1. INTRODUCTION:

1.1 PROJECT OVERVIEW:

Our project “Handwritten Digit Recognition System” can be used to recognize the handwritten digits from image as an input. User can choose the image file from their file explorer and it is given to the model. The model will process the given input and choose the optimized output based on the digit image.

1.2 PURPOSE:

The purpose of this project is to give an application for recognizing the handwritten digits which may solve the problems that are faced by a variety of people/ vendor/ enterprises in the society.

2. LITERATURE SURVEY:

2.1 EXISTING PROBLEM:

It is hard to manually convert a series of handwritten digits in variety of scenarios, one by one. It leads to time consumption, labour expenditure etc. For example, automating the storage of handwritten bills and receipts.

2.2 REFERENCES:

1. Subhasis Mandal, S.R. Mahadeva Prasanna and Suresh Sundaram, review on “Exploration of CNN features for Online Handwriting Recognition”, IEEE, issued in the year 2020.

2. S. Mori, C.Y. Suen and Kamamoto, review on “An Neural Network based Handwritten Character Recognition system”, IEEE, issued in the year 2020.

3. Babu Venkatesh and Chintha, review on “Handwritten Digit Recognition Using K-Nearest Neighbor Classifier”, IEEE, issued in the year 2021.

4. Hazra, T.K Singh, D.P & Daga, review on “Optical Character Recognition using KNN on Custom Image Dataset”, IEEE, issued in the year 2021.

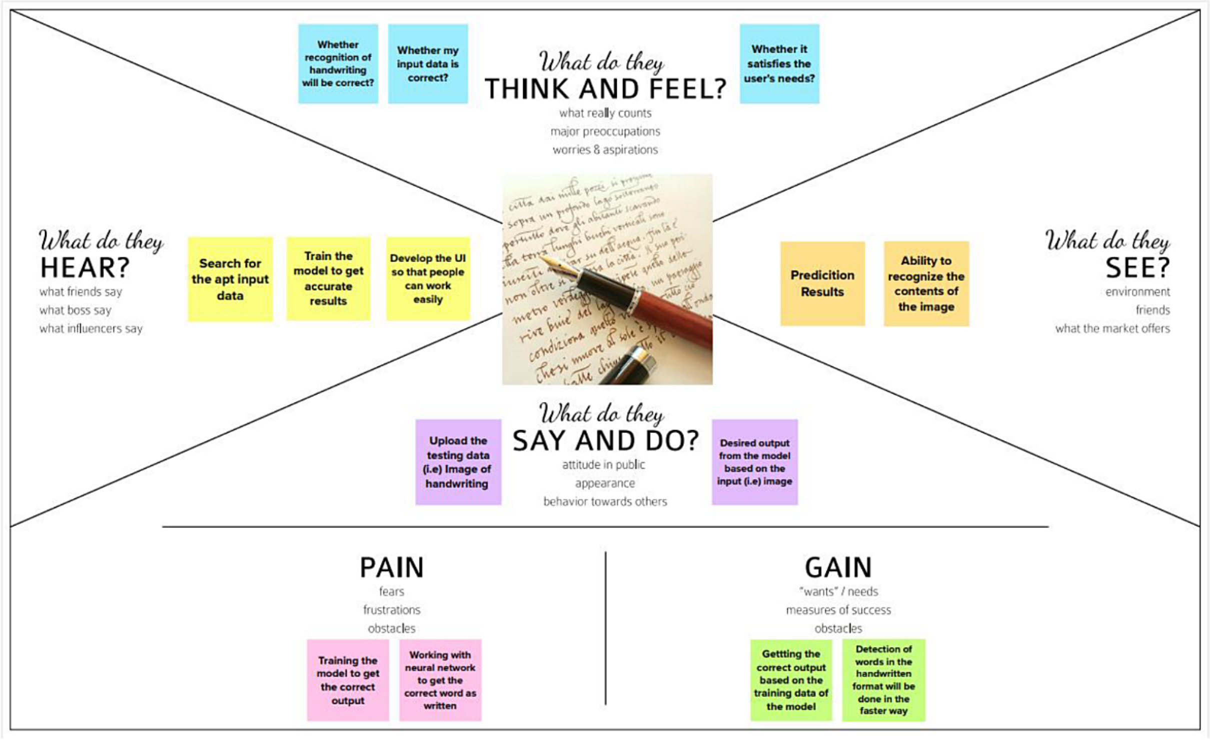
2.3 PROBLEM STATEMENT DEFINITION:

A Handwritten Digit Recognition System helps the helps in converting the image file with a handwritten digit into digital format. Right now, it is possible only to convert manually. This hugely impacts several vendors, businesses and organisations that are on the process of converting their manual records into digital format.

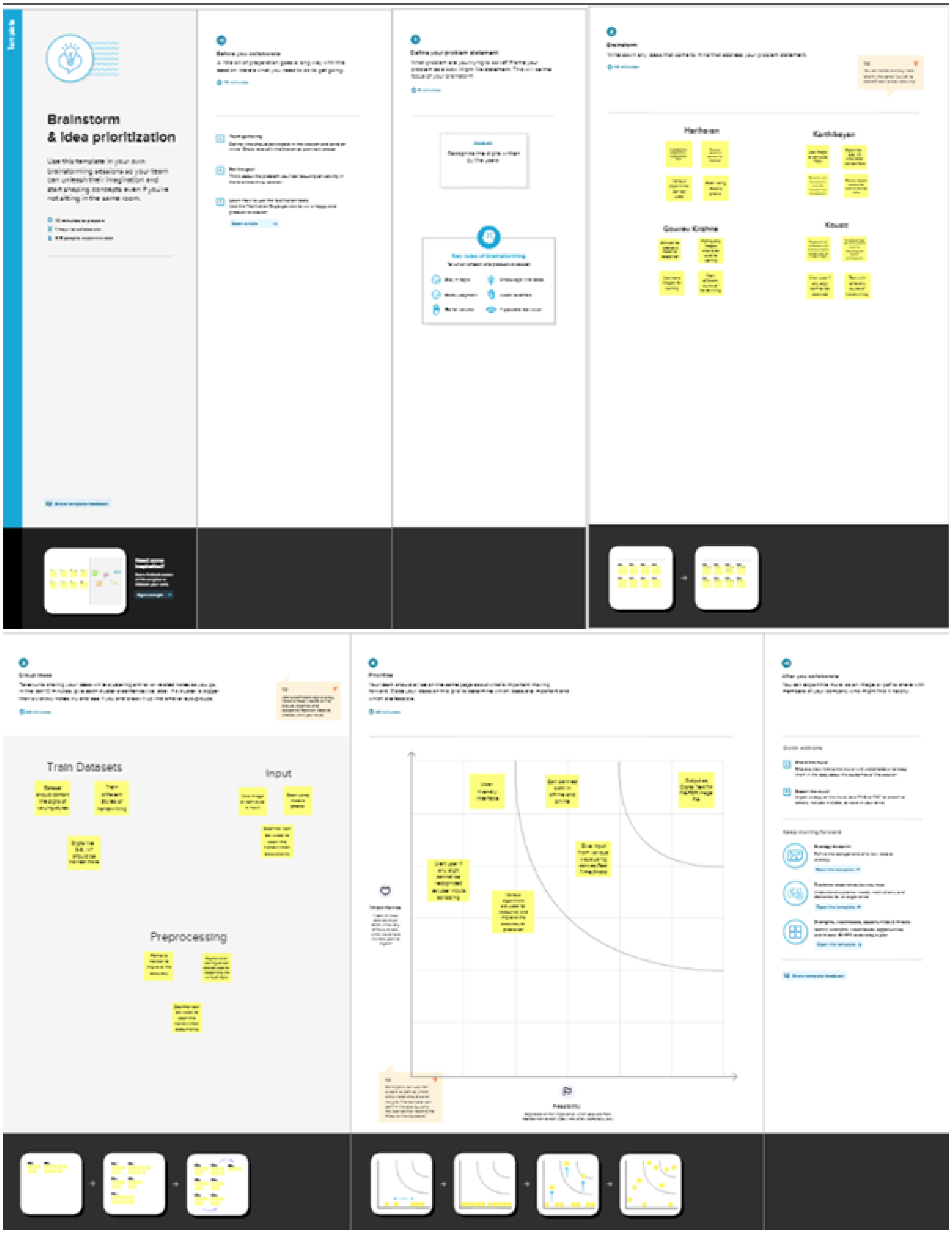
This application solves this problem by easily converting the manual digits from image files.

3. IDEATION AND PROPOSED SOLUTION:

3.1 EMPATHY MAP CANVAS:


Whether recognition of handwriting will be correct?
Whether my input data is correct?
What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Whether it satisfies the user's needs?
What do they
HEAR?
what friends say
what boss say
what influencers say
Search for the apt input data
Train the model to get accurate results
Develop the UI so that people can work easily
Predicition Results
Ability to recognize the contents of the image
What do they
SEE?
environment
friends
what the market offers
Upload the testing data (i.e) Image of handwriting
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
Desired output from the model based on the input (i.e) image
PAIN
fears
frustrations
obstacles
Training the model to get the correct output
Working with neural network to get the correct word as written
GAIN
"wants" / needs
measures of success
obstacles
Gettting the correct output based on the training data of the model
Detection of words in the handwritten format will be done in the faster way

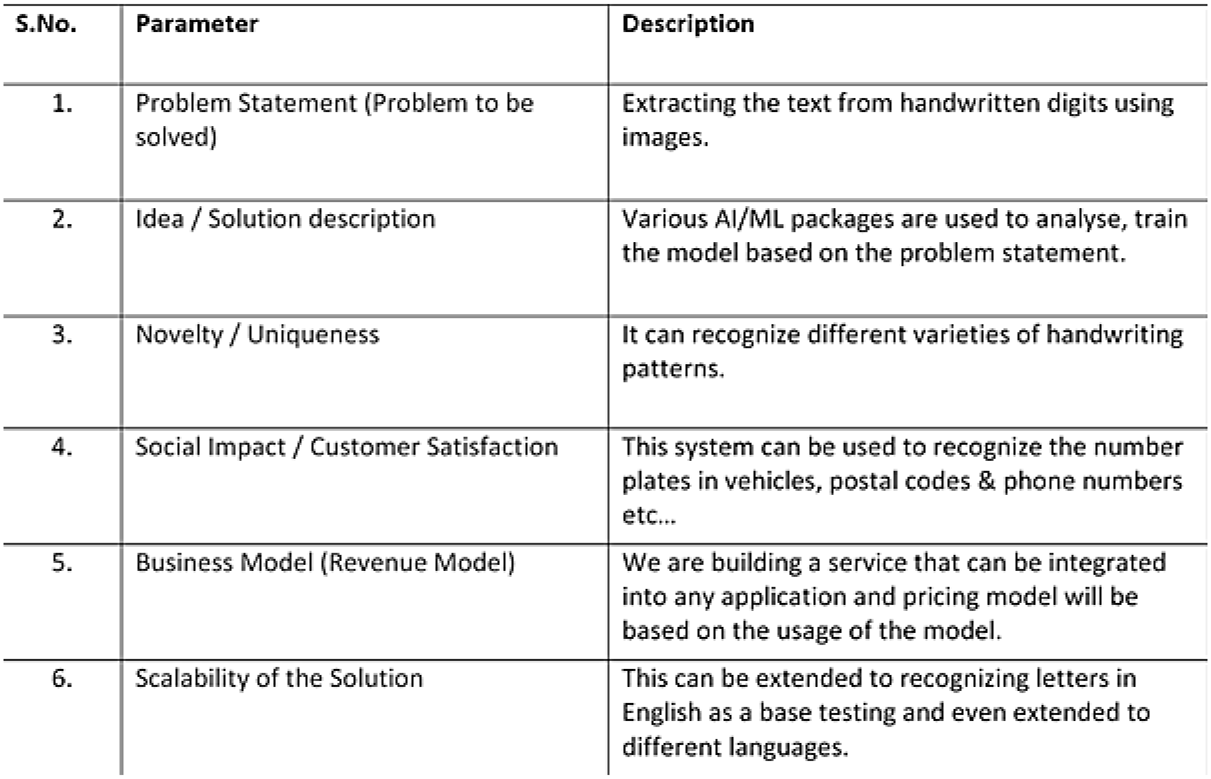
<u>3.2 IDEATION AND BRAINSTROMING</u>:



Brainstorm & Idea prioritization

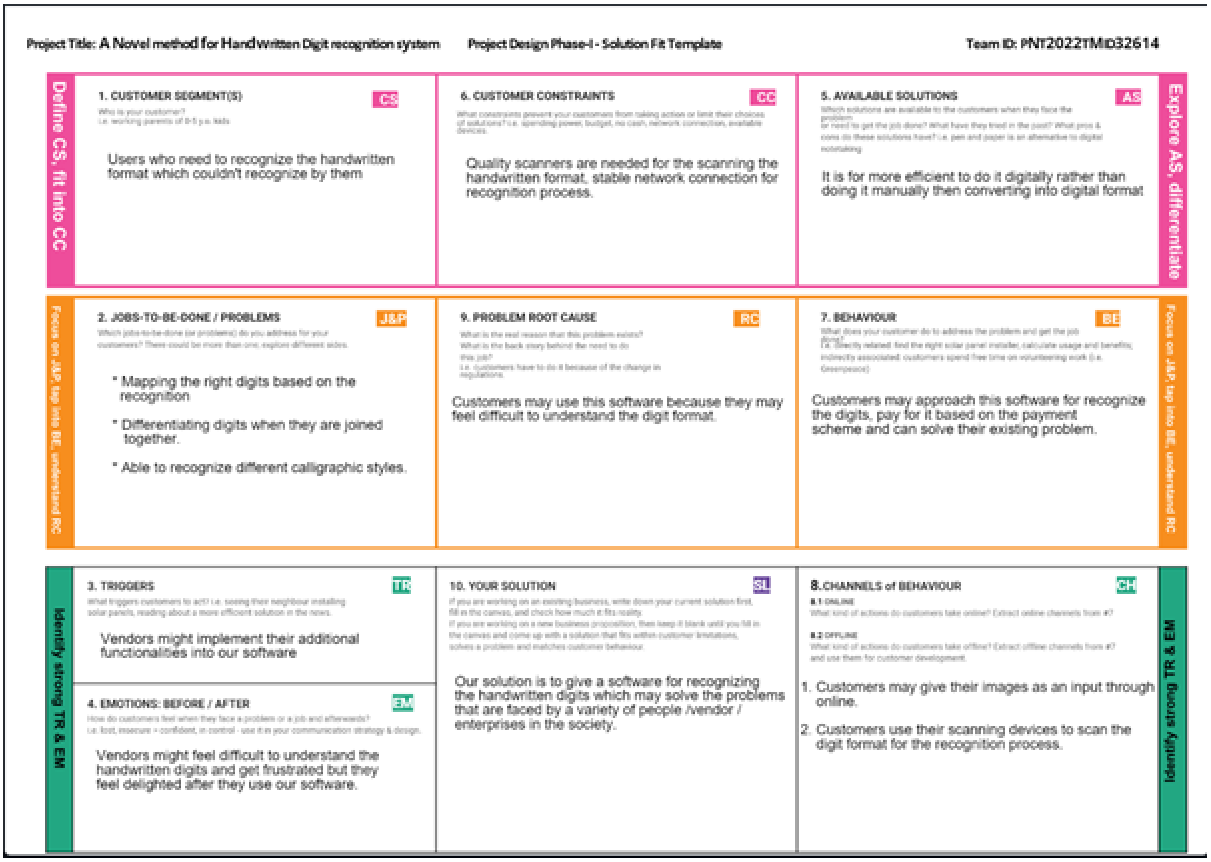
Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

3.3 PROPOSED SOLUTION:



S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Extracting the text from handwritten digits using images.
2.	Idea / Solution description	Various AI/ML packages are used to analyse, train the model based on the problem statement.
3.	Novelty / Uniqueness	It can recognize different varieties of handwriting patterns.
4.	Social Impact / Customer Satisfaction	This system can be used to recognize the number plates in vehicles, postal codes & phone numbers etc...
5.	Business Model (Revenue Model)	We are building a service that can be integrated into any application and pricing model will be based on the usage of the model.
6.	Scalability of the Solution	This can be extended to recognizing letters in English as a base testing and even extended to different languages.

3.4 PROBLEM SOLUTION FIT:



Project Title: A Novel method for Handwritten Digit recognition system

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID32614

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

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

Users who need to recognize the handwritten format which couldn't recognize by them

6. CUSTOMER CONSTRAINTS CC

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

Quality scanners are needed for the scanning the handwritten format, stable network connection for recognition process.

5. AVAILABLE SOLUTIONS AS

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

It is for more efficient to do it digitally rather than doing it manually then converting into digital format

2. JOBS-TO-BE-DONE / PROBLEMS J&P

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

* Mapping the right digits based on the recognition
* Differentiating digits when they are joined together.
* Able to recognize different calligraphic styles.

9. PROBLEM ROOT CAUSE RC

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

Customers may use this software because they may feel difficult to understand the digit format.

7. BEHAVIOUR BE

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

Customers may approach this software for recognize the digits, pay for it based on the payment scheme and can solve their existing problem.

3. TRIGGERS TR

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

Vendors might implement their additional functionalities into our software

4. EMOTIONS: BEFORE / AFTER EM

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

Vendors might feel difficult to understand the handwritten digits and get frustrated but they feel delighted after they use our software.

10. YOUR SOLUTION SL

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

Our solution is to give a software for recognizing the handwritten digits which may solve the problems that are faced by a variety of people /vendor / enterprises in the society.

8.CHANNELS of BEHAVIOUR CH

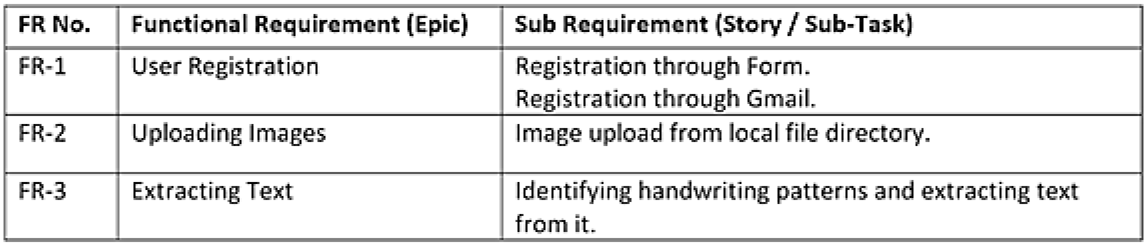
8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

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

1. Customers may give their images as an input through online.
2. Customers use their scanning devices to scan the digit format for the recognition process.

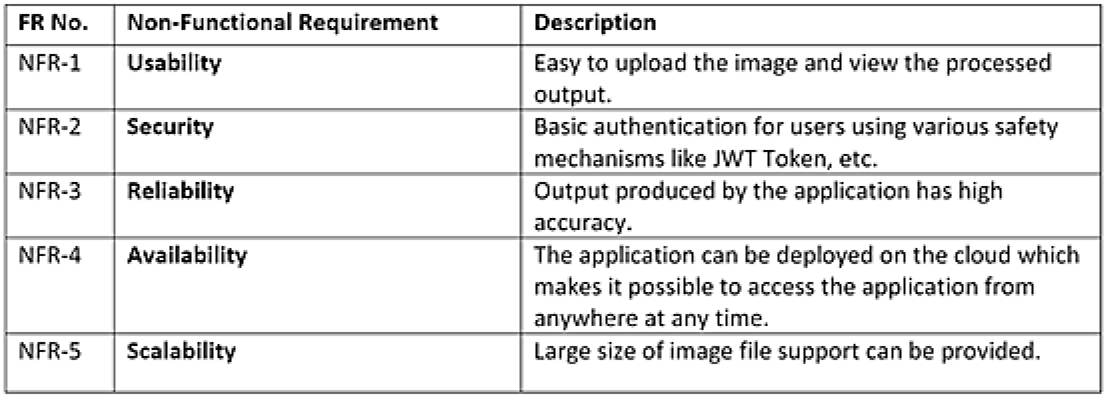
4. REQUIREMENT ANALYSIS:

4.1 FUNCTIONAL REQUIREMENT:



FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form. Registration through Gmail.
FR-2	Uploading Images	Image upload from local file directory.
FR-3	Extracting Text	Identifying handwriting patterns and extracting text from it.

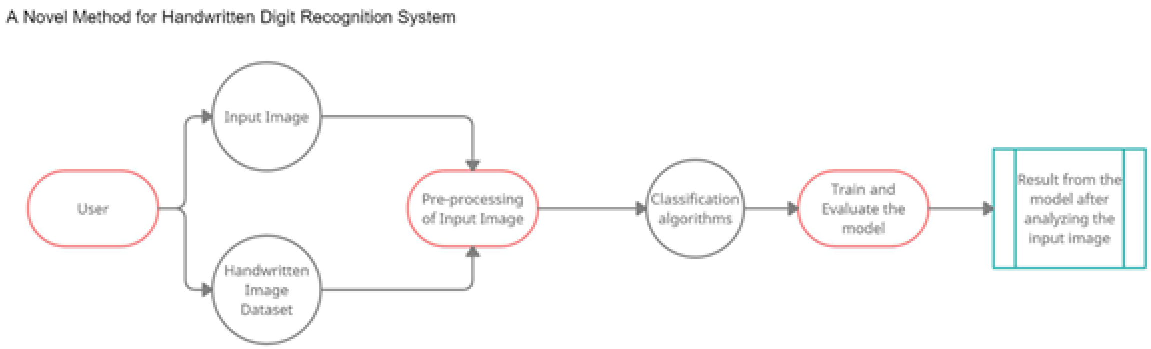
4.2 NON-FUNCTIONAL REQUIREMENT:



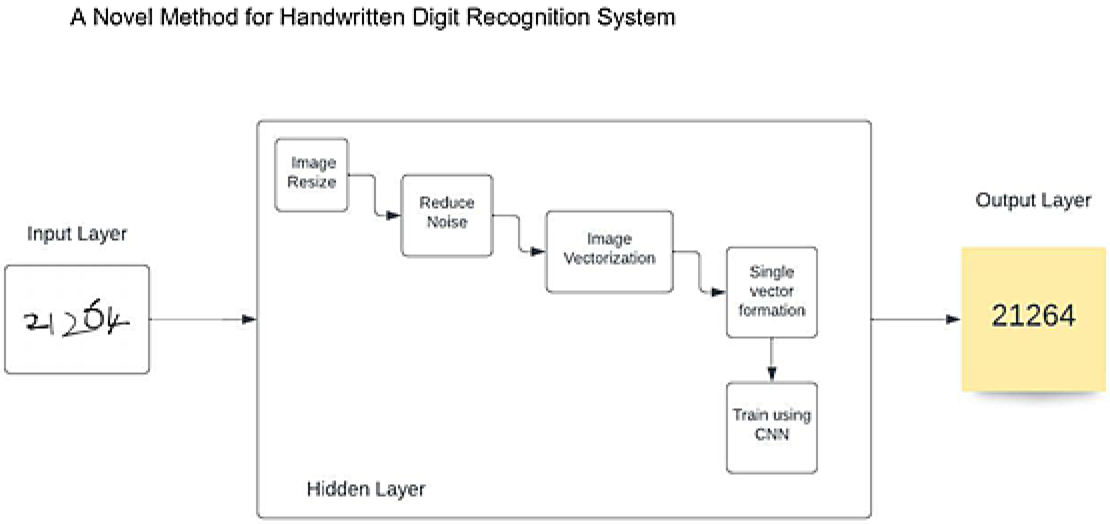
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Easy to upload the image and view the processed output.
NFR-2	**Security**	Basic authentication for users using various safety mechanisms like JWT Token, etc.
NFR-3	**Reliability**	Output produced by the application has high accuracy.
NFR-4	**Availability**	The application can be deployed on the cloud which makes it possible to access the application from anywhere at any time.
NFR-5	**Scalability**	Large size of image file support can be provided.

5. PROJECT DESIGN:

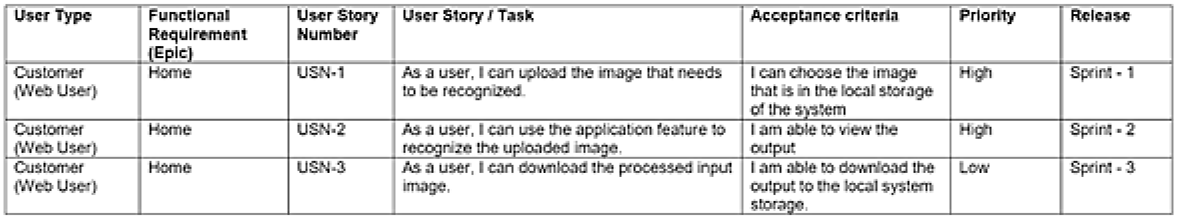
5.1 DATA FLOW DIAGRAM:


A Novel Method for Handwritten Digit Recognition System
User
Input Image
Handwritten Image Dataset
Pre-processing of Input Image
Classification algorithms
Train and Evaluate the model
Result from the model after analyzing the input image

5.2 SOLUTION AND TECHNICAL ARCHITECTURE:



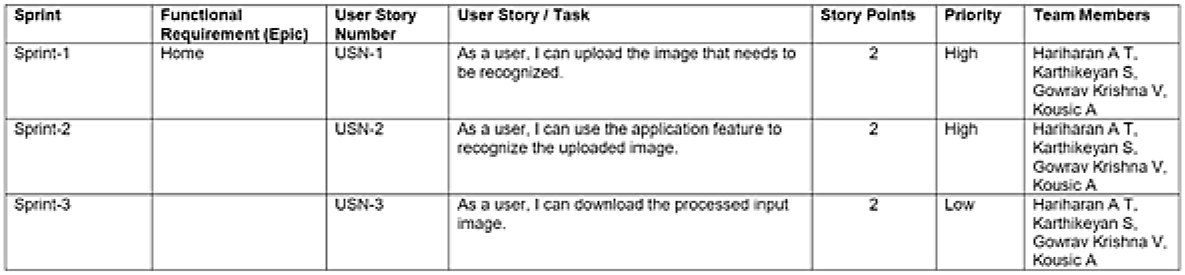
5.3 USER STORIES:



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web User)	Home	USN-1	As a user, I can upload the image that needs to be recognized.	I can choose the image that is in the local storage of the system	High	Sprint - 1
Customer (Web User)	Home	USN-2	As a user, I can use the application feature to recognize the uploaded image.	I am able to view the output	High	Sprint - 2
Customer (Web User)	Home	USN-3	As a user, I can download the processed input image.	I am able to download the output to the local system storage.	Low	Sprint - 3

6. PROJECT PLANNING AND SCHEDULING:

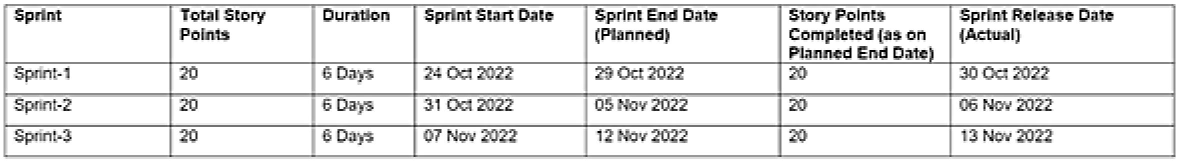
6.1 SPRINT PLANNING AND ESTIMATION:



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Home	USN-1	As a user, I can upload the image that needs to be recognized.	2	High	Hariharan A T, Karthikeyan S, Gowrav Krishna V, Kousic A
Sprint-2		USN-2	As a user, I can use the application feature to recognize the uploaded image.	2	High	Hariharan A T, Karthikeyan S, Gowrav Krishna V, Kousic A
Sprint-3		USN-3	As a user, I can download the processed input image.	2	Low	Hariharan A T, Karthikeyan S, Gowrav Krishna V, Kousic A

ESTIMATION: 8000 rupees for Server hosting charges.

6.2 SPRINT DELIVERY SCHEDULE:



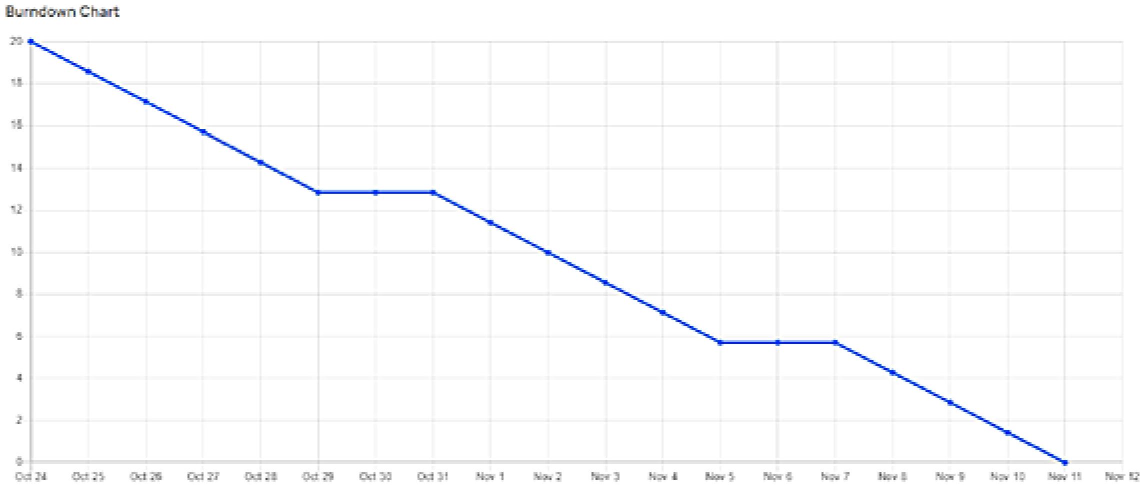
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	30 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	06 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	13 Nov 2022

6.3 REPORTS FROM JIRA:

Velocity:

$$AV = \frac{\text{sprint duration}}{\text{velocity}} = \frac{20}{6} = 3.3\bar{3}$$

Burndown Chart:



7. CODING & SOLUTIONING:

7.1 FEATURE 1:

The user can upload the input in our application through the upload button.

Code:

```
    <form action="/predict" method="POST" enctype="multipart/form-data">
      <div class="upload-btn-wrapper">
            <button class="button-62">Choose</button>
          <input type="file" id="fileButton" name="file" />
      </div>
     <input class="button-62" type="submit">
  </form>
```

7.2 FEATURE 2:

The given input will be sent to the model, processed and the model returns the predicted output through html page.

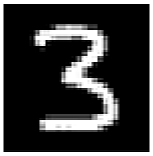
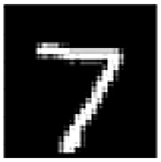
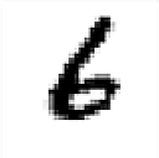
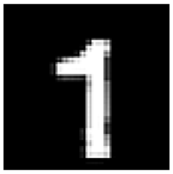
Code:

```
@app.route('/predict', methods=['POST'])
 def upload_image_file():
    if request.method == 'POST':
        model = load_model(r'models/mnistCNN.h5')
        img = Image.open(request.files['file'].stream).convert("L")
```

```
         img = img.resize((28, 28))
         im2arr = np.array(img)
         im2arr = im2arr.reshape(1, 28, 28, 1)
         y_pred = model.predict(im2arr)
         result = np.argmax(y_pred, axis=1)
         print(result)
         if (result == 0):
            return render_template("0.html", showcase=str(result))
       elif (result == 1):
            return render_template("1.html", showcase=str(result))
       elif (result == 2):
            return render_template('2.html', showcase=str(result))
       elif (result == 3):
            return render_template('4.html', showcase=str(result))
       elif (result == 4):
           return render_template('5.html', showcase=str(result))
     elif (result == 6):
         return render_template('6.html', showcase=str(result))
     elif (result == 7):
         return render_template('7.html', showcase=str(result))
     elif (result == 8):
         return render_template('8.html', showcase=str(result))
     else:
         return render_template('9.html', showcase=str(result))
else:
    return None
```

8. TESTING:

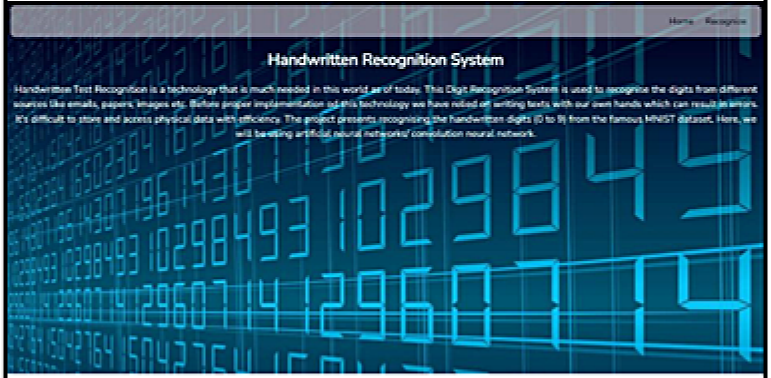
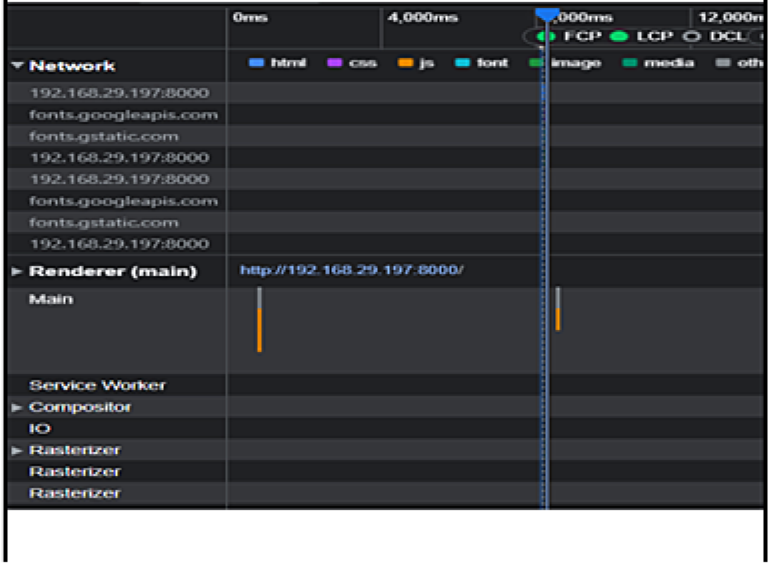
8.1 TEST CASES:

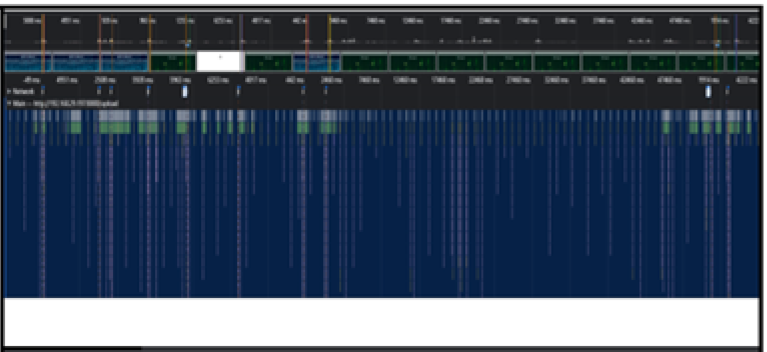
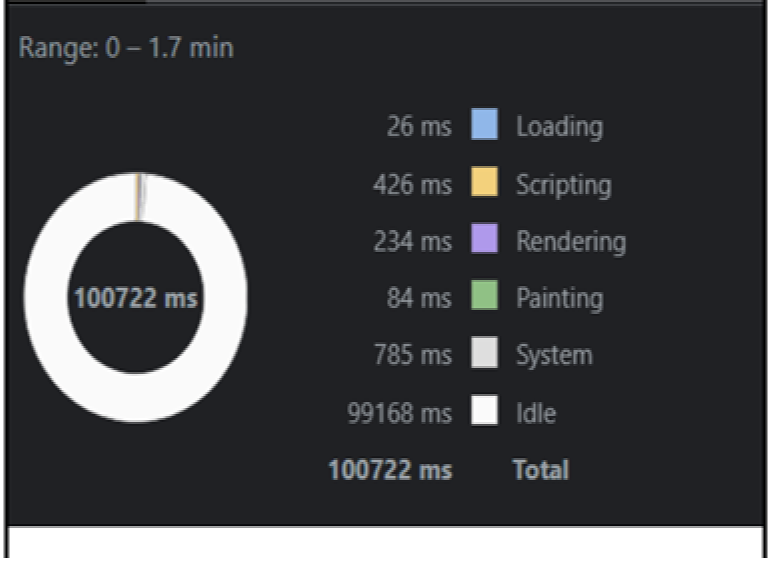
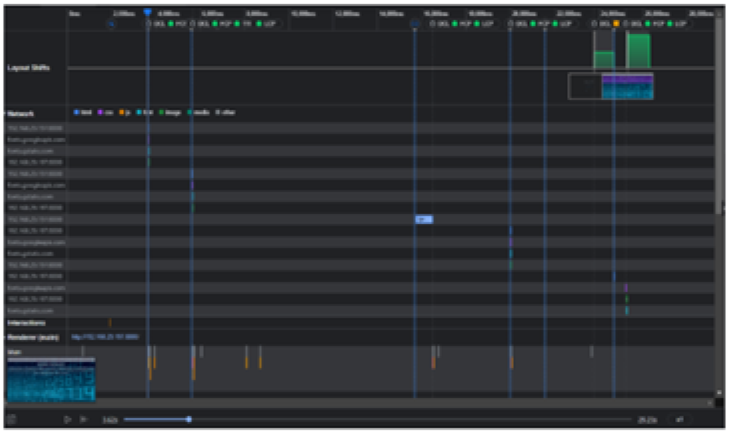
S.No	Input	Expected output	Actual output
1.		3	3
2.		7	7
3.		6	6
4.		1	1

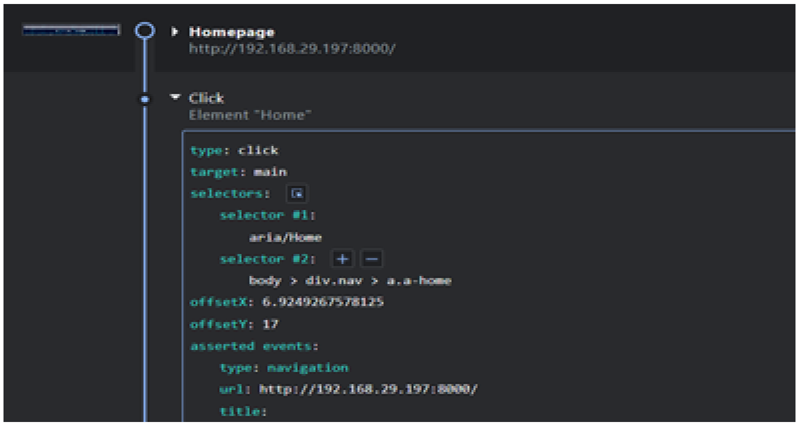
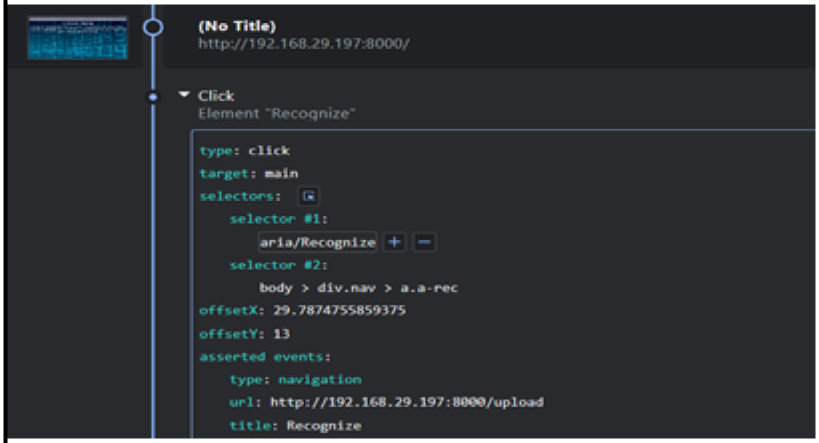
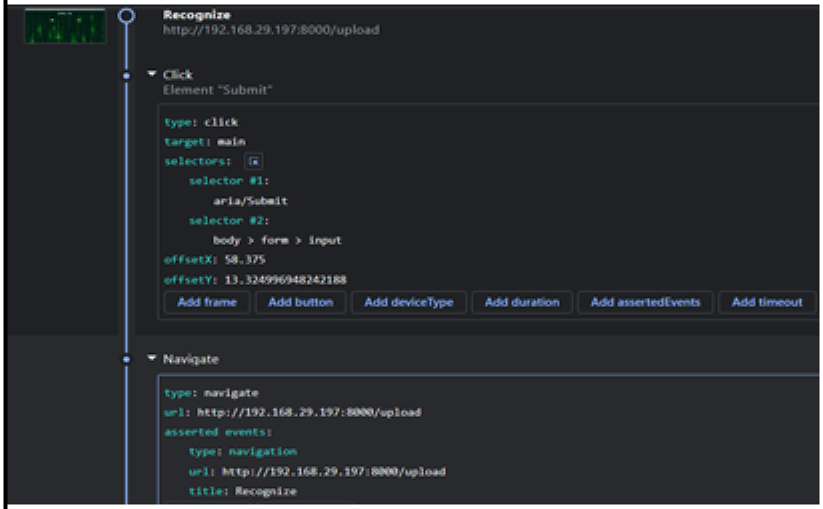
MODEL PERFORMANCE TESTING:

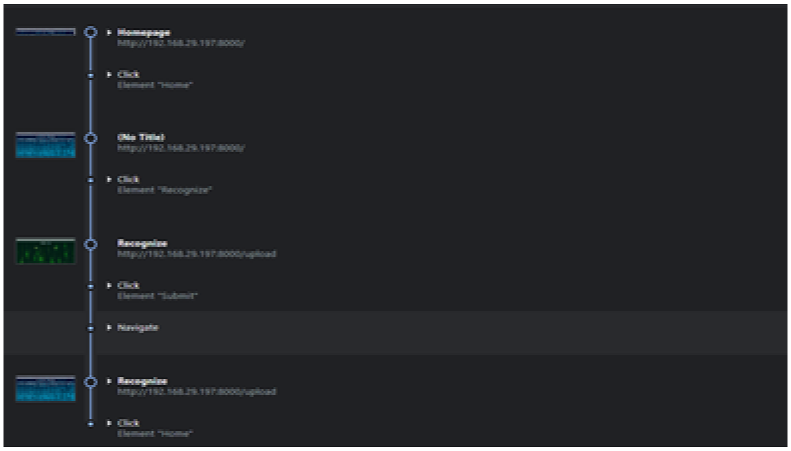
Accuracy	Trained accuracy – 95.4 % Validation accuracy - 90.8 %	(pre-trained model)

DATA ANALYTICS:

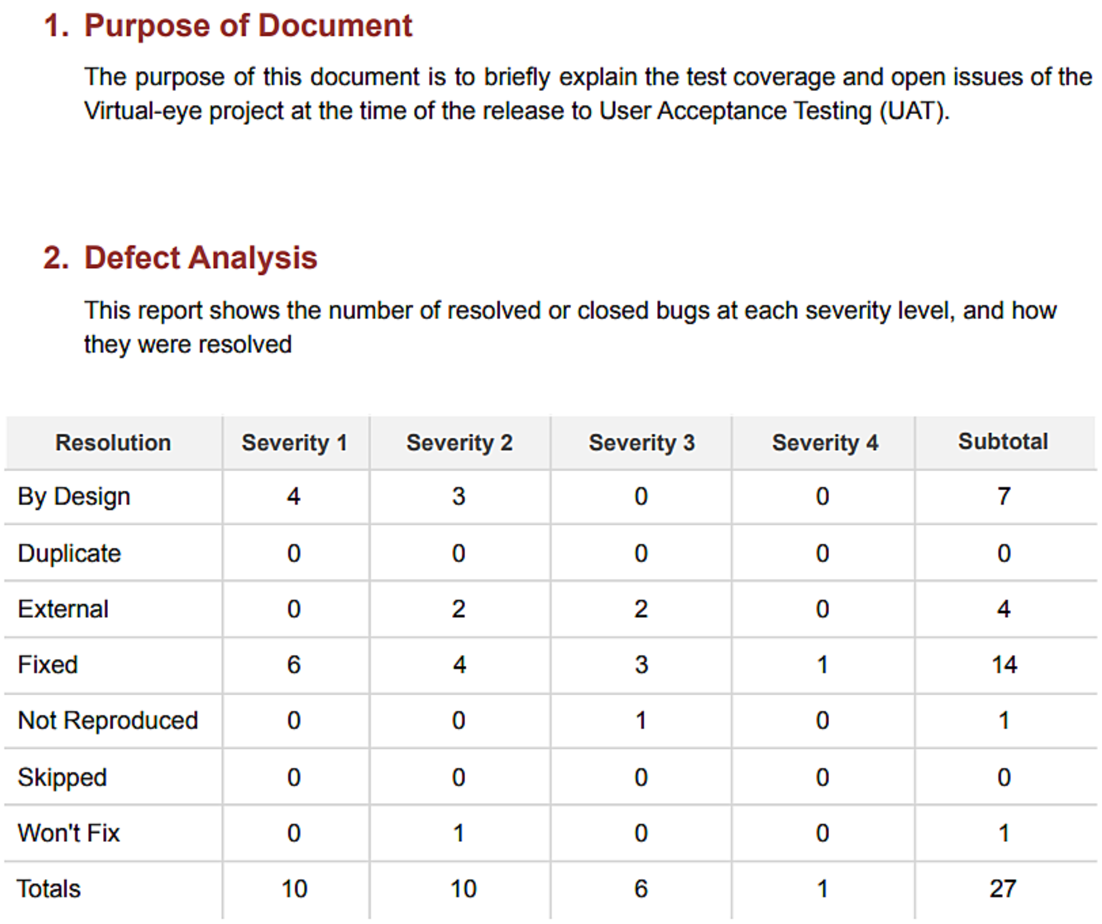
S.No.	Parameter	Screenshot / Values
1.	Dashboard design	No of Visualizations / Graphs - 2  Handwritten Recognition System Handwritten Text Recognition is a technology that is much needed in this world as of today. This Digit Recognition System is used to recognise the digits from different sources like emails, papers, images etc. Before proper implementation of this technology we have relied on writing texts with our own hands which can result in errors. It's difficult to store and access physical data with efficiency. The project presents recognising the handwritten digits (0 to 9) from the famous MNIST dataset. Here, we will be using artificial neural networks/ convolution neural network. 

2.	Data Responsiveness	  Range: 0 – 1.7 min 100722 ms 26 ms 🟦 Loading 426 ms 🟨 Scripting 234 ms 🟪 Rendering 84 ms 🟩 Painting 785 ms ⬜ System 99168 ms ⬜ Idle **100722 ms** **Total**
3.	Amount Data to Rendered (DB2 Metrics)	

4.	Utilization of Data Filters	 Homepage http://192.168.29.197:8000/ Click Element "Home" type: click target: main selectors: selector #1: aria/Home selector #2: body > div.nav > a.a-home offsetX: 6.9249267578125 offsetY: 17 asserted events: type: navigation url: http://192.168.29.197:8000/ title:  (No Title) http://192.168.29.197:8000/ Click Element "Recognize" type: click target: main selectors: selector #1: aria/Recognize selector #2: body > div.nav > a.a-rec offsetX: 29.7874755859375 offsetY: 13 asserted events: type: navigation url: http://192.168.29.197:8000/upload title: Recognize  Recognize http://192.168.29.197:8000/upload Click Element "Submit" type: click target: main selectors: selector #1: aria/Submit selector #2: body > form > input offsetX: 58.375 offsetY: 13.324996948242188 Add frame Add button Add deviceType Add duration Add assertedEvents Add timeout Navigate type: navigate url: http://192.168.29.197:8000/upload asserted events: type: navigation url: http://192.168.29.197:8000/upload title: Recognize

5.	Effective UserStory	No of Scene Added- 1  Homepage http://192.168.29.197:8000/ Click Element "Home" (No Title) http://192.168.29.197:8000/ Click Element "Recognize" Recognize http://192.168.29.197:8000/upload Click Element "Submit" Navigate Recognize http://192.168.29.197:8000/upload Click Element "Home"

8.2 USER ACCEPTANCE TESTING:



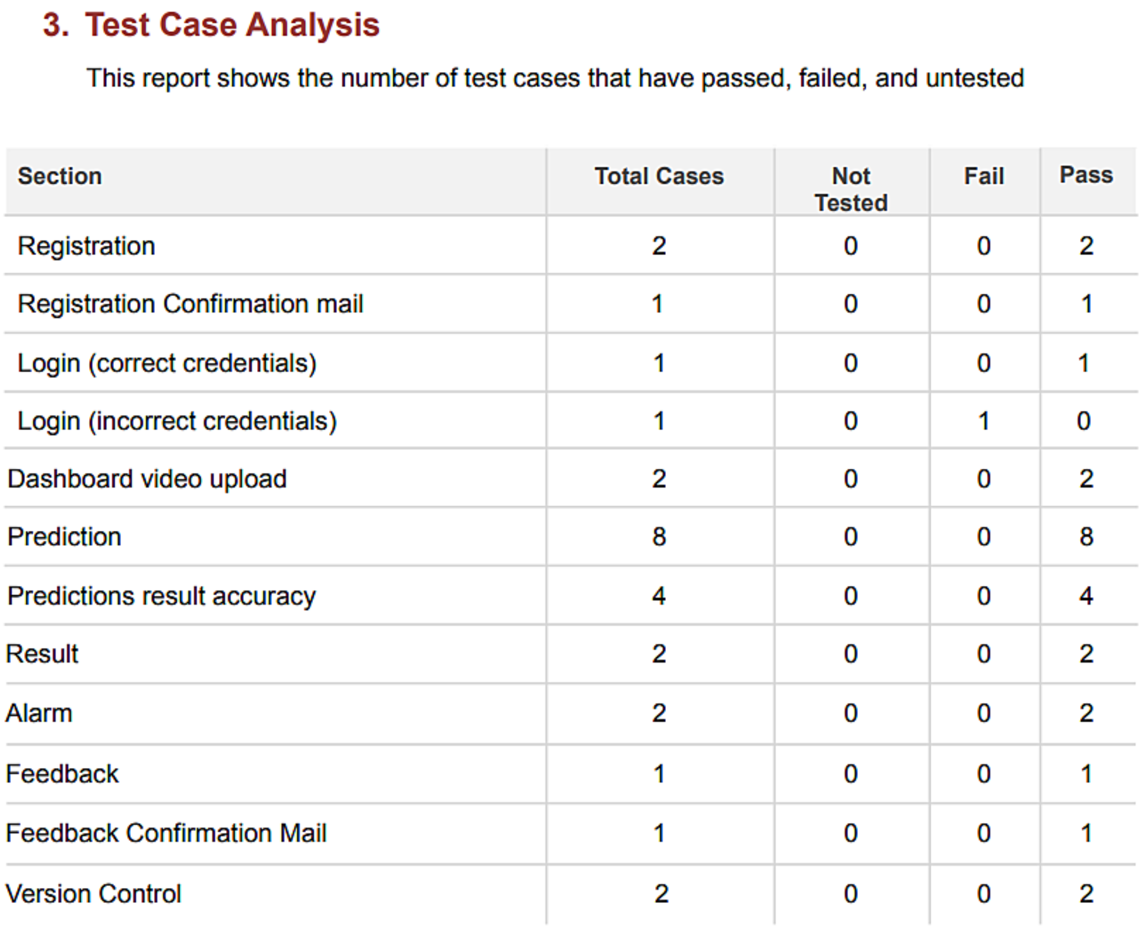
1. Purpose of Document

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

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	4	3	0	0	7
Duplicate	0	0	0	0	0
External	0	2	2	0	4
Fixed	6	4	3	1	14
Not Reproduced	0	0	1	0	1
Skipped	0	0	0	0	0
Won't Fix	0	1	0	0	1
Totals	10	10	6	1	27



3. Test Case Analysis

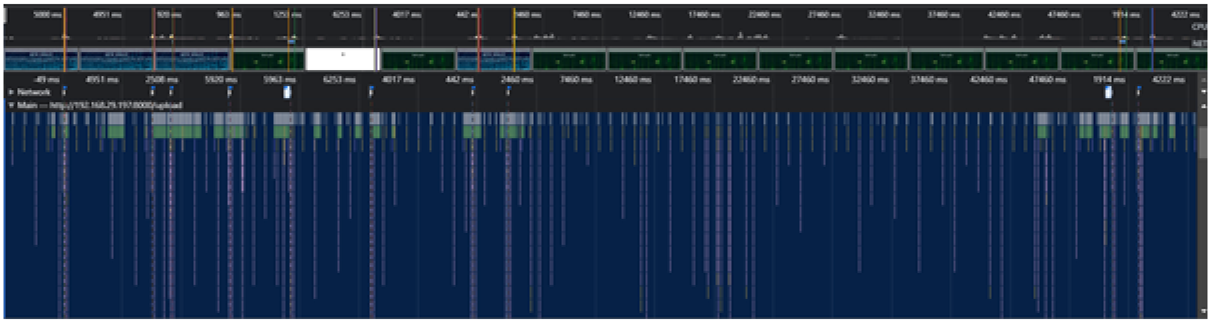
This report shows the number of test cases that have passed, failed, and untested

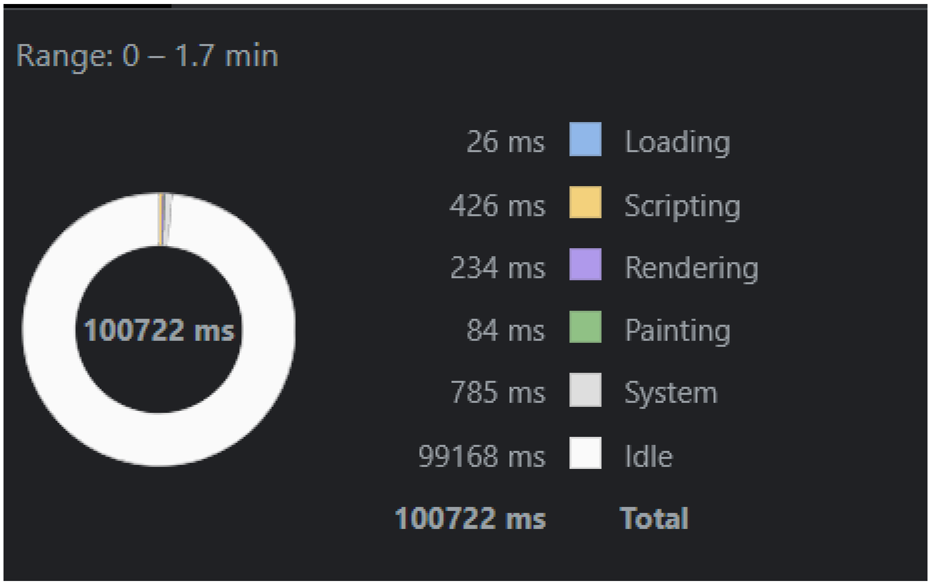
Section	Total Cases	Not Tested	Fail	Pass
Registration	2	0	0	2
Registration Confirmation mail	1	0	0	1
Login (correct credentials)	1	0	0	1
Login (incorrect credentials)	1	0	1	0
Dashboard video upload	2	0	0	2
Prediction	8	0	0	8
Predictions result accuracy	4	0	0	4
Result	2	0	0	2
Alarm	2	0	0	2
Feedback	1	0	0	1
Feedback Confirmation Mail	1	0	0	1
Version Control	2	0	0	2

9. RESULTS:

Thus, this model can be implemented in the areas where the handwritten digits cannot be recognized and it may solve the other problems that are related to the handwritten digits.

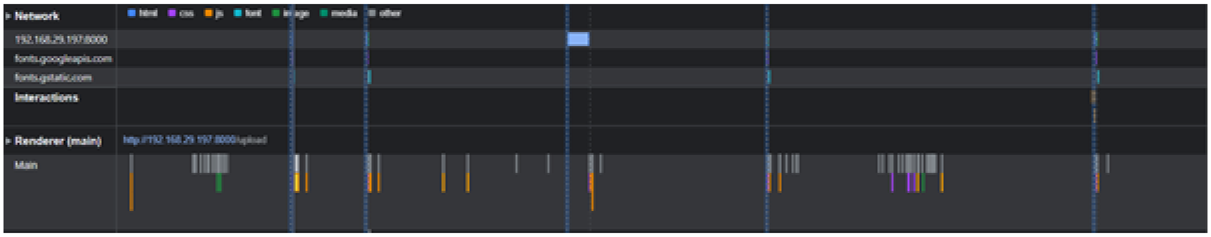
9.1 PERFORMANCE METRICS:





Range: 0 – 1.7 min

26 ms	🟦	Loading
426 ms	🟨	Scripting
234 ms	🟪	Rendering
84 ms	🟩	Painting
785 ms	⬜	System
99168 ms	⬜	Idle
100722 ms		**Total**



10. ADVANTAGES & DISADVANTAGES:

ADVANTAGES:

- Easier data retrieval
- Historical preservation
- Reliable output

DISADVANTAGES:

- Poor images of handwritten digits
- Single image can be recognized

11. CONCLUSION:

Thus our project solves the difficulties that are faced while analyzing the handwritten digits. The model that we developed, will produce the accurate results for the given input image.

12. FUTURE SCOPE:

- Support for large image file.
- Text or other recognition can be supported.
- It can extended to a series of digits to be recognized.

13. APPENDIX:

13.1 SOURCE CODE:

- app.py:

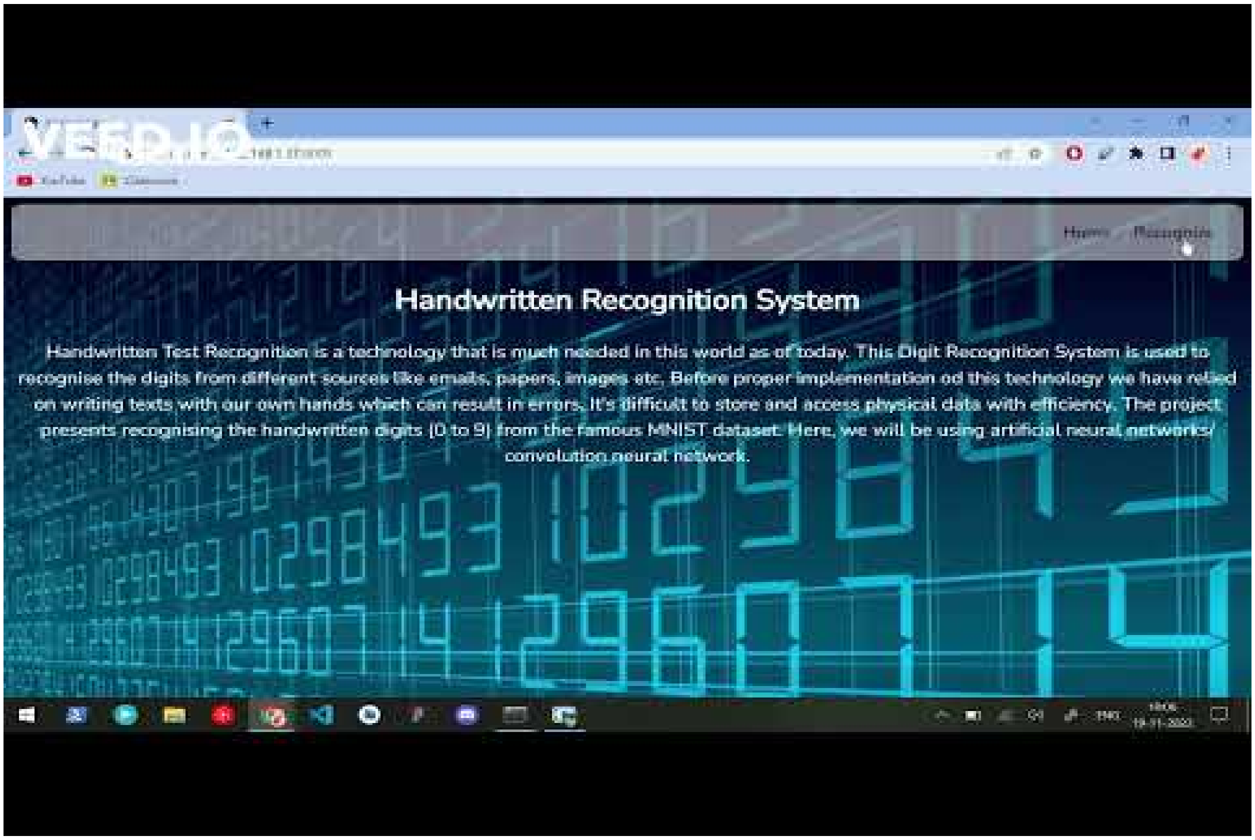
```
from flask import Flask, render_template, request
from PIL import Image
import numpy as np
from tensorflow.keras.models import load_model, Sequential
from tensorflow.keras.models import Sequential
app = Flask(__name__)
```

```
  @app.route('/')
  def upload_file():
     return render_template('index.html')
  @app.route('/upload')
  def upload_file2():
     return render_template('web.html')
 @app.route('/predict', methods=['POST'])
 def upload_image_file():
   if request.method == 'POST':
       model = load_model(r'models/mnistCNN.h5')
       img = Image.open(request.files['file'].stream).convert("L")
       img = img.resize((28, 28))
       im2arr = np.array(img)
         im2arr = im2arr.reshape(1, 28, 28, 1)
         y_pred = model.predict(im2arr)

result = np.argmax(y_pred, axis=1)

 if (result == 0):
           return render_template("0.html", showcase=str(result))
 elif (result == 1):
           return render_template("1.html", showcase=str(result))
 elif (result == 2):
           return render_template('2.html', showcase=str(result))
```

```
                elif (result == 3):
                        return render_template('4.html', showcase=str(result))
                elif (result == 4):
                        return render_template('5.html', showcase=str(result))
                elif (result == 6):
                        return render_template('6.html', showcase=str(result))
                elif (result == 7):
                        return render_template('7.html', showcase=str(result))
                elif (result == 8):
                        return render_template('8.html', showcase=str(result))
                else:
                        return render_template('9.html', showcase=str(result))
        else:
            return None
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8000, debug=True)
```

13.2 GITHUB & PROJECT DEMO LINK:

- Github link: https://github.com/IBM-EPBL/IBM-Project-8744-1658928430
- Project Demo link:



Handwritten Recognition System

Handwritten Test Recognition is a technology that is much needed in this world as of today. This Digit Recognition System is used to recognise the digits from different sources like emails, papers, images etc. Before proper implementation od this technology we have relied on writing texts with our own hands which can result in errors. It's difficult to store and access physical data with efficiency. The project presents recognising the handwritten digits (0 to 9) from the famous MNIST dataset. Here, we will be using artificial neural networks/ convolution neural network.