**­­­­PROJECT REPORT**

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION

**submitted by**

**PNT2022TMID32578**

Arunkumar S - 810019205301

Subashkumar S - 810019205098

Santhoshkumar R - 810019205089

Natesh T -810019205501

**TABLE OF CONTENTS**

1. **INTRODUCTION**
   1. Project Overview
   2. Purpose
2. **LITERATURE SURVEY**
   1. Existing problem
   2. References
   3. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   1. Empathy Map Canvas
   2. Ideation & Brainstorming
   3. Proposed Solution
   4. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   1. Functional requirement
   2. Non-Functional requirements
5. **PROJECT DESIGN**
   1. Data Flow Diagrams
   2. Solution & Technical Architecture
   3. User Stories
6. **PROJECT PLANNING & SCHEDULING**
   1. Sprint Planning & Estimation
   2. Sprint Delivery Schedule
   3. Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   1. Sprint 1
   2. Sprint 2
   3. Sprint 3
8. **TESTING**
   1. Test Cases
   2. User Acceptance Testing
9. **RESULTS**
   1. Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

Source Code

GitHub & Project Demo Link

# CHAPTER 1

## INTRODUCTION

### 1.1 PROJECT OVERVIEW

Handwritten Digit Recognition is the capacity of a computer to interpret the manually written digits from various sources like messages, bank cheques, papers, pictures, and so forth and in various situations for web-based handwriting recognition on PC tablets, identifying number plates of vehicles, handling bank cheques, digits entered in any forms etc. Machine Learning provides various methods through which human efforts can be reduced in recognizing the manually written digits.

Deep Learning is a machine learning method that trains computers to do what easily falls into place for people: learning through examples. With the utilization of deep learning methods, human attempts can be diminished in perceiving, learning, recognizing and in a lot more regions. Using deep learning, the computer learns to carry out classification works from pictures or contents from any document. Deep Learning models can accomplish state-of-art accuracy, beyond the human level performance. The digit recognition model uses large datasets in order to recognize digits from distinctive sources.

### PURPOSE

### The main objective was to actualize a pattern characterization method to perceive the handwritten digits provided in the MINIST data set of images of handwritten digits (0‐9). The goal of our work is to create a model that will be able to recognize and classify the handwritten digits from images by using concepts of Convolution Neural Network. Though the goal of our research is to create a model for digit recognition and classification, it can also be extended to letters and an individual’s handwriting. With high accuracy rates, the model can solve a lot of real life problems.

### The main applications are vehicle license-plate recognition, postal letter-sorting services, Cheque truncation system (CTS) scanning and historical document preservation in archaeology departments, old documents automation in libraries and banks, etc. All these areas deal with large databases and hence demand high recognition accuracy, lesser computational complexity and consistent performance of the recognition system.

### 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. People can struggle to read others’ handwriting. The handwritten digits are not always of the same size, width, orientation as they differ from writing of person to person, so the general problem would be while classifying the digits.

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.

### REFERENCES

[1]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.[2]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.[3]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 o ered 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. The KNN algorithm, however, is a method for handling handwritten digit recognition. The challenges mentioned in this study can be solved more e ectively using the deep learning neural network approach.[4]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 e ective 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 overfitting the dataset and provides biased predictions.

[1]Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN) (2020) Ahlawat, Savita and Choudhary, Amit and Nayyar, Anand and Singh, Saurabh and Yoon, Byungun.

[2]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.

[3] 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, and Zhu, Yixin. [4]Handwritten Digit Recognition Using Machine And Deep Learning Algorithms (2021) Pashine, Samay and Dixit, Ritik and Kushwah, Rishika

### PROBLEM STATEMENT DEFINITION

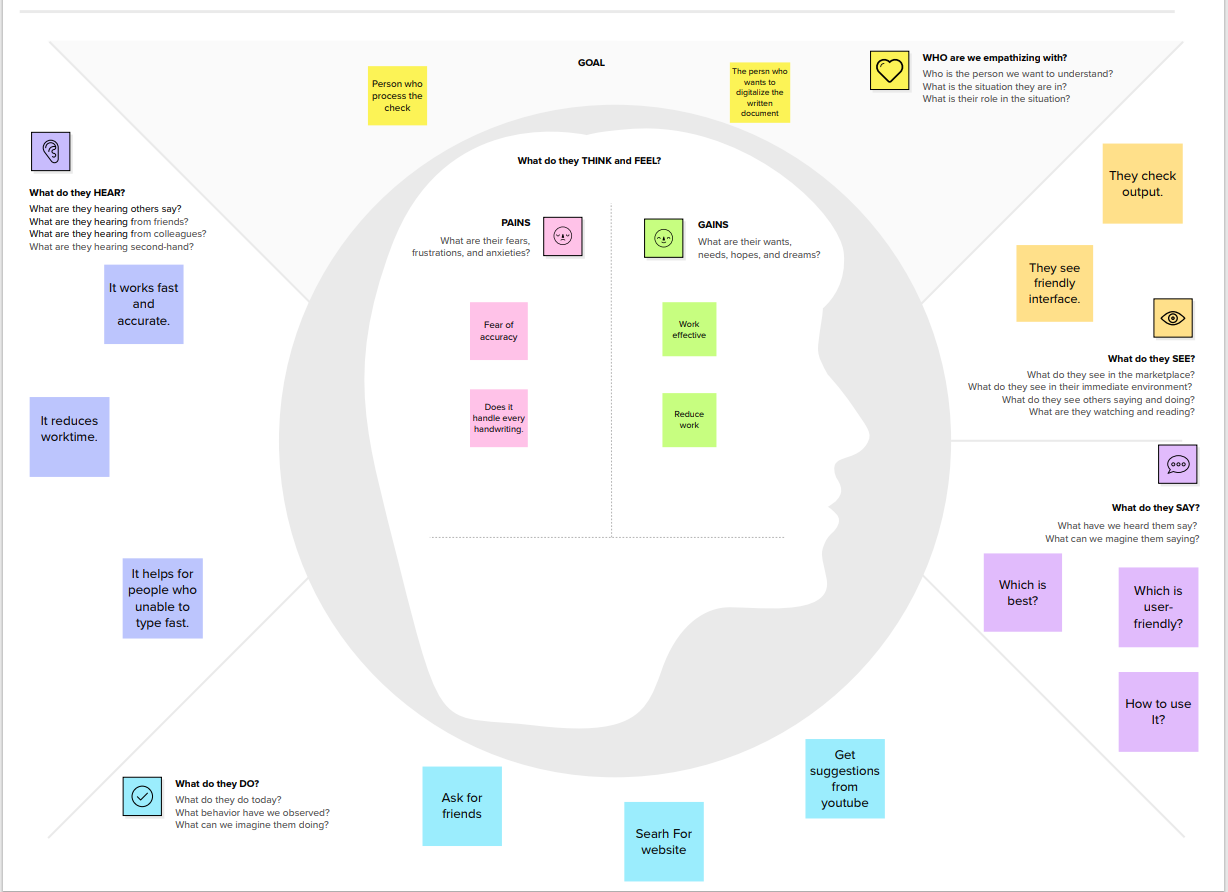
It is easy for the human to perform a task accurately by practicing it repeatedly and memorizing it for the next time. Human brain can process and analyse images easily. Also, recognize the different elements present in the images. In this competition, the goal is to correctly identify digits from a dataset of tens of thousands of handwritten images and experiment with different algorithms to learn first-hand what works well and how techniques compare.

# CHAPTER 3

## IDEATION AND PROPOSED SOLUTION

**­­­­**

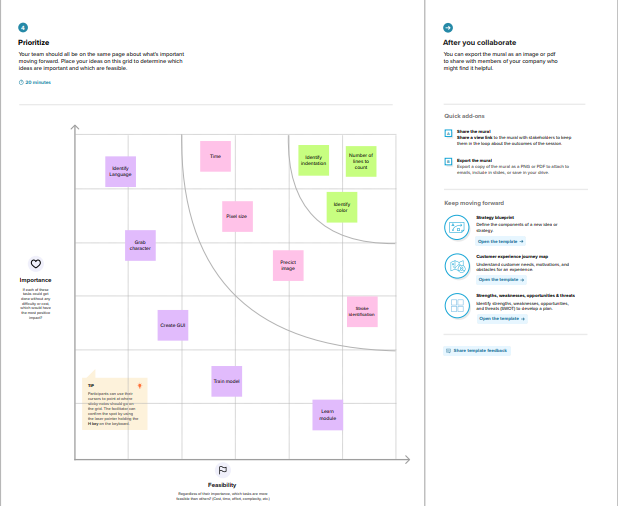
### 3.1 EMPATHY MAP CANVAS

****

­­­­

### IDEATION & BRAINSTORMING

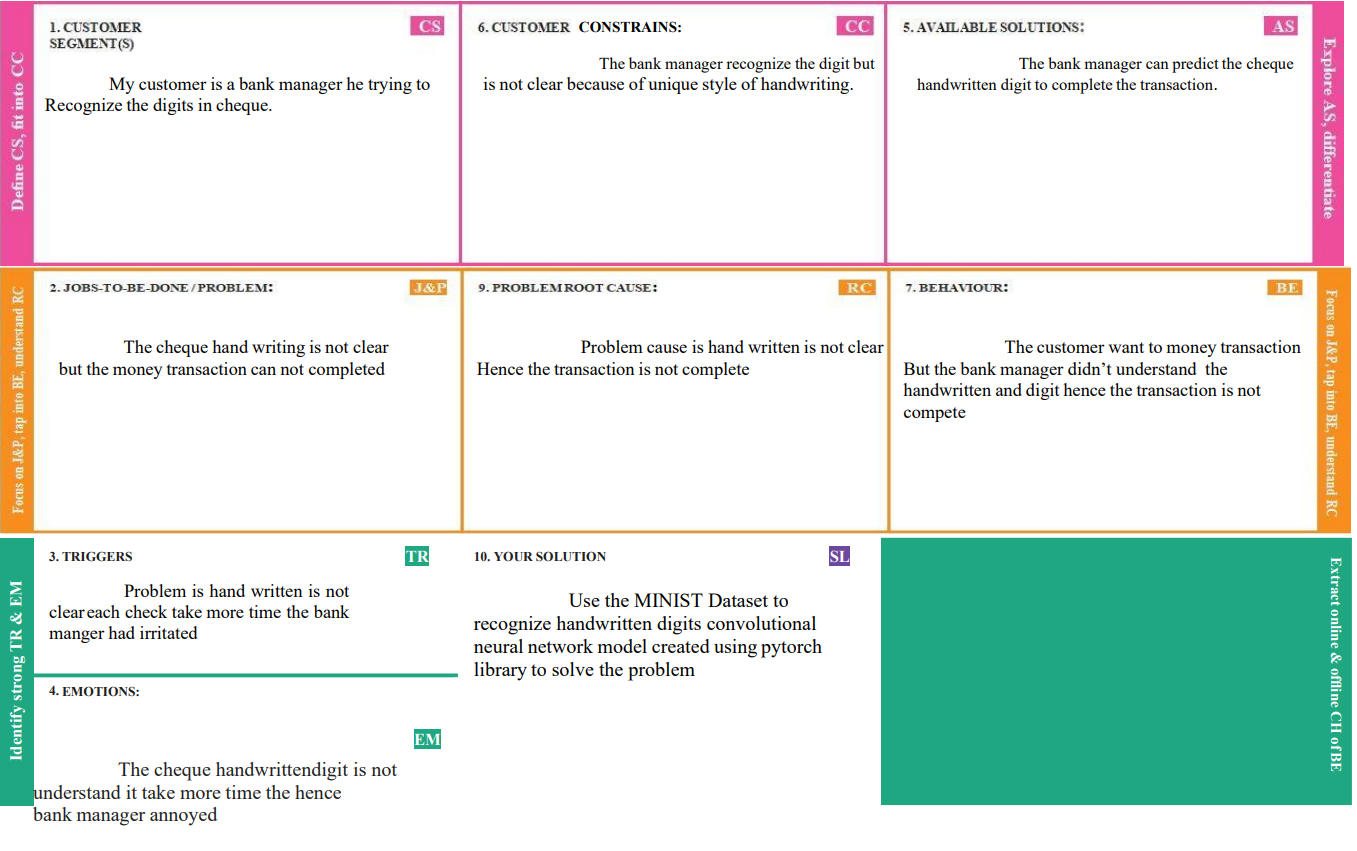
### 



### PROPOSED SOLUTION

|  |  |  |
| --- | --- | --- |
| S.NO | Parameter | Description |
| 1. | Problem Statement (Problem to be solved) | Statement: The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits. Description: It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes |
| 2. | Idea / Solution description | 1. It is the capability of a computer to fete the mortal handwritten integers from different sources like images, papers, and touch defenses. 2. It allows user to translate all those signature and notes into electronic words in a text document format and this data only requires far less physical copies. |
| 3. | Novelty / Uniqueness | Accurately recognize the digits rather than recognizing all the characters like OCR. |
| 4. | Social Impact / Customer  Satisfaction | 1. Artificial Intelligence developed the app called handwritten digit Recognizer. 2. It converts the written word into digital approximations and utilizes complex algorithms to identify characters before churning out a digital approximation |
| 5. | Business Model (Revenue Model) | 1. This system can be integrated with traffic surveillance cameras to recognize the vehicle’s number plates for effective traffic management. |
|  |  | 2. Can be integrated with Postal system to identify and recognize the pin-code details easily. |
| 6. | Scalability of the Solution | 1. Ability to recognize digits in more noisy environments. 2. There is no limit in the number of digits it can be recognized. |

### PROBLEM SOLUTION FIT



# CHAPTER 4

## REQUIREMENT ANALYSIS

### FUNCTIONAL REQUIREMENTS

|  |  |
| --- | --- |
| **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. |

### NON\_FUNCTIONAL REQUIREMENTS

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | One of the very significant problems in pattern recognition applications is the recognition of handwritten characters. Applications for digit recognition include filling out forms, processing bank checks, and sorting mail. |
| NFR-2 | **Security** | 1. The system generates a thorough description of the instantiation parameters, which might reveal information like the writing style, in addition to a categorization of the digit. 2. The generative models are capable of segmentation driven by recognition. 3. The procedure uses a relatively. |
| NFR-3 | **Reliability** | The samples are used by the neural network to automatically deduce rules for reading handwritten digits. Furthermore, the network may learn more about handwriting and hence enhance its accuracy by increasing the quantity of training instances.  Numerous techniques and algorithms, such as  Deep Learning/CNN, SVM, Gaussian Naive Bayes, KNN, Decision Trees, Random Forests, etc., can be used to recognize handwritten numbers. |
| NFR-4 | **Accuracy** | With typed text in high-quality photos, optical character recognition (OCR) technology offers accuracy rates of greater than 99%. However, variances in spacing, abnormalities in handwriting, and the variety of human writing  styles result in less precise character identification. |
| 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

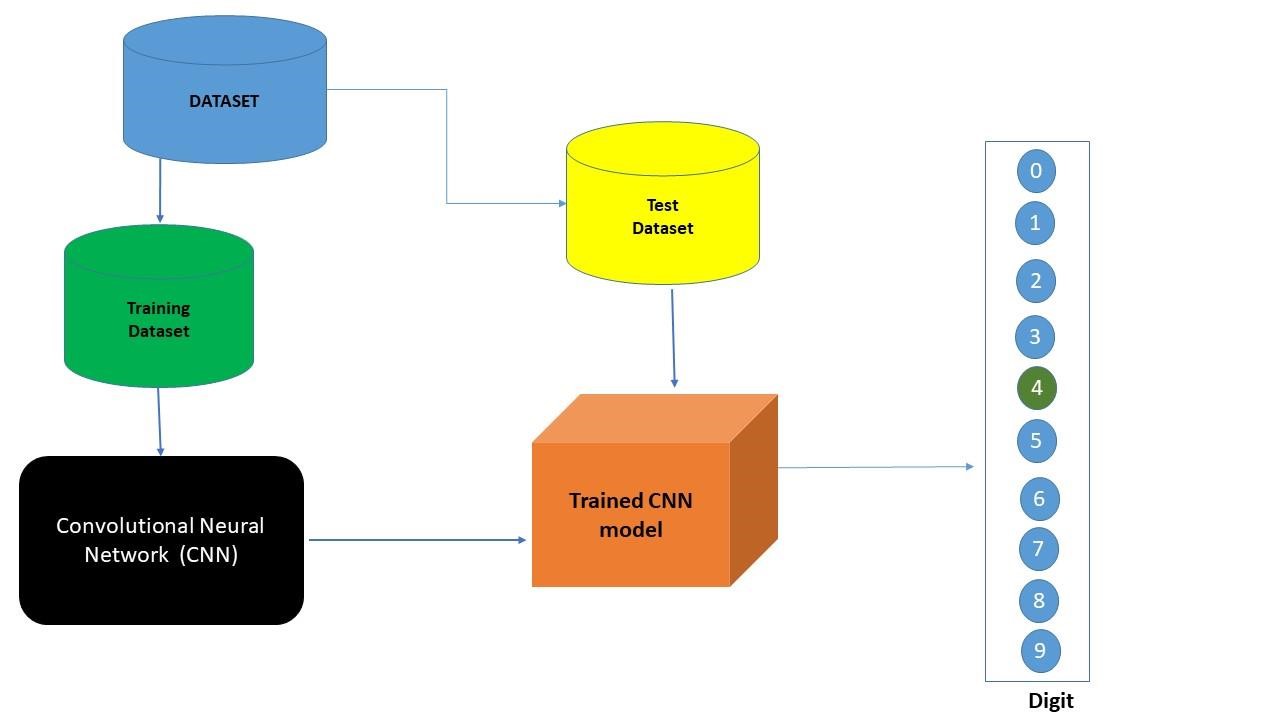
## PROJECT DESIGN

### 5.1 DATA FLOW DIAGRAM

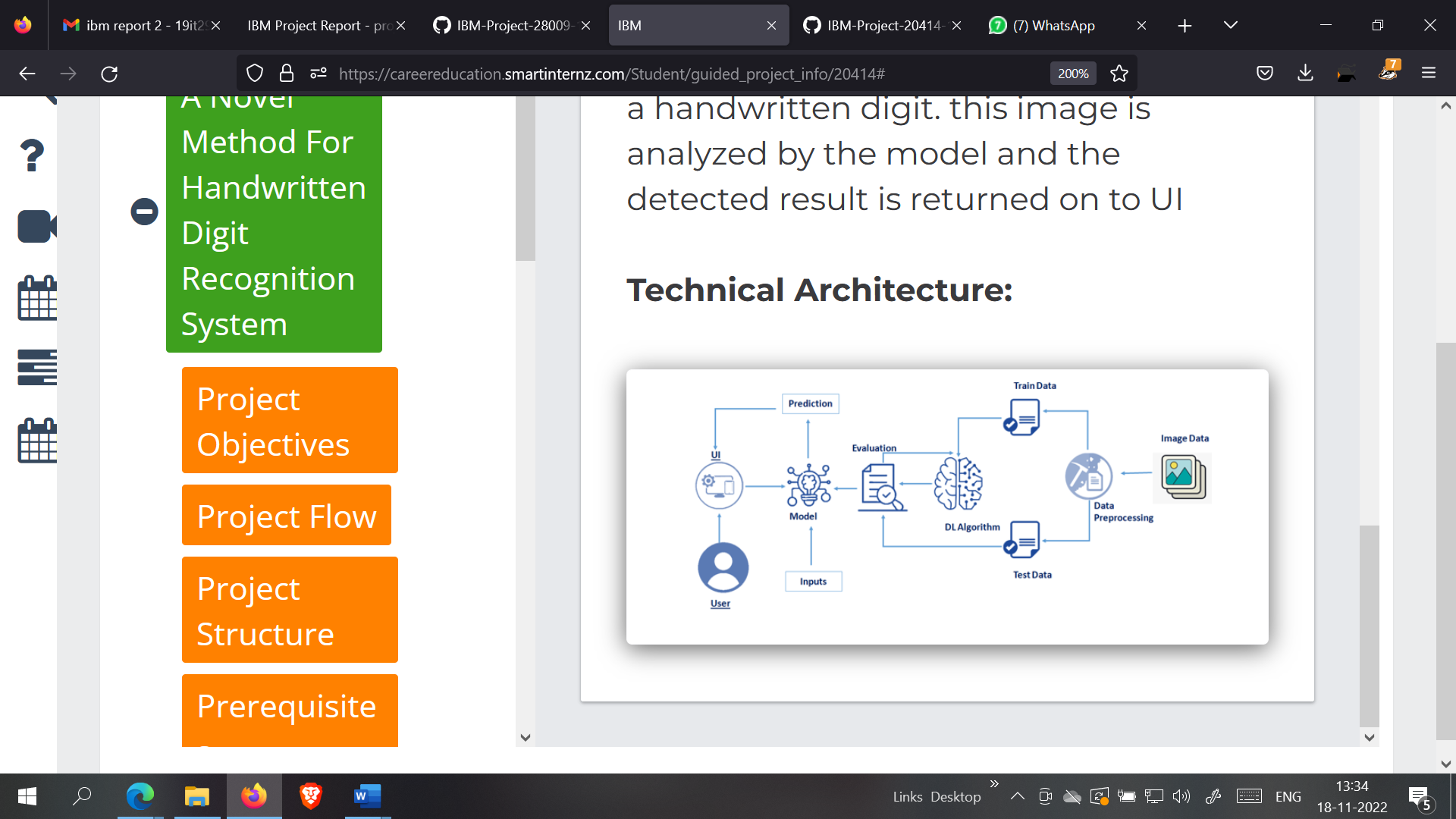
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

**Data Flow Diagrams:**  A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear

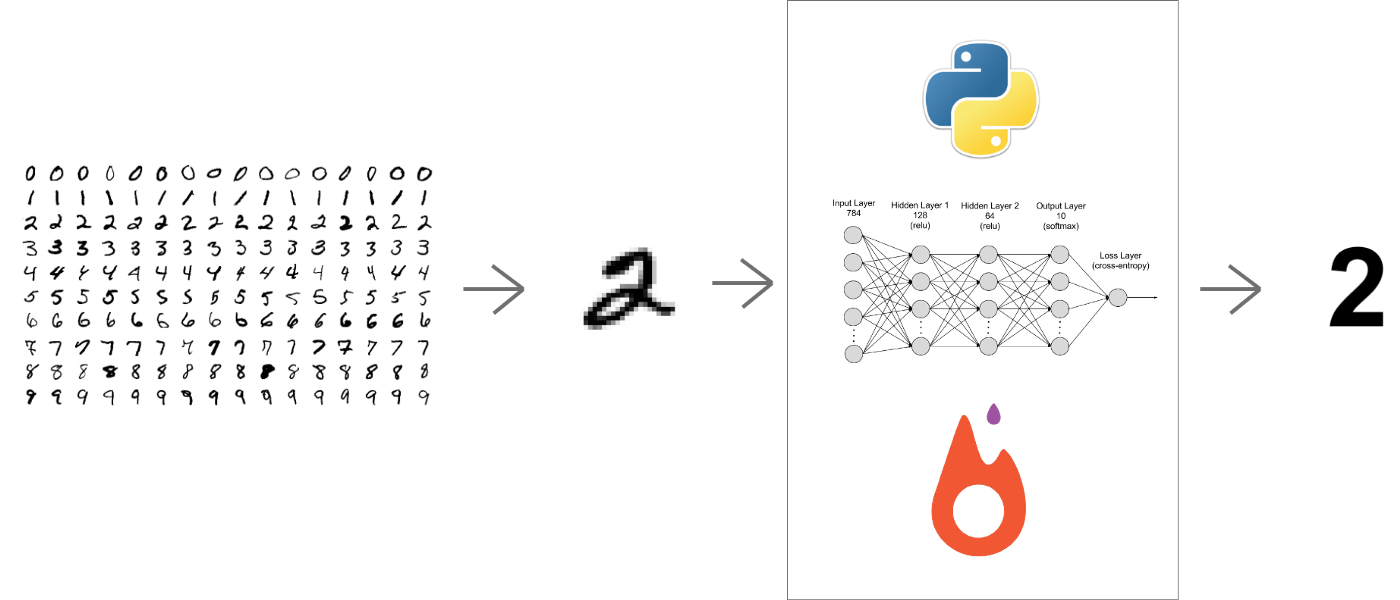
DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information,and where data is stored.



### 5.2 SOLUTION & TECHNICAL ARCHITECTURE



**MNIST DATASET PROCESSING WITH PYTHON**

****

### 

### 

**5.3 COMPONENTS & TECHNOLOGIES:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  |  |  |  |
| 1. | User Interface | How user interacts with application e.g., Mobile Application | HTML, CSS, JavaScript |
|  |  |  |  |
| 2. | Application Logic-1 | Logic for a process in the application | Java / Python |
|  |  |  |  |
| 3. | Application Logic-2 | Logic for a process in the application | IBM Watson STT service |
|  |  |  |  |
| 4. | Application Logic-3 | Logic for a process in the application | IBM Watson Assistant |
|  |  |  |  |
| 5. | Database | Data Type, Configurations etc. | MySQL, NoSQL, etc. |
|  |  |  |  |
| 6. | Cloud Database | Database Service on AI in cloud | IBM DB2 |
|  |  |  |  |
| 7. | File Storage | File storage requirements | IBM Block Storage or Other Storage Service or local file system |
|  |  |  |  |
| 8. | External API-1 | Purpose of External API used in the application | IBM Weather API, etc. |
|  |  |  |  |
| 9. | Internet of Things Model | Purpose of AI Model is for integrating the sensors with a user interface | IBM AI Platform |
|  |  |  |  |
| 10. | Machine Learning Model | Purpose of Machine Learning Model | Digit Recognition Model |
|  |  |  |  |

### 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’m able to use the application in mobile. | I can view the application in mobile. | Low | Sprint-1 |
|  |  | USN-2 | As a user, I can view the guide and awareness to use this application. | I can view the awareness to use this application and its limitations. | Low | Sprint-1 |
|  |  | USN-3 | As a user, I’m allowed to view the guided  video to use the interface of this application. | I can gain knowledge to use this application by a practical method. | Low | Sprint-1 |
|  |  | USN-4 | As a user, I can read the instructions to use this application. | I can read instructions also to use it in a userfriendly method. | Low | Sprint-2 |
|  | Recognize | USN-5 | As a user, In this prediction page I get to choose the image. | I can choose the image from our local system and predict the output. | High | Sprint-2 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Predict | USN-7 | As a user, I’m Allowed to upload and choose the image to be uploaded | I can upload and choose the image from the system storage and also in any virtual storage. | Medium | Sprint-3 |
|  |  | USN-8 | As a user, I will train and test the input to get the maximum accuracy of output. | I can able to train and  test the application until it gets maximum accuracy of the result. | High | Sprint-4 |
|  |  | USN-9 | As a user, I can access the MNIST data set | I can access the MNIST data set to produce the accurate result. | Medium | Sprint-3 |
| **User Type** | **Functional**  **Requirement**  **(Epic)** | **User Story**  **Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Release** |
| Customer  (PC user) | Home | USN-10 | As a user, I can view the guide and awareness to use this application. | I can view the awareness to use this application and its limitations. | Low | Sprint-1 |
|  |  | USN-11 | As a user, I’m allowed to view the guided  video to use the interface of this application. | I can gain knowledge to use this application by a practical method. | Low | Sprint-1 |
|  |  | USN-12 | As a user, I can read the instructions to use this application. | I can read instructions also to use it in a userfriendly method. | Low | Sprint-1 |
|  | Recognize | USN-13 | As a user, I can use the web application virtually anywhere. | I can use the application portably anywhere. | High | Sprint-2 |
|  |  | USN-14 | As it is an open source, can use it cost freely. | I can use it without any payment to be paid for it to access. | Medium | Sprint-2 |
|  |  | USN-15 | As it is a web application, it is installation free | I can use it without the installation of the application or any software. | Medium | Sprint-3 |
|  | Predict | USN-16 | As a user, I’m Allowed to upload and choose the image to be uploaded | I can upload and choose the image from the system storage and also in any virtual storage. | Medium | Sprint-4 |

# CHAPTER 6

## PROJECT PLANNING AND SCHEDULING

### 6.1 SPRINT PLANNING AND ESTIMATION

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SPRINT | USER STORY / TASK | STORY POINTS | PRIORITY | TEAM MEMBERS |
| Sprint - I | Get the dataset | 3 | High | Subashkumar S |
| Explore the data | 2 | Medium | Santhoshkumar R |
| Data Pre-Processing | 3 | High | Arunkumar S |
| Prepare training and testing data | 3 | High | Natesh T |
| Sprint - II | Create the model | 3 | High | Arunkumar S |
| Train the model | 3 | High | Subashkumar S |
| Test the model | 3 | High | Santhoshkumar R |
| Sprint - III | Improve the model | 2 | Medium | Natesh T |
| Save the model | 3 | High | Natesh T Arunkumar S |
|  | Build the Home Page | 3 | High | Subashkumar S |
| Setup a database to store input images | 2 | Medium | Santhoshkumar R |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sprint - IV | Build the results page | 3 | High | Santhoshkumar R |
| Integrate the model with the application | 3 | High | Subashkumar S |
| Test the application | 3 | High | Natesh T, Arunkumar S |

### 6.2 SPRINT DELIVERY SCHEDULE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SPRINT | TOTAL  STORY  POINTS | DURATION | SPRINT START DATE | SPRINT END  DATE  (PLANNED) | STORY  POINTS  COMPLETED  (AS ON  PLANNED DATE) | SPRINT  RELEASE DATE  (ACTUAL) |
| Sprint - I | 11 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 11 | 29 Oct 2022 |
| Sprint - II | 9 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 9 | 05 Nov 2022 |
| Sprint - III | 10 | 6 Days | 07 Oct 2022 | 12 Nov 2022 | 10 | 12 Nov 2022 |
| Sprint - IV | 9 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 9 | 19 Nov 2022 |

**6.3 REPORT FROM JIRA**

**Velocity:**

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit (story points per day)

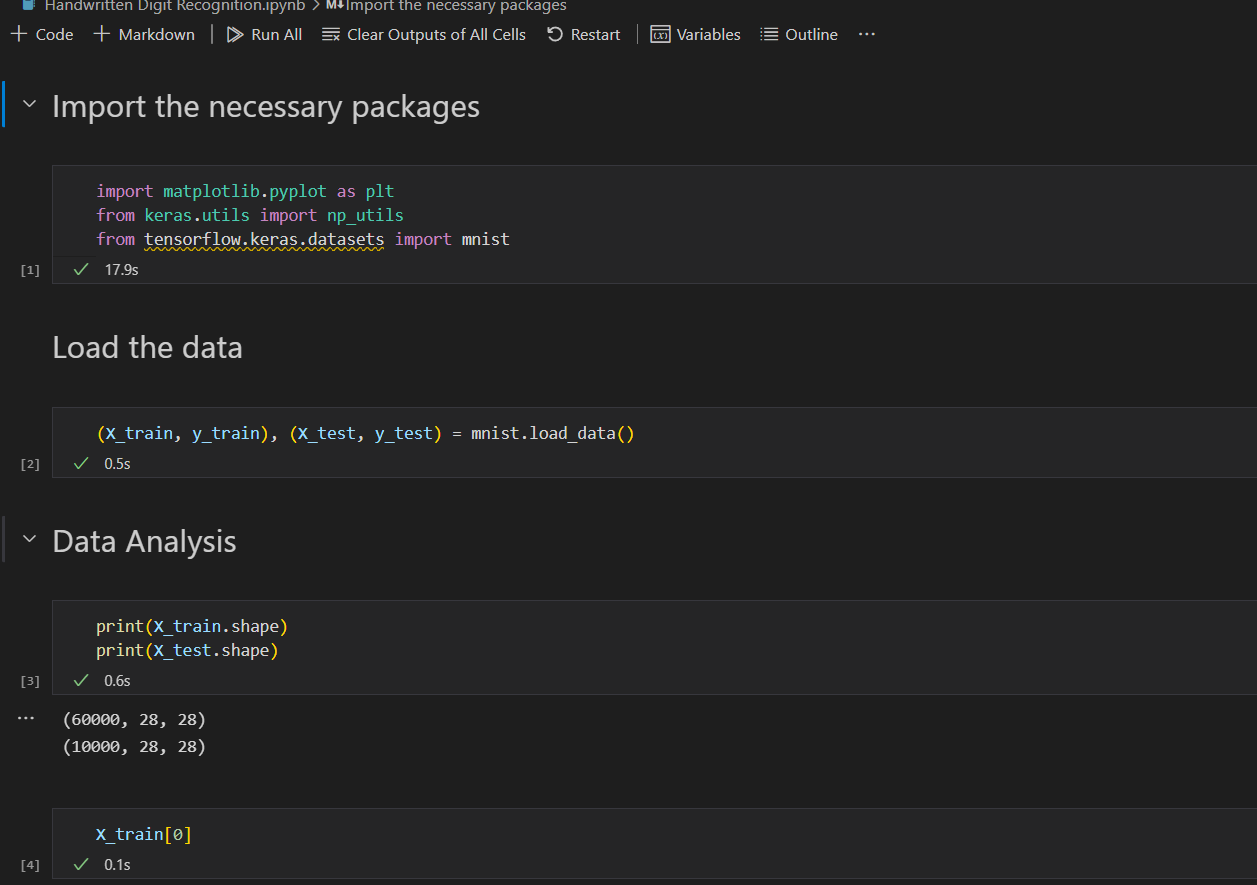
Average Velocity = 20 / 6 = 3.33

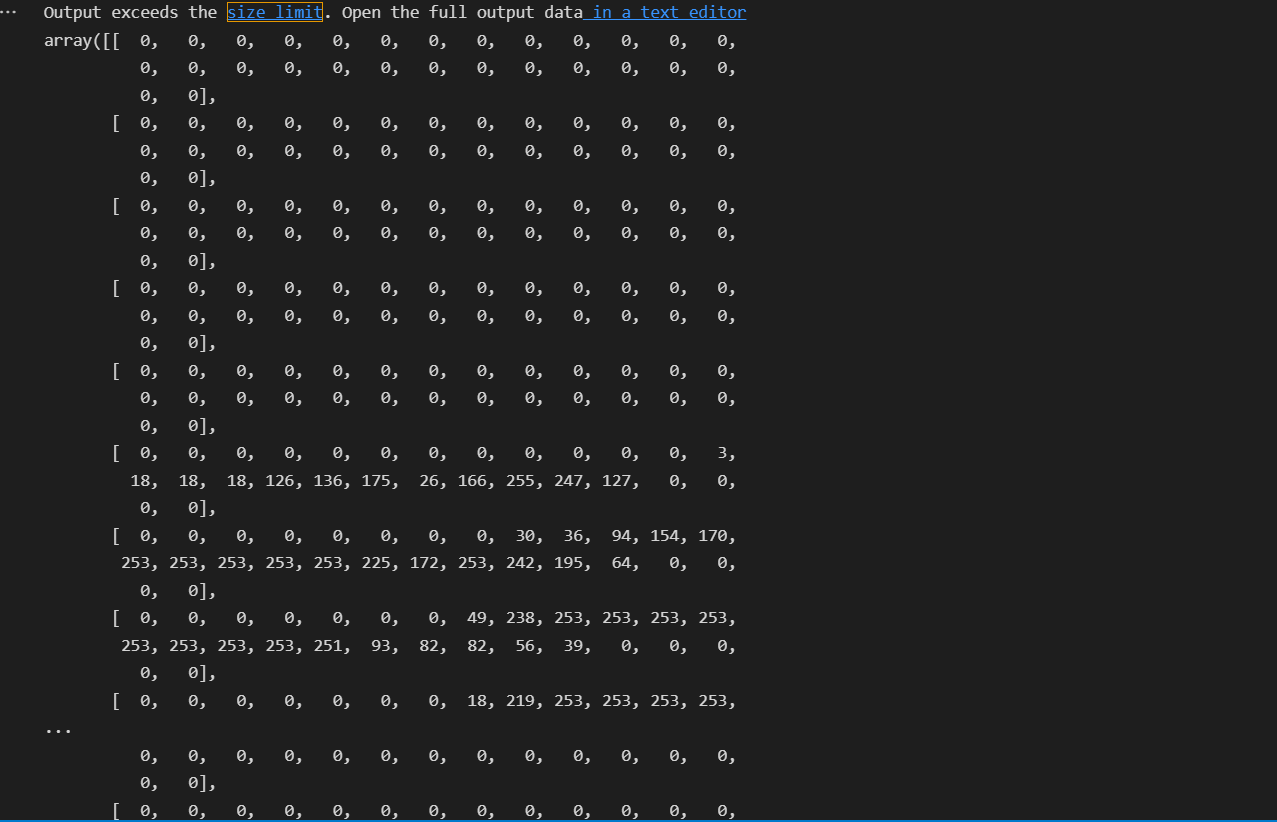
**Burndown Chart:**

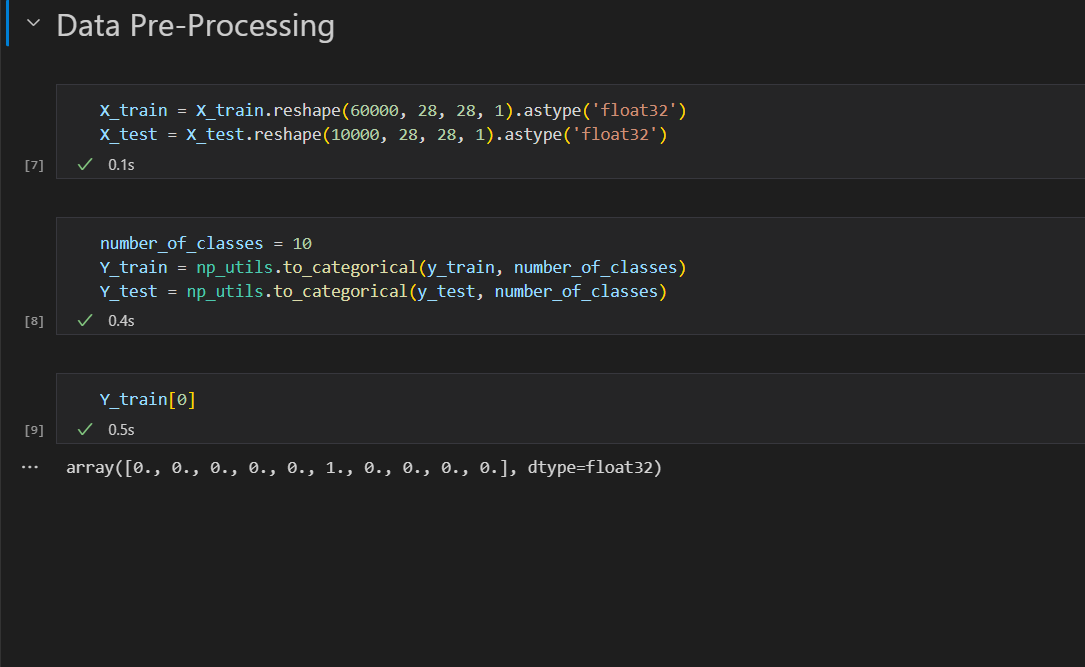
A burn down chart is a graphical representation of work left to do versus time. It is often used in agile[software development](https://www.visual-paradigm.com/scrum/what-is-agile-software-development/) methodologies such as [Scrum](https://www.visual-paradigm.com/scrum/scrum-in-3-minutes/). However, burn down charts can be applied to any project containing measurable progress over time.

**CHAPTER 7**

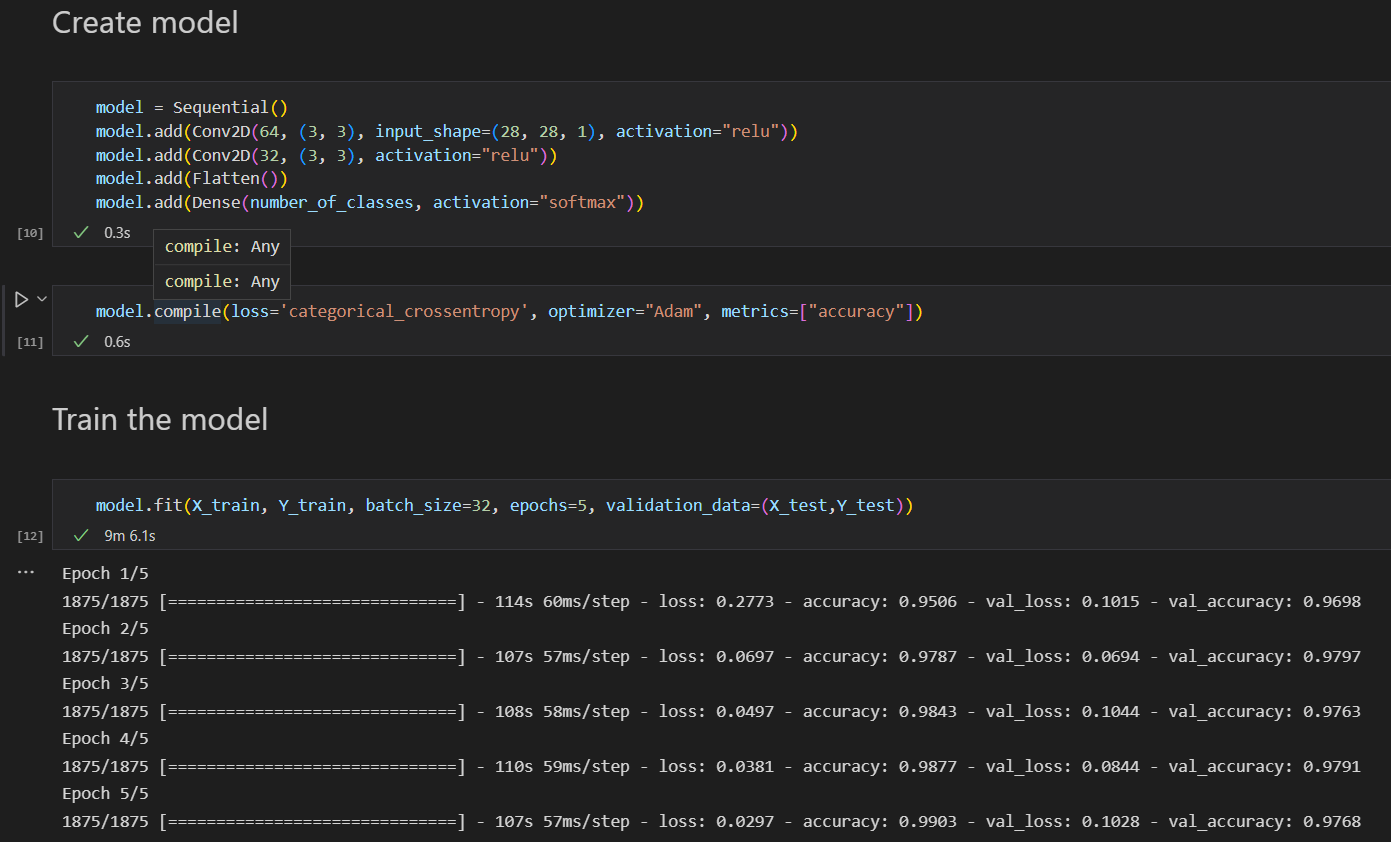
**CODING & SOLUTION**

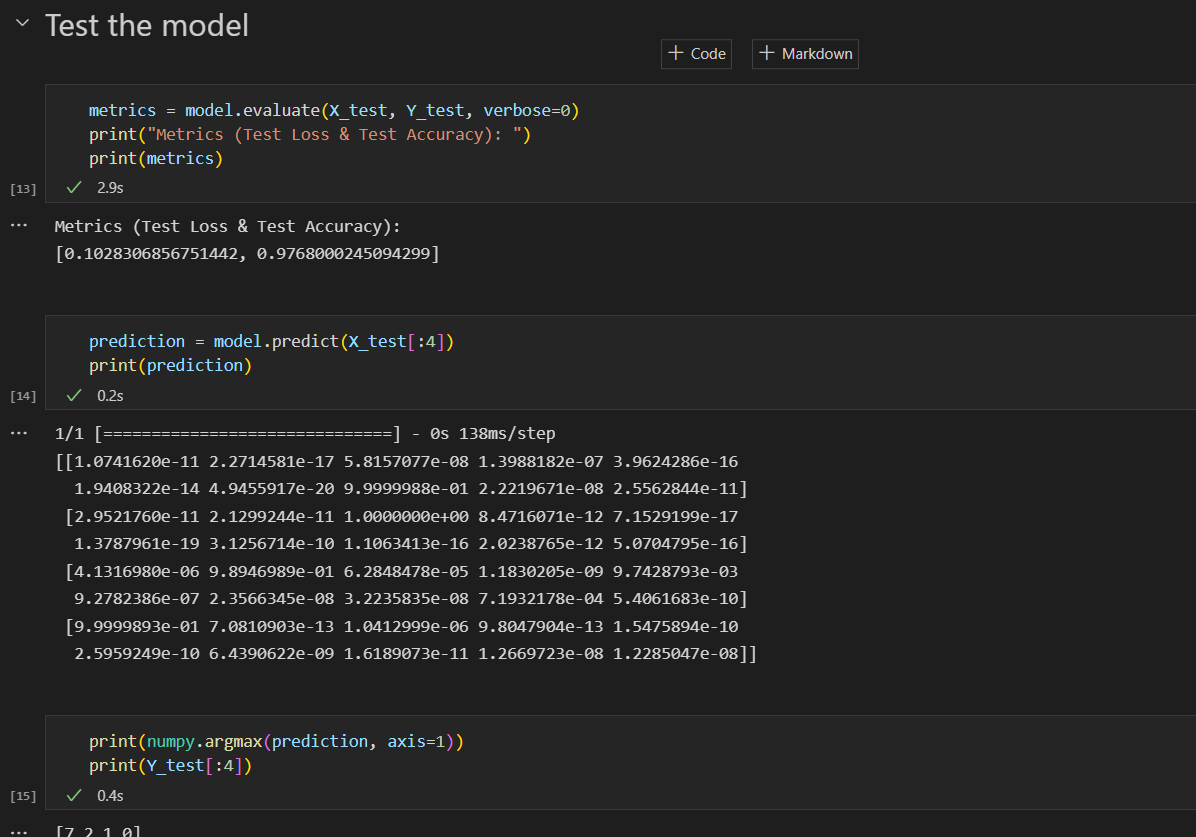
**7.1 SPRINT 1 **

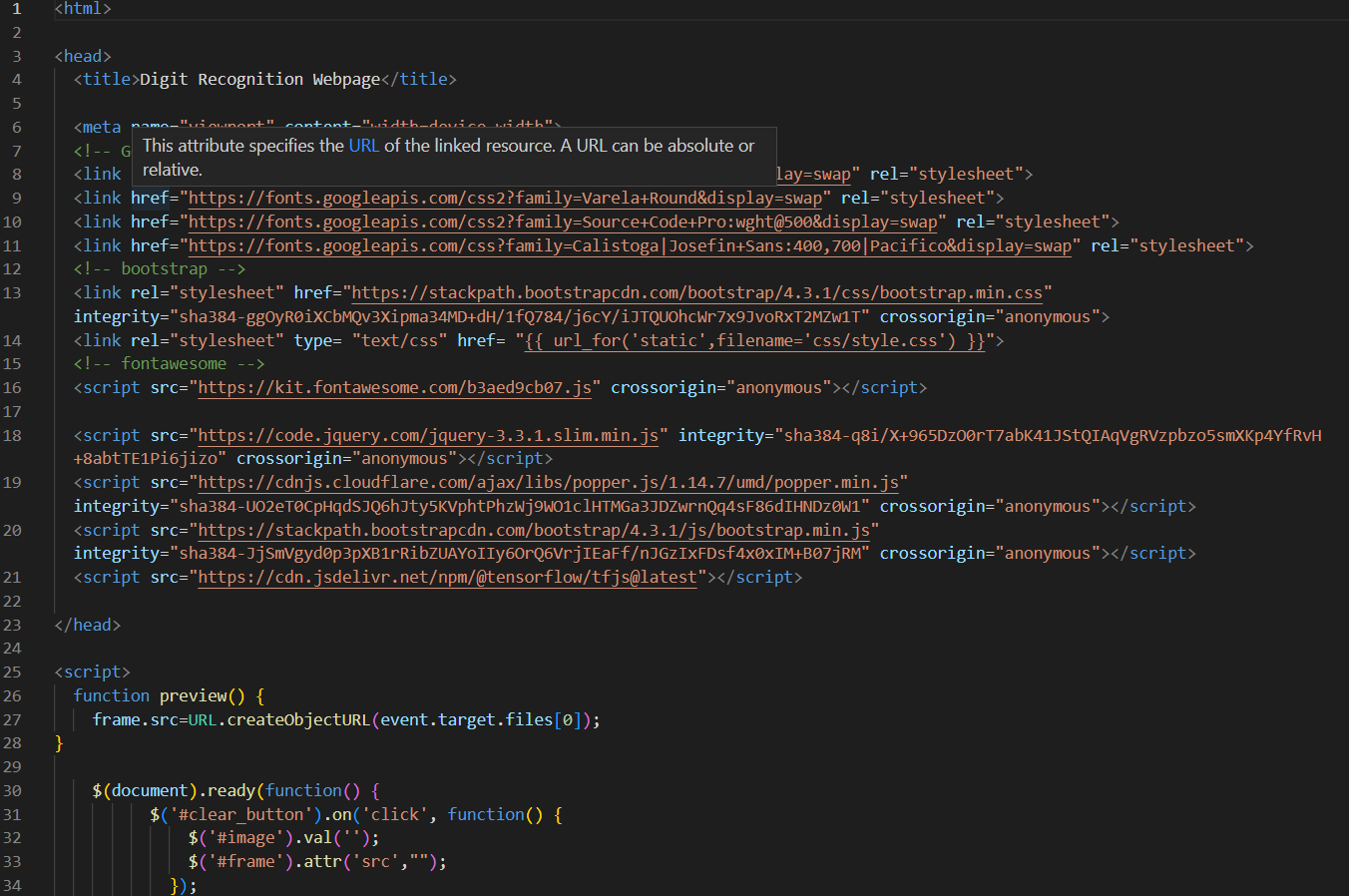
****

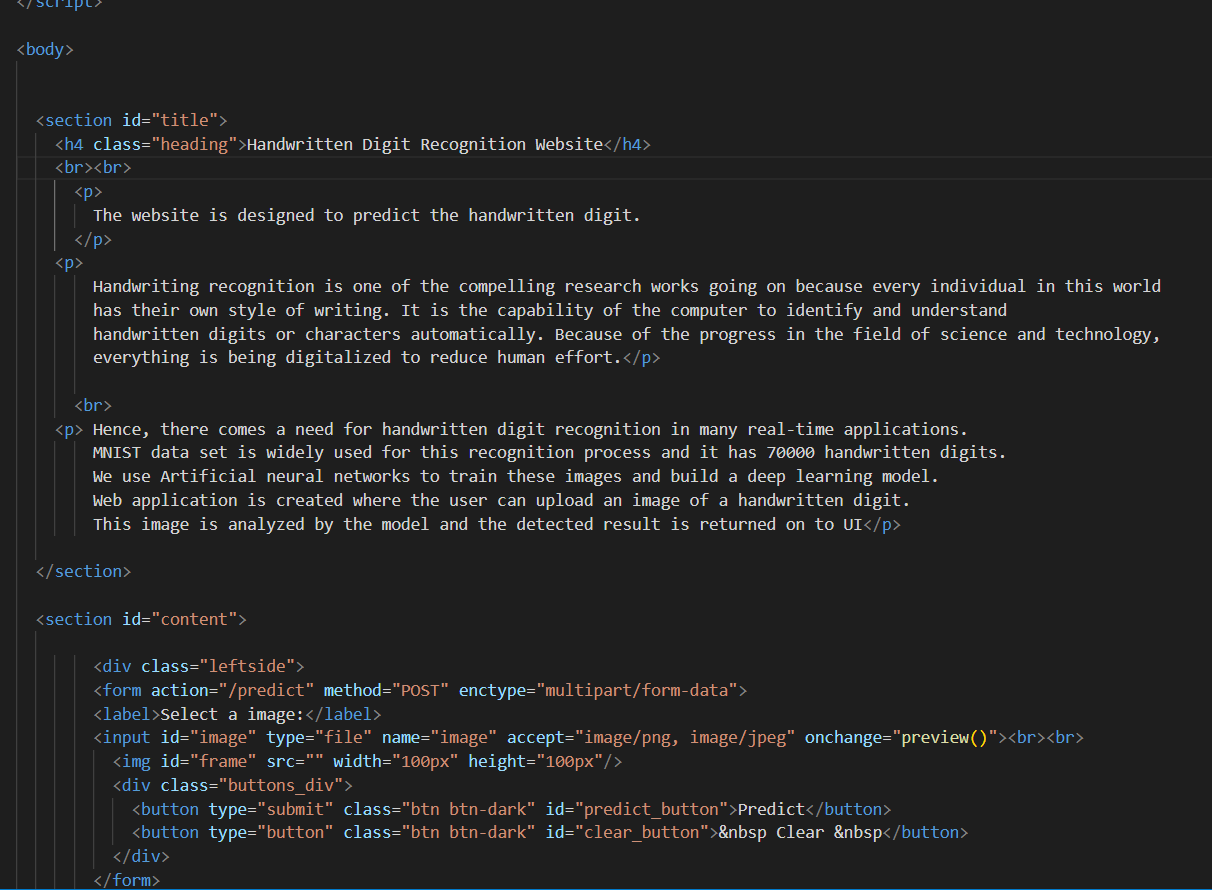
****

**7.2 SPRINT 2**

****

****

**7.3SPRINT 3 **

****

****

# 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 | FAIL |
| 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 | Check if the other predictions are displayed properly | The other predictions should be displayed properly | Working as expected | PASS |

### 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 |

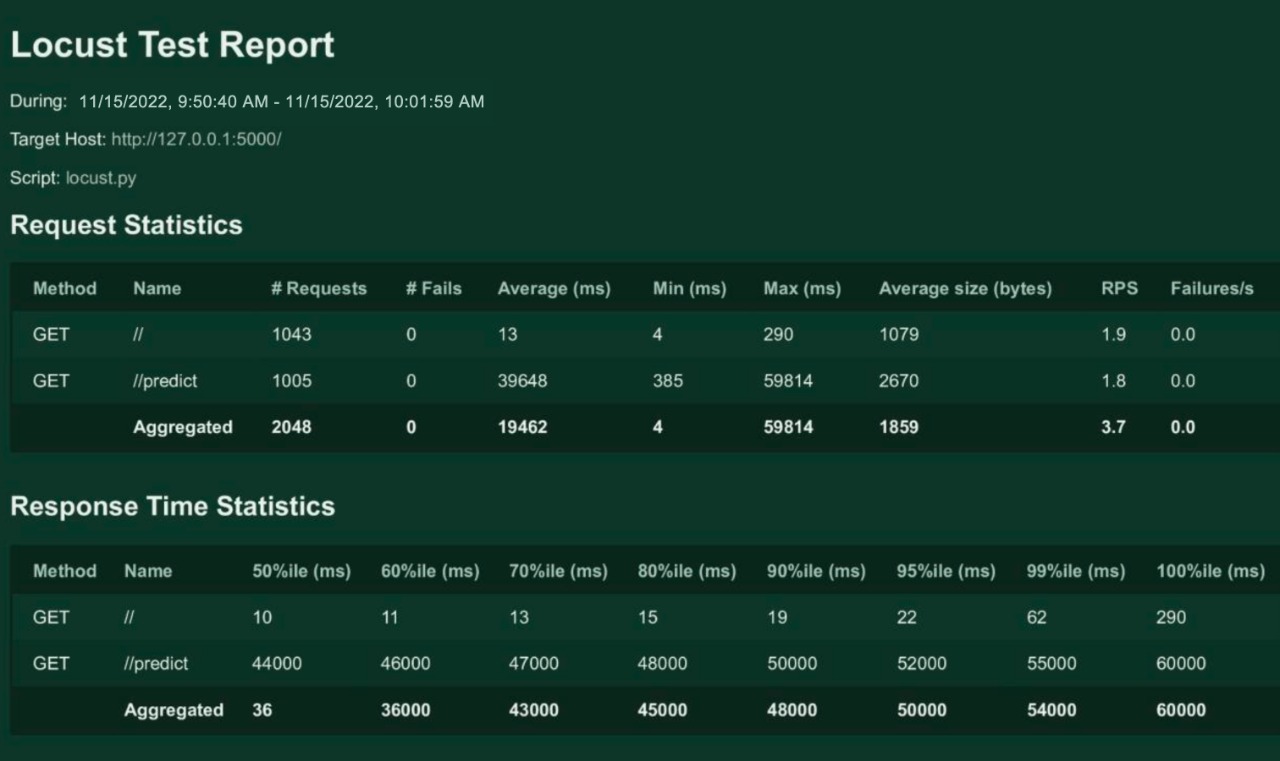
### 8.2.2 TEST CASE ANALYSIS

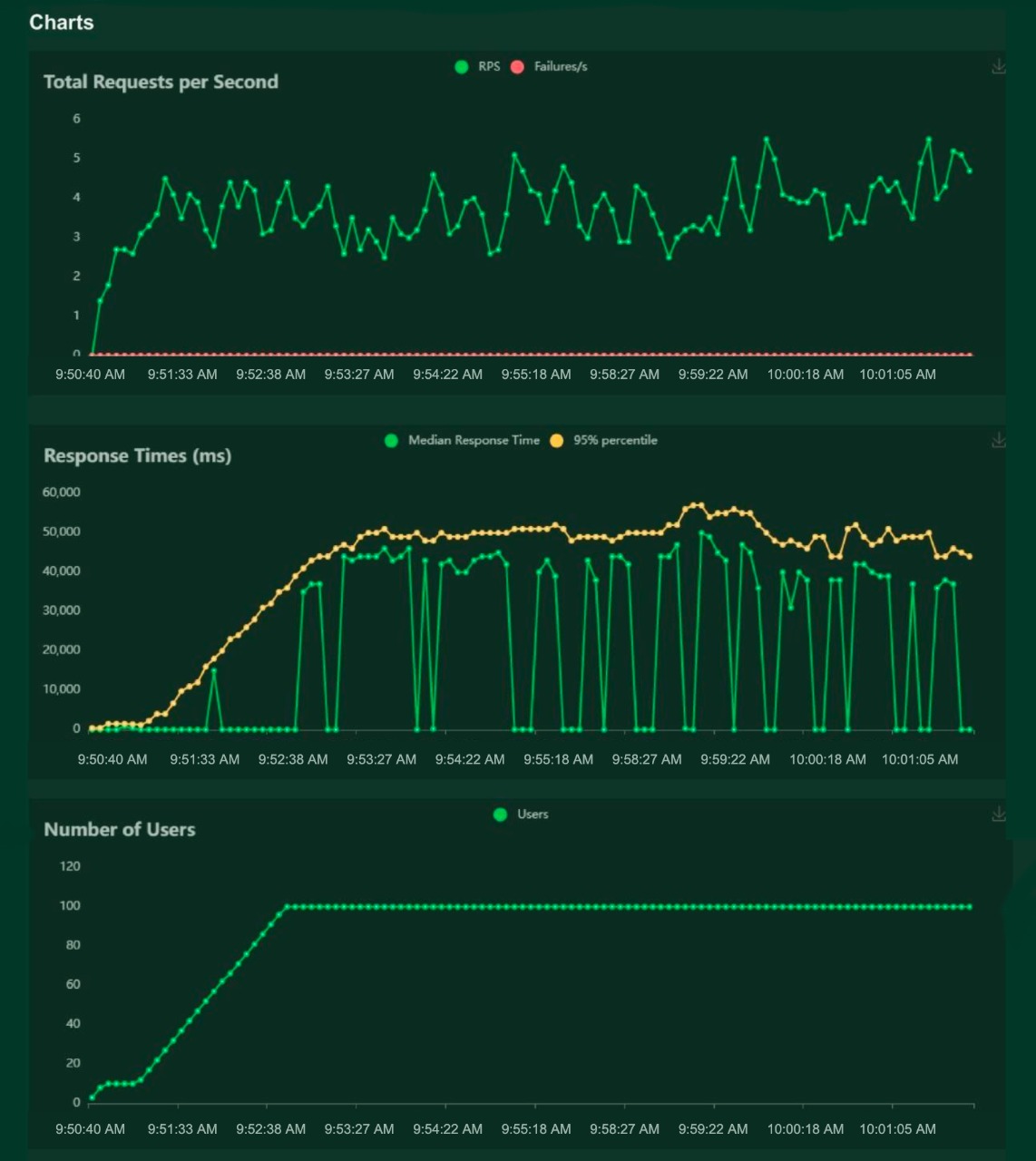
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Section | Total Cases | Not Tested | Fail | Pass |
| Client Application | 10 | 0 | 3 | 7 |
| Security | 2 | 0 | 1 | 1 |
| Performance | 3 | 0 | 1 | 2 |
| Exception Reporting | 2 | 0 | 0 | 2 |

# 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
    - Neural Network is used to train and identify written digits for greater efficiency.
    - The accuracy rate is very high.
    - Speed of data entry
    - It is much easier to dictate the machine than to write
    - Easier data retrieval

### 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
    - There is a wide range of handwriting – good and bad.
    - It is tricky for programmers to provide enough examples of how every character might look.
    - Customers must try with clear image and neat handwriting to get accuracy in digits.
    - Unclear image will not give accurate results.

**CHAPTER 11**

## CONCLUSION

Convolutional Neural Network (CNN) adds its significant improvement to the Manuscript Document Recognition System. This paper tells us the effectiveness of CNN-based classification of data and pre-processing methods. Our model clearly sees handwriting and achieves outgoing predictions of up to 82.16% and accurate predictions of up to 69.16%. However the model can be continuously developed using multiple training samples. This will help the model to learn as well as the generalize better. There are many images in the training set that are completely invisible to the human eye.

This project demonstrated a web application that uses machine learning to recognize 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.

Through extensive evaluation using a MNIST dataset, the present work suggests the role of various hyper-parameters. Fine tuning of hyper-parameters is essential in improving the performance of CNN architecture. We achieved a recognition rate of 99.89% with the Adam optimizer for the MNIST database, which is better than all previously reported results. The effect of increasing the number of convolutional layers in CNN architecture on the performance of handwritten digit recognition is clearly presented through the experiments.

# CHAPTER 12

## FUTURE SCOPE

This project can be enhanced with a great field of machine learning and artificial intelligence. The world can think of a software which can recognize the text from a picture and can show it to the others, for example a shop name detector. Or this project can be extended to a greater concept of all the character sets in the world. This project has not gone for the total English alphabet because there will be more and many more training sets and testing values that the neural network model will not be enough to detect. Think of a AI modeled car sensor going with a direction modeling in the roadside, user shall give only the destination.

All of these enhancement is an application of the texture analysis where advanced image processing, Neural network model for training and advanced AI concepts will come. These applications can be modeled further .As this project is fully done by free and available resources and packages this can be also a limitation of the project. The fund is very important because all machine learning libraries and advanced packages are not available for free. Unless of those the most of the visualizing platforms like on which developers are doing some works like Watson Studio or Aws. These all are mainly paid platforms where a lot of ML projects are going on.

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:

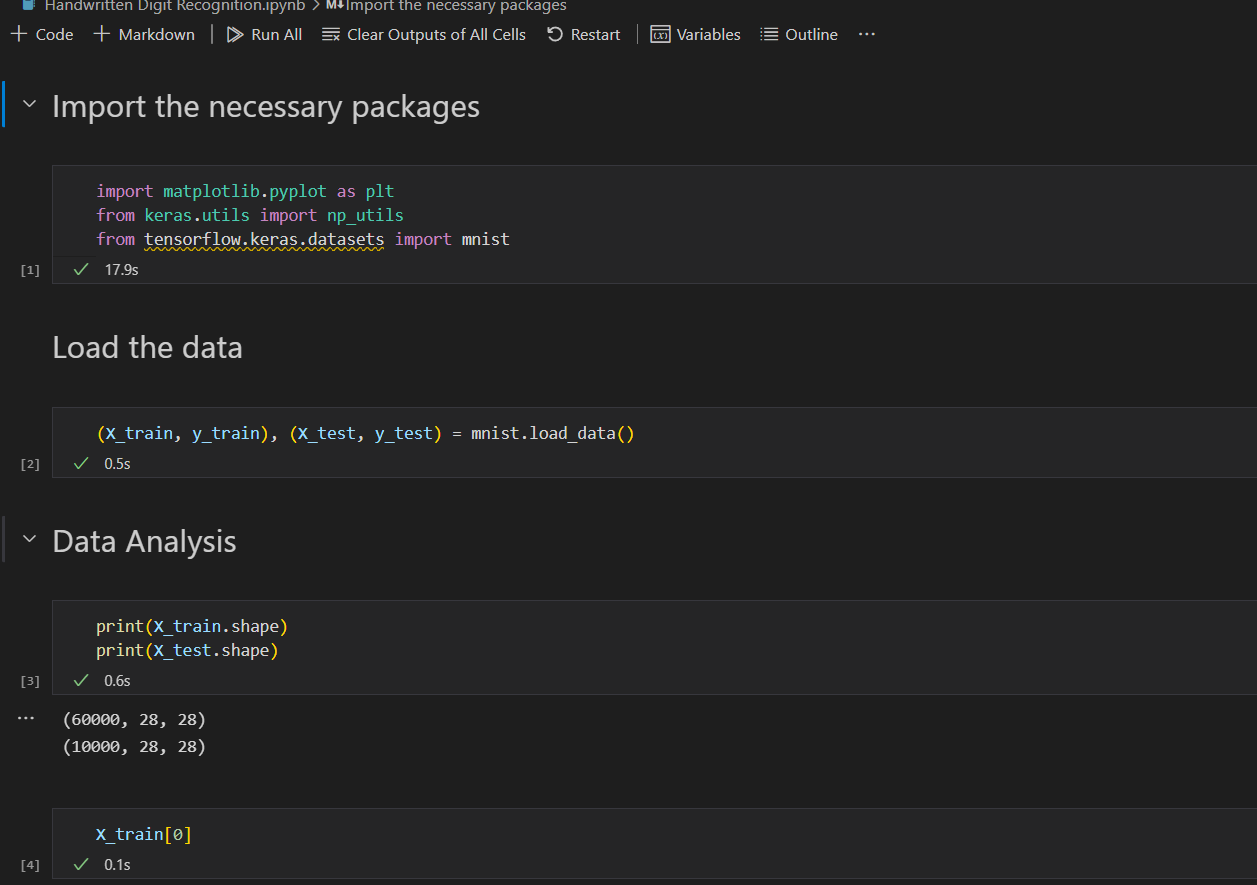
* + - Add support to detect from digits multiple images and save the results
    - Add support to detect multiple digits
    - Improve model to detect digits from complex images
    - Add support to different languages to help users from all over the world

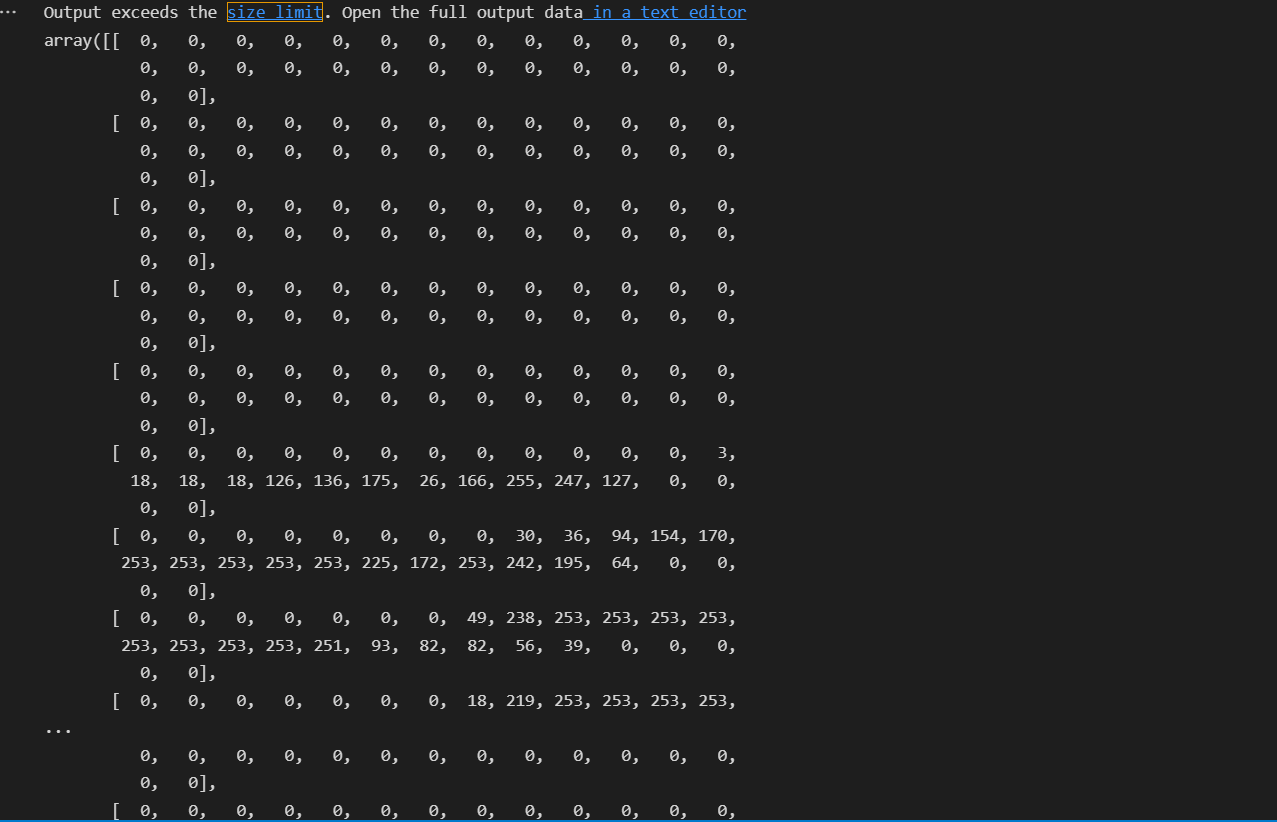
This project has endless potential and can always be enhanced to become better. Implementing this concept in the real world will beneﬁt several industries and reduce the workload on many workers, enhancing overall work efﬁciency.

## APPENDIX

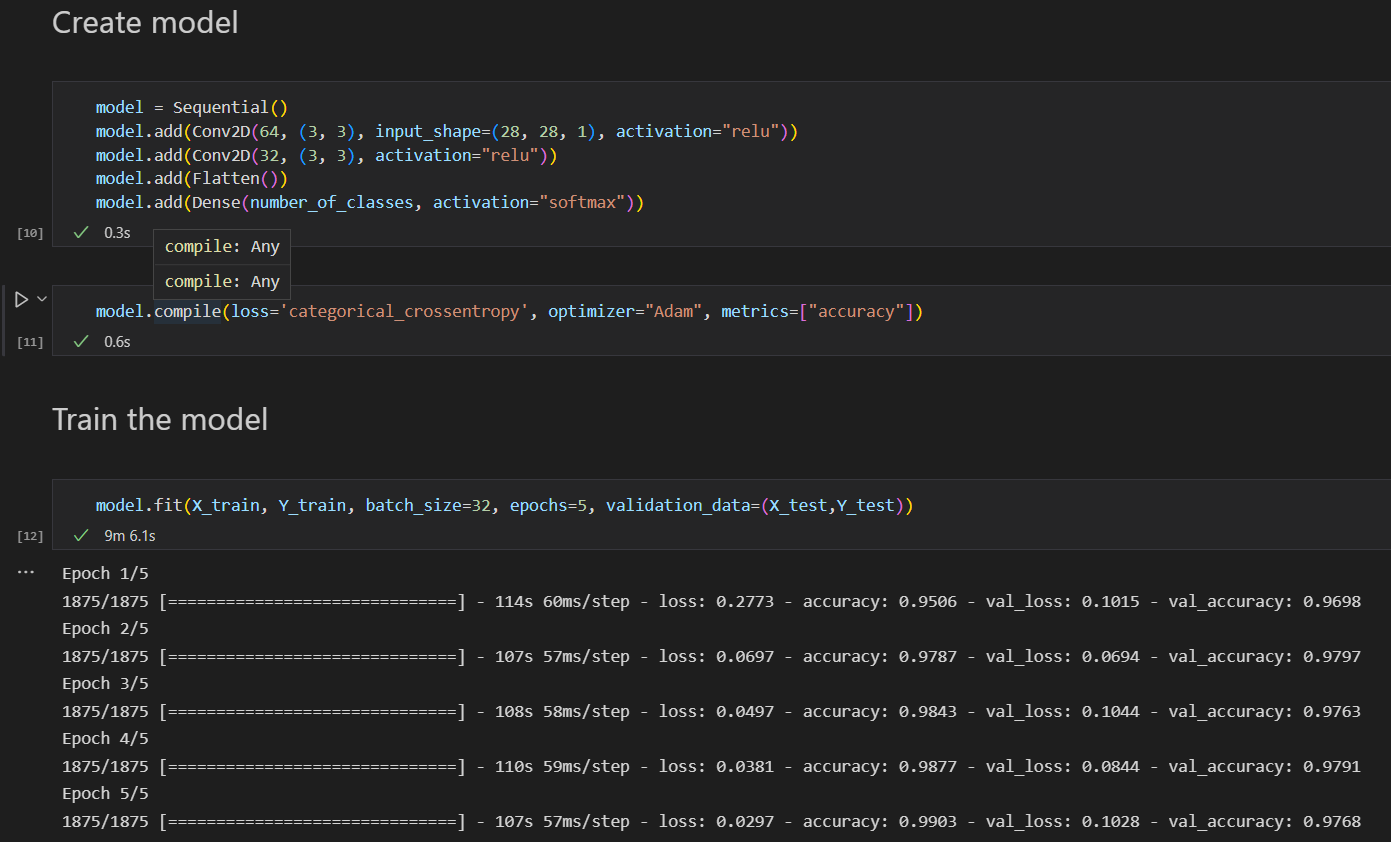
### SOURCE CODE

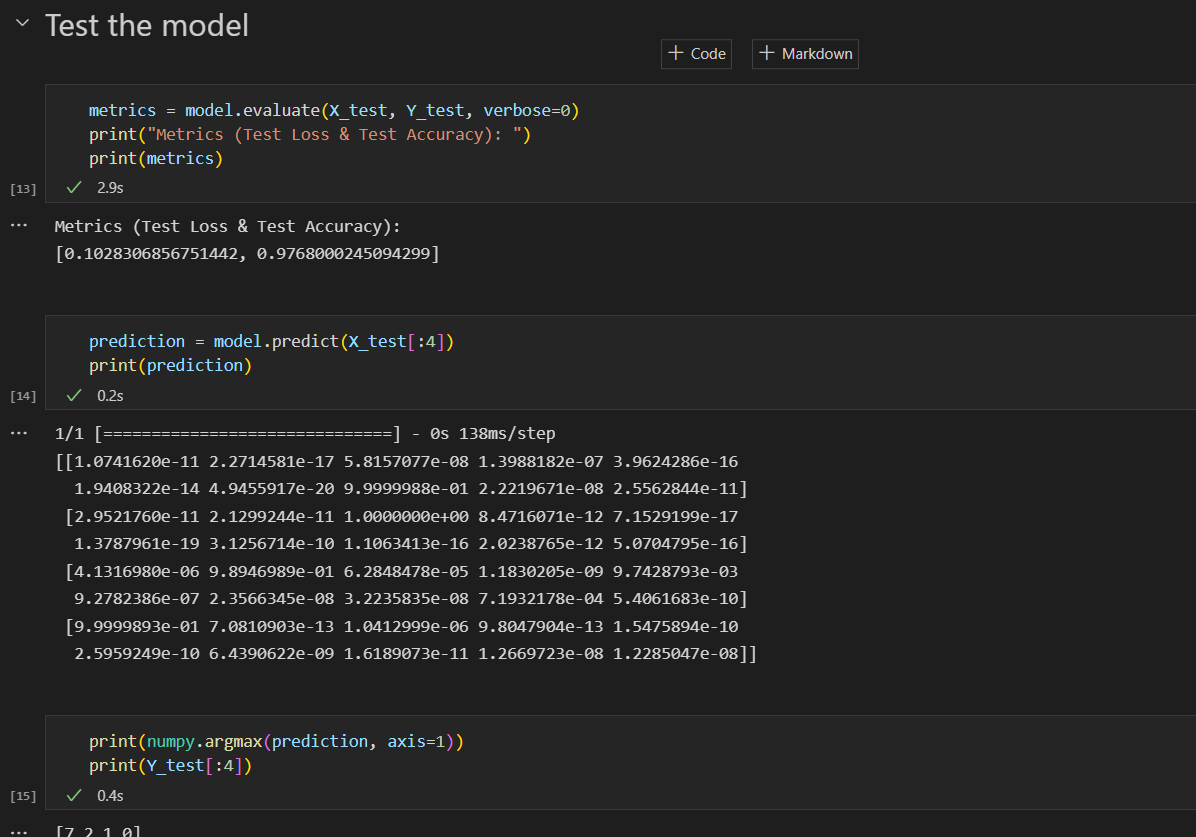
**MODEL CREATION:**

****

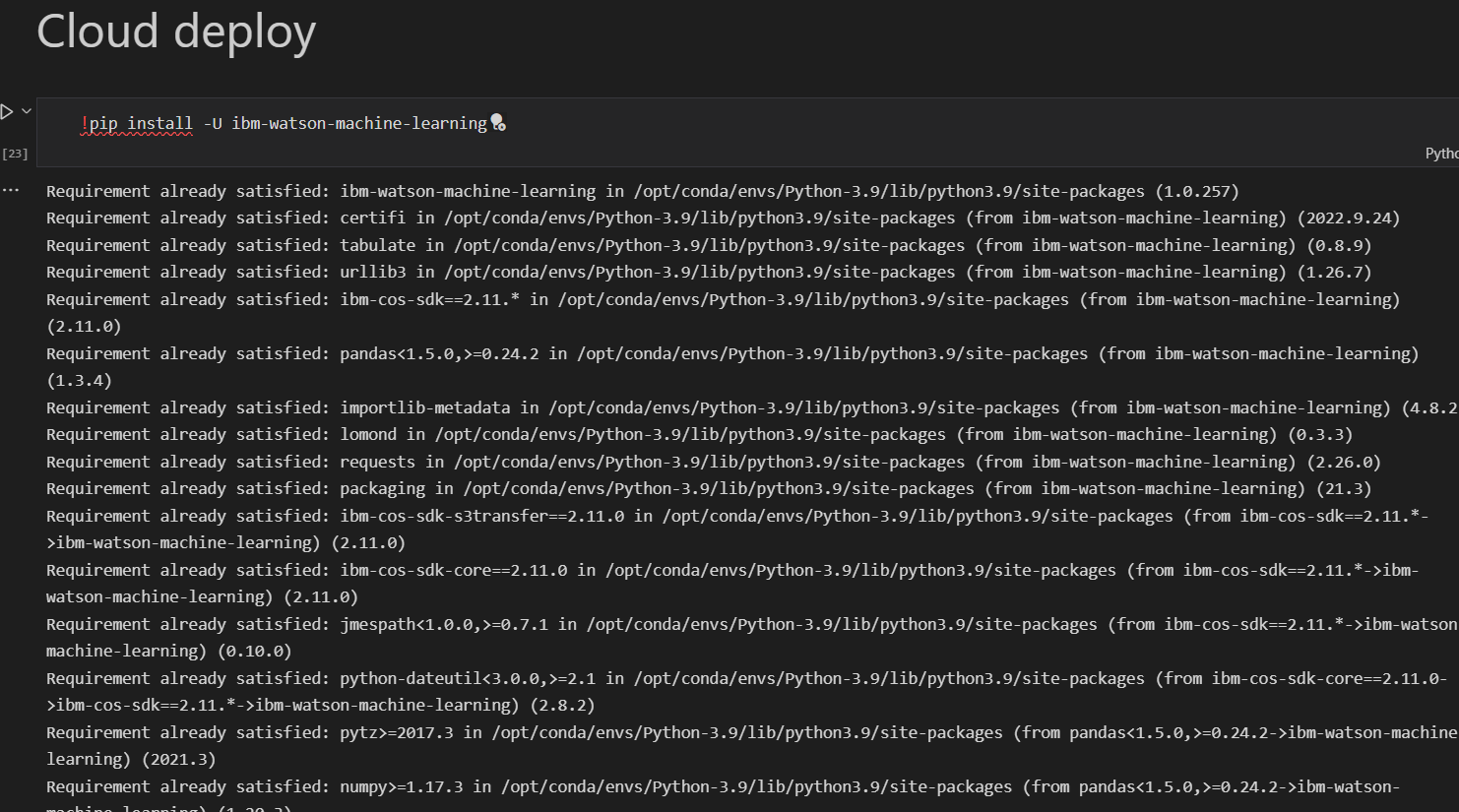


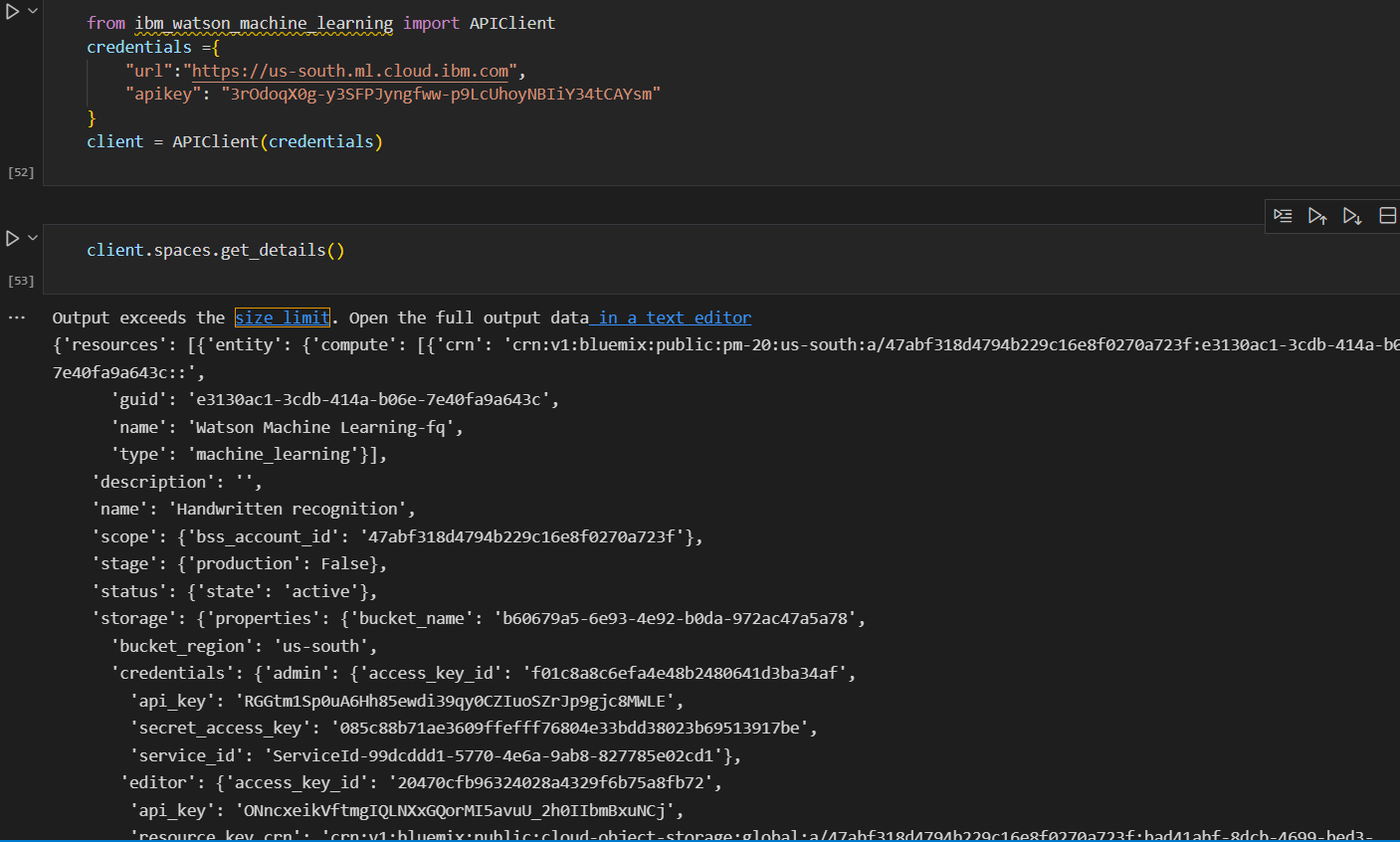


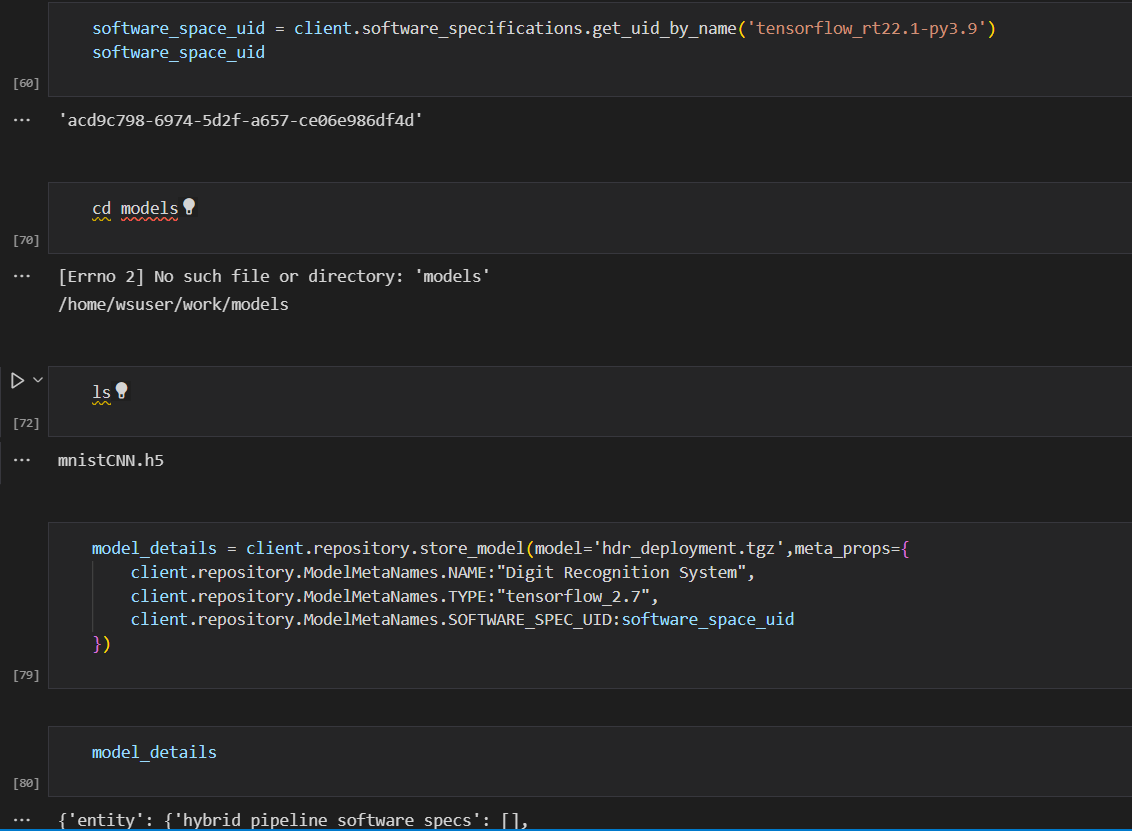
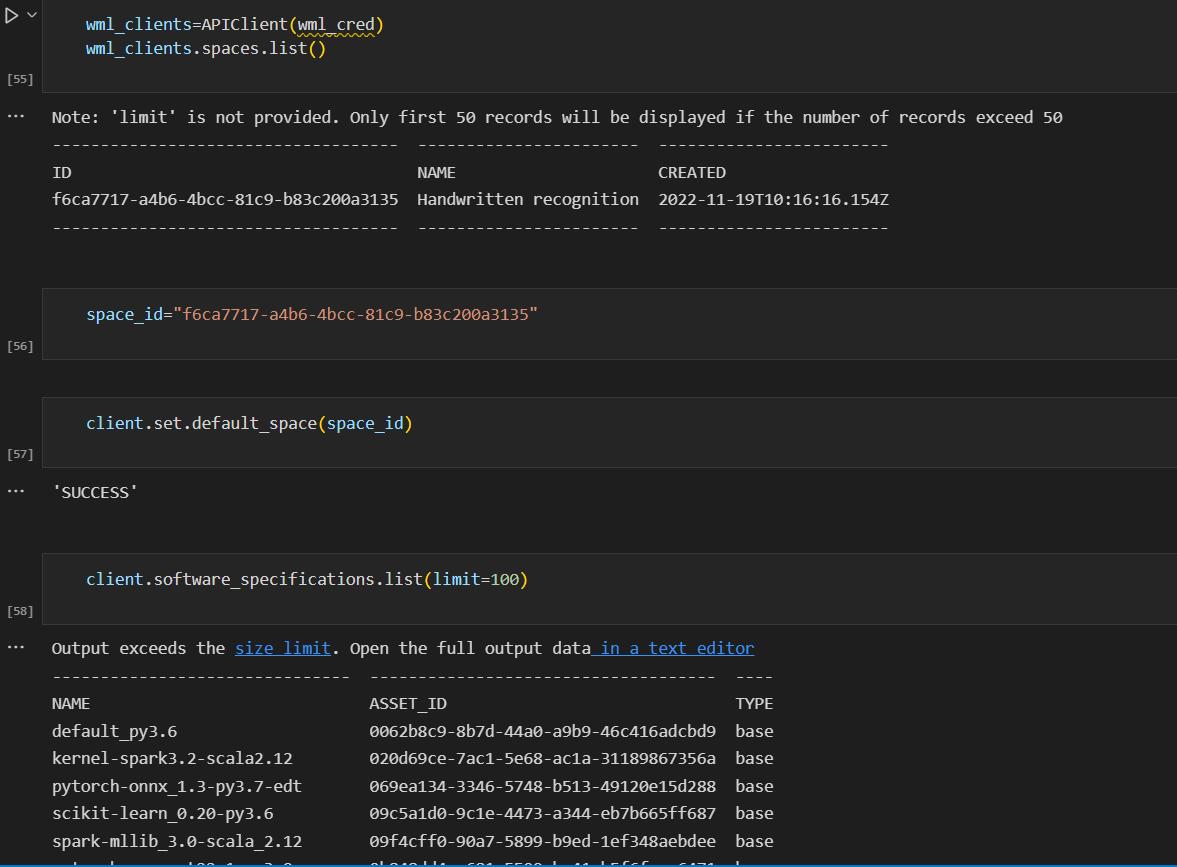


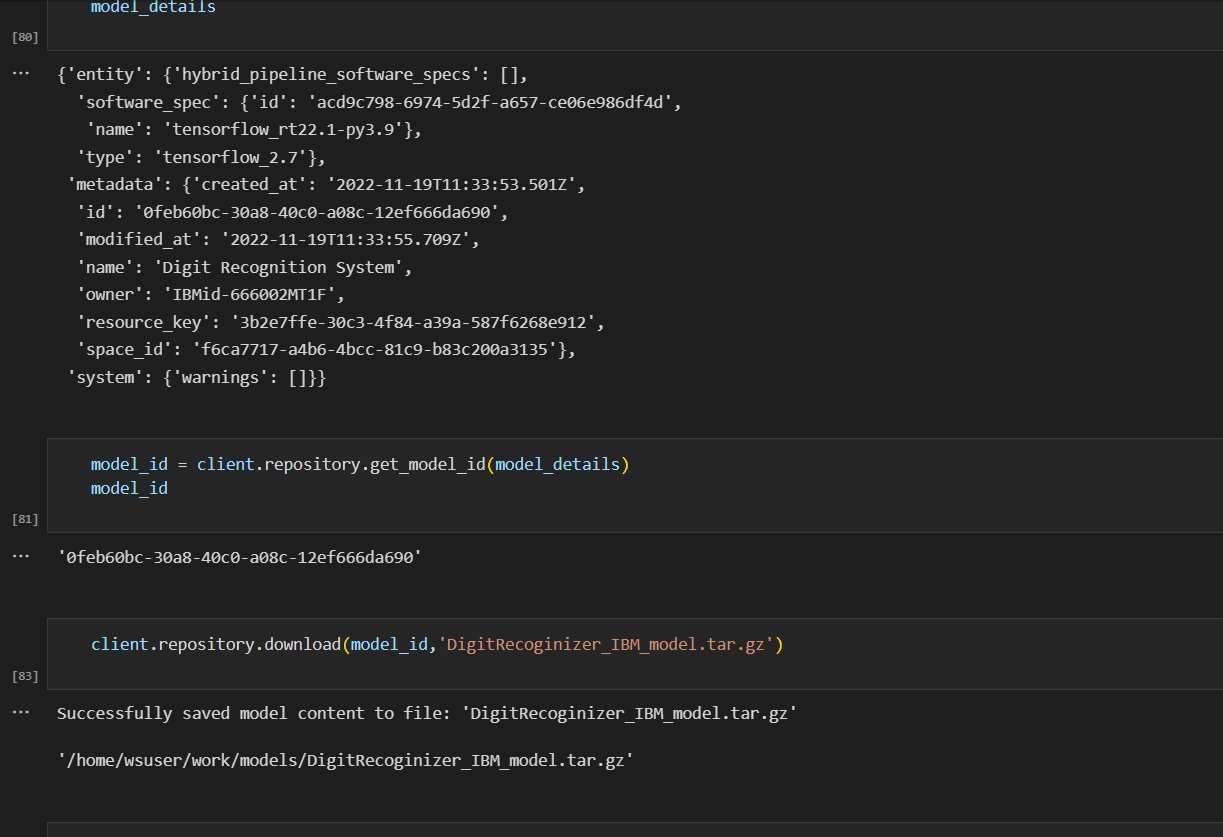


### TRAIN THE MODEL ON IBM:

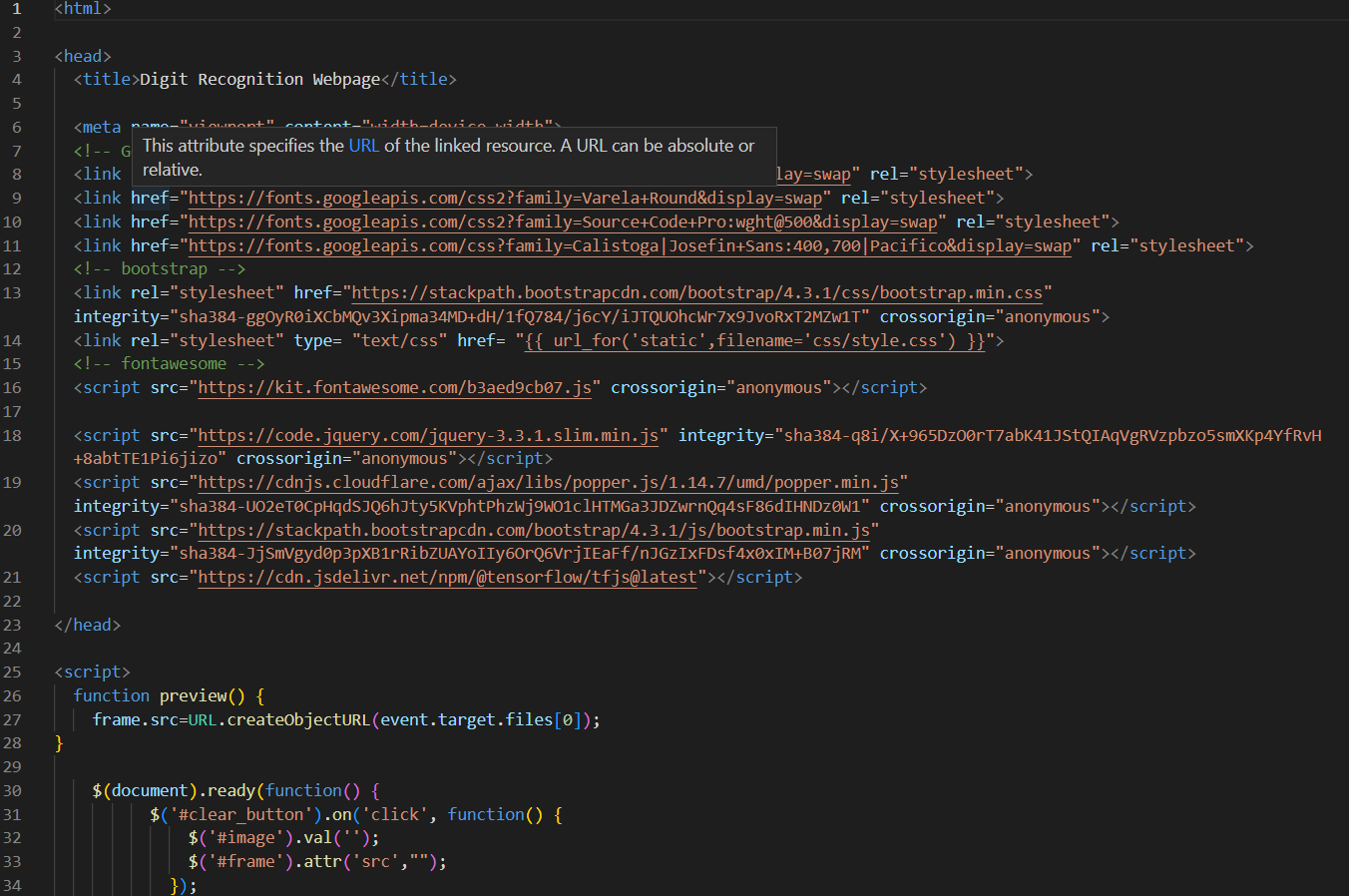


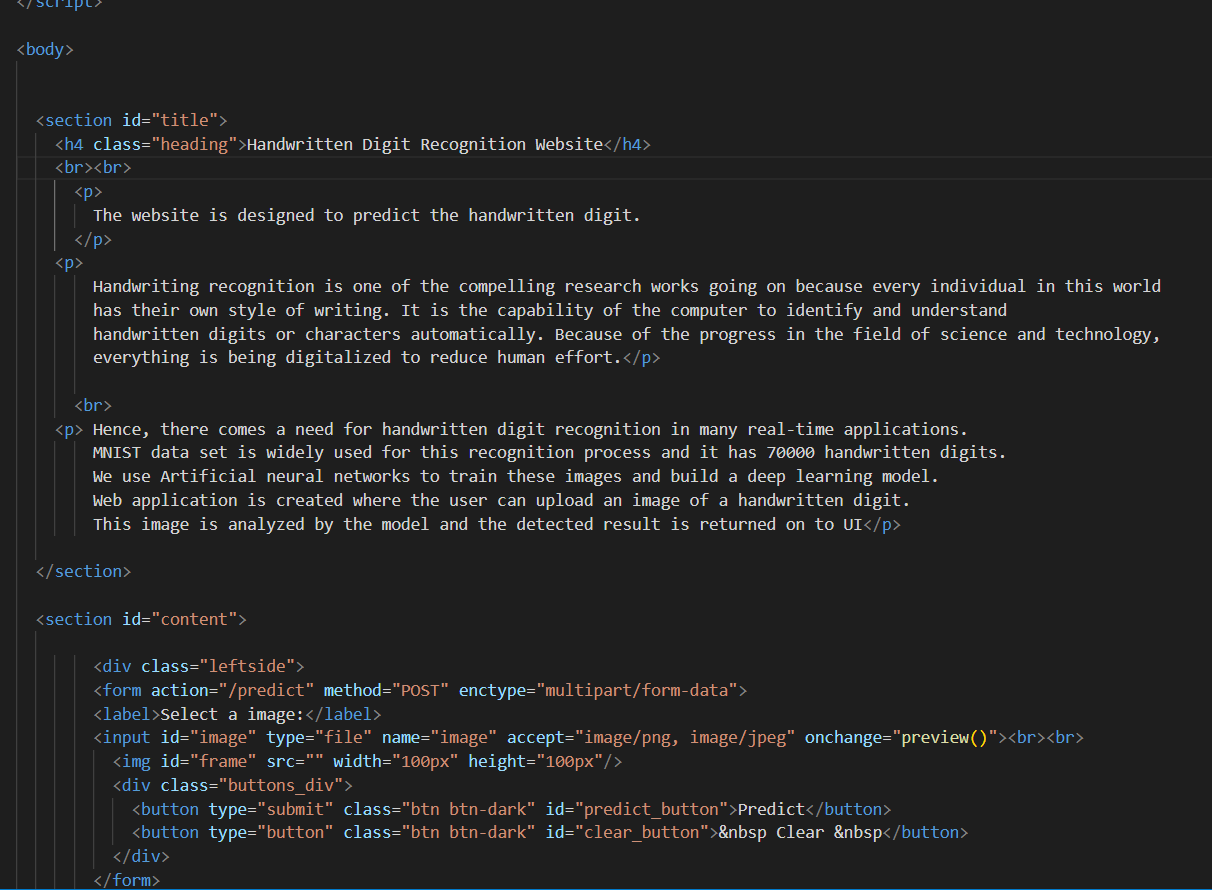






**HOME PAGE(HTML) – index.html**







**HOME PAGE(CSS) – style.css**

#clear\_button{  
 margin-left: 15px;  
 font-weight: bold;  
 color: blue;  
}  
  
#confidence{  
 font-family: 'Josefin Sans', sans-serif;  
 margin-top: 7.5%;  
}  
  
#content{  
 margin: 0 auto;  
 padding: 2% 15%;  
 padding-bottom: 0;  
}  
  
.welcome{  
 text-align: center;  
 position: relative;  
 color: honeydew;  
 background-color: greenyellow;  
 padding-top: 1%;  
 padding-bottom: 1%;  
 font-weight: bold;  
 font-family: 'Prompt', sans-serif;  
}  
  
#team\_id{  
 text-align: right;  
 font-size: 25px;  
 padding-right: 3%;  
}  
  
#predict\_button{  
 margin-right: 15px;  
 color: blue;  
 font-weight: bold;  
}  
  
#prediction\_heading{  
 font-family: 'Josefin Sans', sans-serif;  
 margin-top: 7.5%;  
}  
  
#result{  
 font-size: 5rem;  
}  
  
#title{  
 padding: 1.5% 15%;  
 margin: 0 auto;  
 text-align: center;  
}  
  
.btn {  
 font-size: 15px;  
 padding: 10px;  
 webkit-appearance: none;  
 background: #eee;  
 border: 1px solid #888;  
 margin-top: 20px;  
 margin-bottom: 20px;  
}  
  
.buttons\_div{  
 margin-bottom: 30px;  
 margin-right: 80px;  
}  
  
.heading{  
 font-family: 'Varela Round', sans-serif;  
 font-weight: 700;  
 font-size: 2rem;  
 display: inline;  
}  
  
.leftside{  
 text-align: center;  
 margin: 0 auto;  
 margin-top: 2%;  
 /\* padding-left: 10%; \*/  
}  
  
#frame{  
 margin-right: 10%;  
}  
  
.predicted\_answer{  
 text-align: center;  
 margin: 0 auto;  
 padding: 3% 5%;  
 padding-top: 0;  
 /\* padding-left: 10%; \*/  
}  
  
p{  
 font-family: 'Source Code Pro', monospace, sans-serif;  
 margin-top: 1%;  
}  
  
@media (min-width: 720px) {  
 .leftside{  
 padding-left: 10%;  
 }  
}

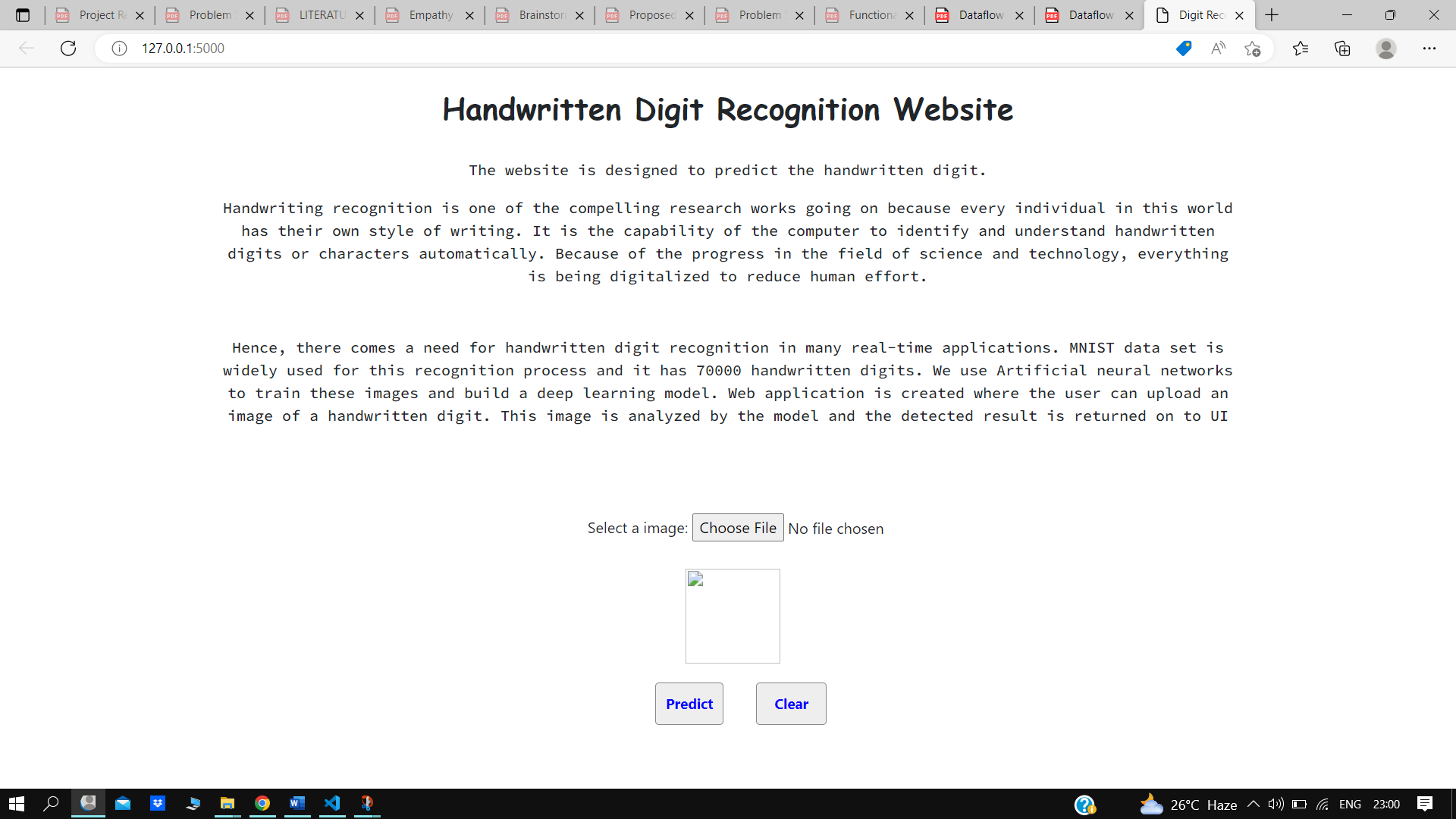
**PREDICT PAGE (HTML) – predict.html**

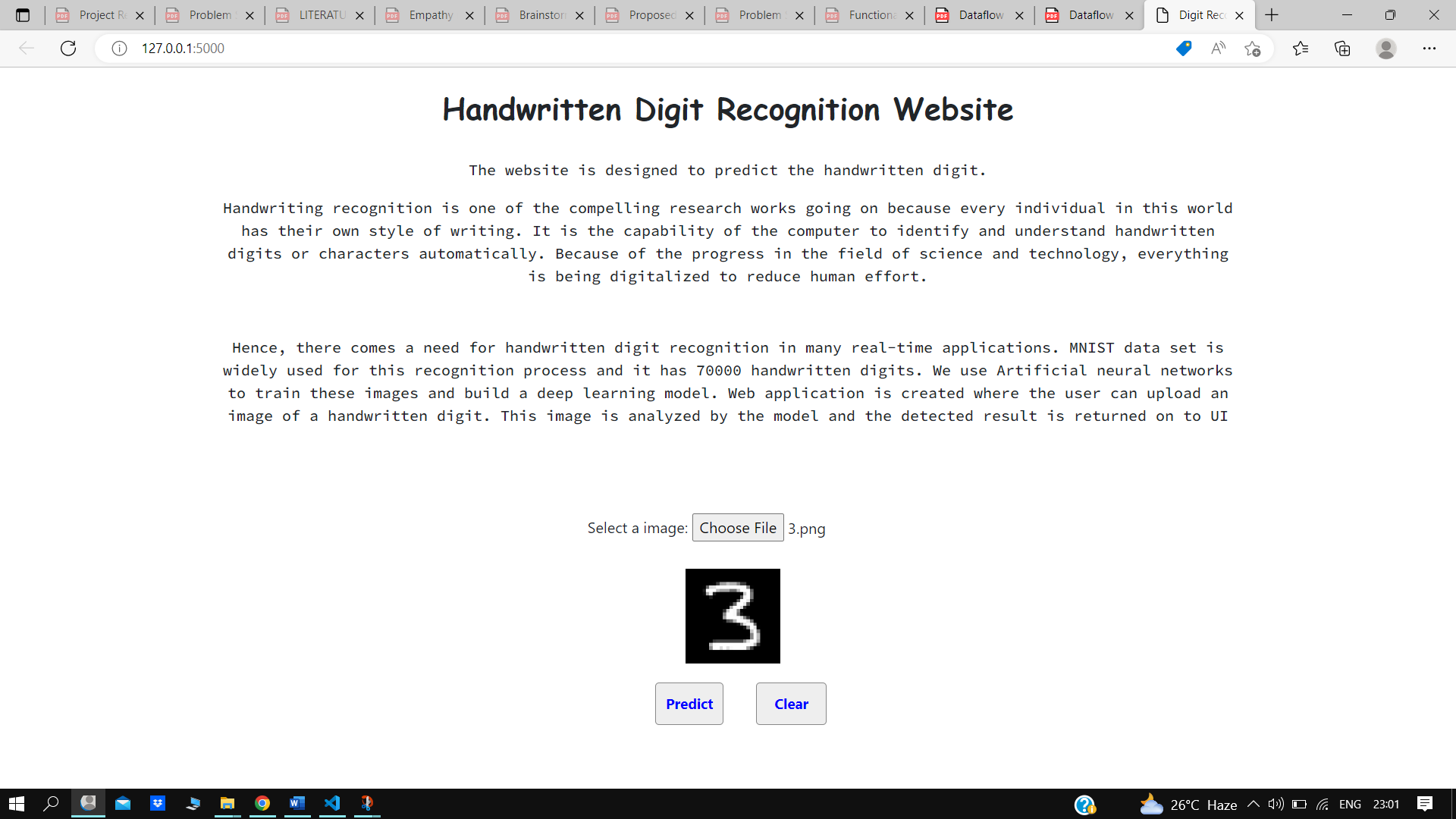
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Prediction</title>  
</head>  
  
<style>  
 body{  
 background-image: url('static/images/index6.jpg');  
 background-repeat: no-repeat;  
 background-size: cover;  
 }  
  
 #rectangle{  
 width:400px;  
 height:150px;  
 background-color: #5796a5;  
 border-radius: 25px;  
 position: absolute;  
 top:25%;  
 left:50%;  
 transform: translate(-50%,-50%);  
 }  
  
 #ans{  
 text-align: center;  
 font-size: 40px;  
 margin: 0 auto;  
 padding: 3% 5%;  
 padding-top: 15%;  
 color: white;  
 }  
  
</style>  
<body>  
 <div id="rectangle">  
 <h1 id="ans">Predicted Number : {{num}}</h1>  
 </div>  
</body>  
</html>

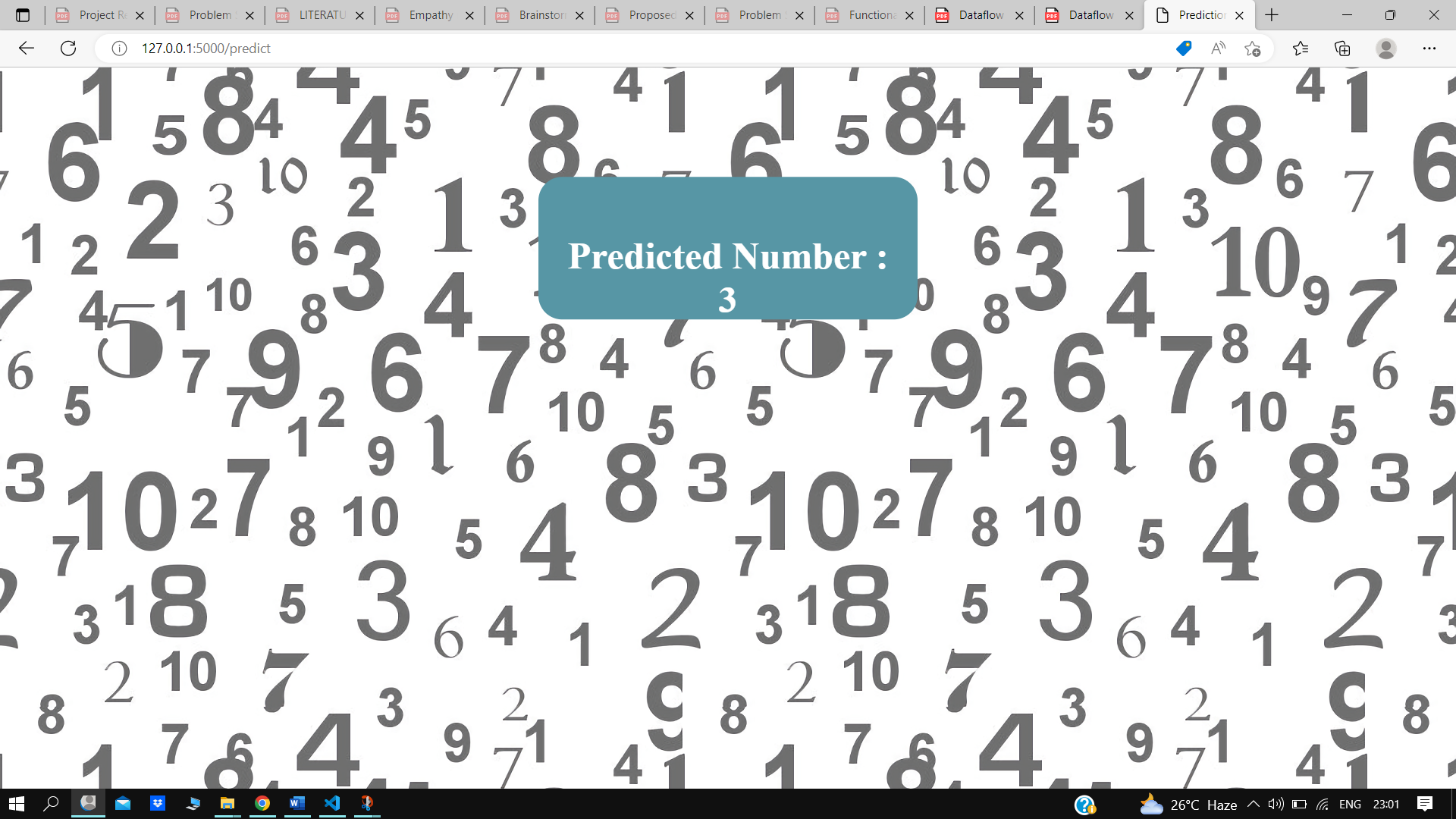
**FLASK APP - app.py**

import numpy as np  
import os  
from PIL import Image  
from flask import Flask, request, render\_template  
from werkzeug.utils import secure\_filename  
from keras.models import load\_model  
  
UPLOAD\_FOLDER = 'C:/Users/Dell/PycharmProjects/A-novel-method-for-digit-recognition-system/flask\_app/uploads'  
  
  
app = Flask(\_\_name\_\_)  
app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER  
  
model = load\_model("mnistCNN.h5")  
  
  
@app.route('/')  
def index():  
 return render\_template('index.html')  
  
  
@app.route('/predict', methods=['GET', 'POST'])  
def upload():  
 if request.method == "POST":  
 f = request.files["image"]  
 filepath = secure\_filename(f.filename)  
 f.save(os.path.join(app.config['UPLOAD\_FOLDER'], filepath))  
  
 upload\_img = os.path.join(UPLOAD\_FOLDER, filepath)  
 img = Image.open(upload\_img).convert("L") # convert image to monochrome  
 img = img.resize((28, 28)) # resizing of input image  
  
 im2arr = np.array(img) # converting to image  
 im2arr = im2arr.reshape(1, 28, 28, 1) # reshaping according to our requirement  
  
 pred = model.predict(im2arr)  
  
 num = np.argmax(pred, axis=1) # printing our Labels  
  
 return render\_template('predict.html', num=str(num[0]))  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=True, threaded=False)

**SCREENSHOTS:**

****

****

****

**GITHUB LINK:**

[**Click to git repository**](https://github.com/IBM-EPBL/IBM-Project-11680-1659338866)