(x),
    "area": "",
    "amount": 0
                }
       # tmpData = {
       #     "filename": str(x),
       #     "area": "",
       #     "amount": 0
```

```
        # }
        result = prediction(str(data['uid']),str(x))
        print(x)
        listData['area'] = (result / 1000)
```

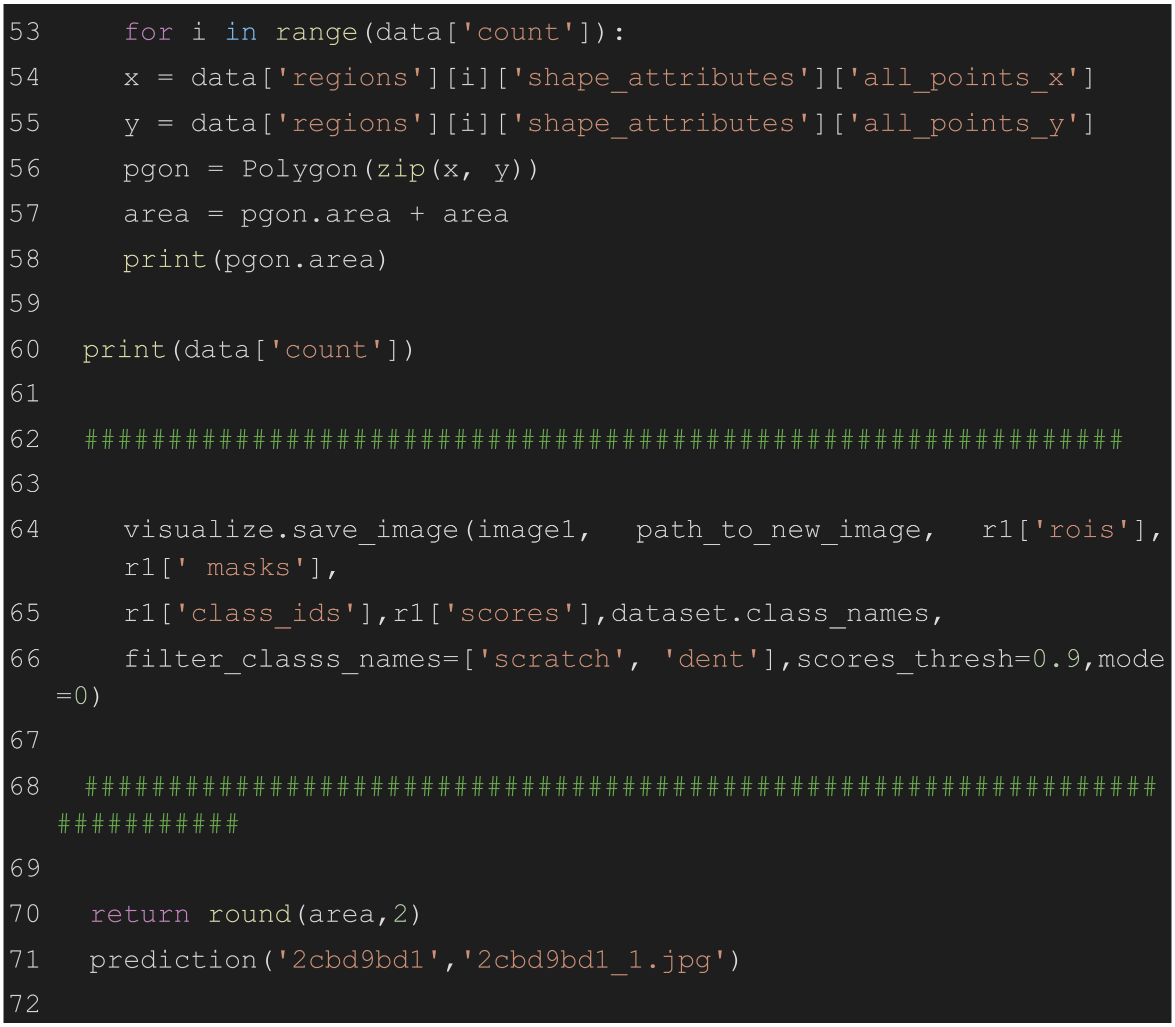
```
      claimAmount += listData['amount']
      analyzed.append(listData)
      insertValues.append((data['uid'],str(listData['area']),str(listData['amount'])))
      print(analyzed)
      print(insertValues)

    tuple_of_tuples = tuple([tuple(x) for x in insertValues])

    sql_stmt = f"INSERT INTO images VALUES(?,?,?)"
    stmt = ibm_db.prepare(conn, sql_stmt)

        try:
        exe = ibm_db.execute_many(stmt,tuple_of_tuples)
        print(exe)
        sql_stmt_2 = f"update users set claimamount='{str(claimAmount)}' where uid='{data['uid']}'"
        res = ibm_db.exec_immediate(conn, sql_stmt_2)
        resp = jsonify(analyzed)
        resp.headers.add('Access-Control-Allow-Origin', '*')
        return resp
        except:
        resp = jsonify({"uid": 'error'})
        resp.headers.add('Access-Control-Allow-Origin', '*')
        return resp
        print(ibm_db.stmt_error)


@app.route('/viewImage', methods = ['POST','GET'])
def viewImg():
    uid = request.args['uid']
    filename = request.args['filename']
    # uid = request.args['uid']
    # print(list(uid))
    return send_from_directory(ROOT_DIR + '/server/public/'+uid+'/
```

9.2 Feature 2

```
def prediction(uid,filename):
# print(uid)
area = 0
# path_to_new_image = '/content/drive/MyDrive/pythoncode/insuranceApp/dataset/test/test3.jpg'
path_to_new_image = os.path.join(ROOT_DIR + '/server/public/' + uid +'/'+ filename)
print(path_to_new_image)
```



```
  image1 = mpimg.imread(path_to_new_image)
  print(np.shape(image1.shape))
  if np.shape(image1.shape)[0] < 3:
  image1 = image1.reshape((*image1.shape, 1))

# Run object detection
#print(len([image1]))
results1 = model.detect([image1], verbose=0)

# Display results
# ax = get_ax(1)
r1 = results1[0]

    def save_co_ordinates(image, boxes, masks, class_ids, class_name s):
    image = image.split("/")[-1]
    image_data = []

    for i in range(boxes.shape[0]):

            mask = masks[:, :, i]
            padded_mask = np.zeros(
            (mask.shape[0] + 2, mask.shape[1] + 2), dtype=np.uint8)
            padded_mask[1:-1, 1:-1] = mask
            contours = find_contours(padded_mask, 0.5)
            for verts in contours:
            verts = np.fliplr(verts) - 1
            list_co_ordinates = np.moveaxis(verts, 1, 0).tolist()

                      region = {"shape_attributes":
                      {"all_points_x": list_ co_ordinates[0],
                      "all_points_y": list_c o_ordinates[1]},
                      }
                      image_data.append(region)
```



```
    for i in range(data['count']):
    x = data['regions'][i]['shape_attributes']['all_points_x']
    y = data['regions'][i]['shape_attributes']['all_points_y']
    pgon = Polygon(zip(x, y))
    area = pgon.area + area
    print(pgon.area)

  print(data['count'])

  ##############################################################

    visualize.save_image(image1,  path_to_new_image,  r1['rois'], r1[' masks'],
    r1['class_ids'],r1['scores'],dataset.class_names,
    filter_classs_names=['scratch', 'dent'],scores_thresh=0.9,mode=0)

  ############################################################################

  return round(area,2)
  prediction('2cbd9bd1','2cbd9bd1_1.jpg')

```

8.TESTING

8.1 Test Cases

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

Characteristics of a good test case:

- Accurate: Exacts the purpose.

- Economical: No unnecessary steps or words.
- Traceable: Capable of being traced to requirements.
- Repeatable: Can be used to perform the test over and over.
- Reusable: Can be reused if necessary.

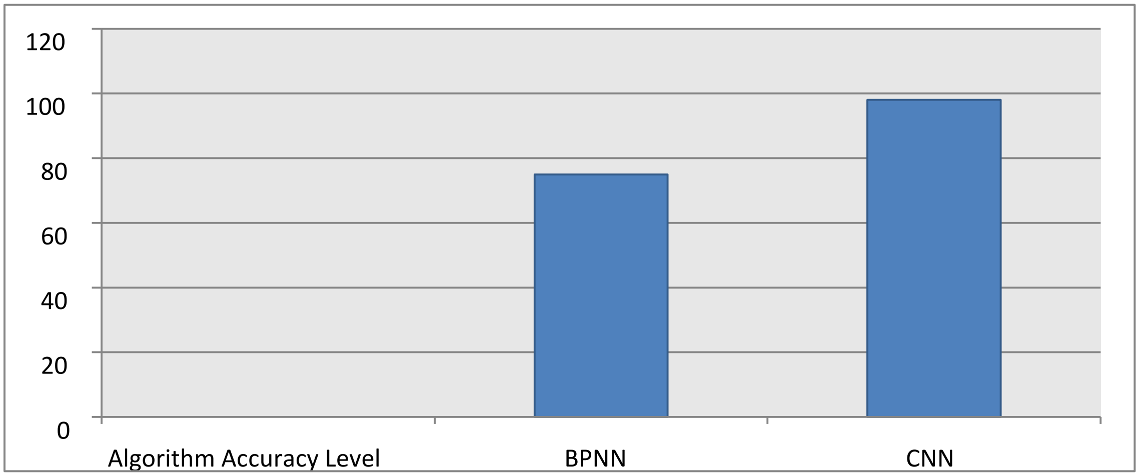
S.NO	**Scenario**	**Input**	**Excepted output**	**Actual output**
1	User login	User name and password	Login	Login success.
2	Upload Image	Upload damaged vehicle image as a input	Detecting object and analyze for claim insurance	Details are stored in a database.

8.2 User Acceptance Testing

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

9. RESULTS

9.1 Performance Metrics



10. ADVANTAGES & DISADVANTAGES

ADVANTAGE :

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

DISADVANTAGE

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

11. CONCLUSION

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

12. FUTURE SCOPE

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

13.

APPENDIX

Source

Code

Index:

```
<template>
```

```
  <div class="flex h-screen bg-screen">
      <div class="m-auto">
        <div class="w-96 h-3/4 bg-white rounded-lg px-8 py-8">
          <div v-if="page == 'login'">
            <LoginPage />
          </div>
          <div v-if="page == 'reg'">
            <RegPage />
          </div>
        </div>
      </div>
      <div v-if="isLoading">
        <div class="flex justify-center items-center h-screen fixed top-0 left0 right-0 bottom-0 w-full z-50 overflow-hidden bg-gray-700 opacity-75">
          <LoadingPage />
        </div>
      </div>
    </div>
</template>

<script>

import LoginPage from './login.vue' import
RegPage from './register.vue' import
LoadingPage from './loading.vue' import
ProfilePage from './profile.vue'
```

```
export default {
data: () => ({
isWarning: false,
isLoading: false,
page: 'login',
posts: [],     isValid:
false,     uid: ",
    username: "",
password: "",     carName:
"",     regNo: "",
regDate: "",     url:
'http://127.0.0.1:5000/logi
n?'
  }),
  name: 'IndexPage',
components: {
LoginPage,
    RegPage,
    LoadingPage,
    ProfilePage,
  },
  mounted() {
      },   methods: {
changePage(page) {
return this.page = page
```

```
    },
    setLoading(action) {
return this.isLoading = action
    },

    changeAfterReg(uid,username,carname,regno,regdate) {
this.uid = uid       this.username = username
this.carName = carname
      this.regNo = regno
this.regDate = regdate       var
tmpdata = {
        'uid': uid,
        'username': username,
        'carname': carname,
        'regno': regno,
        'regdate': regdate
      }
    }
  },    created() {
this.$root.$refs.index = this;
  }
}
</script>
```

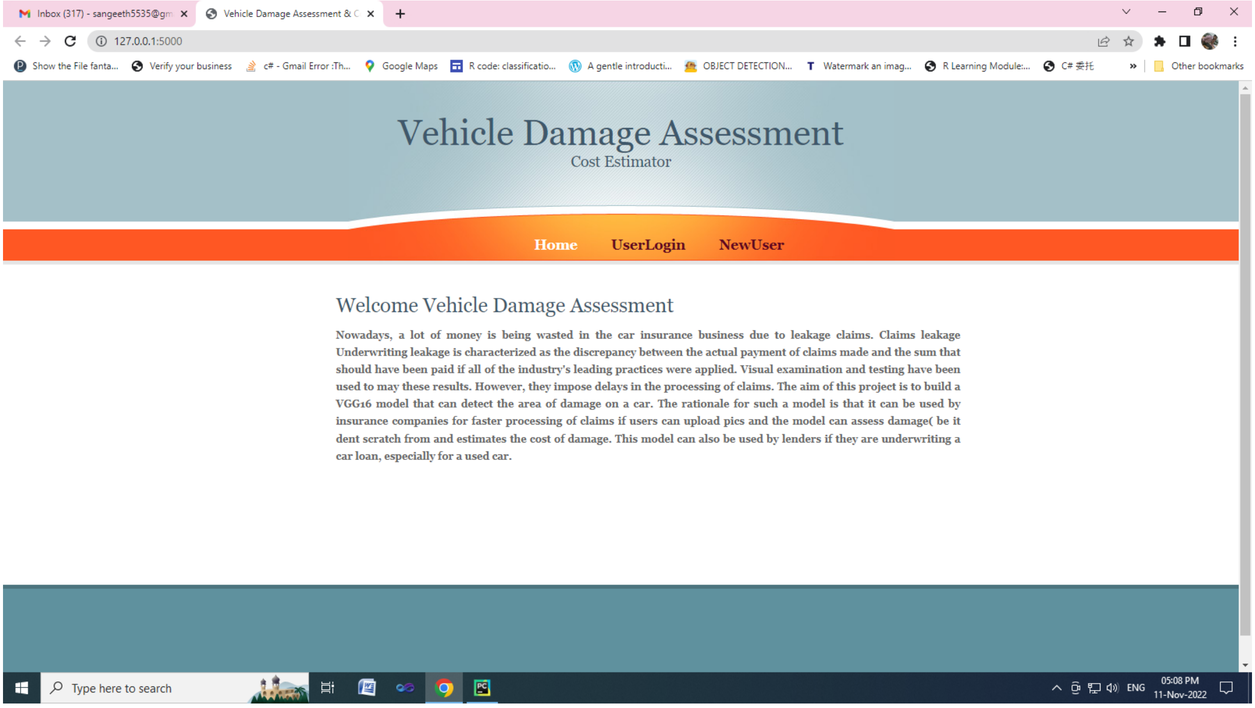
```
</template>
<script>

import moment from 'moment'

export default {
name: 'RegPage',
components: {

    },     data: () => ({         posts: [],
tmpDate: "",         datePh: "Select
Date",         isValid: false,
username: "",         password: "",
carName: "",         regNo: "",
regDate: "",         url:
'http://127.0.0.1:5000/register?'
    }),
    mounted() {
document.getElementById('username').addEventListener('input', (e)
=> {             if(this.username.length > 0 && this.carName.length >
0 && this.regNo.length > 0 && this.regDate.length > 0 &&
this.password.length > 0) {
                this.isValid = true
            } else {
this.isValid = false
            }
            console.log(this.regDate)
    })
    document.getElementById('carname').addEventListener('input', (e)
=> {          if(this.username.length > 0 && this.carName.length >
0 && this.regNo.length > 0 && this.regDate.length > 0 &&
this.password.length > 0) {
        this.isValid = true
      } else {
this.isValid = false
      }
      console.log(this.regDate)
    })
    document.getElementById('regno').addEventListener('input', (e) =>
{
      if(this.username.length > 0 && this.carName.length > 0 &&
this.regNo.length > 0 && this.regDate.length > 0 &&
this.password.length > 0) {
        this.isValid = true
      } else {
this.isValid = false
      }
      console.log(this.regDate)
    })
    document.getElementById('password').addEventListener('input', (e)
=> {          if(this.username.length > 0 && this.carName.length >
0 && this.regNo.length > 0 && this.regDate.length > 0 &&
this.password.length > 0) {
        this.isValid = true
      } else {
this.isValid = false
      }
      console.log(this.regDate)
    })
  },
  methods: {      loginpage() {
this.$root.$refs.index.changePage('login');
   },
   handleChange(e) {
    var date = moment(e).format('MM/DD/YYYY')
this.regDate = date        console.log(this.regDate)
if(this.username.length > 0 && this.carName.length > 0 &&
this.regNo.length > 0 && this.regDate.length > 0 &&
this.password.length > 0) {           this.isValid = true
    } else {
this.isValid = false
    }
   },
   async register(username,carName,regNo,regDate,password) {
this.$root.$refs.index.setLoading(true);
    const res = await fetch(this.$server.url + 'register?' + new
URLSearchParams({username: username.toLowerCase(), carName:
carName, regNo: regNo, regDate: regDate, password:
password}),{method: "POST"})
```

```
    if(res.status == 200) {
this.posts =  await res.json()
if(this.posts['success'] != false) {
console.log('success')                var resp
= {
          'uid': this.posts['success'],
          'username': this.username.toUpperCase(),
          'carname': this.carName.toUpperCase(),
          'regno': this.regNo.toUpperCase(),
          'regdate': this.regDate,
          'claim': "
        }
        this.$root.$refs.index.setLoading(false);
this.$router.push({name: 'profile', params: {data: resp}})
      } else {
console.log('failed')
      }
    }
   },
  },
}
</script>
<style lang="">

</style>
```

```
        '_id': x[1],
        'name': x[0],
        'psw': x[2]
    }
    print(data)
    query = {'_id': {'Seq': data['_id']}}
    docs = my_database.get_query_result(query)
    print(docs)
    print(len(docs.all()))
    if (len(docs.all()) == 0):
        url = my_database.create_document(data)
        return render_template('goback.html', data="Register, please login using your details")
    else:
        return render_template('goback.html', data="You are already a member, please login using your details")
@app.route("/userlog", methods=['GET', 'POST'])
def userlog():
    if request.method == 'POST':
        user = request.form['_id']
        passw = request.form['psw']
        print(user, passw)
        query = {'_id': {'$eq': user}}
        docs = my_database.get_query_result(query)
        print(docs)
        print(len(docs.all()))
        if (len(docs.all()) == 0):
            return render_template('goback.html', pred="The username is not found.")
        else:
            if ((user == docs[0][0]['_id'] and passw == docs[0][0]['psw'])):
                return render_template("userhome.html")
            else:
                return render_template('goback.html',data="user name and password incorrect")
```

```
@app.route("/predict", methods=['GET', 'POST'])
def predict():
    if request.method == 'POST':
        file = request.files['fileupload']
        file.save('static/Out/Test.jpg')
        import warnings
        warnings.filterwarnings('ignore')
        import tensorflow as tf
        classifierLoad = tf.keras.models.load_model('body.h5')
        import numpy as np
        from keras.preprocessing import image
        test_image = image.load_img('static/Out/Test.jpg', target_size=(200, 200))
        img1 = cv2.imread('static/Out/Test.jpg')
        # test_image = image.img_to_array(test_image)
        test_image = np.expand_dims(test_image, axis=0)
        result = classifierLoad.predict(test_image)
        result1 = ''
        if result[0][0] == 1:
            result1 = "front"
        elif result[0][1] == 1:
            result1 = "rear"
        elif result[0][2] == 1:
            result1 = "side"
        file = request.files['fileupload1']
        file.save('static/Out/Test1.jpg')
        import warnings
        warnings.filterwarnings('ignore')
        import tensorflow as tf
        classifierLoad = tf.keras.models.load_model('level.h5')
        import numpy as np
        from keras.preprocessing import image
        test_image = image.load_img('static/Out/Test1.jpg', target_size=(200, 200))
        img1 = cv2.imread('static/Out/Test1.jpg')
        # test_image = image.img_to_array(test_image)
```

```
          test_image = np.expand_dims(test_image, axis=0)
          result = classifierLoad.predict(test_image)
          result2 = ''
          if result[0][0] == 1:
             result2 = "minor"
          elif result[0][1] == 1:
             result2 = "moderate"
          elif result[0][2] == 1:
             result2 = "severe"
          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('userhome.html', prediction=value)
    if __name__ == '__main__':
app.run(debug=True, use_reloader=True)
```

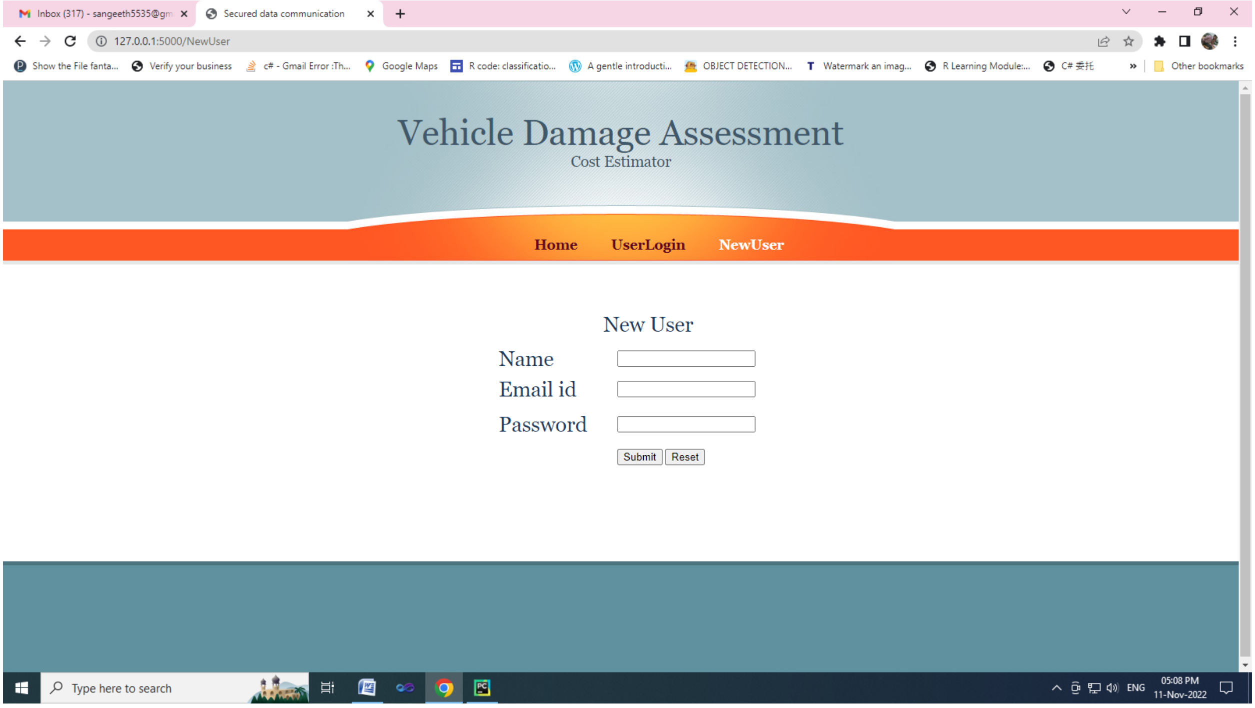


Vehicle Damage Assessment

Cost Estimator

Welcome Vehicle Damage Assessment

Nowadays, a lot of money is being wasted in the car insurance business due to leakage claims. Claims leakage Underwriting leakage is characterized as the discrepancy between the actual payment of claims made and the sum that should have been paid if all of the industry's leading practices were applied. Visual examination and testing have been used to may these results. However, they impose delays in the processing of claims. The aim of this project is to build a VGG16 model that can detect the area of damage on a car. The rationale for such a model is that it can be used by insurance companies for faster processing of claims if users can upload pics and the model can assess damage(be it dent scratch from and estimates the cost of damage. This model can also be used by lenders if they are underwriting a car loan, especially for a used car.



Vehicle Damage Assessment

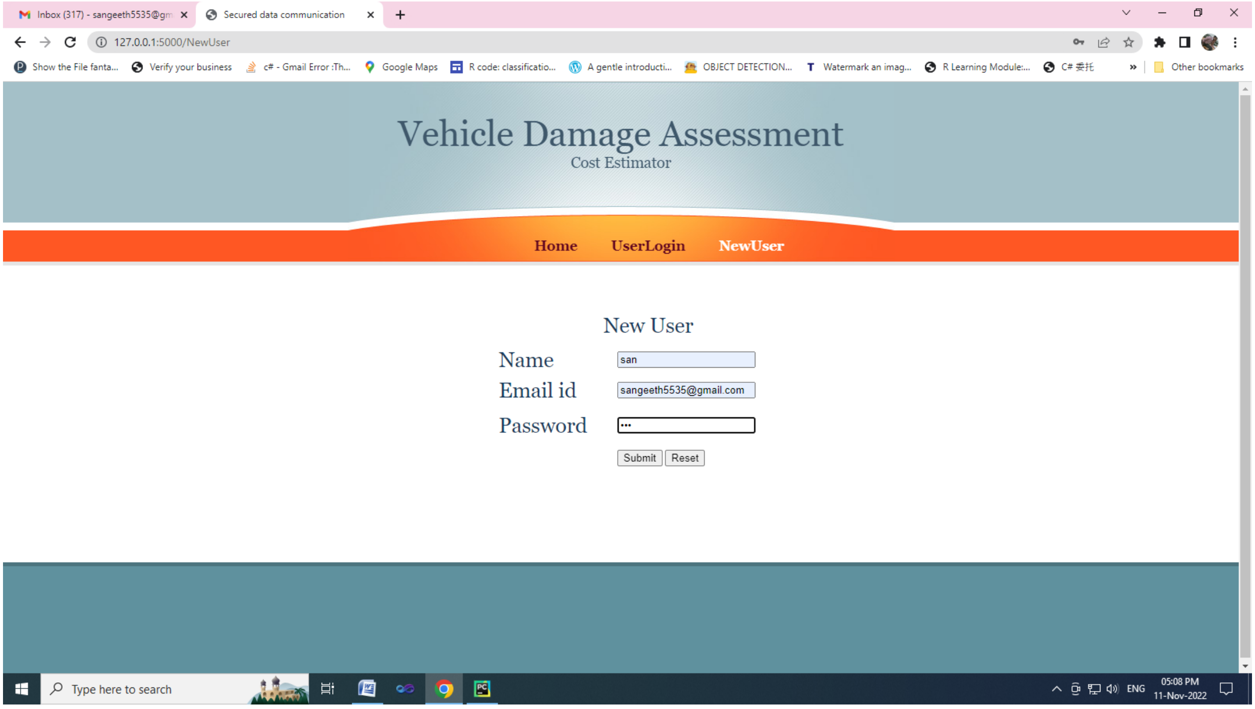
Cost Estimator

New User

Name

Email id

Password



127.0.0.1:5000/NewUser

Vehicle Damage Assessment

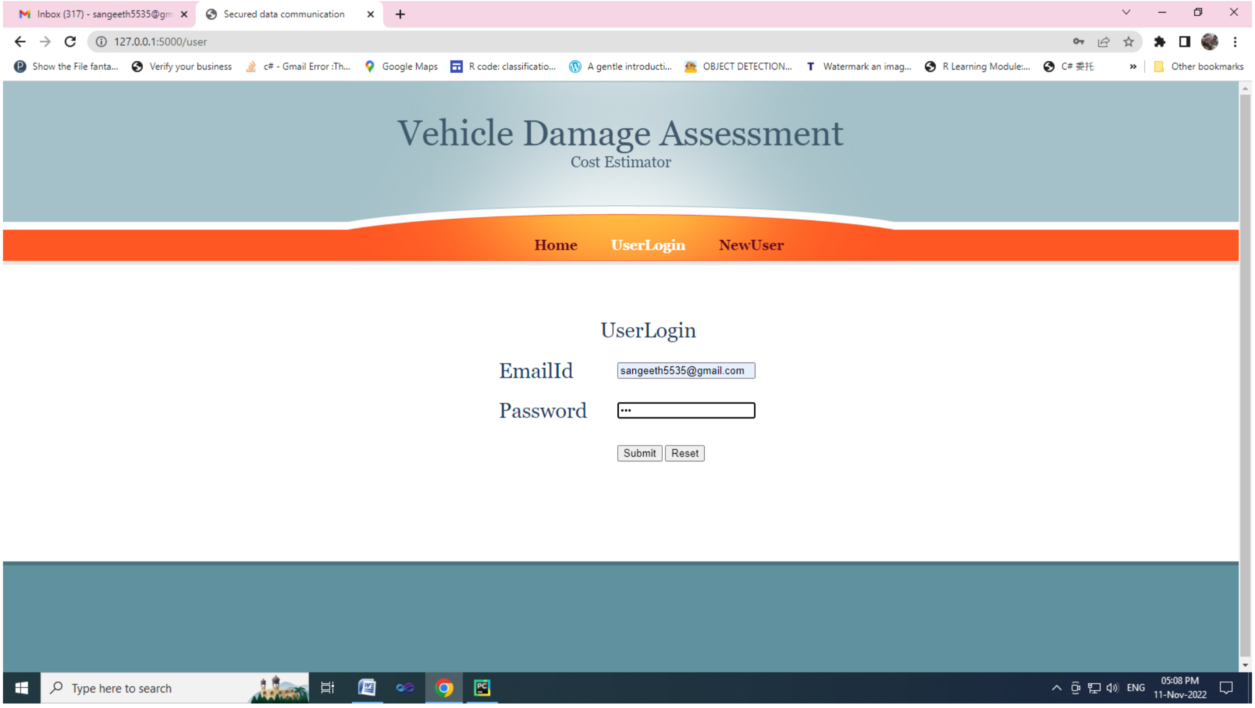
Cost Estimator

New User

Name san

Email id sangeeth5535@gmail.com

Password •••



127.0.0.1:5000/user

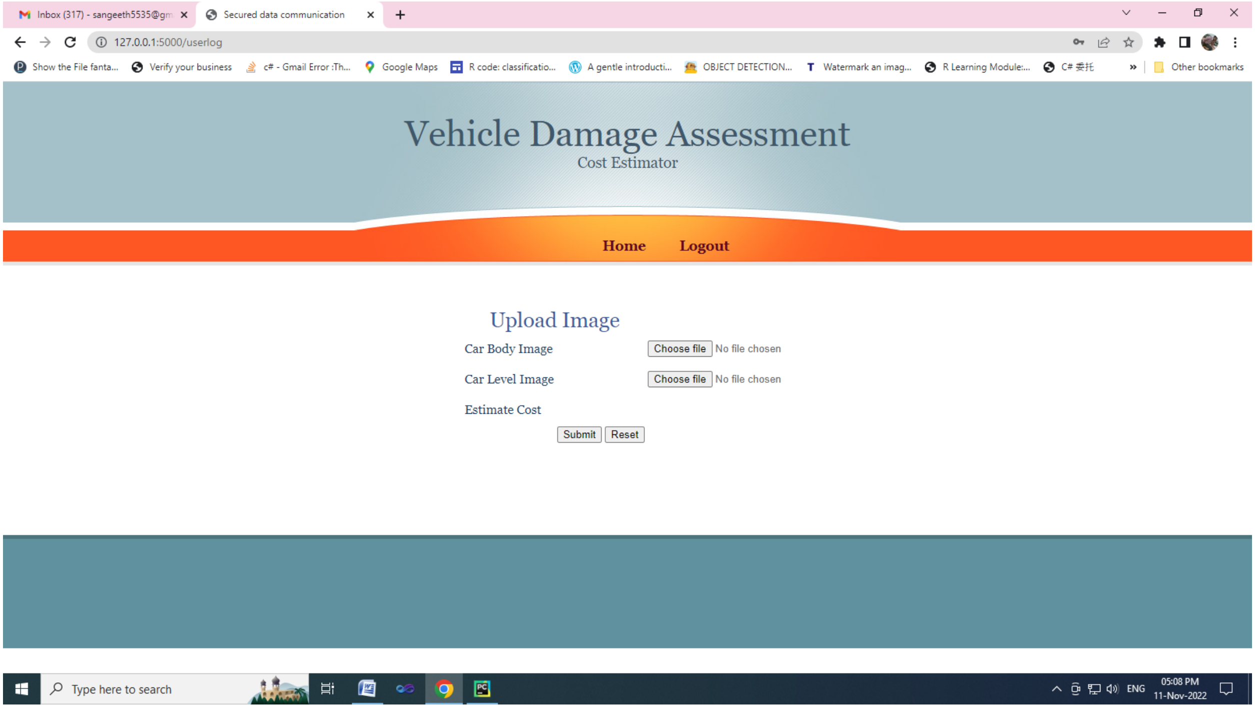
Vehicle Damage Assessment

Cost Estimator

UserLogin

EmailId sangeeth5535@gmail.com

Password •••



Inbox (317) - sangeeth5535@gm
Secured data communication
127.0.0.1:5000/userlog
Show the File fanta... Verify your business c# - Gmail Error :Th... Google Maps R code: classificatio... A gentle introducti... OBJECT DETECTION... Watermark an imag... R Learning Module:... C# 委托 Other bookmarks
Type here to search
ENG 05:08 PM 11-Nov-2022

Vehicle Damage Assessment

Cost Estimator

Home Logout

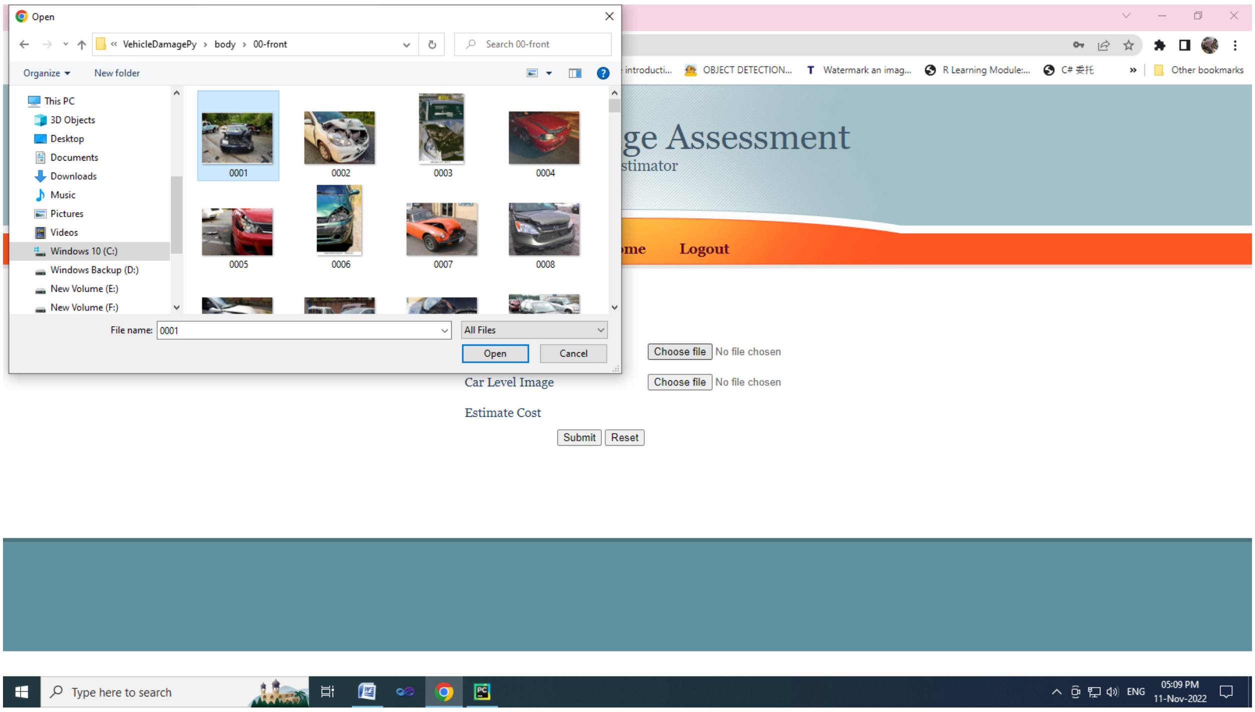
Upload Image

Car Body Image Choose file No file chosen

Car Level Image Choose file No file chosen

Estimate Cost

Submit Reset



ge Assessment

stimator

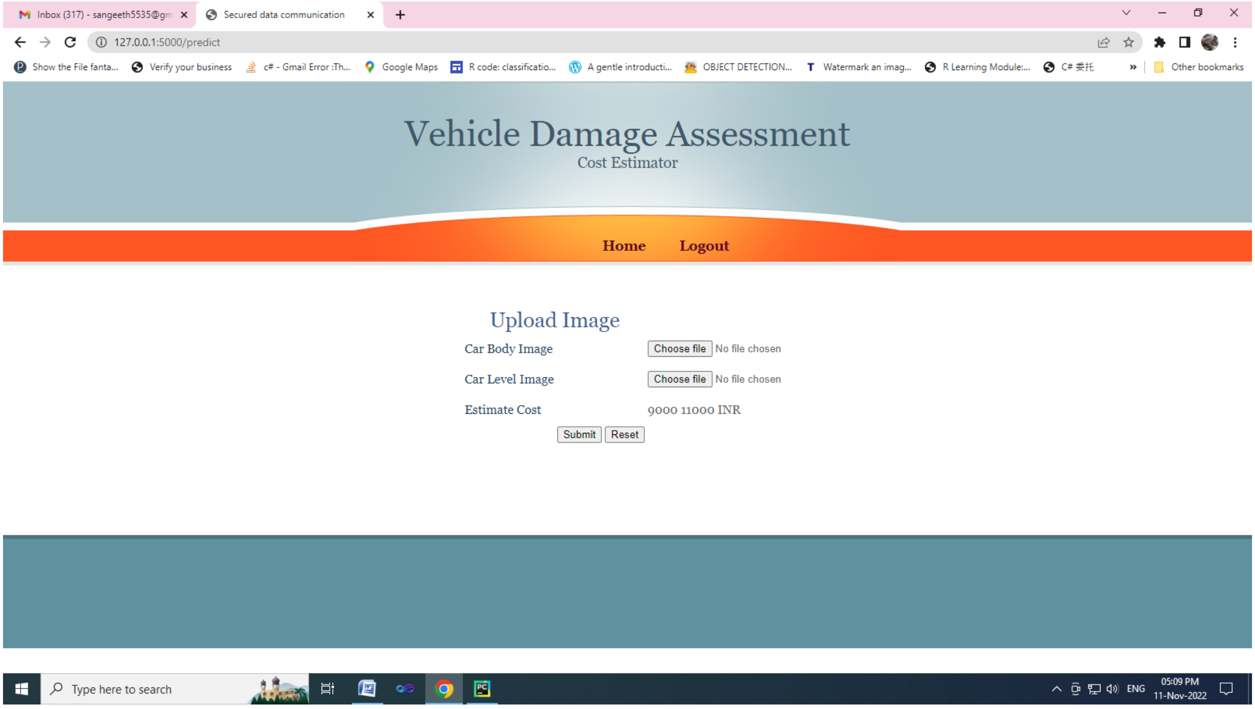
me Logout

Choose file No file chosen

Car Level Image Choose file No file chosen

Estimate Cost

Submit Reset



Vehicle Damage Assessment

Cost Estimator

Upload Image

Car Body Image — Choose file No file chosen

Car Level Image — Choose file No file chosen

Estimate Cost — 9000 11000 INR

GitHub & Project Demo Link

Github link: https://github.com/IBM-EPBL/IBM-Project-6033-1658822308.git