PROJECT REPORT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION

submitted by

PNT2022TMID20903

Sandhya L - 412419205079

Roshma Eugin E R - 412419205075

Swathy M-412419205092

Sai Sanjana N- 412419205076

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 PROJECT OVERVIEW

Machine learning and deep learning play an important role in computer technology and artificial intelligence. With the use of deep learning and machine learning, human effort can be reduced in recognizing, learning, predictions and in many more areas.

Handwritten Digit Recognition is the ability of computer systems to recognise handwritten digits from various sources, such as images, documents, and so on. This project aims to let users take advantage of machine learning to reduce manual tasks in recognizing digits.

1.2 PURPOSE

Digit recognition systems are capable of recognizing the digits from different sources like emails, bank cheque, papers, images, etc. and in different real-world scenarios for online handwriting recognition on computer tablets or system, recognize number plates of vehicles, processing bank cheque amounts, numeric entries in forms filled up by hand (tax forms) and so on.

CHAPTER 2
LITERATURE SURVEY

2.1 EXISTING PROBLEM

The fundamental problem with handwritten digit recognition is that handwritten digits do not always have the same size, width, orientation, and margins since they vary from person to person. Additionally, there would be issues with identifying the numbers because of similarities between numerals like 1 and 7, 5 and 6, 3 and 8, 2 and 5, 2 and 7, etc. Finally, the individuality and variation of each individual's handwriting influence the structure and appearance of the digits.

2.2 REFERENCES

Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN) (2020)

Ahlawat, Savita and Choudhary, Amit and Nayyar, Anand and Singh, Saurabh and Yoon, Byungun

This paper's primary goal was to enhance handwritten digit recognition ability. To avoid difficult pre-processing, expensive feature extraction, and a complex ensemble (classifier combination) method of a standard recognition system, they examined different convolutional neural network variations. Their current work makes suggestions on the function of several hyper-parameters through thorough evaluation utilizing an MNIST dataset. They also confirmed that optimizing

hyper-parameters is crucial for enhancing CNN architecture performance. With the Adam optimizer for the MNIST database, they were able to surpass many previously published results with a recognition rate of 99.89%. Through the trials, it is made
abundantly evident how the performance of handwritten digit recognition is affected by the number of convolutional layers in CNN architecture. According to the paper, evolutionary algorithms can be explored for optimizing convolutional filter kernel sizes, CNN learning parameters, and the quantity of layers and learning rates.

An Efficient And Improved Scheme For Handwritten Digit Recognition Based On Convolutional Neural Network (2019)

Ali, Saqib and Shaukat, Zeeshan and Azeem, Muhammad and Sakhawat, Zareen and Mahmood, Tariq and others

This study uses rectified linear units (ReLU) activation and a convolutional neural network (CNN) that incorporates the Deeplearning4j (DL4J) architecture to recognize handwritten digits. The proposed CNN framework has all the necessary parameters for a high level of MNIST digit classification accuracy. The system's training takes into account the time factor as well. The system is also tested by altering the number of CNN layers for additional accuracy verification. It is important to note that the CNN architecture consists of two convolutional layers, the first with 32 filters and a 5x5 window size and the second with 64 filters and a 7x7 window size. In comparison to earlier proposed systems, the experimental findings show that the proposed CNN architecture for the MNIST dataset demonstrates great performance in terms of time and accuracy. As a result, handwritten numbers are detected with a recognition rate of 99.89% and high precision (99.21%) in a short amount of time.

Improved Handwritten Digit Recognition Using Quantum K-Nearest Neighbor

Algorithm (2019)

Wang, Yuxiang and Wang, Ruijin and Li, Dongfen and Adu-Gyamfi, Daniel and Tian, Kaibin and Zhu, Yixin

The KNN classical machine learning technique is used in this research to enable quantum parallel computing and superposition. They used the KNN algorithm with quantum acceleration to enhance handwritten digit recognition. When dealing with more complicated and sizable handwritten digital data sets, their suggested method considerably lowered the computational time complexity of the traditional KNN algorithm. The paper offered a theoretical investigation of how quantum concepts can be applied to machine learning. Finally, they established a fundamental operational concept and procedure for machine learning with quantum acceleration.

Handwritten Digit Recognition Using Machine And Deep Learning Algorithms (2021)

Pashine, Samay and Dixit, Ritik and Kushwah, Rishika

In this study, they developed three deep and machine learning-based models for handwritten digit recognition using MNIST datasets. To determine which model was the most accurate, they compared them based on their individual properties. Support vector machines are among the simplest classifiers, making them faster than other algorithms and providing the highest training accuracy rate in this situation. However, due to their simplicity, SVMs cannot categorize complicated and ambiguous images as accurately as MLP and CNN algorithms can. In their research, they discovered that CNN produced the most precise outcomes for handwritten digit recognition. This led them to the conclusion that CNN is the most effective
solution for all types of prediction issues, including those using picture data.

Next, by comparing the execution times of the algorithms, they determined that increasing the number of epochs without changing the configuration of the algorithm is pointless due to the limitation of a certain model, and they discovered that beyond a certain number of epochs, the model begins over-fitting the dataset and provides biased predictions.

2.3 PROBLEM STATEMENT DEFINITION

For years, the traffic department has been combating traffic law violators. These offenders endanger not only their own lives, but also the lives of other individuals. Punishing these offenders is critical to ensuring that others do not become like them. Identification of these offenders is next to impossible because it is impossible for the average individual to write down the license plate of a reckless driver. Therefore, the goal of this project is to help the traffic department identify these offenders and reduce traffic violations as a result.

CHAPTER 3
IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

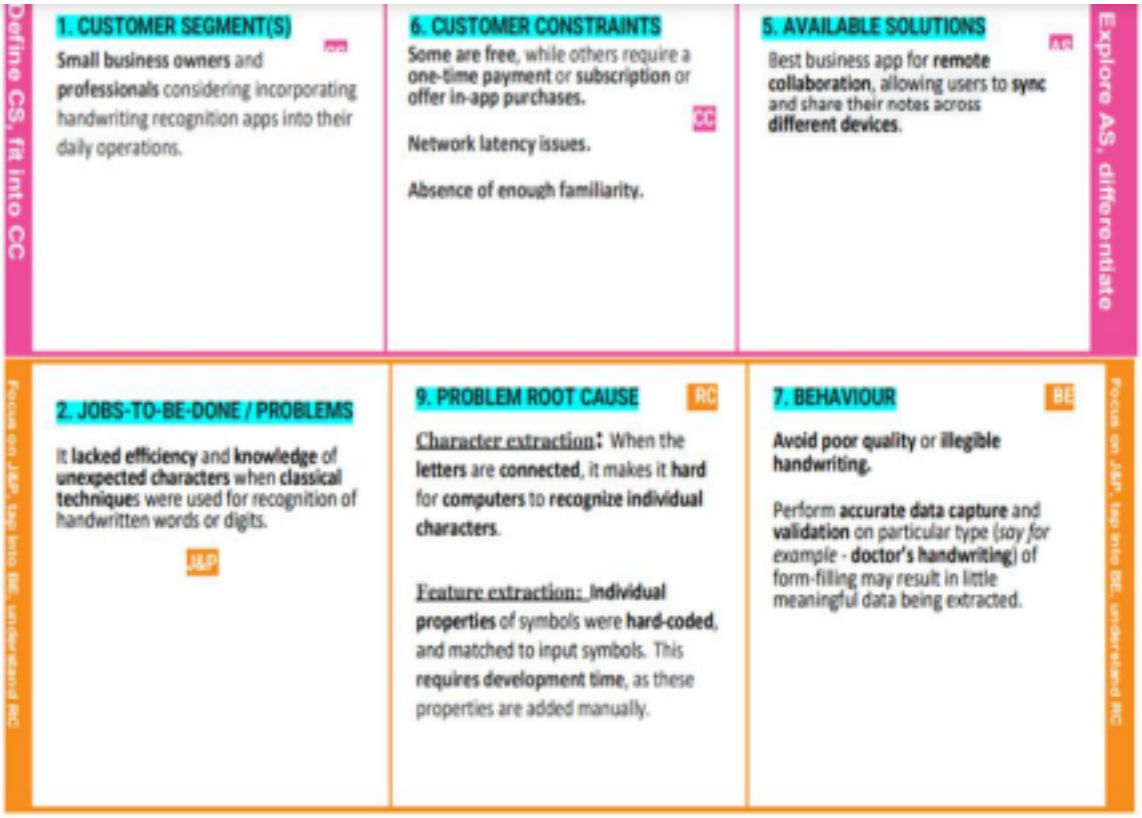
3.2 IDEATION & BRAINSTORMING

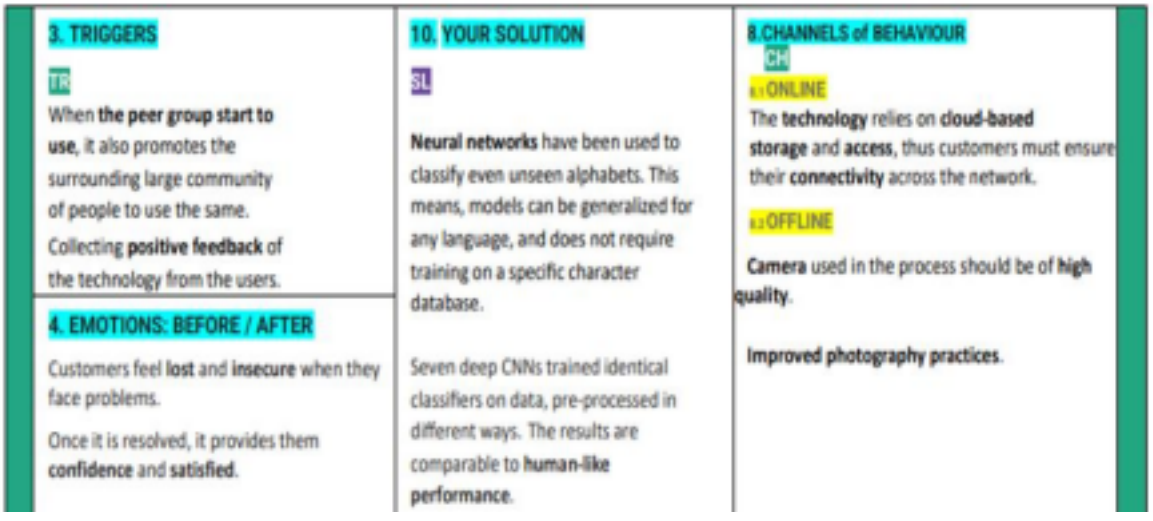
3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically. Because of the progress in the field of science and technology, everything is being digitized to reduce human effort. Hence, there comes a need for handwritten digit recognition in many real-time applications. Most of the banks are still relying on manual cheque processing which is both time consuming and error prone. Handwriting recognition systems with reliable accuracy can make an impact in these business fields.
2.	Idea / Solution description	The MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. We use Artificial neural networks to train these images and build a deep learning model. We opt to use multi-layer neural networks as deep NN. Due to the fact that data is Image, the best type of neural network satisfying our goal is **Convolutional Neural Networks**. As we have to do for most of the data, normalization plays an important role in our process. Before doing any tasks, pre-processing images (our data-set) is highly recommended. Consequently better accuracy will be achieved by pre-processed data. After pre-processing and normalizing, the prepared data set could be used as input to our deep convolutional neural network. Then deep NN will be run and fit to our data and the result will be produced by that.
3.	Novelty / Uniqueness	Web application is created where the user can upload an image of a handwritten digit. This image is analyzed by the model and the detected result is returned to the UI. One of the major decisions had to be made was choosing

		the suitable programming language satisfying our goal for extracting knowledge from our data. After some searching the suitable decision has been made by selecting Python as the project programming language. Due to the fact that, a lot of tools and frameworks are available for Python to create powerful Artificial Neural Networks. Also IBM Watson helps to predict future outcomes,automate complex processes,and optimize user's time. And also the result accuracy will be increased from 70% which is the accuracy of the test results that the previous developed codes produced.
4.	Social Impact / Customer Satisfaction	Can ensure road safety by identifying the owner of the speeding vehicle by using the registration number of that vehicle. Can automate data entry jobs and speed up cheque approval process
5.	Business Model (Revenue Model)	Can collaborate with banks to speedup the cheque approval process which improves customer experience.
6.	Scalability of the Solution	This project will help us to detect digits more precisely. Also we can develop this model to recognize alphabets.

3.4 PROBLEM SOLUTIONs





CHAPTER 4

4.1 FUNCTIONAL REQUIREMENTS

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Sub Requirement (Story / Sub-Task)
FR-1	Image Data: Handwritten digit recognition refers to a computer's capacity to identify human handwritten digits from a variety of sources, such as photographs, documents, touch screens, etc., and categorize them into ten established classifications (0-9). In the realm of deep learning, this has been the subject of countless studies.
FR-2	Website: Web hosting makes the code, graphics, and other items that make up a website accessible online. A server hosts every website you've ever visited. The type of hosting determines how much space is allotted to a website on a server. Shared, dedicated, VPS, and reseller hosting are the four basic varieties.
FR-3	Digit Classifier Model: To train a convolutional network to predict the digit from an image, use the MNIST database of handwritten digits and get the training and validation data first.
FR-4	Cloud: The cloud offers a range of IT services, including virtual storage, networking, servers, databases, and applications. In plain English, cloud computing is described as a virtual platform that enables unlimited storage and access to your data over the internet.
FR-5	Modified National Institute of Standards and Technology dataset: The abbreviation MNIST stands for the MNIST dataset. It is a collection of 60,000 tiny square grayscale photographs, each measuring 28 by 28, comprising handwritten single digits between 0 and 9.

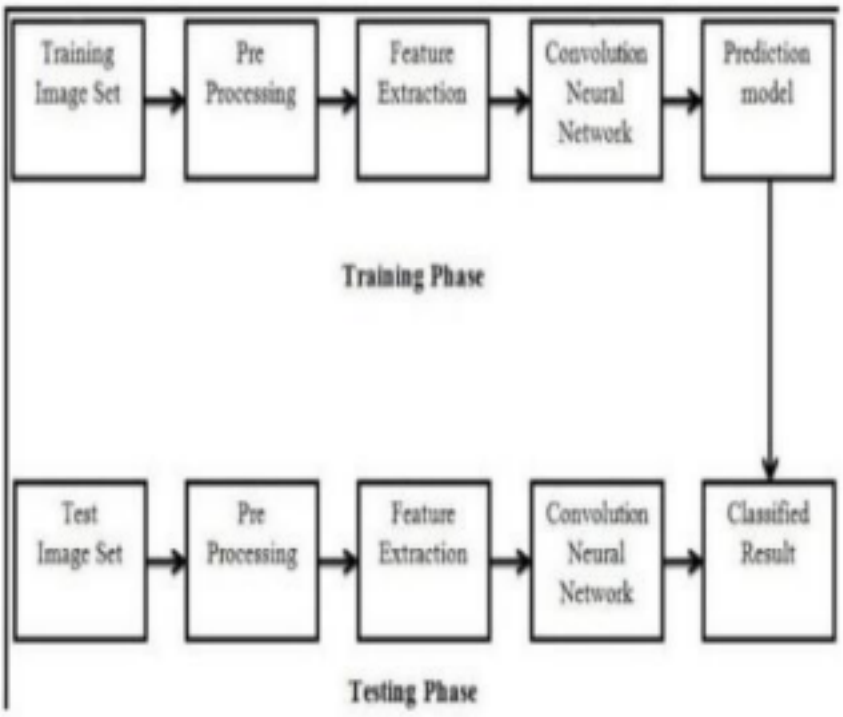
4.2 NON FUNCTIONAL REQUIREMENt

NFR-5	**Availability**	The features for handwritten digit recognition have been Acquainted. These features are based on shape analysis of the digit image and extract slant or slope information. They are effective in obtaining good recognition of accuracy.
NFR-6	**Scalability**	The scalability in the task of handwritten digit recognition, using a classifier, has great importance and it makes use of online handwriting recognition on computer tablets, recognizing zip codes on mail for postal mail sorting, processing bank check amounts, numeric entries in forms filled up manually(for example - tax forms) and so on.

CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 SOLUTION & TECHNICAL ARCHITECTURE

Technical Architecture for Handwritten Digit Recognition System:

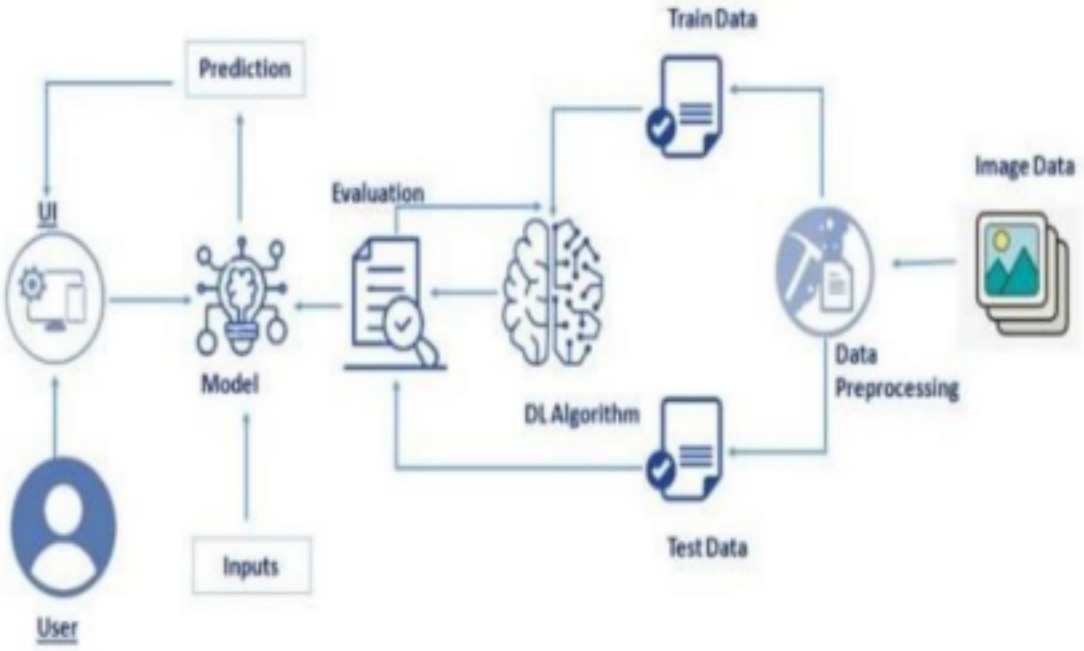


Table-1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	User interacts with the application using a web app	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic	Login to access the application	Java / Python
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	Storage of user files of handwritten image	IBM Block Storage or Other Storage Service or Local Filesystem
10.	Machine Learning Model	Machine learning model is used to identify the handwritten image uploaded by users	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / AI Local Server Configuration AI Server Configuration	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Machine learning frameworks is used to train a predictive model	PyTorch, Open-cv
2.	Security Implementations	The system should automatically be able to authenticate all users with their unique username and password	Password based login, Authorization
3.	Scalable Architecture	The website traffic limit must be scalable enough to support 2 lakhs users at a time	3-tier
4.	Availability	The system functionality and services are available for use with all operations.	distributed servers
5.	Performance	The application can give response to requests within 5 sec. It uses fewer features to train the neural network, which results in faster convergence.	number of requests per sec

User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Home	USN-1	As a user, I can able to know about the application and read the instruction to usage of mobile app.	I can view the instruction about application.	High	Sprint-1
		USN-2	As a user, I am allowed to view Demo video for using the application.	I can gain Knowledge from Demo Video.	High	Sprint-4
		USN-3	As a user, I can access the MNIST dataset from my Drive Files.	I can access the MNIST dataset to get the output.	Low	Sprint-2
	Upload	USN-4	As a user, I have access to upload the dataset from my Drive Files or from other Files.	I can upload the image from System Storage.	Medium	Sprint-1
	Result	USN-5	As a user, I can able to view the result of uploaded image as my predicted output.	I can able to view the result of uploaded image.	High	Sprint-1
Customer (Web View)	Home	USN-6	As a user, I can read the information about the Web application.	I can read and gain knowledge about the web application.	High	Sprint-1
	Pre-Processing	USN-7	As a user, I will train and test the input.	I can able to train and test the input data	High	Sprint-4
	Recognize	USN-8	As a user, I can recognize how the input is evaluated.	I can able to know the Evolution of input.	Low	Sprint-2
	Predict	USN_9	As a user, I am able to predict the image.	I can able to predict the image.	Medium	Sprint-3
	Accuracy	USN_10	As a user, I can see the accuracy of my input image as output result.	I can able to view the resulted output.	High	Sprint-1

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Product Backlog, Sprint Schedule, and Estimation

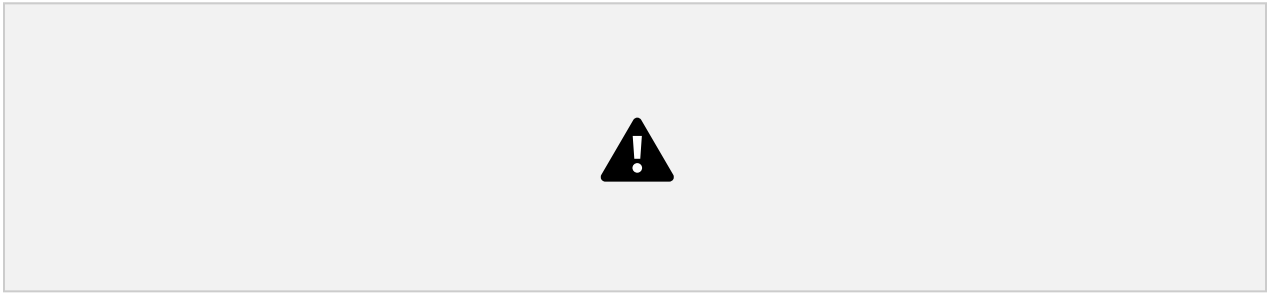
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release	Team Members
Customer (Mobile user)	Home	USN-1	As a user, I can able to know about the application and read the instruction to usage of mobile app.	I can view the instruction about application.	High	Sprint-1	TM-1
		USN-2	As a user, I am allowed to view Demo video for using the application.	I can gain Knowledge from Demo Video.	High	Sprint-4	TM-2,TM-1
		USN-3	As a user, I can access the MNIST dataset from my Drive Files.	I can access the MNIST dataset to get the output.	Low	Sprint-2	TM-3
	Upload	USN-4	As a user, I have access to upload the dataset from my Drive Files or from other Files.	I can upload the image from System Storage.	Medium	Sprint-1	TL
	Result	USN-5	As a user, I can able to view the result of uploaded image as my predicted output.	I can able to view the result of uploaded image.	High	Sprint-1	TM-1,TM-2
Customer (Web View)	Home	USN-6	As a user, I can read the information about the Web application.	I can read and gain knowledge about the web application.	High	Sprint-1	TM-3,TL
	Pre-Processing	USN-7	As a user, I will train and test the input.	I can able to train and test the input data	High	Sprint-4	TM-1,TM-2
	Recognize	USN-8	As a user, I can recognize how the input is evaluated.	I can able to know the Evolution of input.	Low	Sprint-2	TM=3
	Predict	USN_9	As a user, I am able to predict the image.	I can able to predict the image.	Medium	Sprint-3	TL
	Accuracy	USN_10	As a user, I can see the accuracy of my input image as output result.	I can able to view the resulted output.	High	Sprint-1	TM-2,TM-3

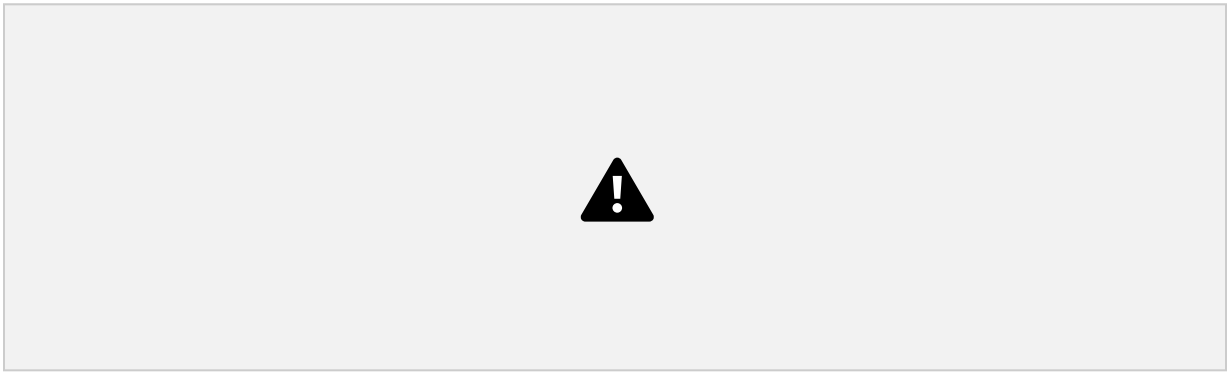
6.2 SPRINT DELIVERY SCHEDULE

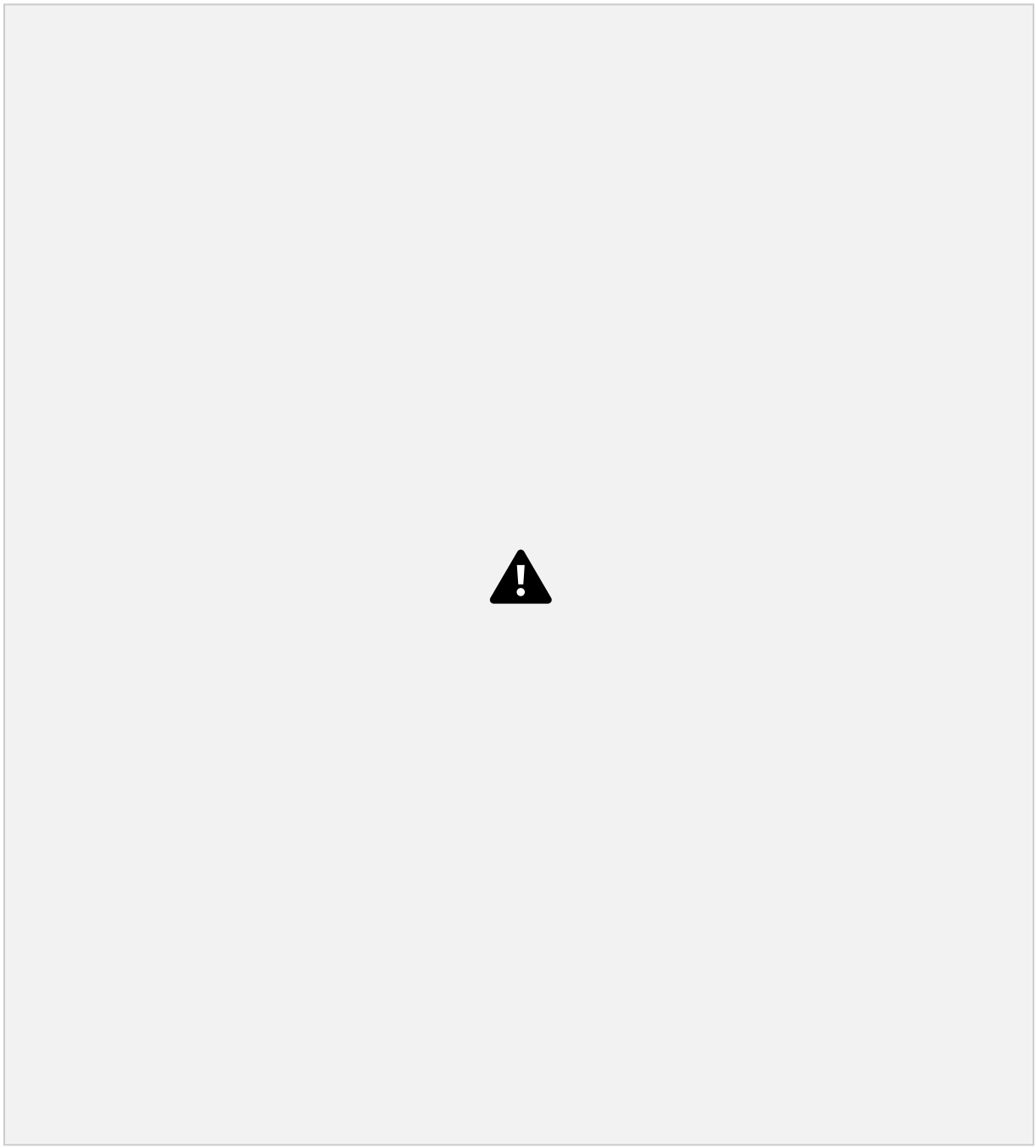
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	30 NOV 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	5 NOV 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	11 NOV 2022
Sprint-4	20	3 Days	11 Nov 2022	19 Nov 2022	20	13 NOV 2022

CHAPTER 7

CODING & SOLUTIONING







CHAPTER 8

TESTING

8.1 TEST CASES

Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status

HP_TC_001	UI	Home Page	Verify UI elements in the Home Page	The Home page must be displayed properly	Working as expected	PASS
HP_TC_002	UI	Home Page	Check if the UI elements are displayed properly in different screen sizes	The Home page must be displayed properly in all sizes	The UI is not displayed properly in screen size 2560 x 1801 and 768 x 630	FAIL
HP_TC_003	Functional	Home Page	Check if user can upload their file	The input image should be uploaded to the application successfully	Working as expected	PASS
HP_TC_004	Functional	Home Page	Check if user cannot upload unsupported files	The application should not allow user to select a non image file	User is able to upload any file	FAIL
HP_TC_005	Functional	Home Page	Check if the page redirects to the result page once the input is given	The page should redirect to the results page	Working as expected	PASS

BE_TC_001	Functional	Backend	Check if all the routes are working properly	All the routes should properly work	Working as expected	PASS

M_TC_001	Functional	Model	Check if the model can handle various image sizes	The model should rescale the image and predict the results	Working as expected	PASS
M_TC_002	Functional	Model	Check if the model predicts the digit	The model should predict the number	Working as expected	PASS
M_TC_003	Functional	Model	Check if the model can handle complex input image	The model should predict the number in the complex image	The model fails to identify the digit since the model is not built to handle such data	FAIL
RP_TC_001	UI	Result Page	Verify UI elements in the Result Page	The Result page must be displayed properly	Working as expected	PASS
RP_TC_002	UI	Result Page	Check if the input image is displayed properly	The input image should be displayed properly	The size of the input image exceeds the display container	FAIL
RP_TC_003	UI	Result Page	Check if the result is displayed properly	The result should be displayed properly	Working as expected	PASS
RP_TC_004	UI	Result Page		The other predictions should be displayed properly	Working as expected	PASS

8.2 USER ACCEPTANCE TESTING

8.2.1 DEFECT ANALYSIS

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Total
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	1	1
Won't Fix	1	0	1	0	2
Total	6	1	4	3	14

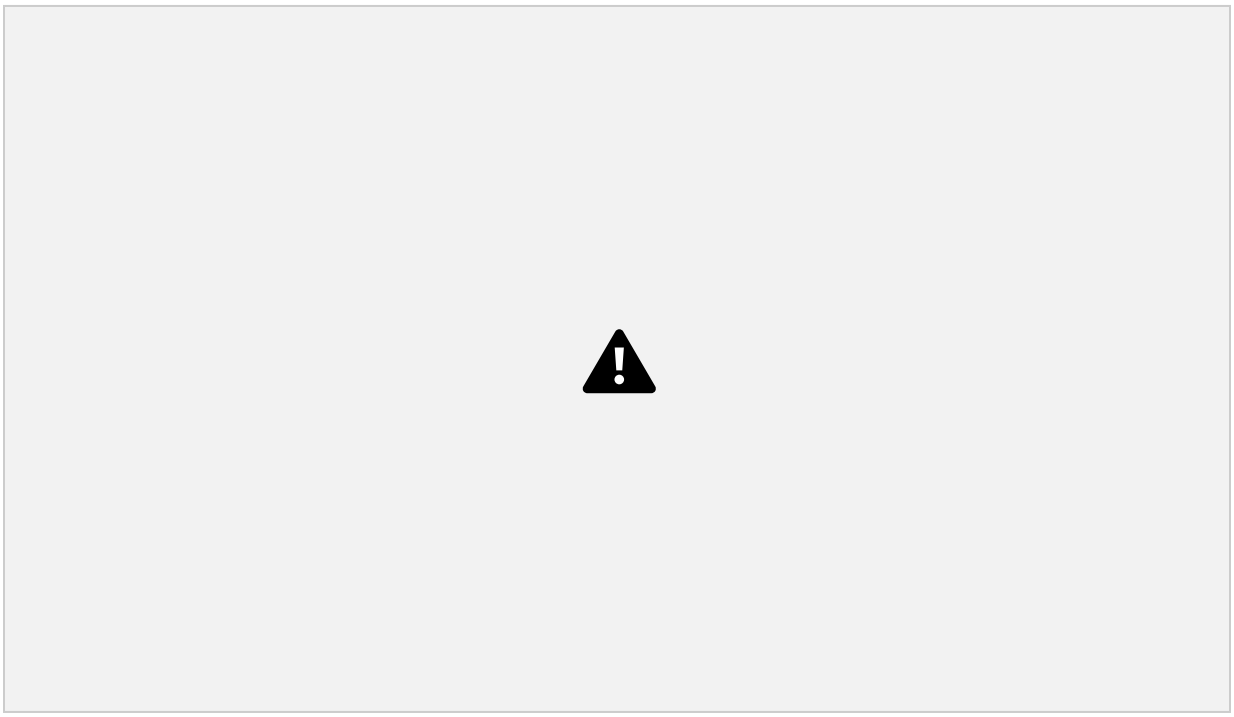
8.2.2 TEST CASE ANALYSIS

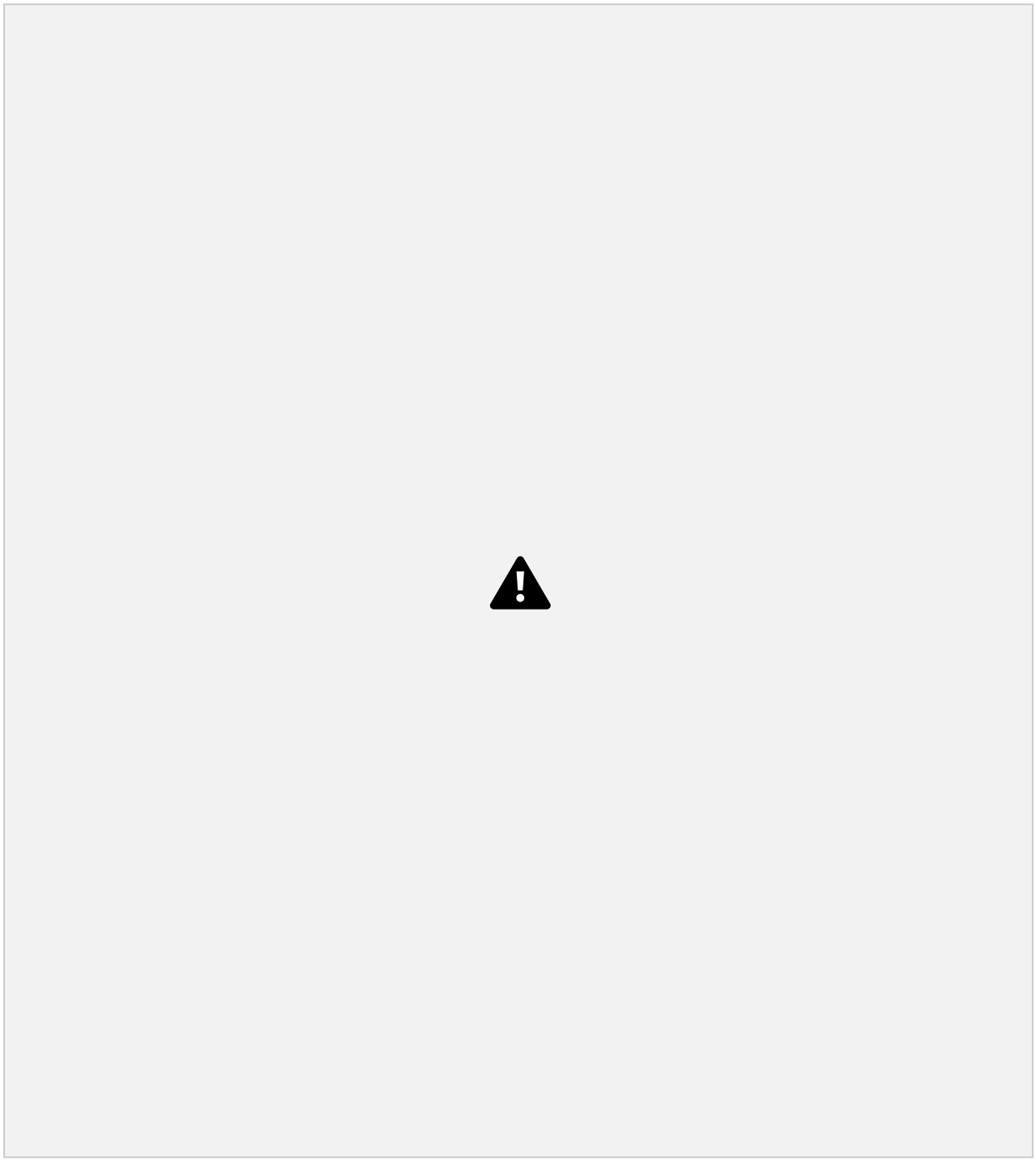
Section	Total Cases	Not Tested	Fail	Pass
Client Application	10	0	3	7
Security	2	0	1	1
Performance	3	0	1	2
Exception Reporting	2	0	0	2

CHAPTER 9

RESULTS

9.1 PERFORMANCE METRICS





CHAPTER 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES

- Reduces manual work
- More accurate than average human

- Capable of handling a lot of data
- Can be used anywhere from any device

DISADVANTAGES

- Cannot handle complex data
- All the data must be in digital format
- Requires a high performance server for faster predictions
- Prone to occasional errors

CHAPTER 11
CONCLUSION

This project demonstrated a web application that uses machine learning to recognise handwritten numbers. Flask, HTML, CSS, JavaScript, and a few other technologies were used to create this project. The model predicts the handwritten digit using a CNN network. During testing, the model achieved a 99.61%

recognition rate. The proposed project is scalable and can easily handle a huge number of users. Since it is a web application, it is compatible with any device that can run a browser. This project is extremely useful in real-world scenarios such as recognizing number plates of vehicles, processing bank cheque amounts, numeric entries in forms filled up by hand (tax forms) and so on. There is so much room for improvement, which can be implemented in subsequent versions.

CHAPTER 12
FUTURE SCOPE

This project is far from complete and there is a lot of room for improvement. Some of the improvements that can be made to this project are as follows:

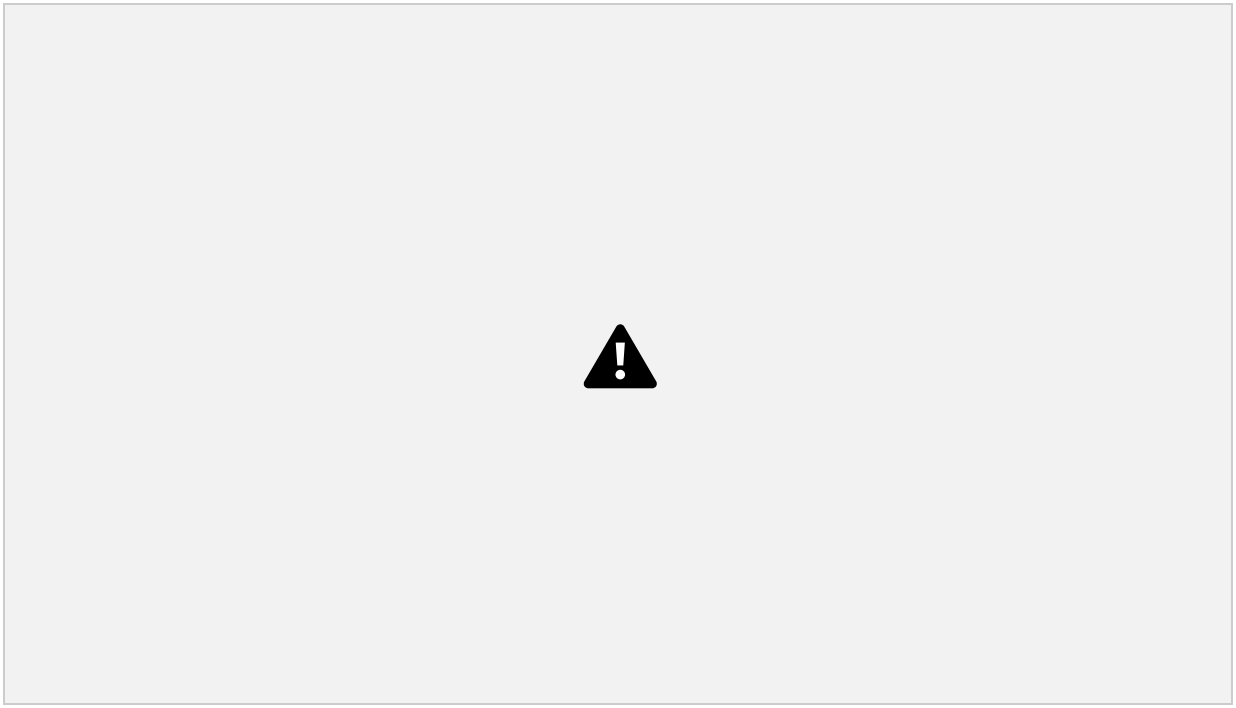
- Add support to detect from digits multiple images and save the results
- Add support to detect multiple digits
- Improve model to detect digits from complex images
- Add support to different languages to help users from all over the world

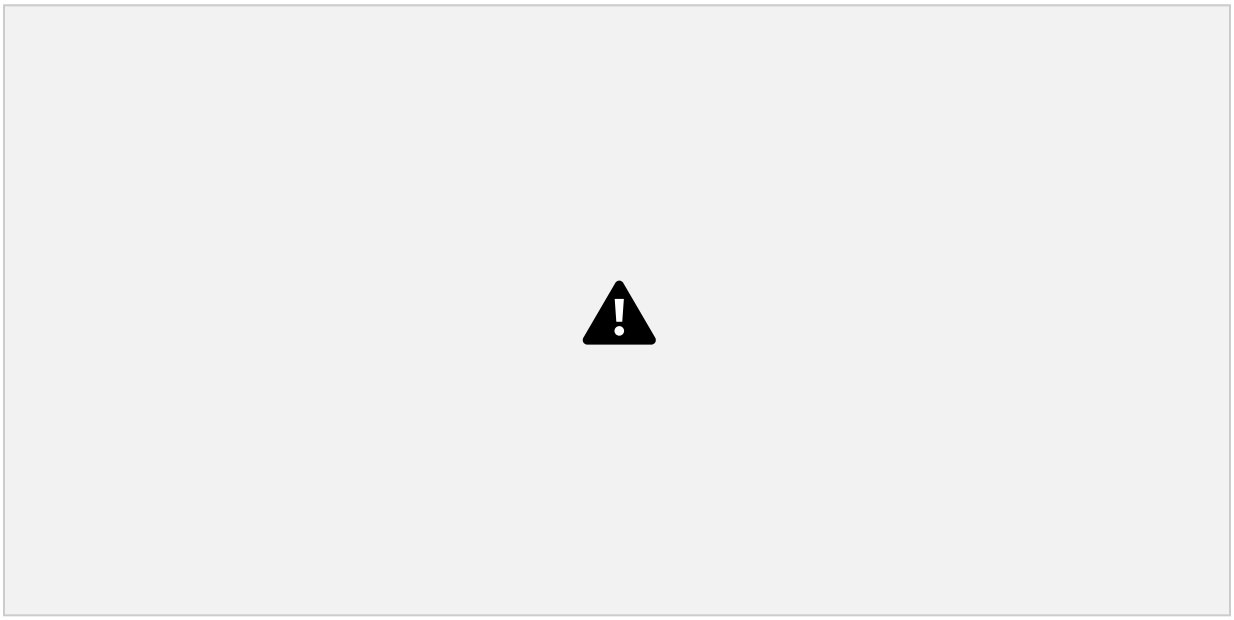
This project has endless potential and can always be enhanced to become better. Implementing this concept in the real world will benefit several industries and reduce the workload on many workers, enhancing overall work efficiency.

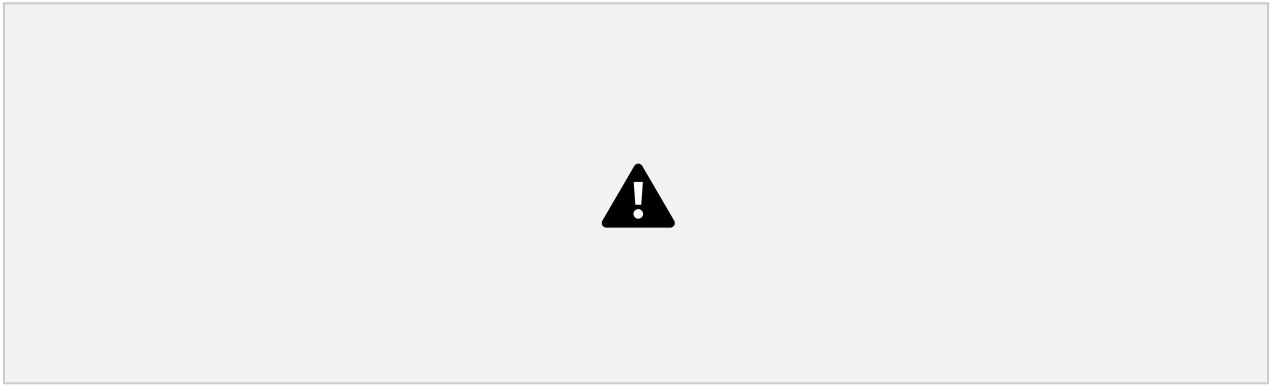
APPENDIX

SOURCE CODE

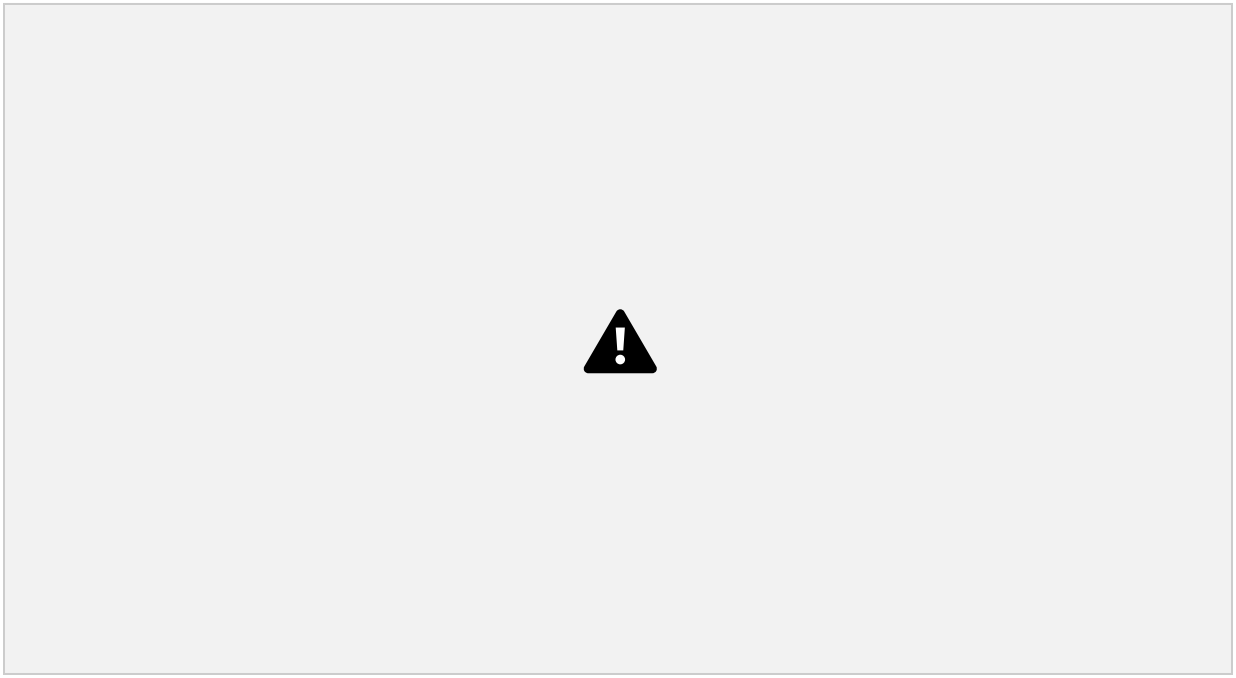
MODEL CREATION



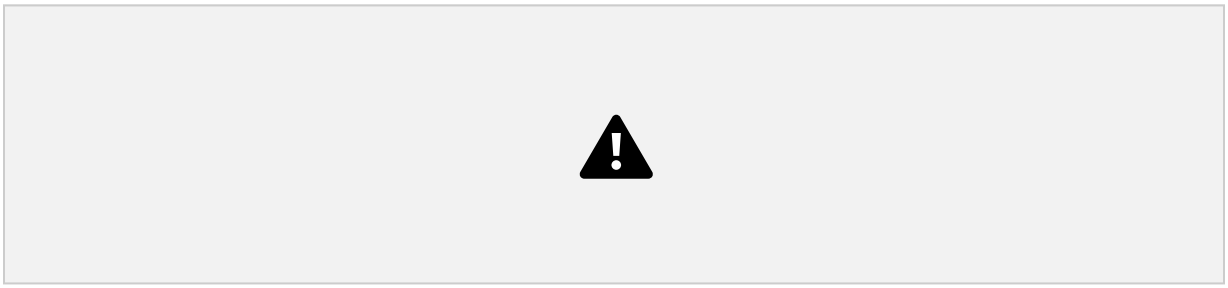


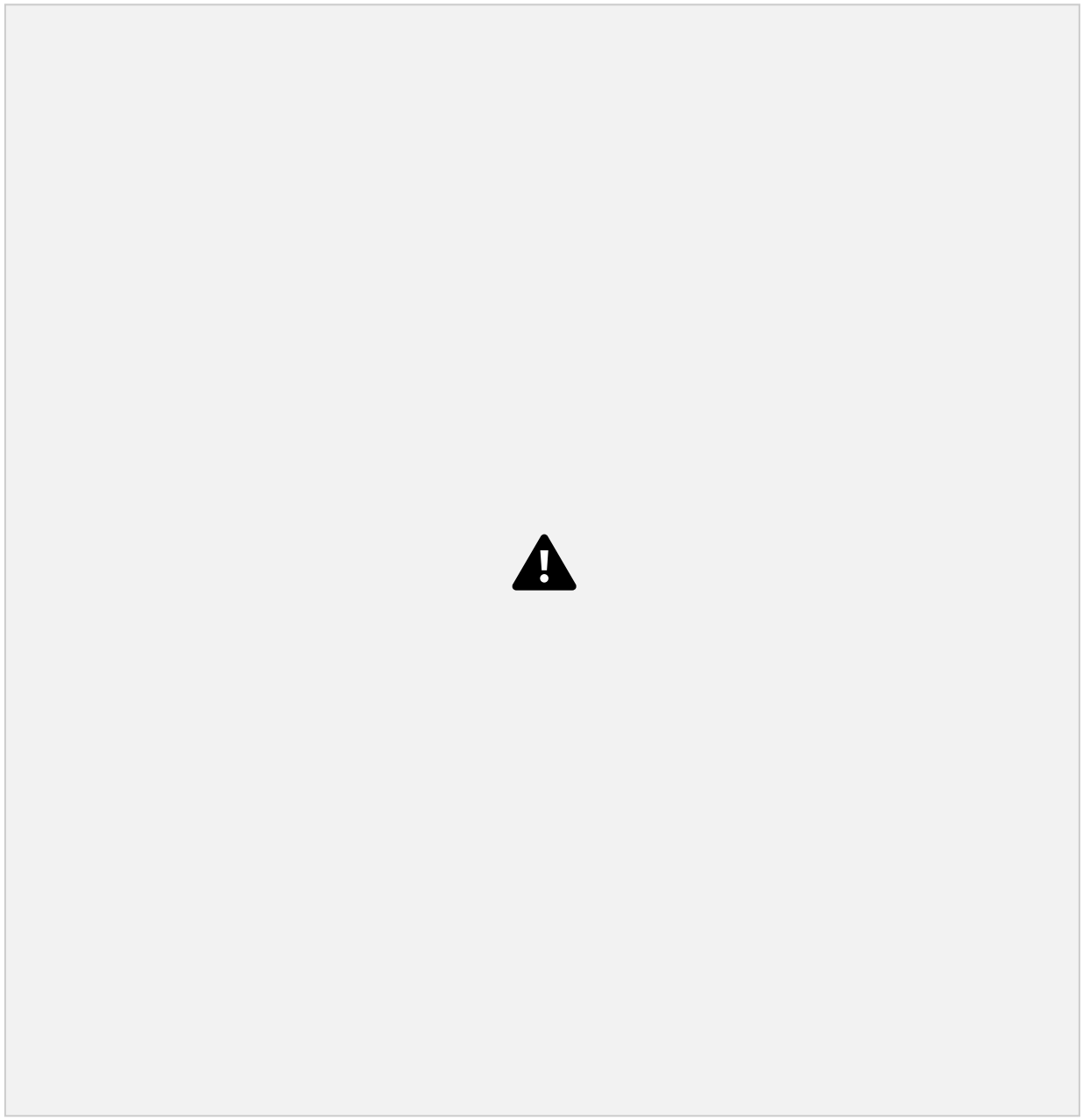


FLASK APP



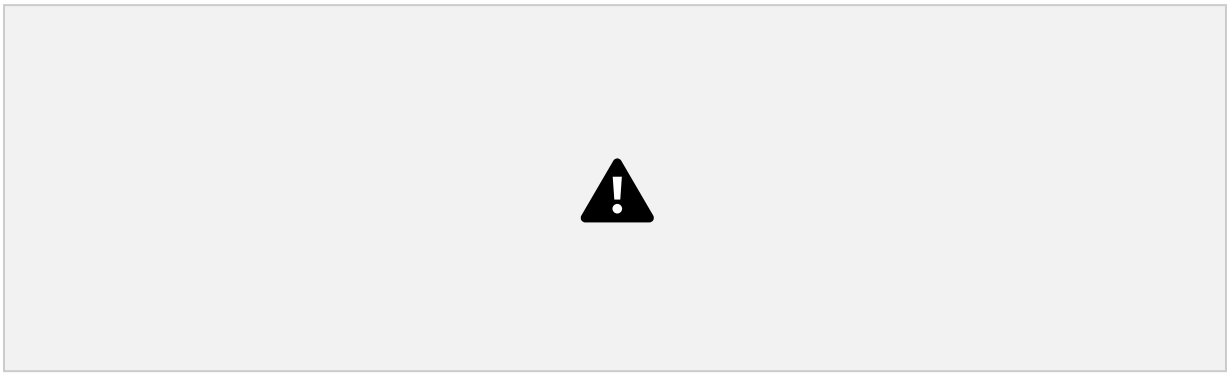
RECOGNIZER

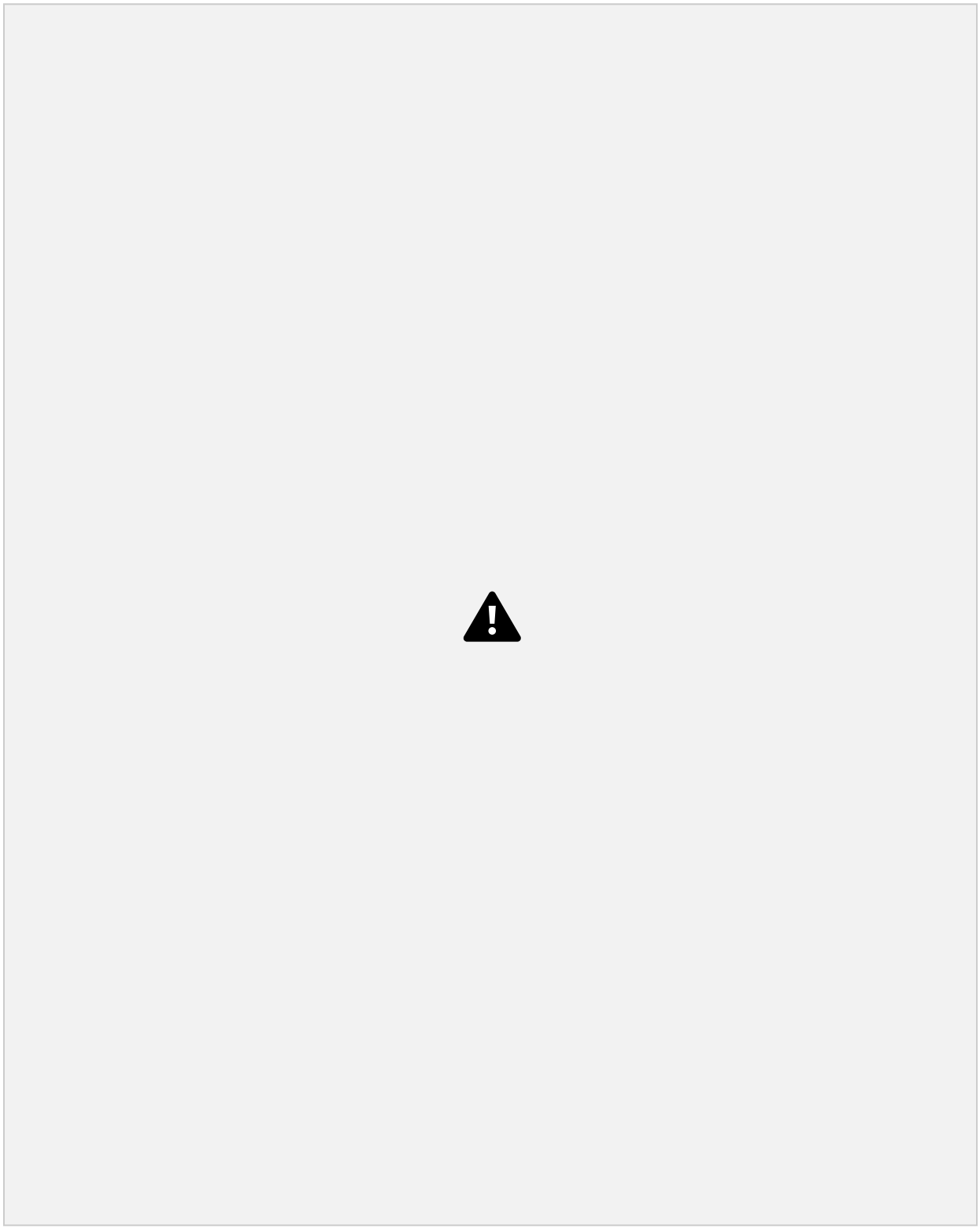




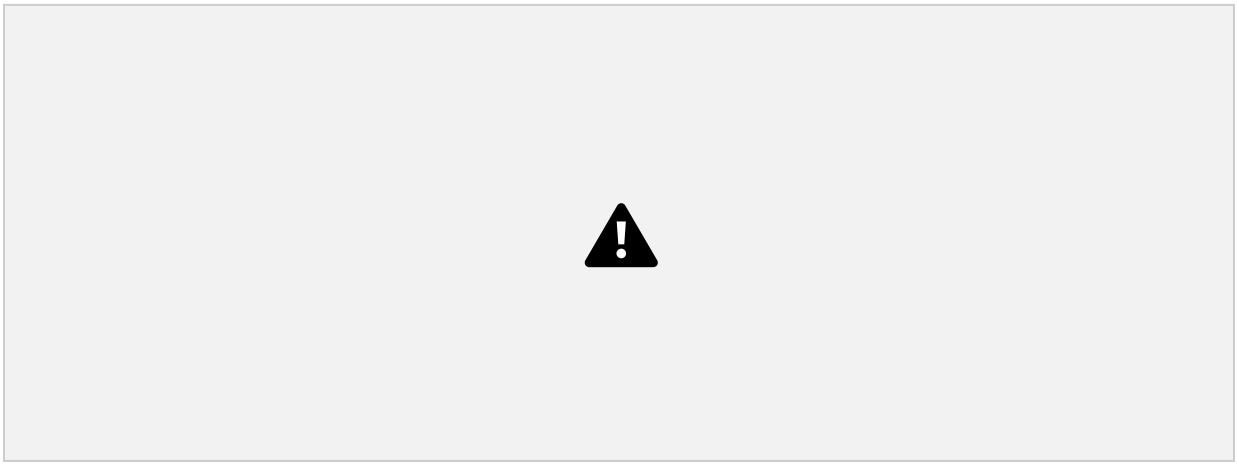
3

HOME PAGE (HTML)

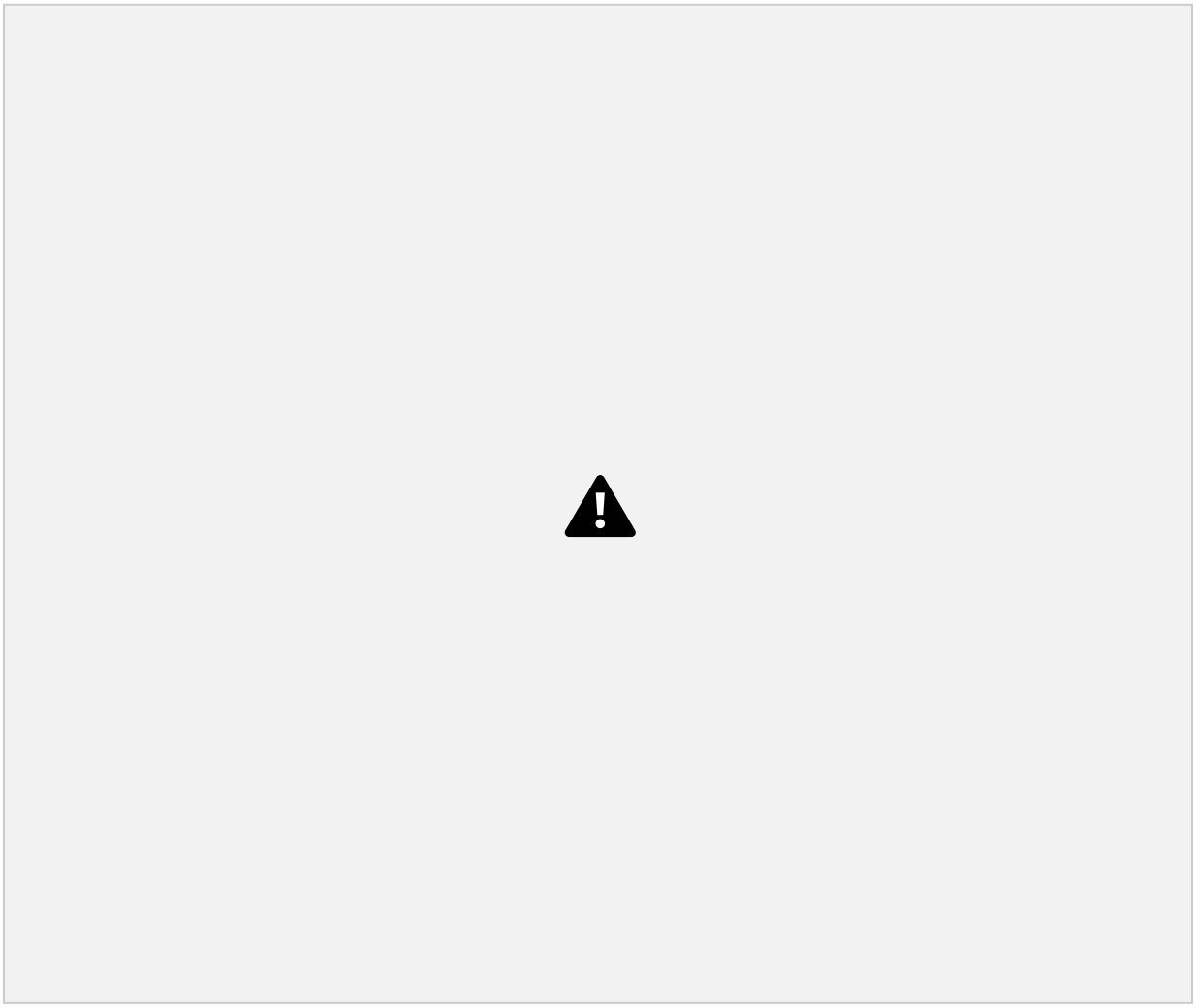


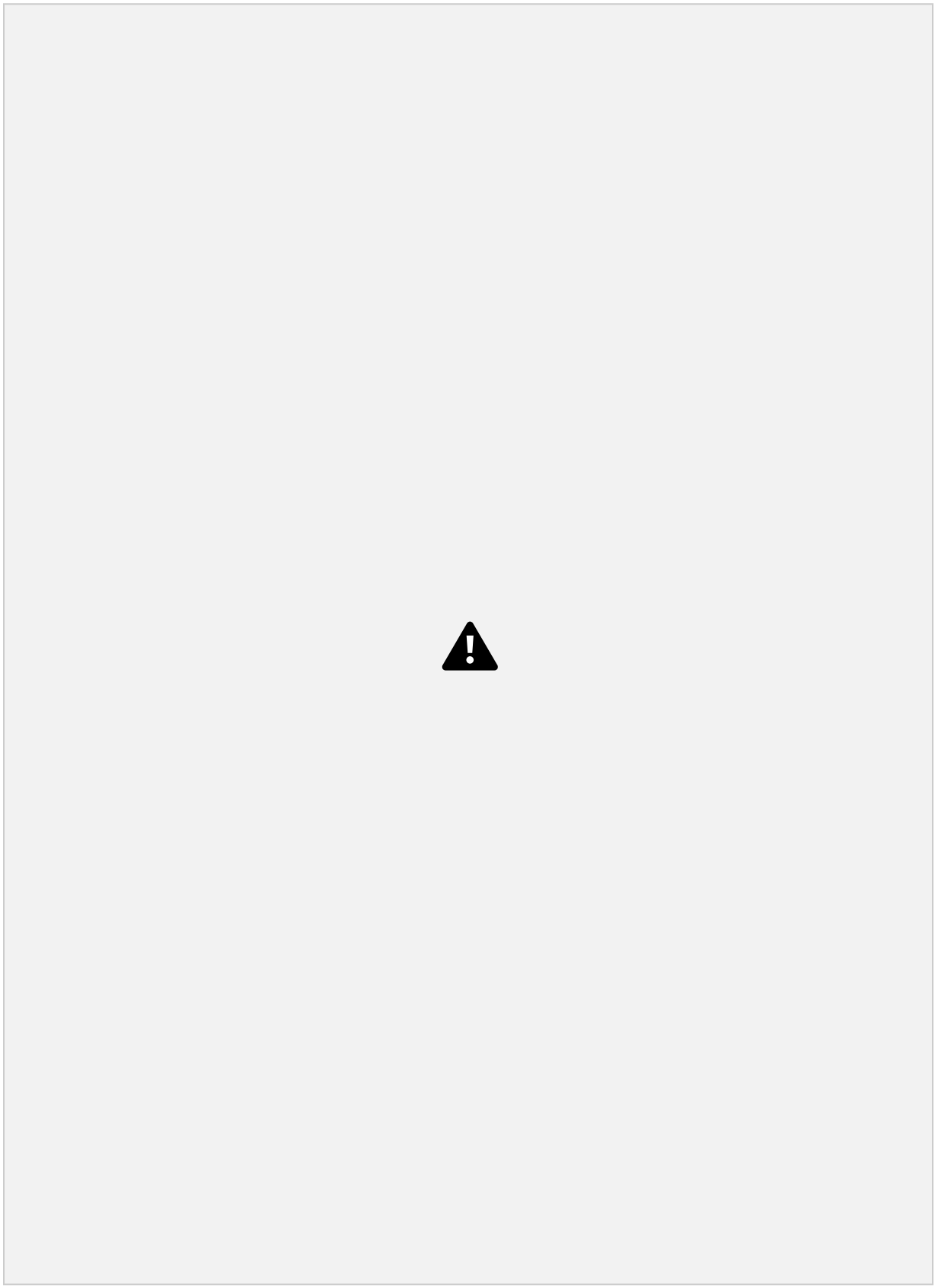


HOME PAGE (JS)



PREDICT PAGE (HTML)





GITHUB

https://github.com/IBM-EPBL/IBM-Project-822-1658324485

PROJECT DEMO

https://drive.google.com/file/d/1pu7JzhNPt_xWMqZmV1vScBOev6dPLlhH/view?usp=drivesdk