**DETECTING OF PARKINSON’S DISEASE USING MACHINE LEARNING**

**A PROJECT  REPORT**

**Submitted by**

HARIHARAN M

HARIPRASATH B

HARI PRIYA D

RAJA M

**TEAM ID :** PNT2022TMID06357

**TABLE OF CONTENTS :**

|  |  |
| --- | --- |
| **S.NO** | **TITLE** |
| **1** | **INTRODUCTION** |
| 1.1 | Project Overview |
| 1.2 | Purpose |
| **2** | **LITERATURE SURVEY** |
| 2.1 | Existing Problem |
| 2.2 | References |
| 2.3 | Problem Statement |
| **3** | **IDEATION AND PROPOSED SOLUTION** |
| 3.1 | Empathy Map Canvas |
| 3.2 | Ideation and Brainstorming |
| 3.3 | Proposed Solution |
| 3.4 | Problem Solution Fit |
| **4** | **REQUIREMENT ANALYSIS** |
| 4.1 | Functional Requirement |
| 4.2 | Non Functional Requirement |
| **5** | **PROJECT DESIGN** |
| 5.1 | Data Flow Diagram |
| 5.2 | Solution and Technical Architecture |
| 5.3 | User Stories |
| **6** | **PROJECT PLANNING AND SCHEDULING** |
| 6.1 | Sprint Planning and Estimation |
| **S.NO** | **TITLE** |
| 6.2 | Sprint Delivery Schedule |
| 6.3 | Reports From JIRA |
| **7** | **CODING AND SOLUTIONING** |
| 7.1 | Feature 1 |
| 7.2 | Feature 2 |
| 7.3 | Database Schema |
| **8** | **TESTING** |
| 8.1 | Test Cases |
| 8.2 | User Acceptance Testing |
| **9** | **RESULTS** |
| 9.1 | Performance Metrics |
| **10** | **ADVANTAGES AND DISADVANTAGES** |
| **11** | **CONCLUSION** |
| **12** | **FUTURE SCOPE** |
| **13** | **APPENDIX**  Source Code  GitHub and Project Demo Link |

**1. INTRODUCTION**

1.1 **Project Overview**

The recent report of the World Health Organization shows a visible increase in the number and health burden of Parkinson’s disease patients increases rapidly. In China, this disease is spreading so fast and estimated that it reaches half of the population in the next 10 years. Classification algorithms are mainly used in the medical field for classifying data into different categories according to the number of characteristics. Parkinson’s disease is the second most dangerous neurological disorder that can lead to shaking, shivering, stiffness, and difficulty walking and balance. It caused mainly due by the breaking down of cells in the nervous system. Parkinson’s can have both motor and non-motor symptoms. The motor symptoms include slowness of movement, rigidity, balance problems, and tremors. If this disease continues, the patients may have difficulty walking and talking. The non-motor symptoms include anxiety, breathing problems, depression, loss of smell, and change in speech. If the above-mentioned symptoms are present in the person then the details are stored in the records. In this paper, the author considers the speech features of the patient, and this data is used for predicting whether the patient has Parkinson’s disease or not. Neurodegenerative disorders are the results of progressive tearing and neuron loss in different areas of the nervous system. Neurons are functional units of the brain. They are contiguous rather than continuous. A good healthy looking neuron as shown in fig 1 has extensions called dendrites or axons, a cell body, and a nucleus that contains our DNA. DNA is our genome and a hundred billion neurons contain our entire genome which is packaged into it. When a neuron gets sick, it loses its extension and hence its ability to communicate which is not good for it and its metabolism becomes low so it starts to accumulate junk and it tries to contain the junk in the little packages in little pockets. When things become worse and if the neuron is a cell culture it completely loses its extension, becomes round and full of vacuoles. This work deals with the prediction of Parkinson’s disorder which is now a day is tremendously increasing incurable disease. Parkinson’s disease is a most spreading disease which gets its name from James Parkinson who earlier described it as a paralysis agitans and later gave his surname was known as PD. It generally affects the neurons which are responsible for overall body movements. The main chemicals are dopamine and acetylcholine which affect the human brain. There is a various environmental factor which has been implicated in PD below are the listed factor which caused Parkinson’s disease in an individual.

**1.1.1 Parkinson’s disease symptoms**

The symptoms of Parkinson’s disease broadly divided into two categories.

• Motor symptoms: This is a symptom where any voluntary action involved. It indicates the movement-related disorders like tremors, rigidity, freezing, Bradykinesia or any voluntary muscle movement.

• Non-Motor symptoms: Non motor symptoms include disorders of mood and affect with apathy, cognitive dysfunction as well as complex behavioral disorders. There are two other categories of PD which are divided by doctors: Primary symptom and Secondary symptom.

• Primary symptoms: It is the most important symptom. Primary symptoms are rigidity, tremor and slowness of movement.

• Secondary symptoms: It is a symptom that directly impacts the life of an individual. These can be either motor or non-motor. Its effect depends on person to person. A very wide range of symptoms is associated with Parkinson‘s,. Besides these symptoms, there are some other symptoms found that lead to Parkinson’s disease. These symptoms are micrographic, decreased olfaction & postural instability, slowing of the digestive system, constipation, fatigue, weakness, and Hypotension. Speech difficulties i.e. dysphonia (impaired speech production) and dysarthria (speech articulation difficulties) are found in patients with Parkinson’s.

**1.2 Purpose**

The main aim is to predict the prediction efficiency that would be beneficial for the patients who are suffering from Parkinson and the percentage of the disease will be reduced. Generally in the first stage, Parkinson's can be cured by the proper treatment. 10 So it‘s important to identify the PD at the early stage for the betterment of the patients. The main purpose of this research work is to find the best prediction model i.e. the best machine learning technique which will distinguish the Parkinson’s patient from the healthy person. The techniques used in this problem are KNN, Naïve Bayes, and Logistic Regression. The experimental study is performed on the voice dataset of Parkinson’s patients which is downloaded from the Kaggle. The prediction is evaluated using evaluation metrics like confusion matrix, precision, recall accuracy, and f1-score. The author used feature selection where the important features are taken into consideration to detect Parkinson’s.

**2. LITERATURE SURVEY**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **JOURNAL NAME** | **AUTHORS** | **OBSERVATION** |
| 1 | Bio composite Multiple Uses foraNew Approach in the Diagnosisof Parkinson’s Disease Using a Machine Learning Algorithm | AbdallahAl- Husban Mustafa Mahdi Abdulridha  A.A.Hamad Mohamad | These results were obtained when the data set in the whole group was classified with the most relevant features of 45% (accuracy: 85%, sensitivity:0.94, specificity:00.78,  measure: 0.86, kappa: 0.72,  Acu: 0.86). |
| 2 | Imperative Role ofMachine Learning Algorithm for Detection of  Parkinson’s Disease: Review, Challenges and Recommendatio ns | Arti Rana Ankur Dumka Rajesh Singh  Manoj Kumar Panda  Neeraj Priyadarshi | PD was obtained by L1-Norm SVM with K- fold cross- validation, with 99%; in handwritten patterns, it was obtained by bagging ensemble, with 97.96%; and for gait analysis, it was obtained by SVM with 100%. This reviewaddressedvarious challenges and also provided some futurerecommendations and opportunities, as we observed that there is still a lot of work that has to be performed in the future. |
| 3 | Predicting SeveriOf Parkinson’s DiseaseUsing Deep Learning. | Sridhti Grover  SaloniBhartia  Akshama  Abhilasha  Yadav  Seeja K.R | we have used a dataset of 5875 instances, the accuracy of our approach can be further improved by implementing it on a larger dataset, having a greater number of instances of each severity class as well as on a combined database of patients’ voice data and other patient . |

|  |  |  |  |
| --- | --- | --- | --- |
| 4 | Detection of Parkinson’s Disease Using Machine Learning Algorithm | Shrihari K Kulkarni  K R Sumana | The results for algorithms based on accuracy are like this: Decision Tree 93.25%,  Logistic Regression 91.25%, Naive Bayes 94.5%, and RNN 88.75%. |
| 5 | Parkinson’s Disease Detection using machine Learning Techniques | C K Gomathy | Wecanpredict the parkinson’s disea se in patient’s body using machine learning technology and this method makes the process easy to our user. Our analysis provides very accurate performance in detecting Parkinson's disease using XGBOOST algorithm |
| 6 | Parkinson disease onset detection Using Machine Learning | Sonia Singla | Parkinson disease data and find out XGBoost is the best Algorithm to predict the onset of the disease which will enable early treatment and save a life. |
| 7 | A Survey of Detection of Parkinson Disease using DeepLearning Technique | Sakshi Jadhav Seema Thorat Sakshi Fokane Rahul Chakre | The work is mainly focusing on advancement of predictive models to achieve good accuracy in predicting valid disease outcomes using deep learning methods like prediction based on Artificial Neural Network (ANN). In this paper, Deep Learning techniques are proposed for the prediction of Parkinson Disease in early stage. |
| 8 | Machine Learning for the Diagnosis of | Jei Mei Christian Deserosiers | Machine learning approaches therefore have the potential to provide clinicians with |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Parkinson's Disease |  | additional tools to screen, detect or diagnose PD. |
| 9 | Parkinson’s disease rating scales | Jamir Pitton Rissardo AnaLeticia Fornari Caprara | Instead of developing new scales, the studies should focus on the clinimetric evaluation, and assess these scales in different ethnic origins translating the scales and evaluating their quality. |
| 10 | Parkinson Disease diagnosis using machine learning | Anila M  Dr. G.Pradeepini | Developing a very faster classifier using novel architecture of neural network combined with specific approach may work better. |

**2.1 Existing problem :**

The main idea behind the implementation is to classify a person as Healthy or having Parkinson’s disease by looking at the Spiral Drawing made by the person. The Spiral Drawing created by The project aims at presenting a solution for Parkinson’s disease detection using Spiral Drawings and CNN. a healthy person will look almost similar to a standard spiral shape. However, a spiral drawn by a person with Parkinson’s disease will highly deviate from a perfect spiral shape and look distorted due to slow motor movements and decreased coordination between hand and brain**.**

**2.2 References :**

1.Mahlknecht, P.; Krismer, F.; Poewe, W.; Seppi, K. Meta-Analysis of Dorsolateral Nigral Hyperintensity on MagneticResonance Imaging as a Marker for Parkinson’s Disease.Mov. Disord. 2017, 32, 619–623.

2. Dickson, D.W. Neuropathology of Parkinson disease.Parkinsonism Relat. Disord. 2018, 46 (Suppl. 1), S30–S33.

3. Aich, S., Sain, M., Park, J., Choi, K.W. and Kim, H.C., 2017, November.A mixed classification approach for the prediction of Parkinson's disease

4. M. Abdar and M. Zomorodi-Moghadam, “Impact of Patients’ Gender on Parkinson’s disease using Classification Algorithms” Journal of AI and Data Mining, vol. 6, 2018.

5. Satish Srinivasan, Michael Martin & Abhishek Tripathi, “ANN based Data Mining Analysis of Parkinson’s Disease” International Journal of Computer Applications, vol. 168, June 2017.

6. Ramzi M. Sadek et al., “Parkinson’s Disease Prediction using Artificial Neural Network” International Journal of Academic Health and Medical Research, vol. 3, Issue 1, January 2019.

7. Arvind Kumar Tiwari, “Machine Learning based Approaches for Prediction of Parkinson’s Disease”, Machine Learning and Applications: An International Journal (MLAU) vol. 3, June 2016.

8. Dr. Anupam Bhatia and Raunak Sulekh, “Predictive Model for Parkinson’s Disease through Naive Bayes Classification” International Journal of Computer Science & Communication vol. 9, Dec. 2017, pp. 194- 202, Sept 2017 - March 2018.

9. Dragana Miljkovic et al, “Machine Learning and Data Mining Methods for Managing Parkinson’s Disease” LNAI 9605, pp 209-220, 2016.

10. R. P. Duncan, A. L. Leddy, J. T. Cavanaugh et al., “Detecting and predicting balance decline in Parkinson disease: a prospective cohort study,” Journal of Parkinson’s Disease, vol. 5, no. 1, pp. 131–139, 2015.

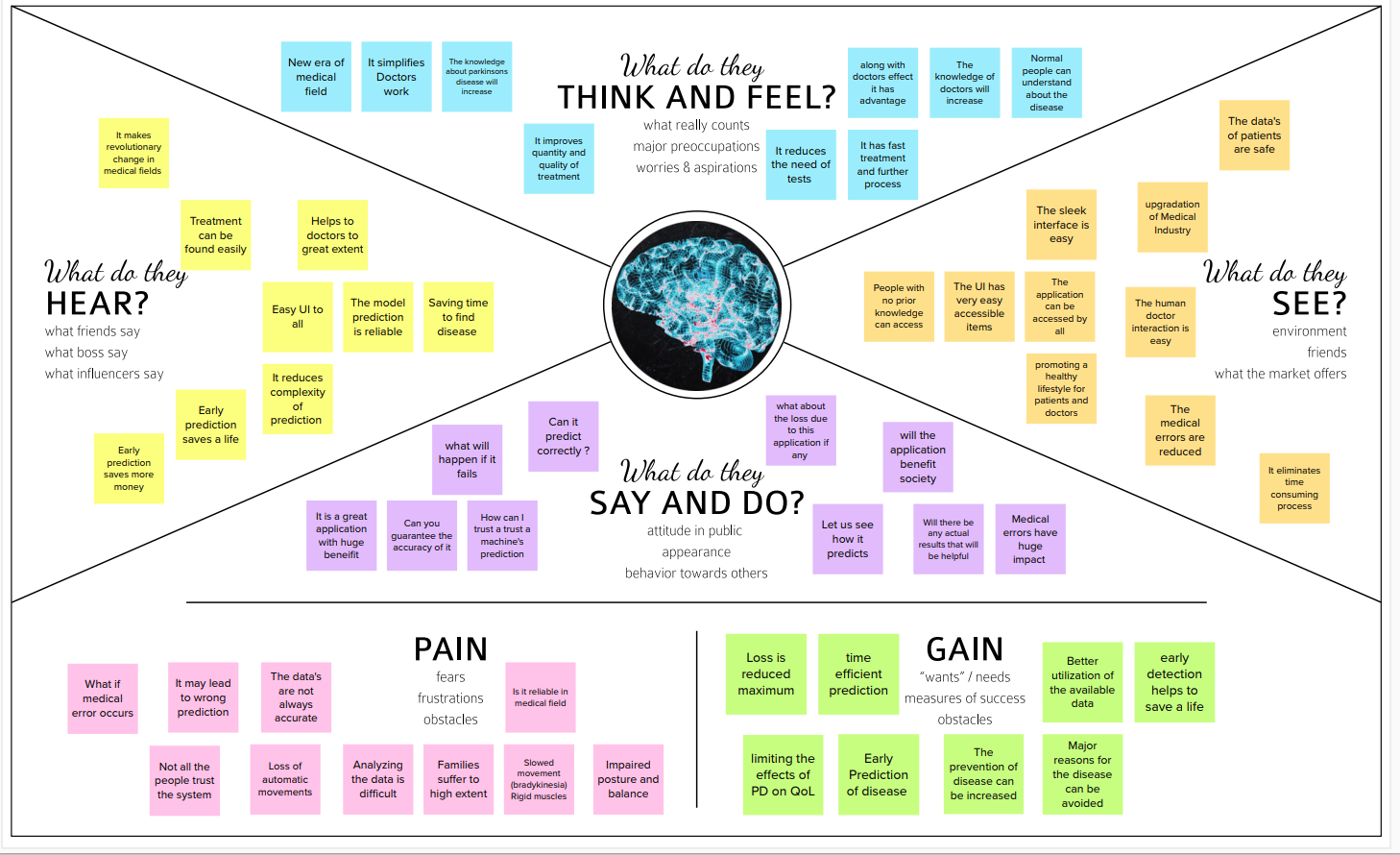
**2.3 Problem Statement :**

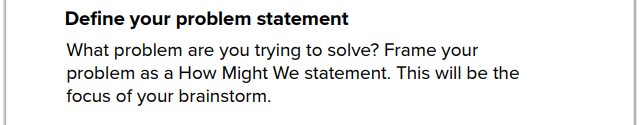
The main aim is to predict the prediction efficiency that would be beneficial for the patients who are suffering from Parkinson and the percentage of the disease will be reduced. Generally in the first stage, Parkinson's can be cured by the proper treatment. 10 So it‘s important to identify the PD at the early stage for the betterment of the patients. The main purpose of this research work is to find the best prediction model i.e. the best machine learning technique which will distinguish the Parkinson’s patient from the healthy person. The techniques used in this problem are KNN, Naïve Bayes, and Logistic Regression. The experimental study is performed on the voice dataset of Parkinson’s patients which is downloaded from the Kaggle. The prediction is evaluated using evaluation metrics like confusion matrix, precision, recall accuracy, and f1-score. The author used feature selection where the important features are taken into consideration to detect Parkinson’s.

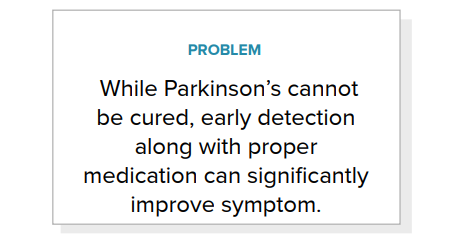
**3. IDEATION & PROPOSED SOLUTION**

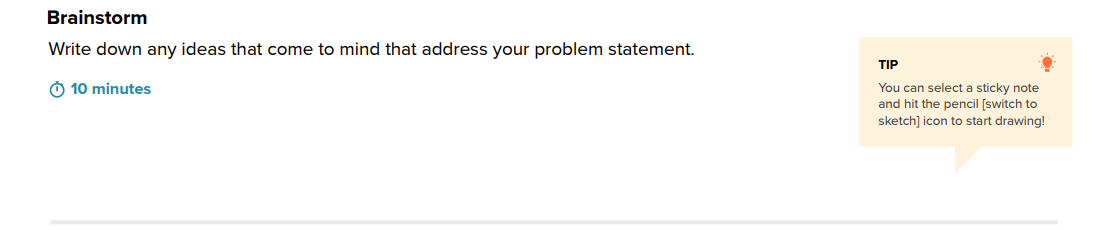
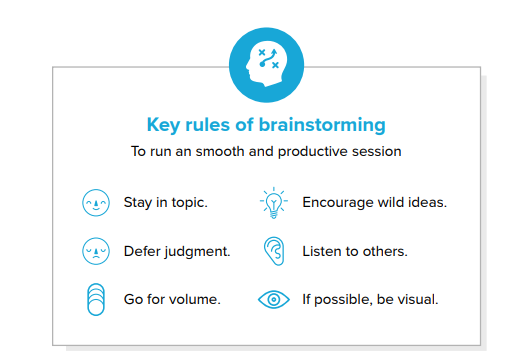
**3.1 Empathy Map Canvas :**

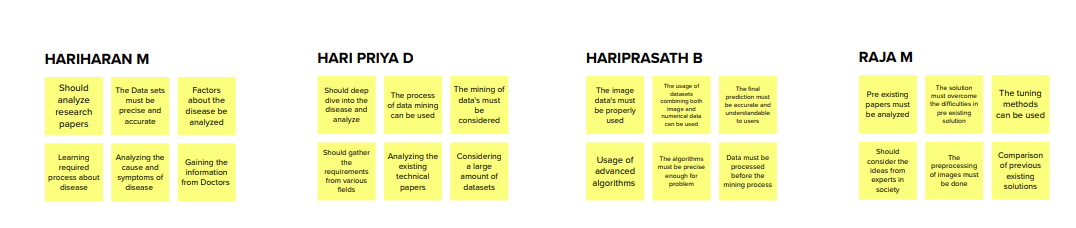
More than 10 million people are living with Parkinson’s Disease worldwide, according to the Parkinson’s Foundation. While Parkinson’s cannot be cured, early detection along with proper medication can signifcantly improve symptoms and quality of life

**3.2 Ideation & Brainstorming :**





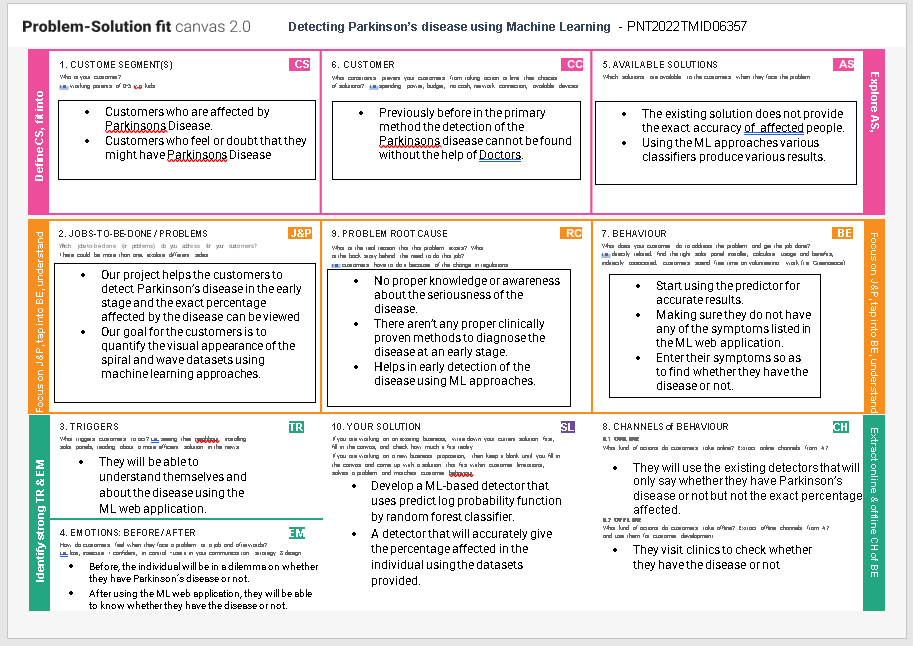




**3.3 Proposed Solution :**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Parameter** | **Description** |
|  | Problem Statement (Problem to be solved) | More than 10 million people are living with Parkinson’s Disease worldwide, according to the Parkinson ‘s Foundation. While Parkinson’s cannot be cured, early detection along with proper medication can significantly improve symptoms and quality of life. |
|  | Idea / Solution description | Prediction of Parkinson’s disease with higher accuracy and estimation using web application which will help stakeholders such as the government and health insurance companies . It can identify patients at risk of disease or health conditions. |
|  | Novelty / Uniqueness | It identifies patients at risk of disease or health conditions at early stages . The use of OpenCV techniques to eliminate even the use of paper for the drawing test also contributes to the novelty factor. The application in case of a prediction leaning to a confirmation of the condition can provide awareness and various information about the condition including location and other details of treatment centres and specialists. Since the application must work with the patients physical and personal information, the security factor is of paramount importance. The usage of OTP verified authentication means is a novelty factor |
|  | Social Impact / Customer Satisfaction | The prediction of disease can effectively control and prevent large scale outbreaks and epidemics .It also detects the abnormal distribution of disease in prior and hence the customers could be saved from traumas which will lead them to the road of happiness. |
|  | Business Model (Revenue Model) | The platform is free. It can used by people pertaining to all age groups with limited technical knowledge and can be prescribed to others .The model helps to increase economic status and has easy user interface. |
|  | Scalability of the Solution | Highly scalable .Produces accurate results with small and large amount of data . It may accessed by any number of people and advancement of chatbots can be introduced |

**3.4 Problem Solution fit :**



|  |  |  |
| --- | --- | --- |
| **S.NO** | **Functional**  **Requirement (Epic)** | **Sub Requirement**  **( Story / Sub-Task)** |
| 1 | User Authentication | The users must be registered first and can be only able to access the web application . This is to ensure that the web application is used for a good reason. |
| 2 | Web Service Management Process | Web Service Management process by Web Portal admin in registering web client to do SSO or member data communication. The web page is hosted in cloud. |
| 3 | Data Management | The Web server and Portal manager can have access to data to edit and update again to server. |
| 4 | Testing | Applying the algorithms on the test data |
| 5 | Confirmation | Display the result with the description of having Parkinson’s or not |

**4.REQUIREMENT ANALYSIS**

**4.1 Functional Requirement :**

**4.2 Non-Functional Requirements :**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Non-Functional**  **Requirements** | **Description** |
| 1 | Usability | The webpage loading for users submitting their image input details at the web application must be loaded fast than rendering more time. |
| 2 | Security | Authorization access scenarios and definitions, hand-over procedures for patient records. The image and other inputs of patients must be highly secured and can’t be accessible to others. |
| 3 | Reliability | The prediction of the system must be with higher accuracy so that the output from the application can be trusted by the users without any doubts and can be used for further dragonising process with Doctors. |
| 4 | Performance | The landing page supporting 5,000 users per hour must provide 6 second or less response time in a Chrome desktop browser, including the rendering of text and images and over an LTE connection and the uploading of Data (image) must also should be fast and the output page should be rendered within seconds. |
| 5 | Availability | The web application should be available to all Doctors across the globe and can be implemented in every hospital so that the patients can use it effectively . |
| 6 | Scalability | The System must function using Cloud and during a down process also it must satisfy the maximum number of clients. . The system must use higher RAM and CPU processing in Server to handle multiple request at same time. |

**5. PROJECT DESIGN**

**5.1 Data Flow Diagrams :**

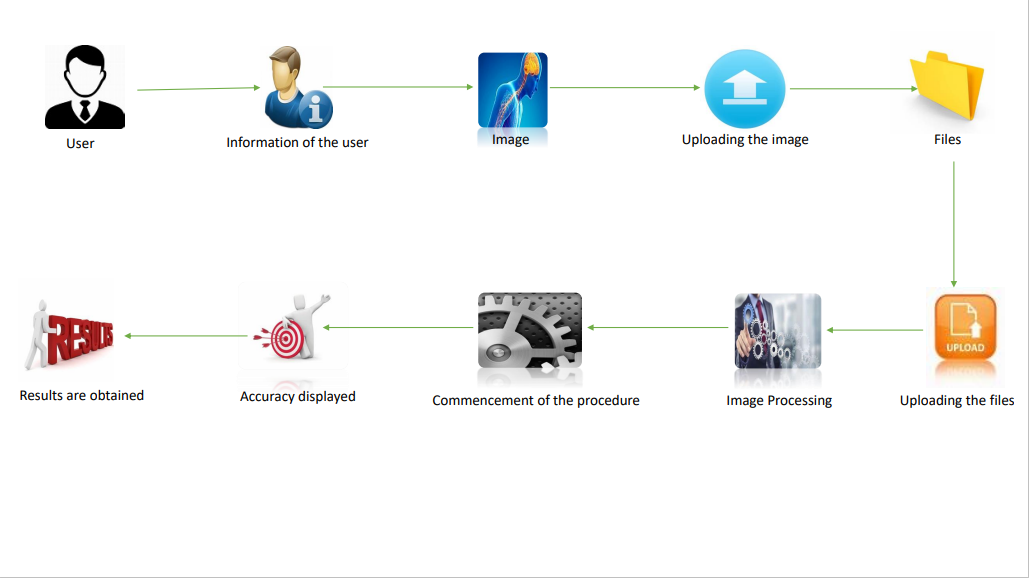
• DFD is the abbreviation for Data Flow Diagram

• The flow of data of a system or a process is represented by DFD.

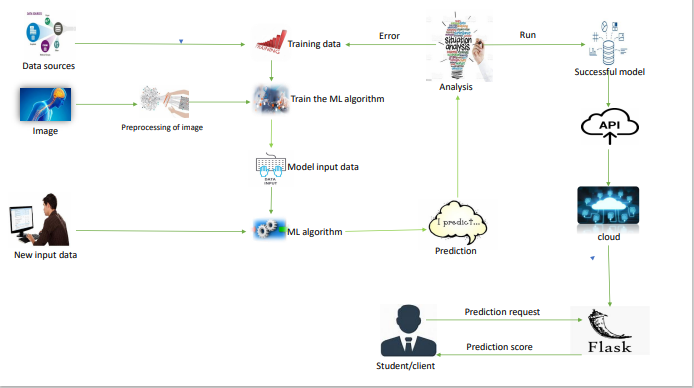
• It also gives insight into the inputs and outputs of each entity and the process itself

• DFD does not have control flow and no loops or decision rules are present

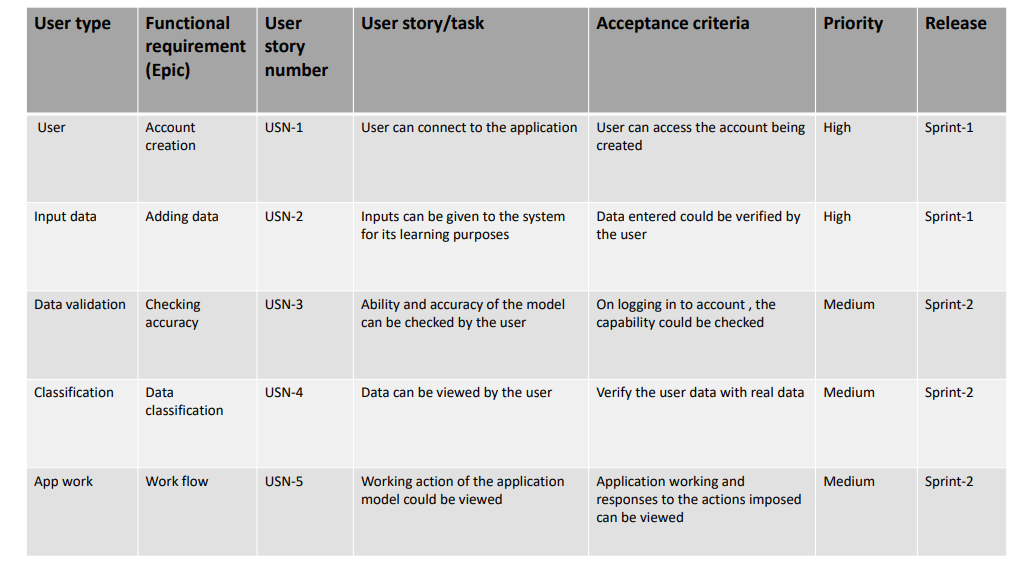
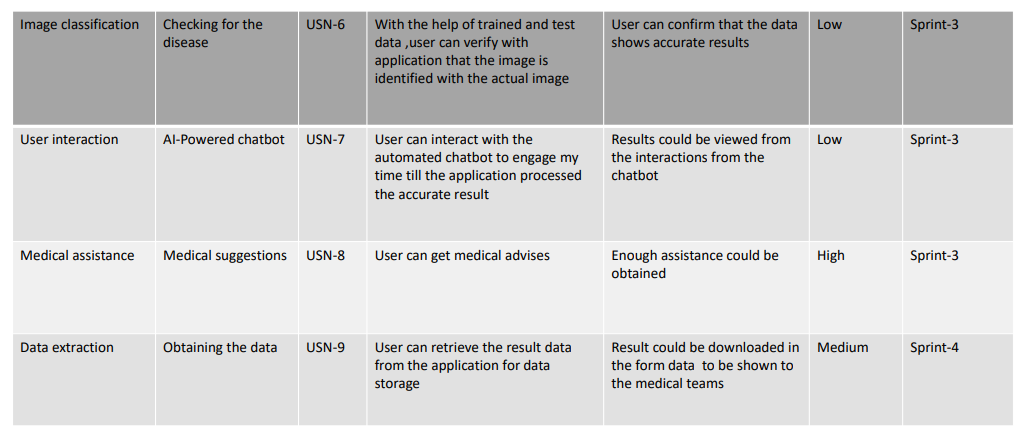
• Specific operations depending on the type of data can be explained by a flowchart



**5.2 Solution& Technical Architecture:**



**5.3 User Stories:**



**6.PROJECT PLANNING & SCHEDULING:**

**6.1 Sprint Planning & Estimation:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Tas** | **Story Points** | **Priority** | **Team Members** |
| Sprint-2 | homepage | USN-1 | Description about Parkinson's disease. | 8 | Low | Hariharan M Raja M |
| Sprint-2 |  | USN-2 | Details about the test vitals required for the testing. | 13 | Low | Raja M Hariprasath B |
| Sprint-3 | Registration | USN-3 | As a user, I can register for the application by entering my username, email, phone number, and password, and confirming my password | 5 | Medium | Hari Priya D Hariharan M |
| Sprint-3 | Login | USN-4 | As a user, I can log in to the web application by entering my email id & password. | 5 | Medium | Hari Priya D Raja M |
| Sprint-3 | Main Page(Test vitals) | USN-5 | As a user, I submit the required image for the prediction | 5 | Medium | Hariharan M Hariprasath B |
| Sprint-3 | Result | USN-6 | Results will be displayed | 8 | High | Hariprasath B Raja M Hari Priya D |
| Sprint-1 | Data collection | USN-7 | Collect the required data for the detection of Parkinson's disease | 8 | High | Hariharan M Hari Priya D Raja M |
| Sprint-1 | Data preprocessing | USN-8 | Clean and analyze the data to avoid noise and duplications | 8 | High | Hari Priya D Hariharan M Hariprasath B |
| Sprint-1 | Model Building | USN-9 | Build the model using a Random forest classifier to classify the images. | 5 | High | Raja M Hariprasath B Hariharan M |
| Sprint-4 | Deploy the model | USN-10 | Deployment of ML model using IBM Watson Studio, object storage. | 13 | High | Hariharan M Hari Priya D Hariprasath B Raja M |
| Sprint-4 | Integrate the web app with the IBM model | USN-11 | Use flask for the integration purpose | 8 | Medium | Hari Priya D Hariprasath B |

**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 | 21 | 6 Days | 04 Nov 2022 | 29 Oct 2022 | 21 | 29 Oct 2022 |
| Sprint-2 | 21 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 21 | 02 Nov 2022 |
| Sprint-3 | 21 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 21 | 12 Nov 2022 |
| Sprint-4 | 21 | 6 Days | 18 Nov 2022 | 19 Nov 2022 | 21 | 18 Nov 2022 |

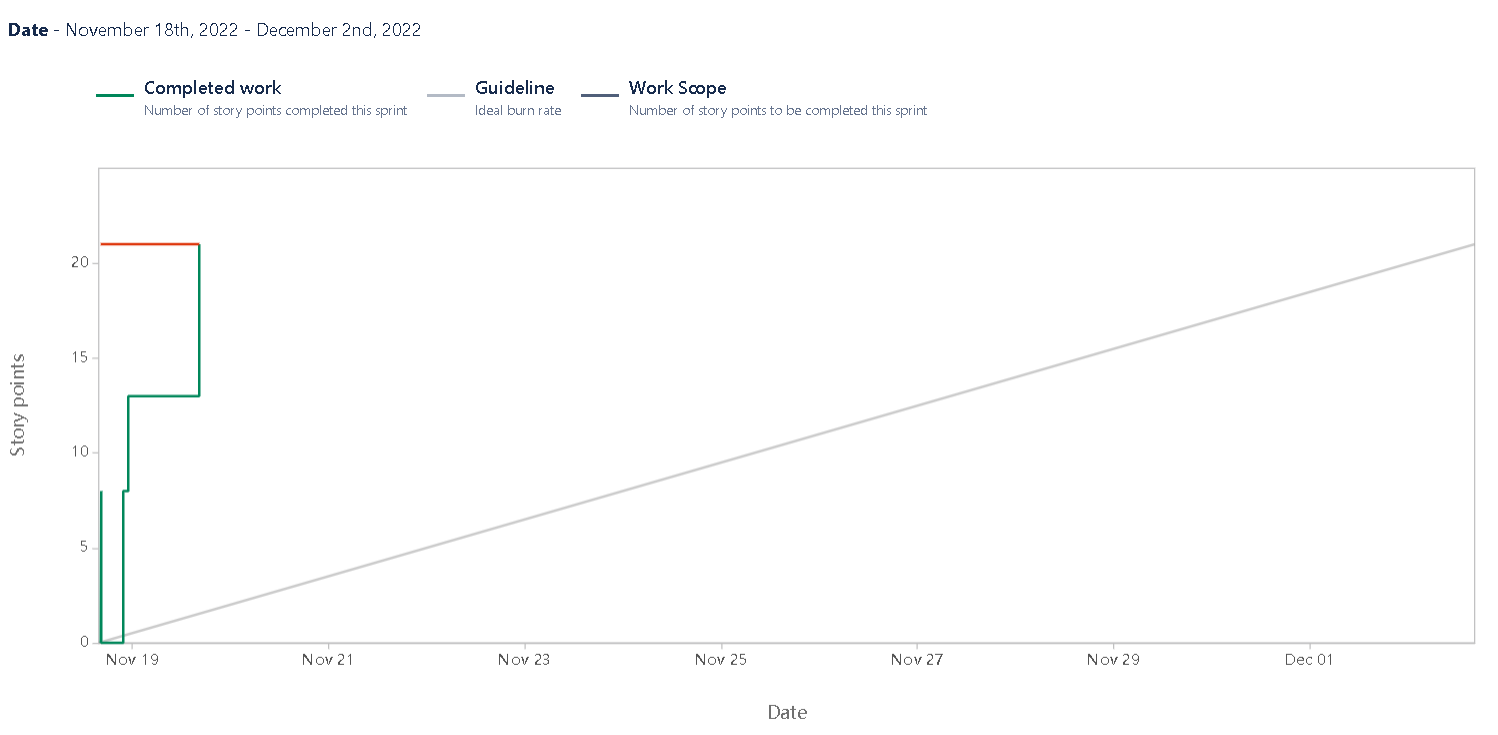
**Velocity:** we have a 6-day sprint duration, and the velocity of the team is 20 (points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit (story points per day).

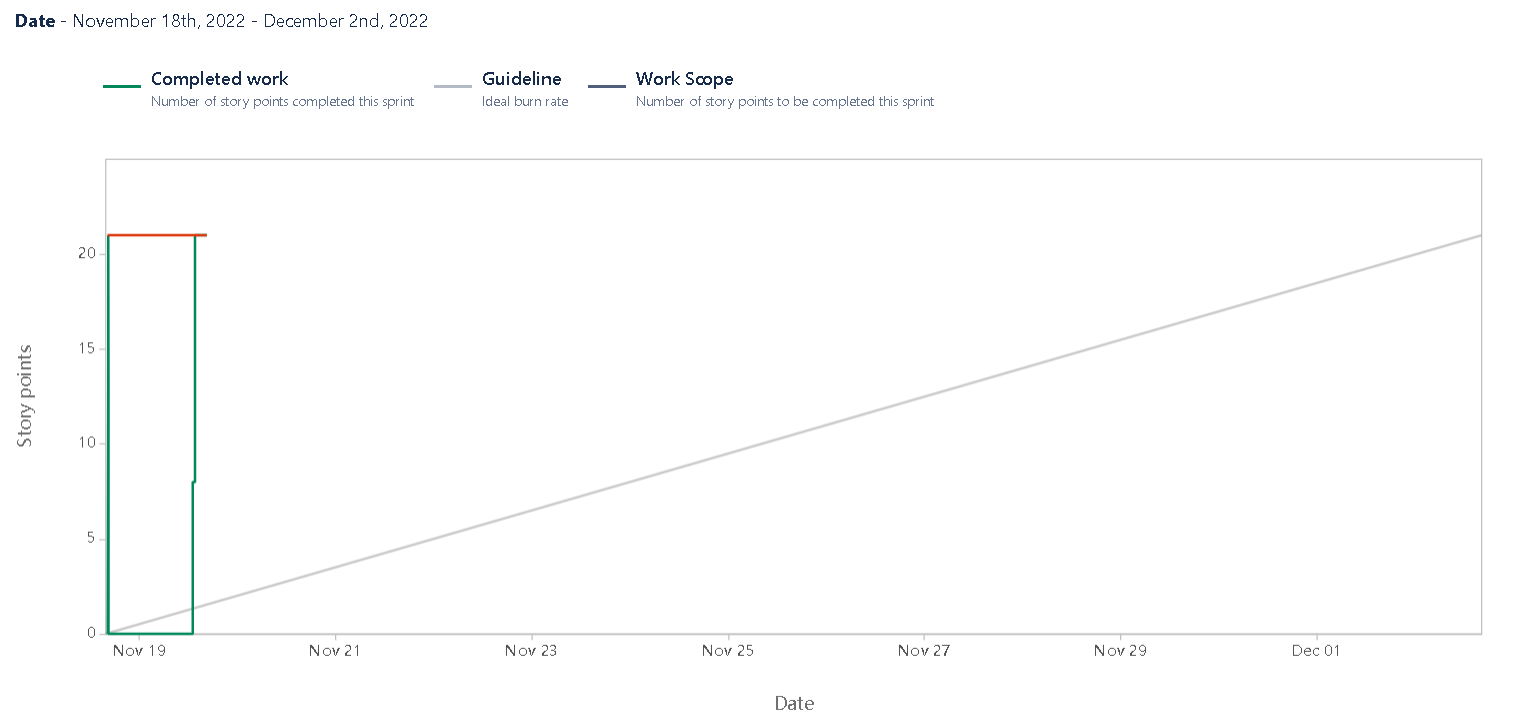
**AV = sprint duration = 21/6 = 3.5 velocity.**

**6.3 Reports from JIRA :**

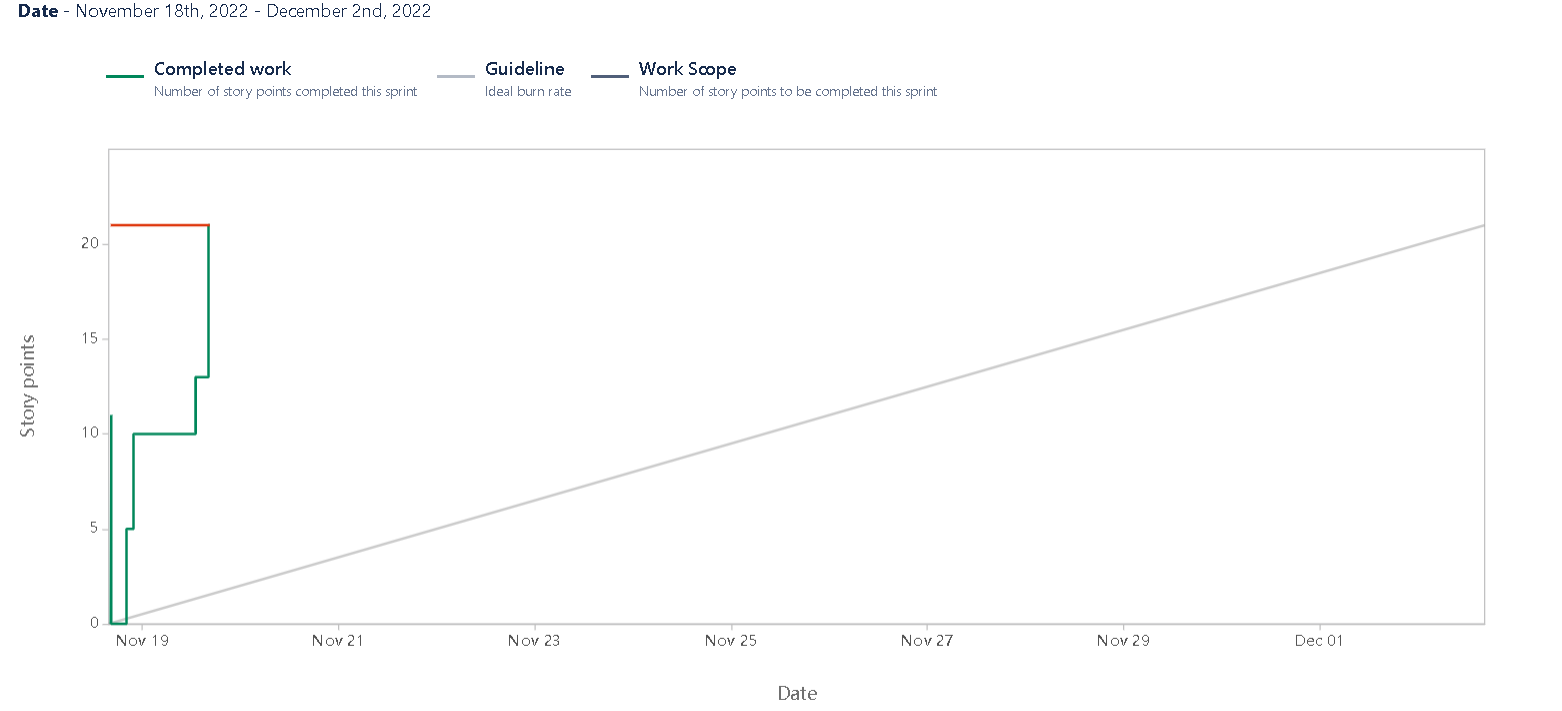
**Burnup Report**

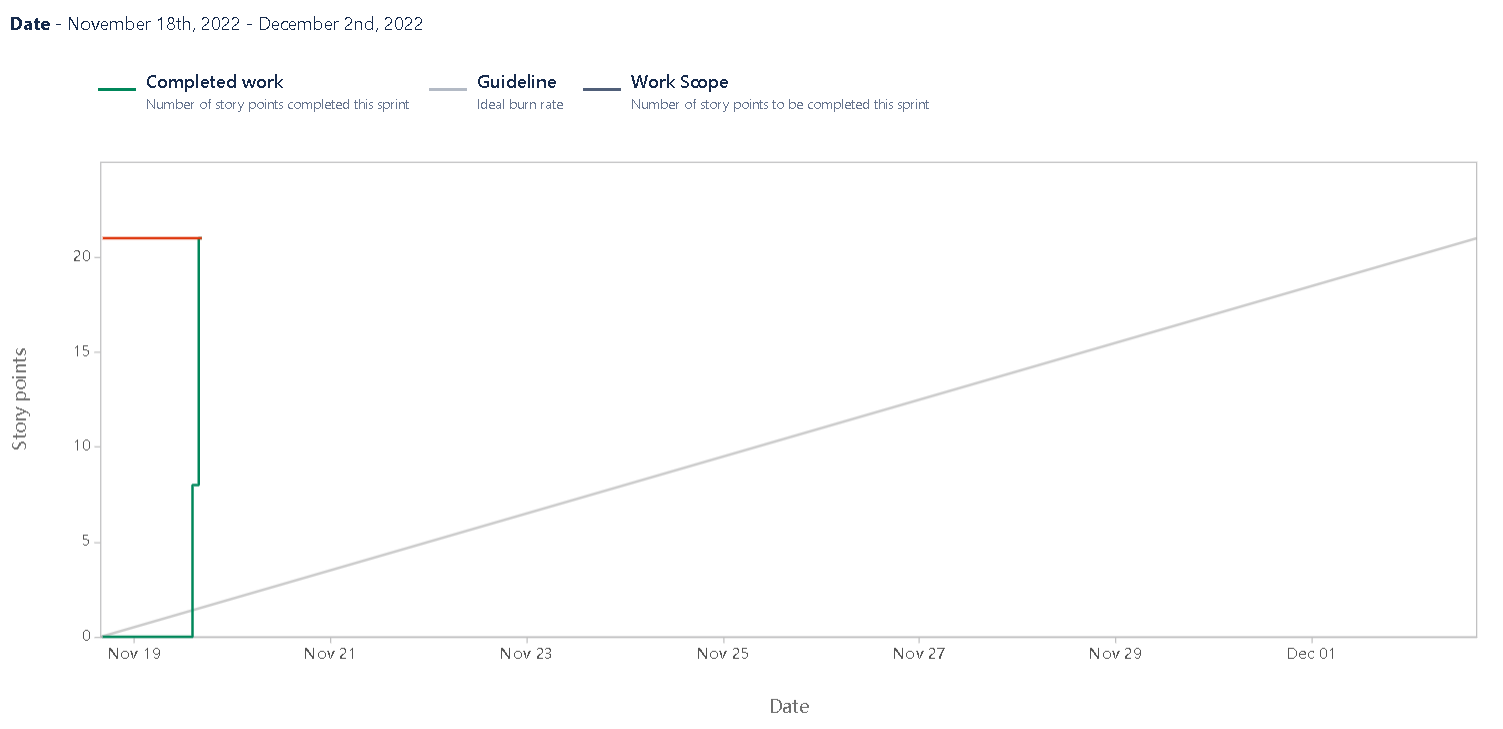
**Sprint 1 :**



**Sprint 2 :**

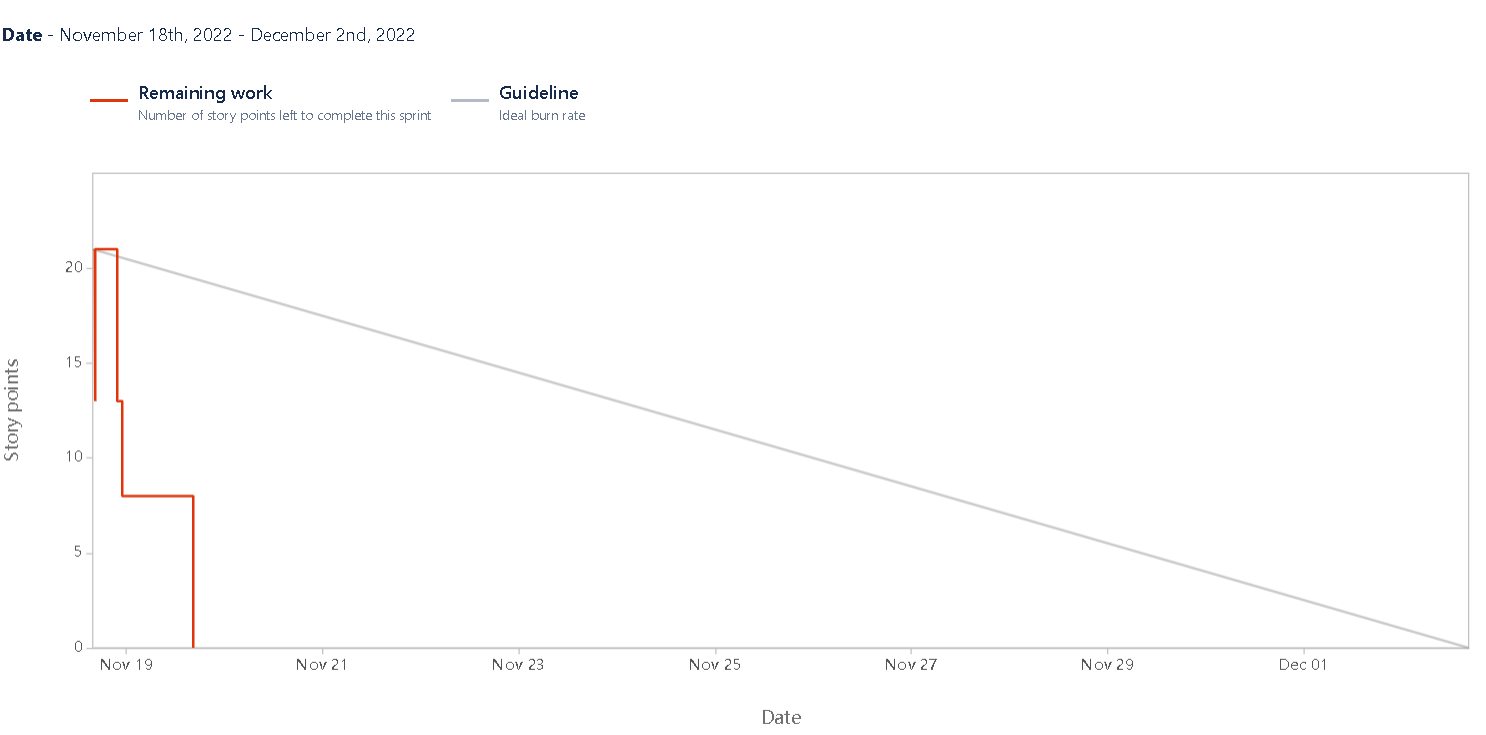
**Sprint 3 :**

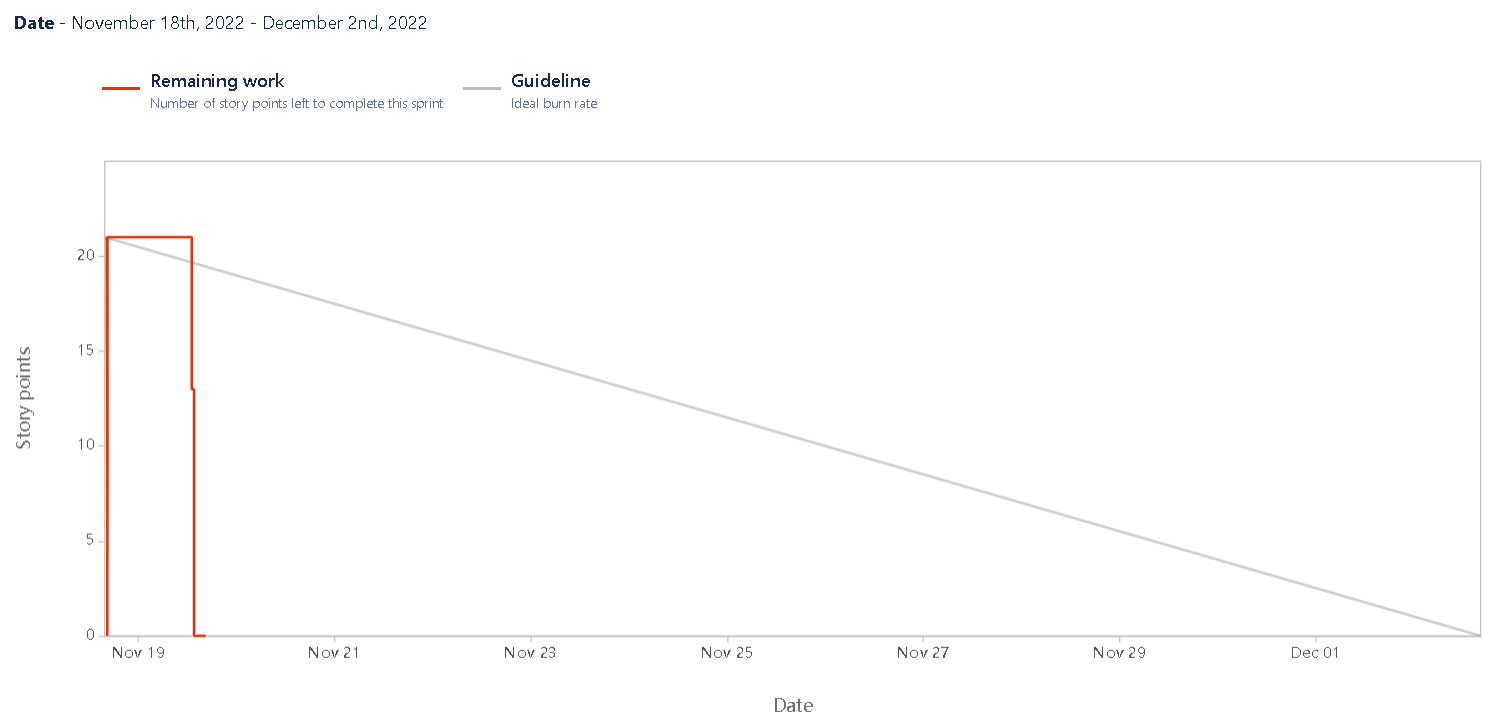


**Sprint 4 :**

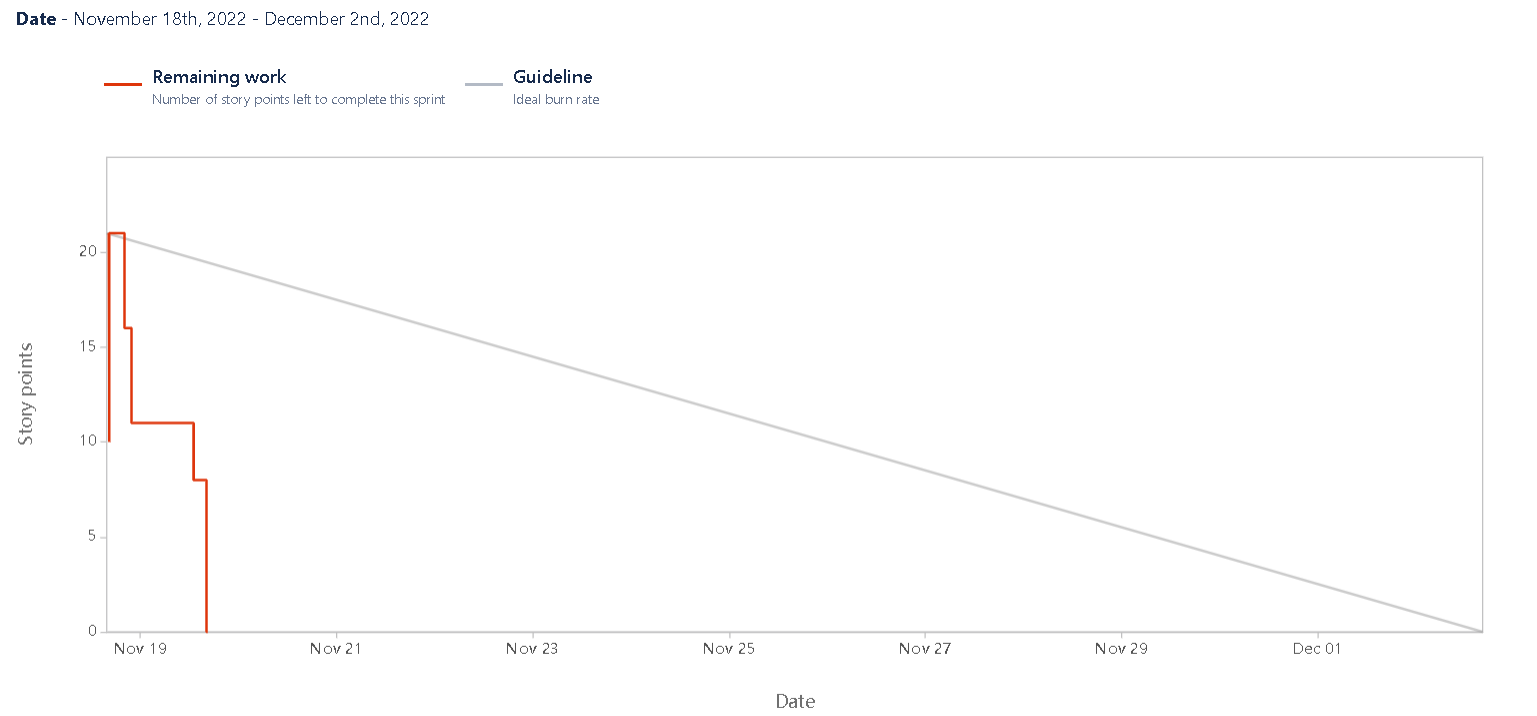
**Sprint burndown chart**

**Sprint 1 :**



**Sprint 2 :**

**Sprint 3 :**

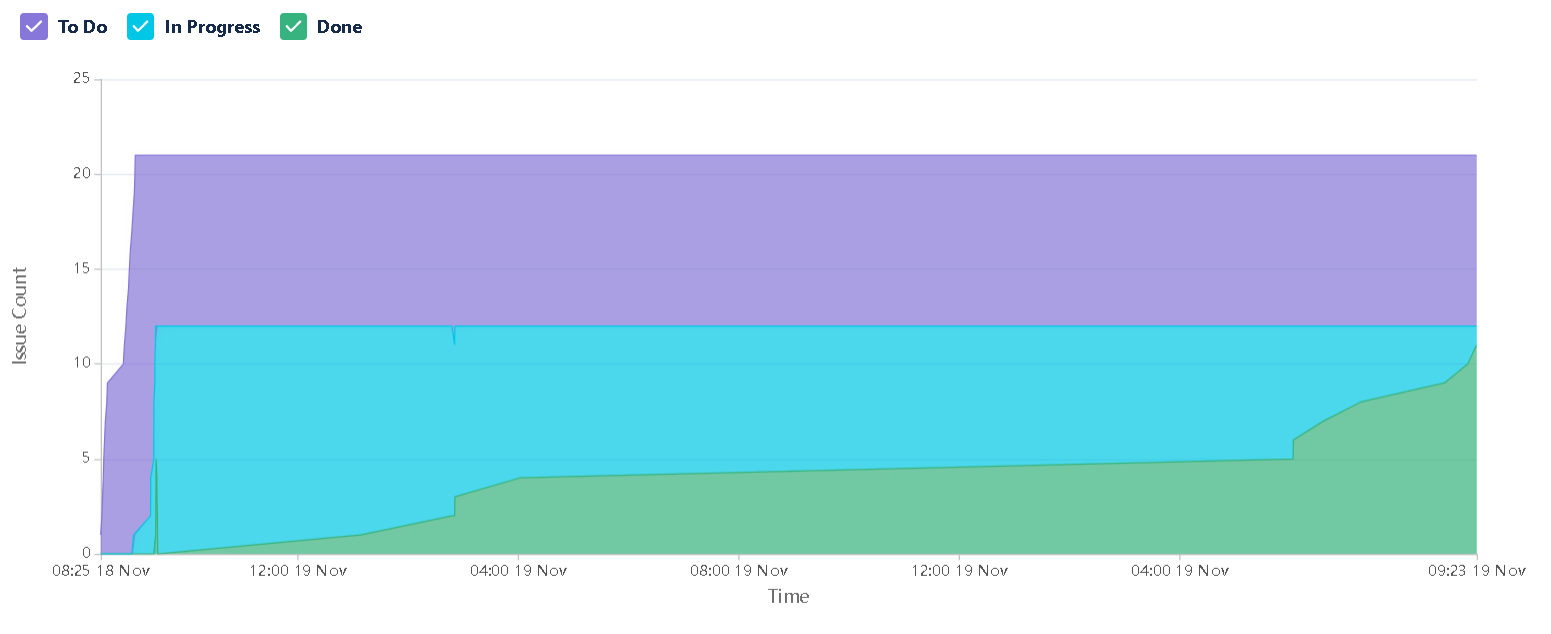


**Sprint 4:**

**Velocity report**

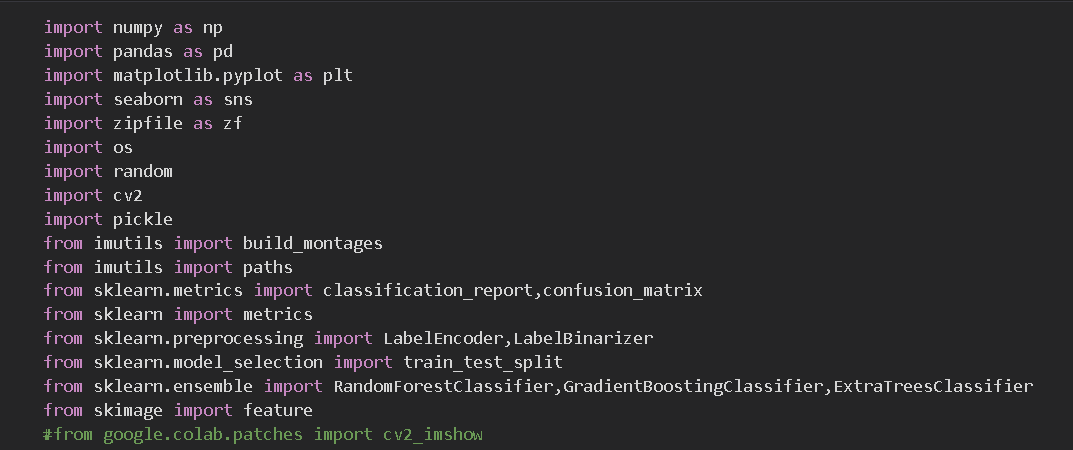


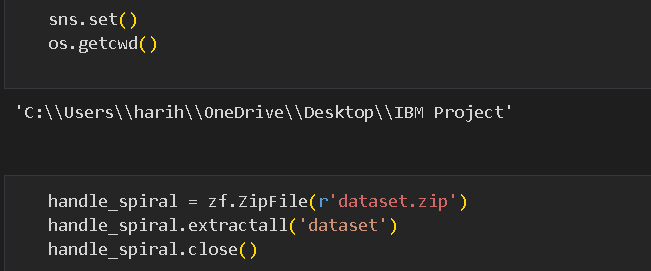
**Cumulative flow diagram**



**7.CODINGS &SOLUTIONING (Explain the features added in the project along with code)**

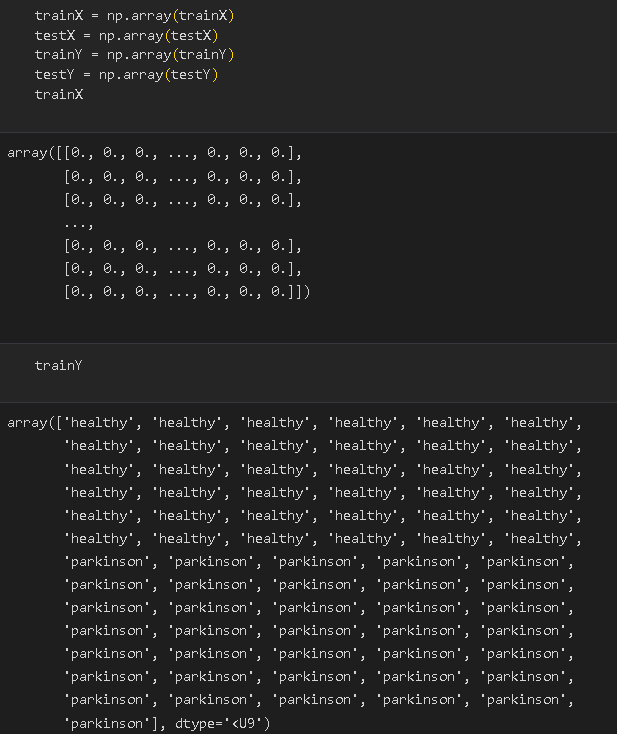
**7.1 Features 1**

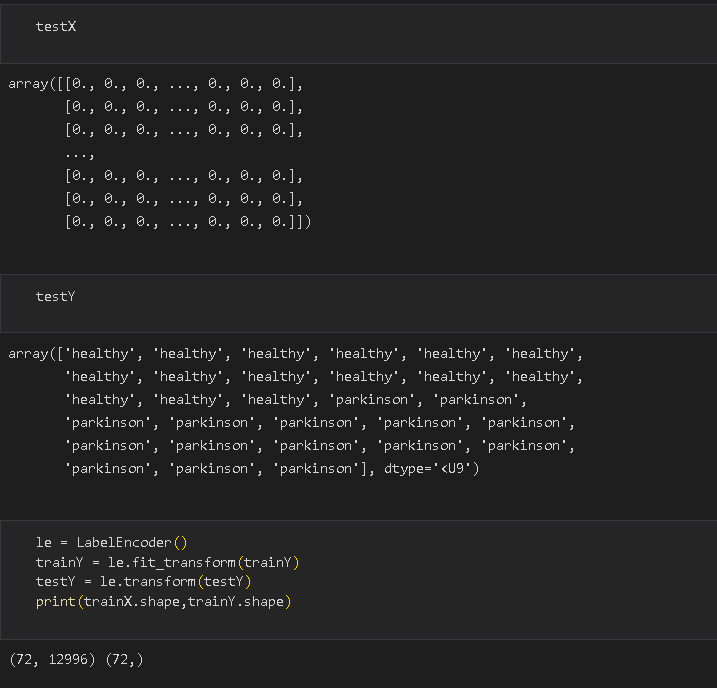
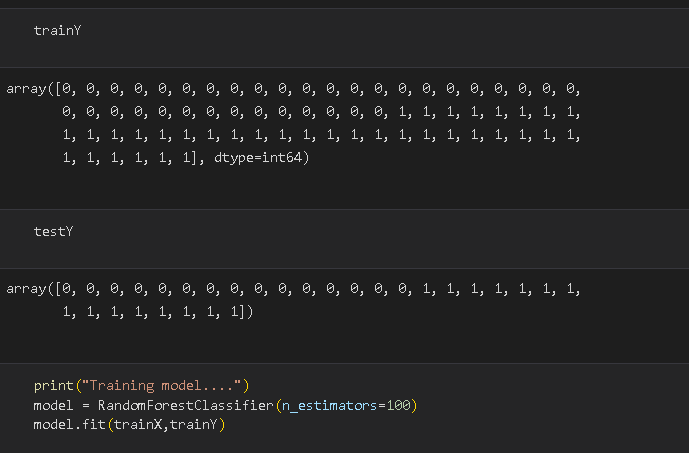


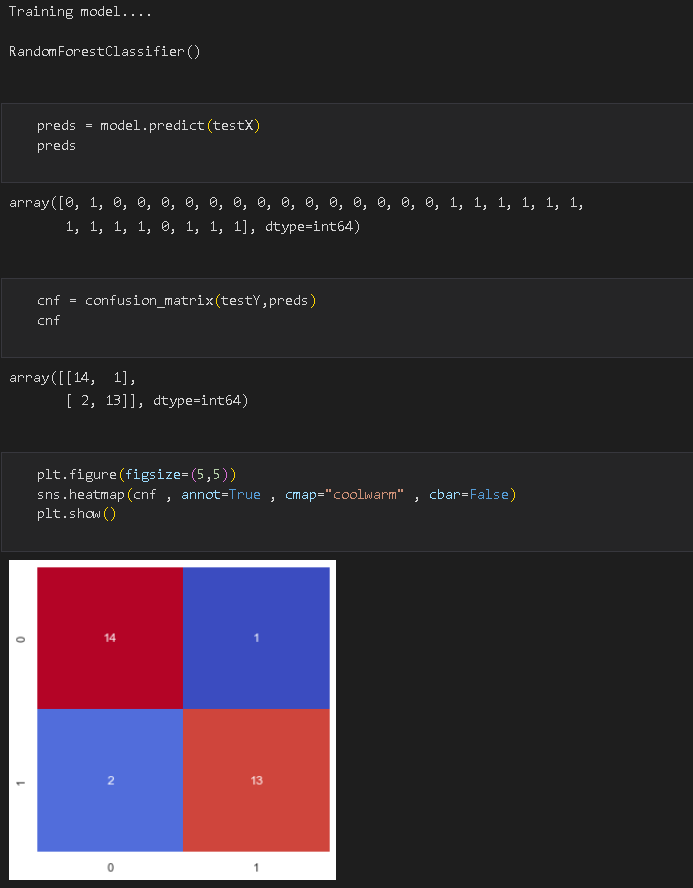


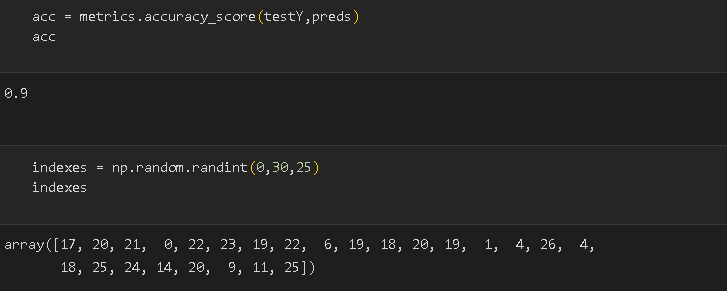


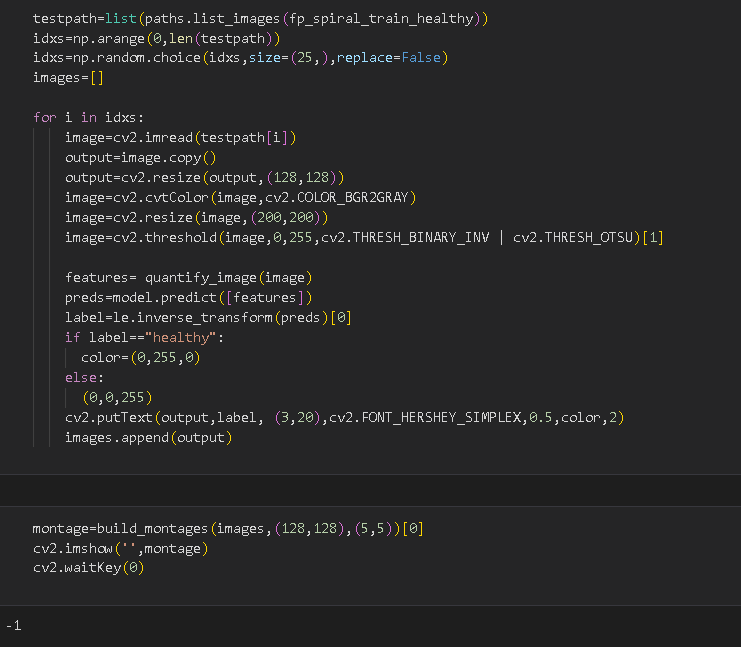


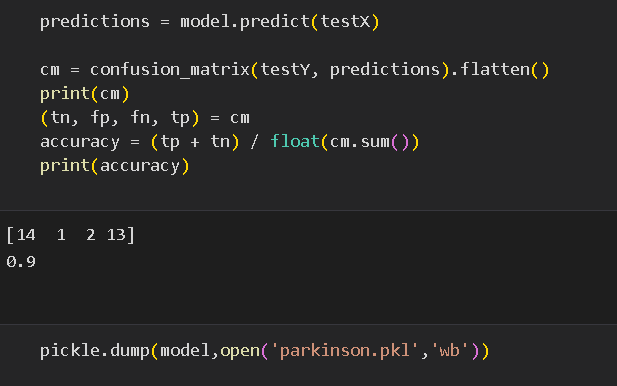




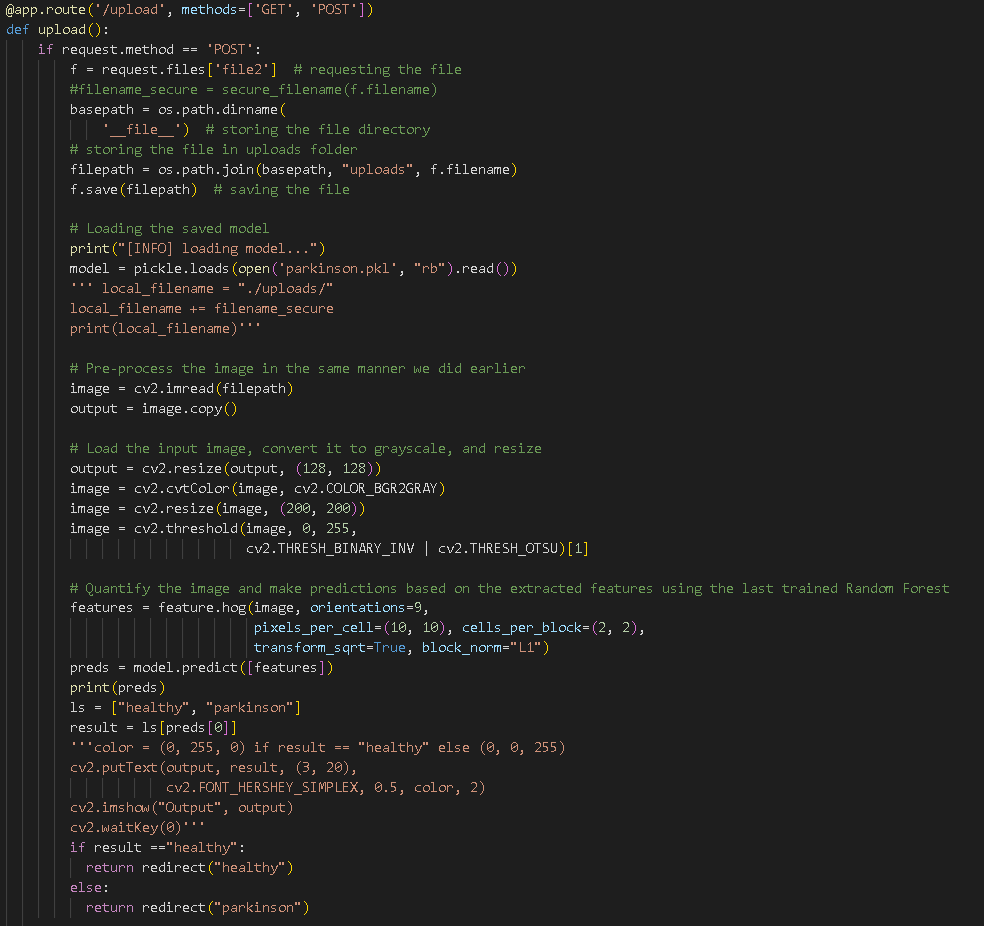


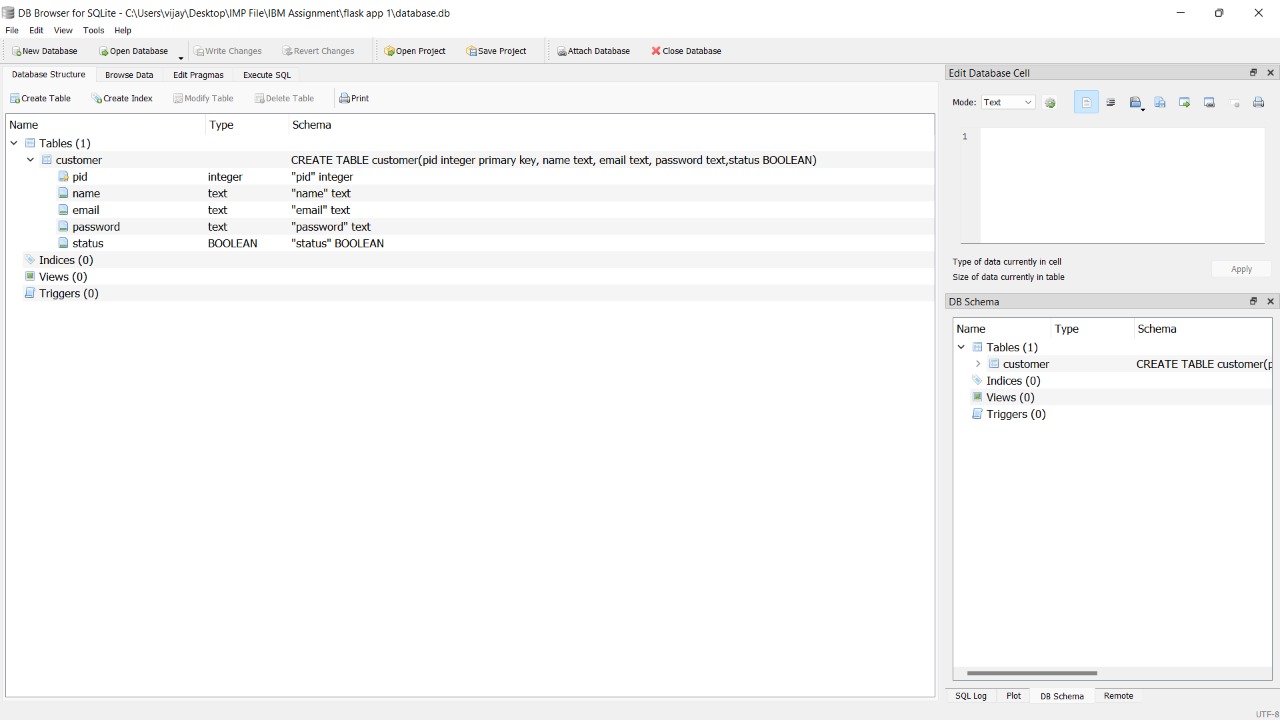


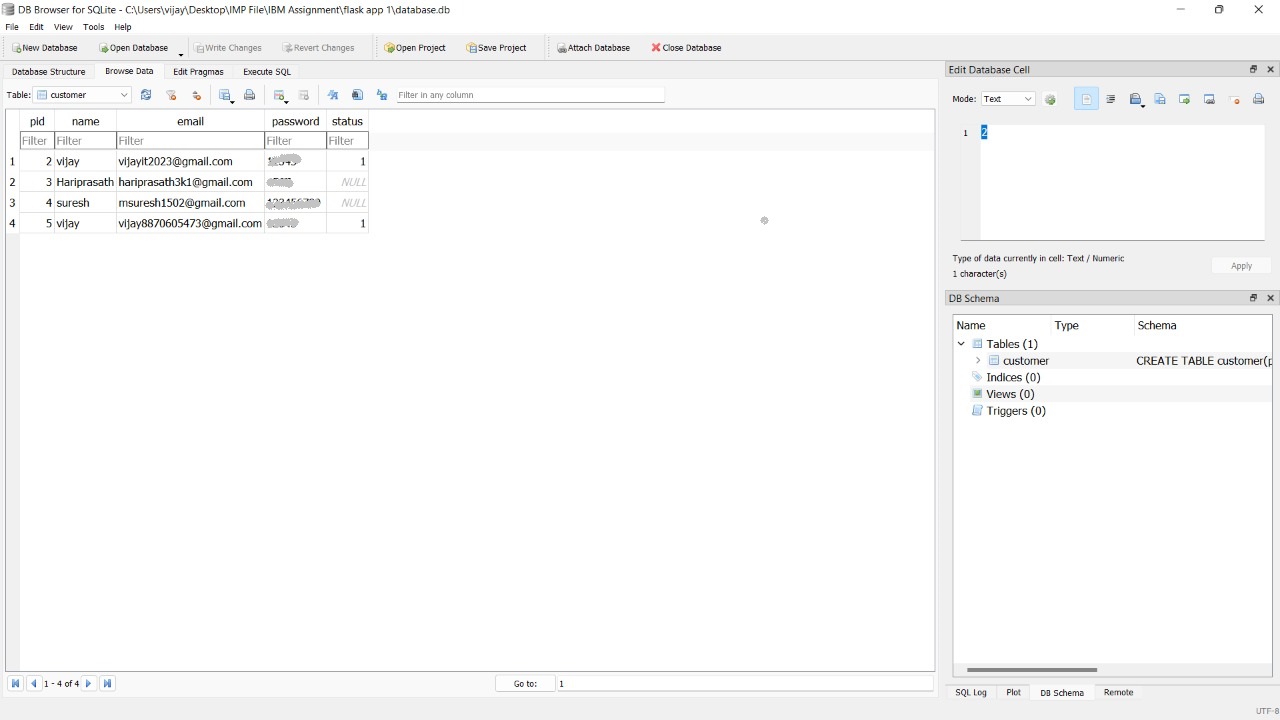




**7.2 Features 2**

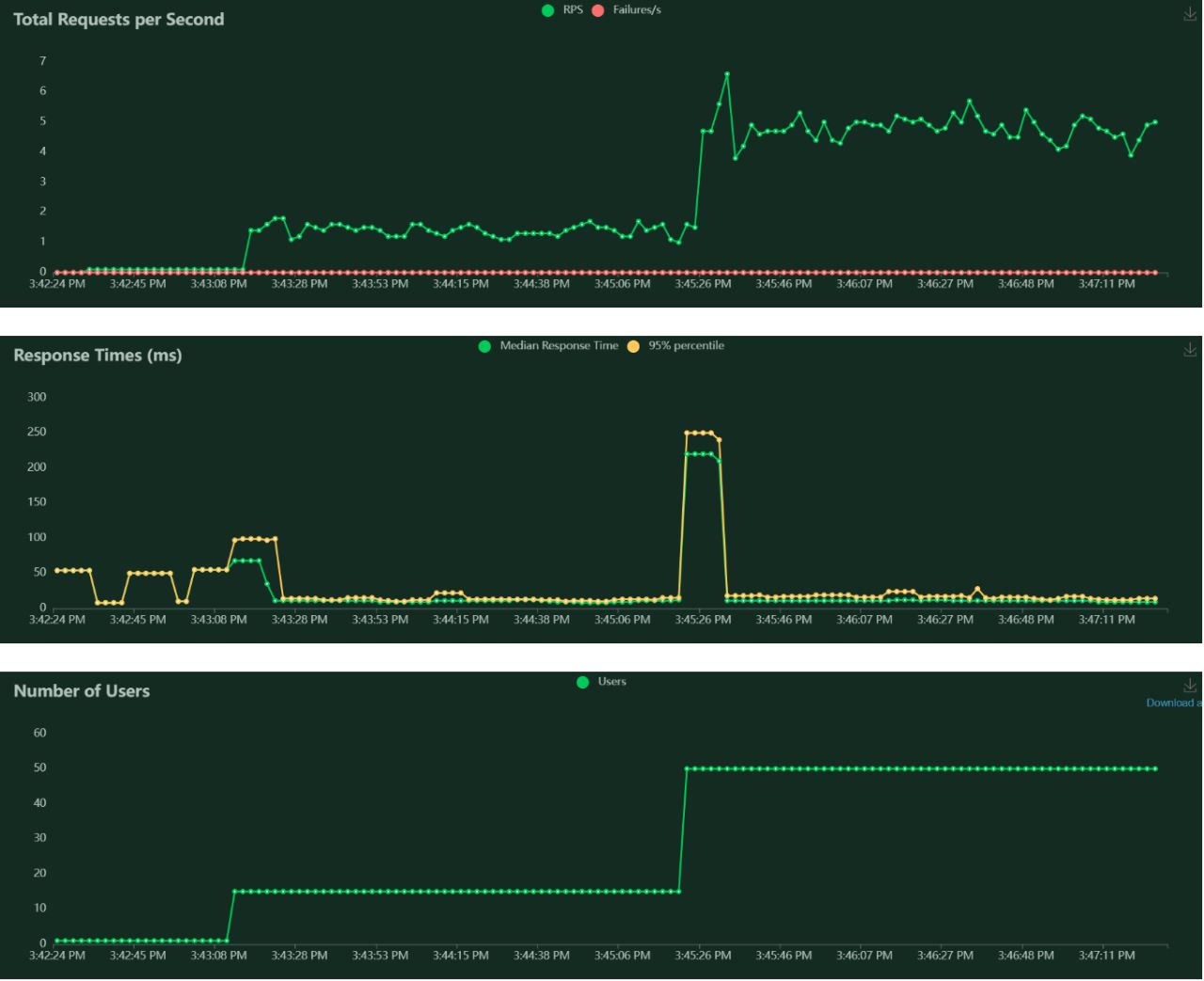


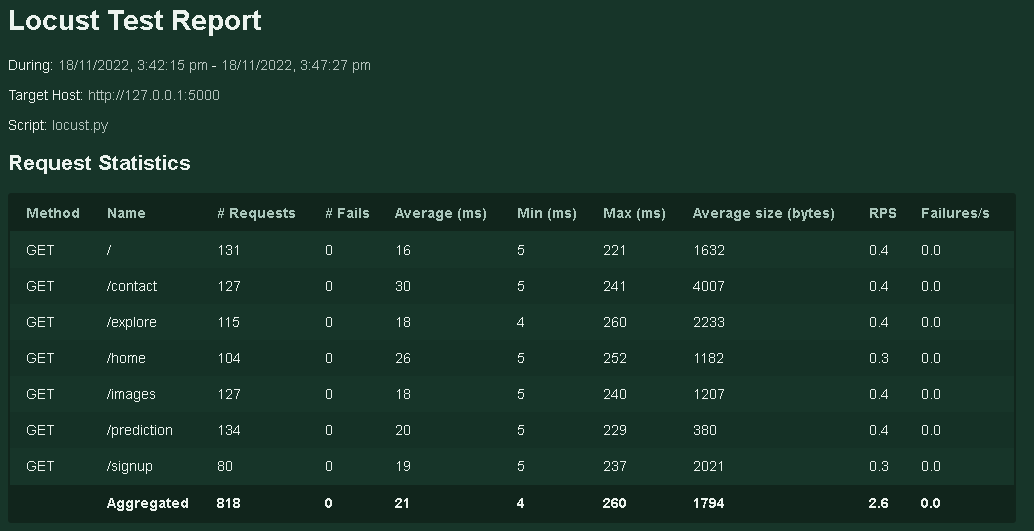
**7.3 Database schema (if applicable)**



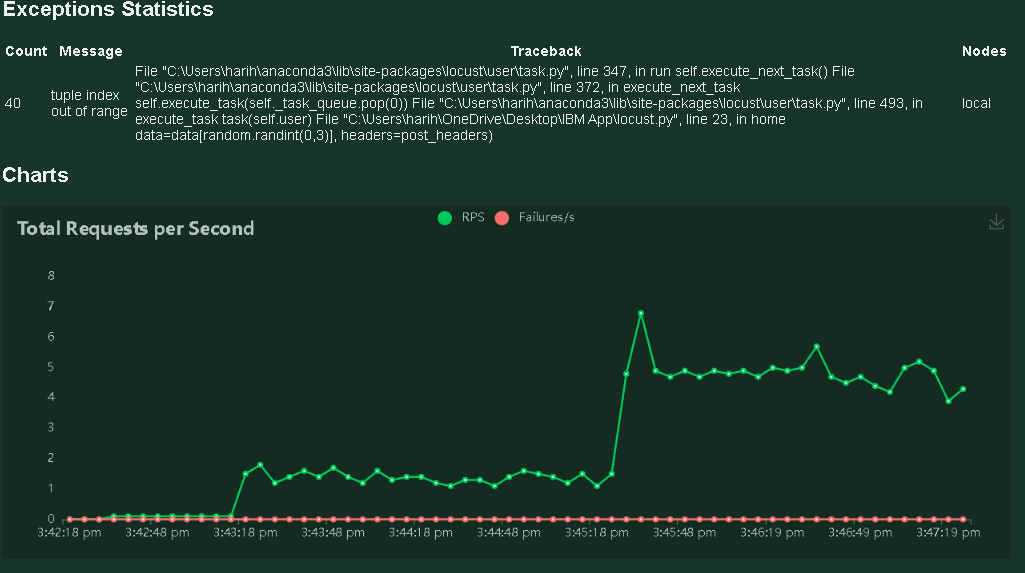
**8.TESTING**

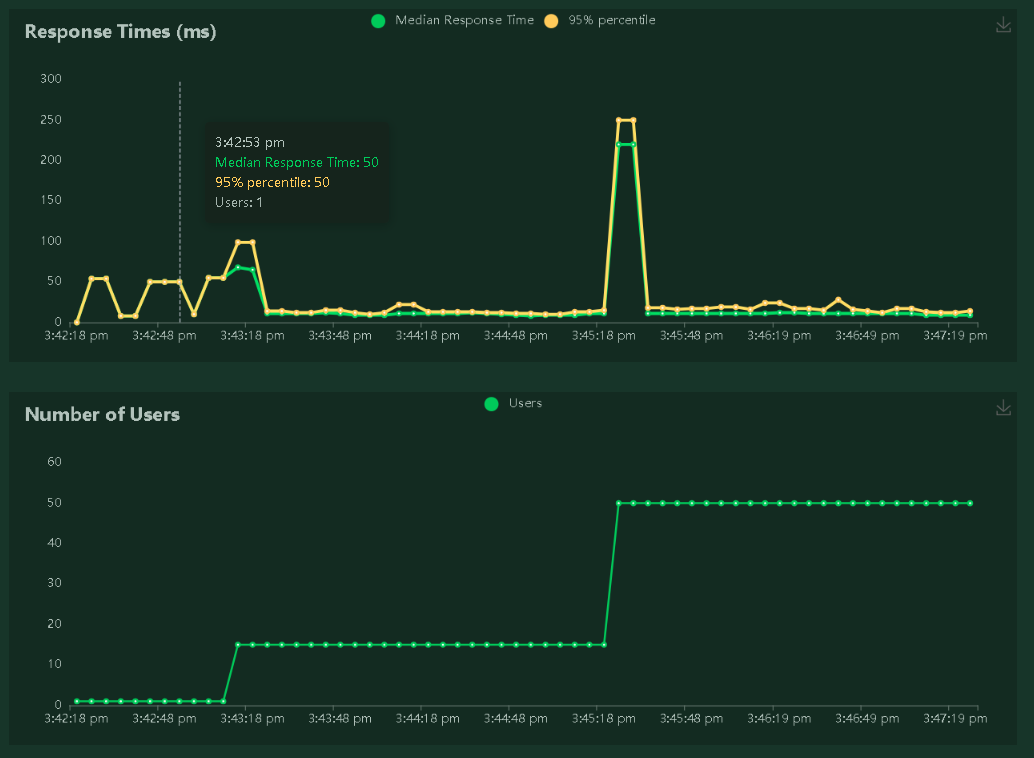
**8.1 Test cases**

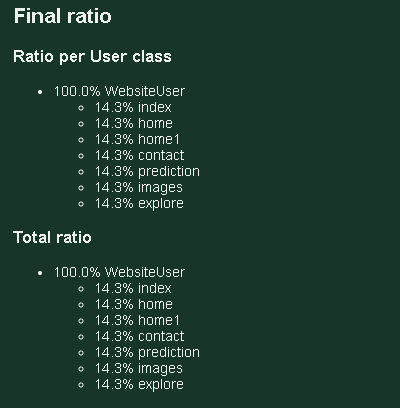


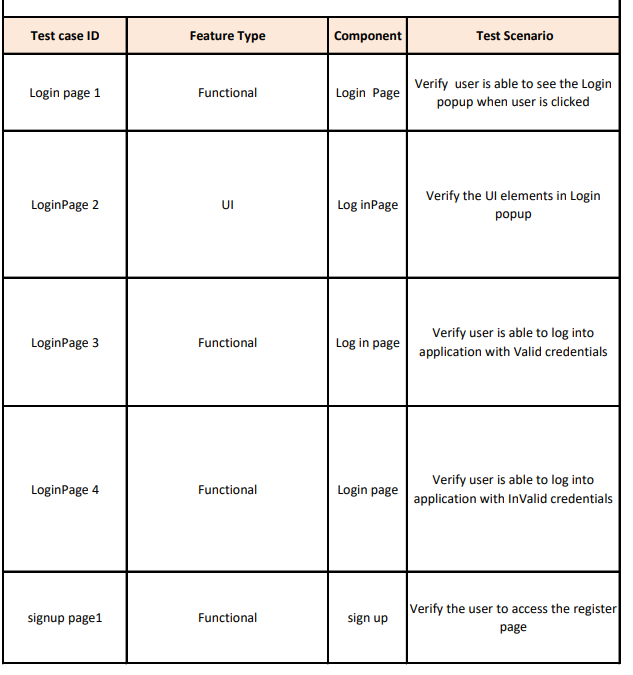


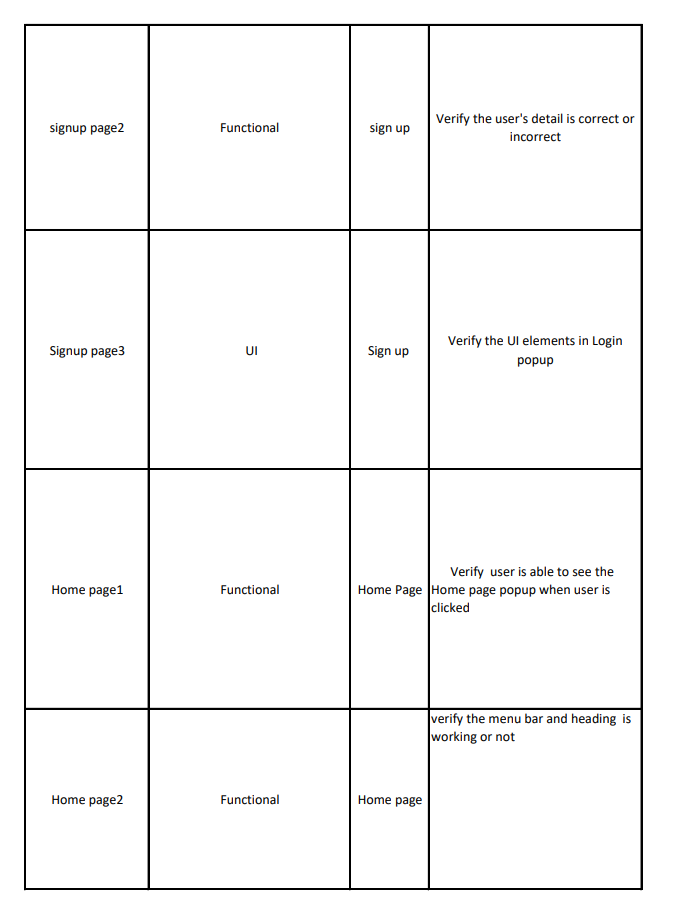


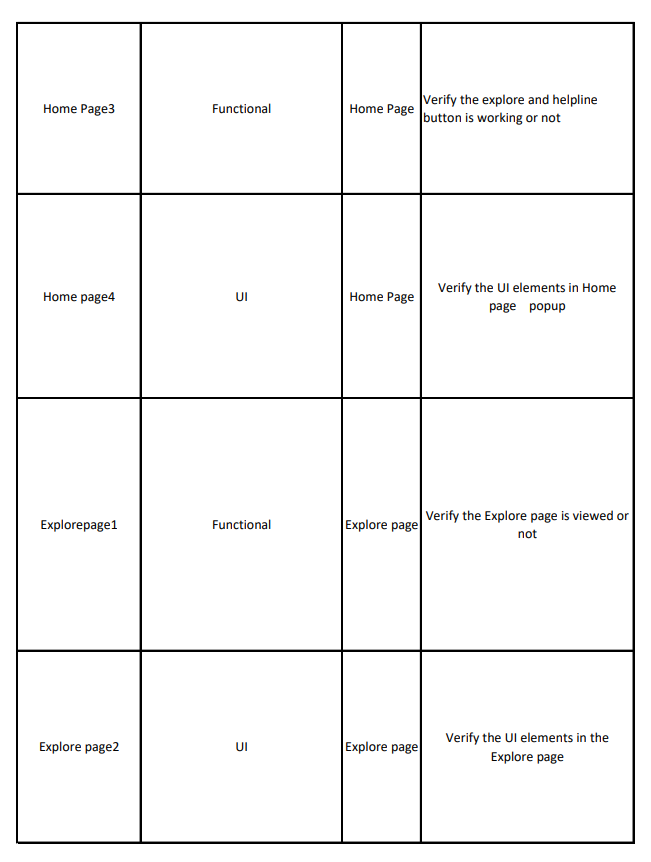


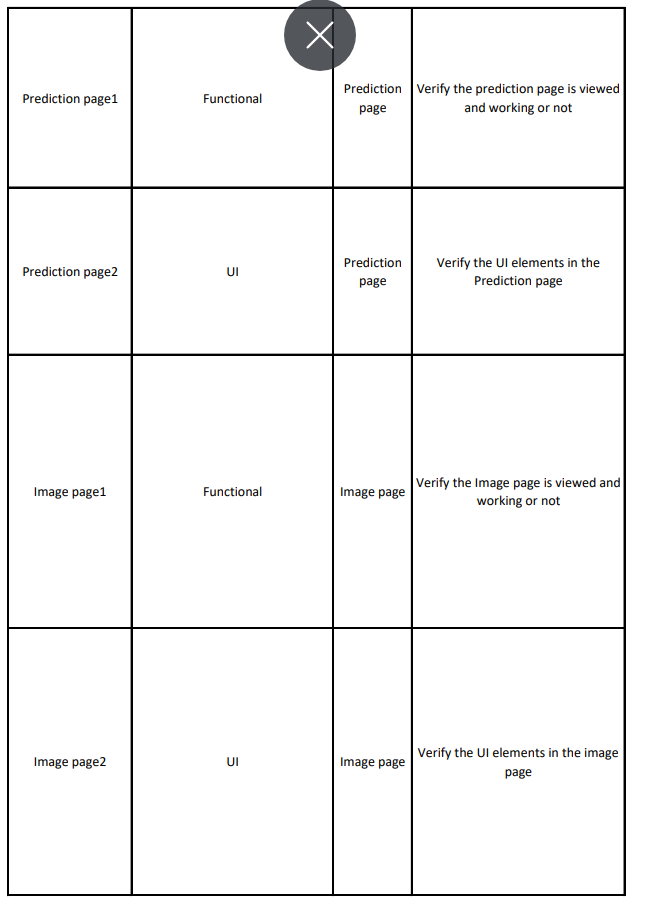


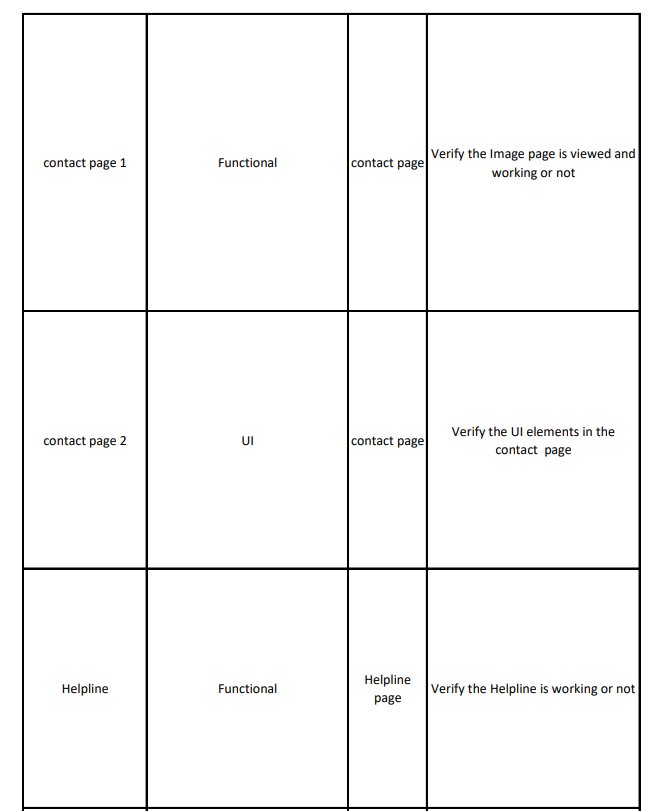


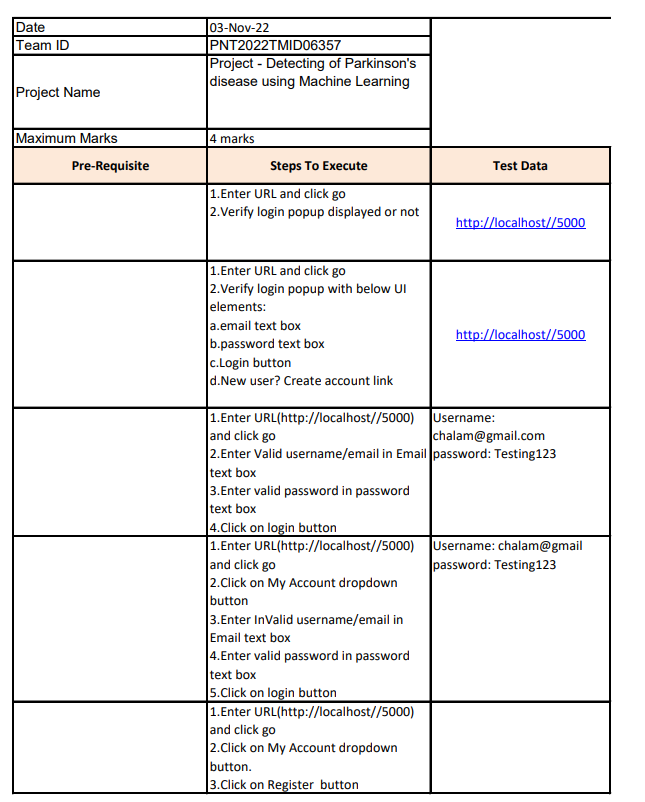
**8.2 User Acceptance Testing ;**

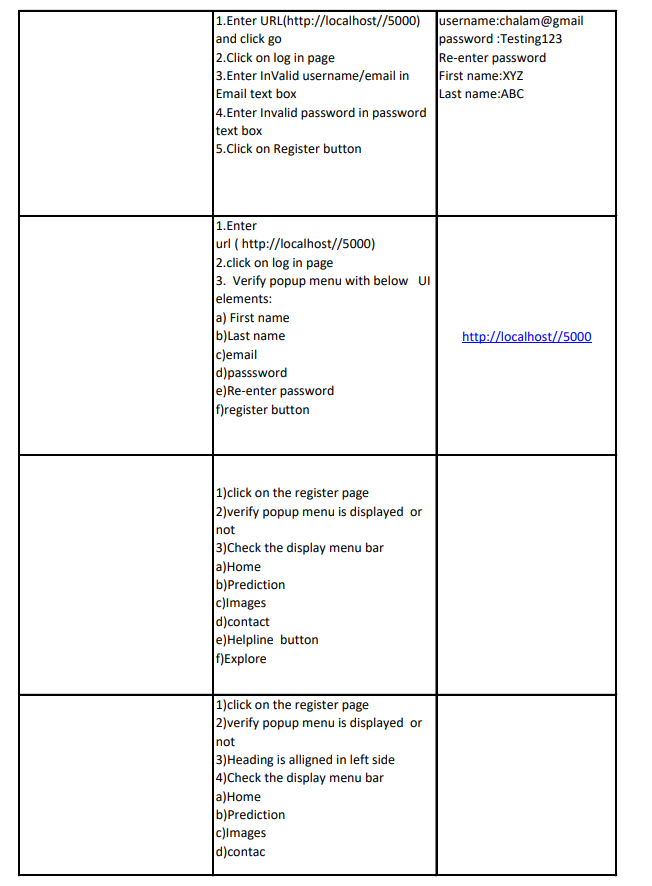


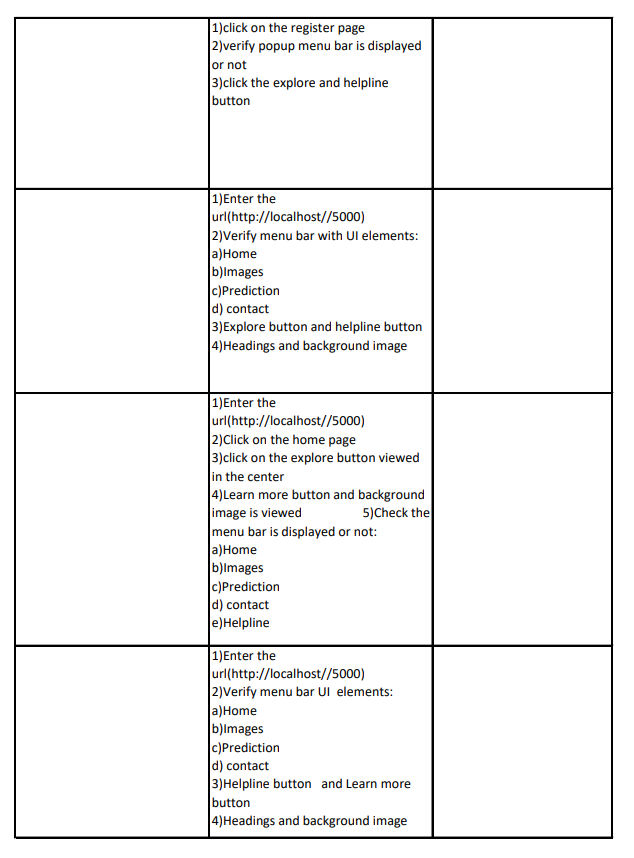


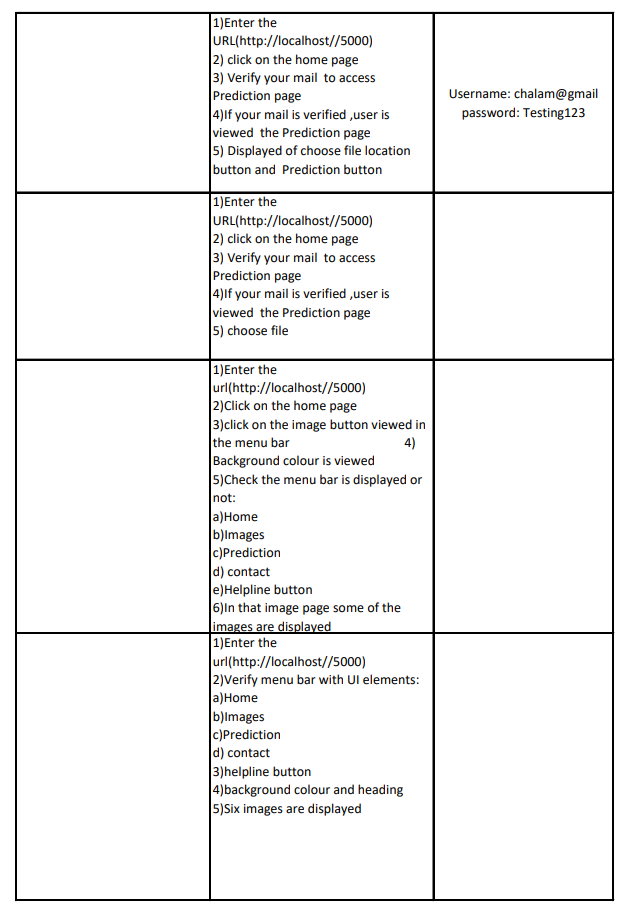


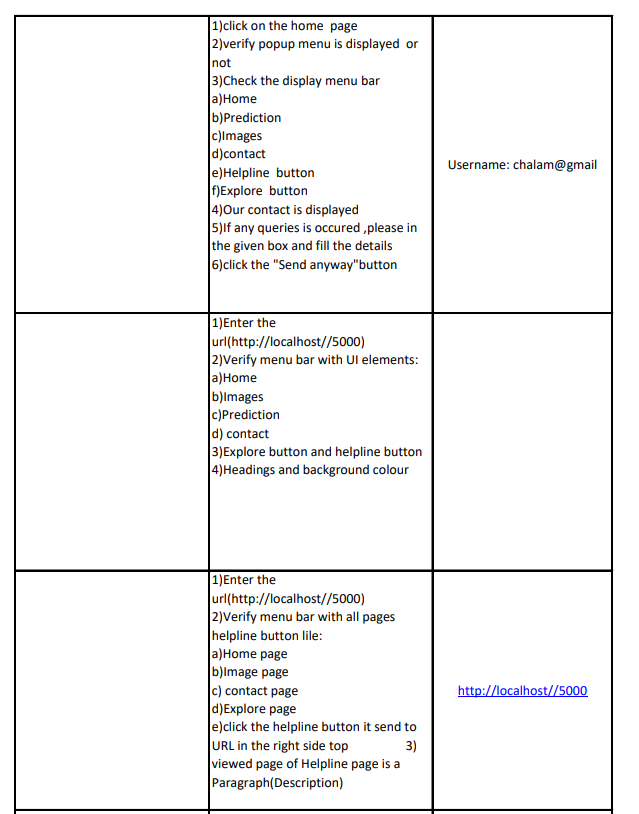


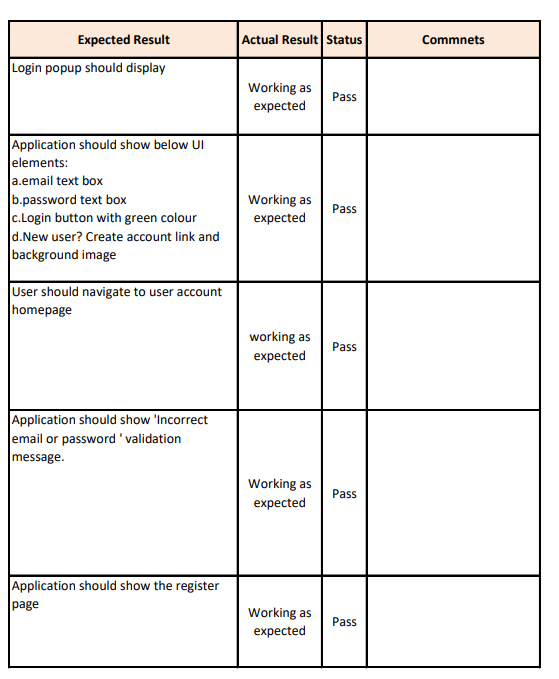


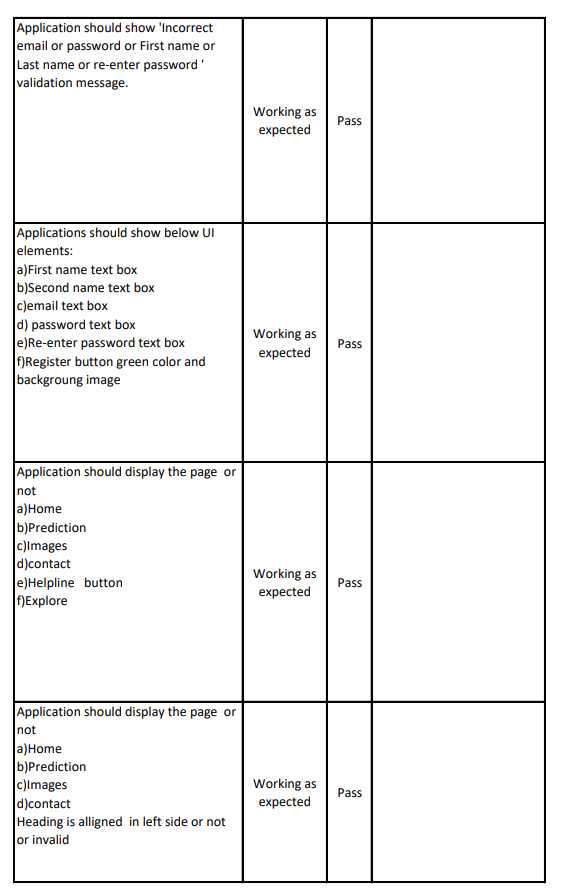


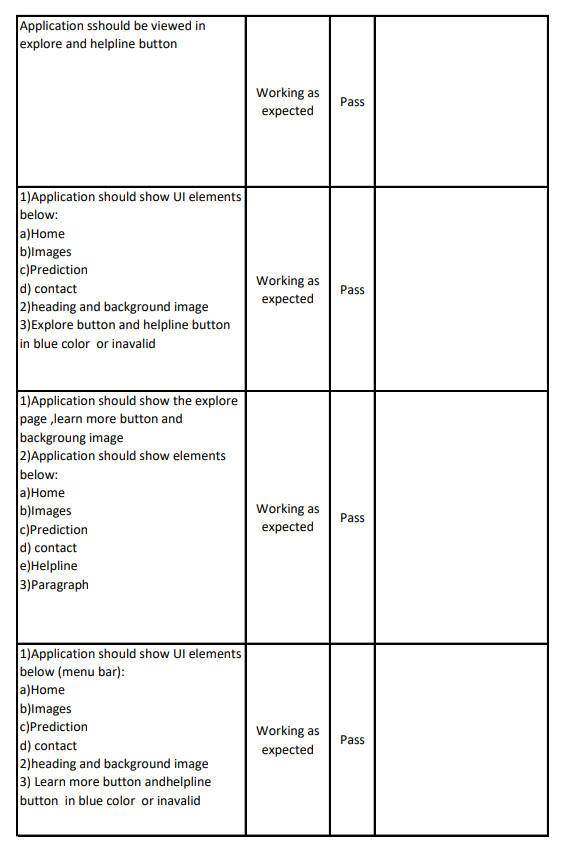


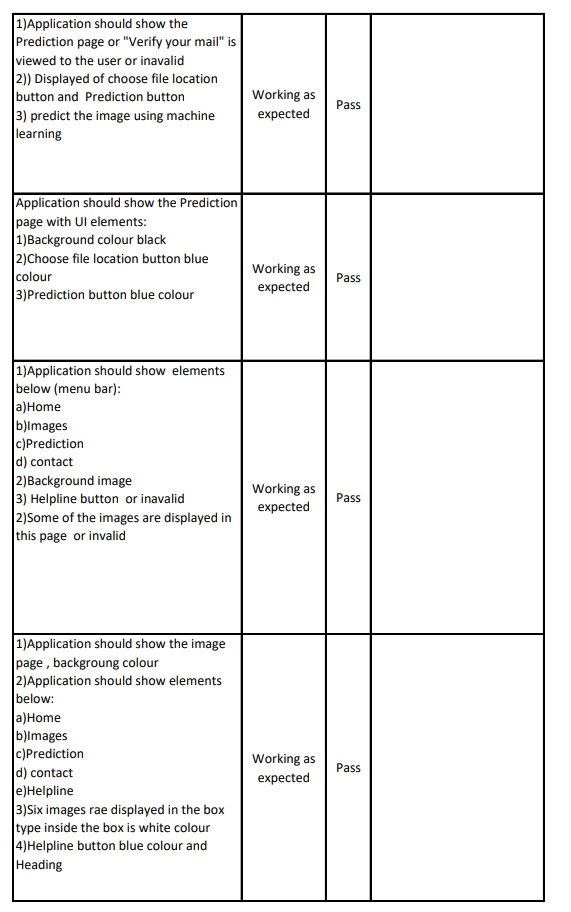


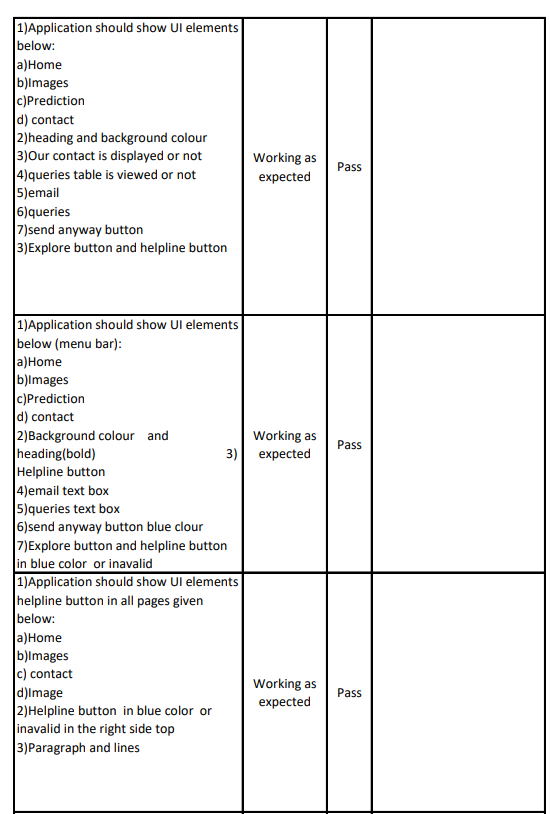


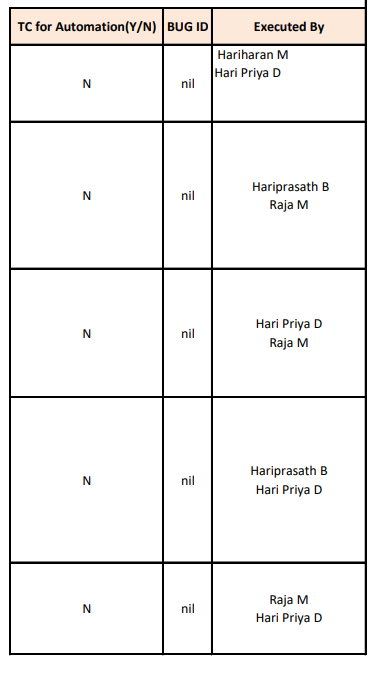


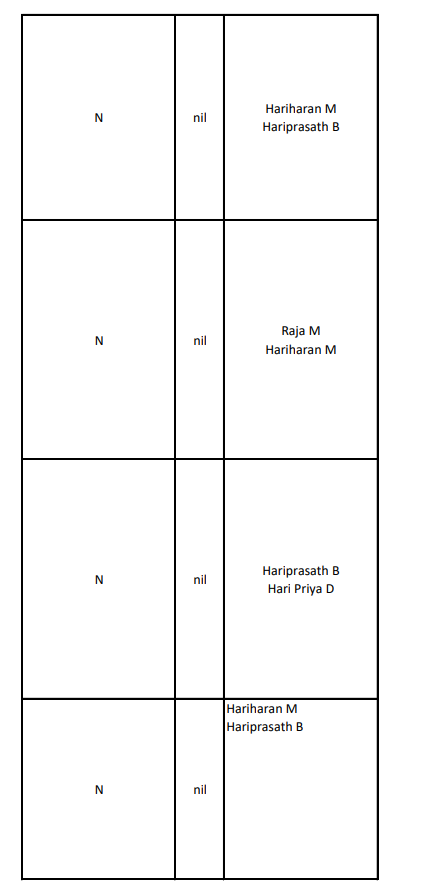


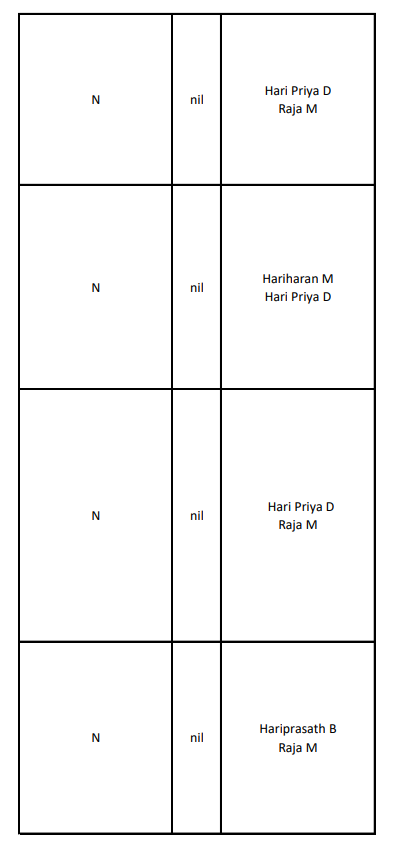


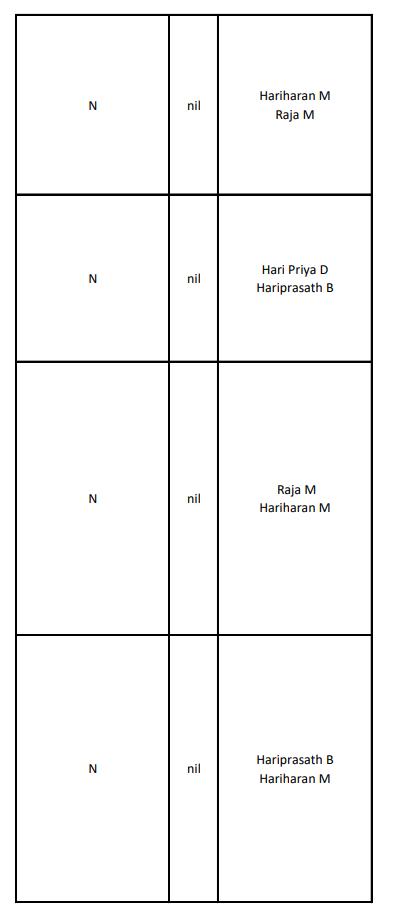
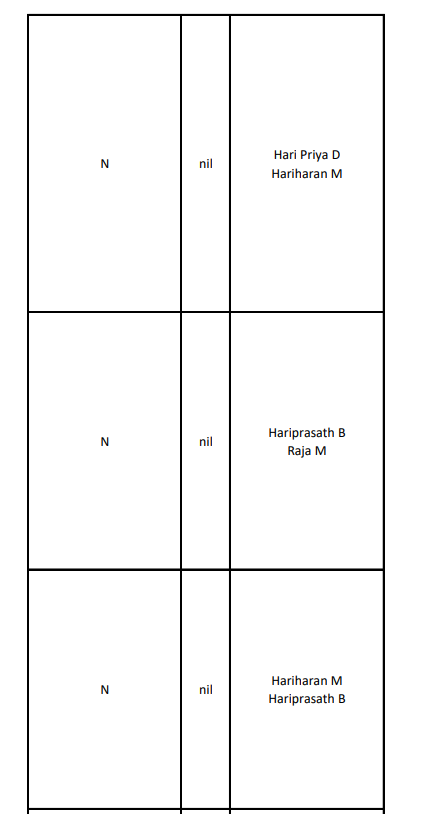




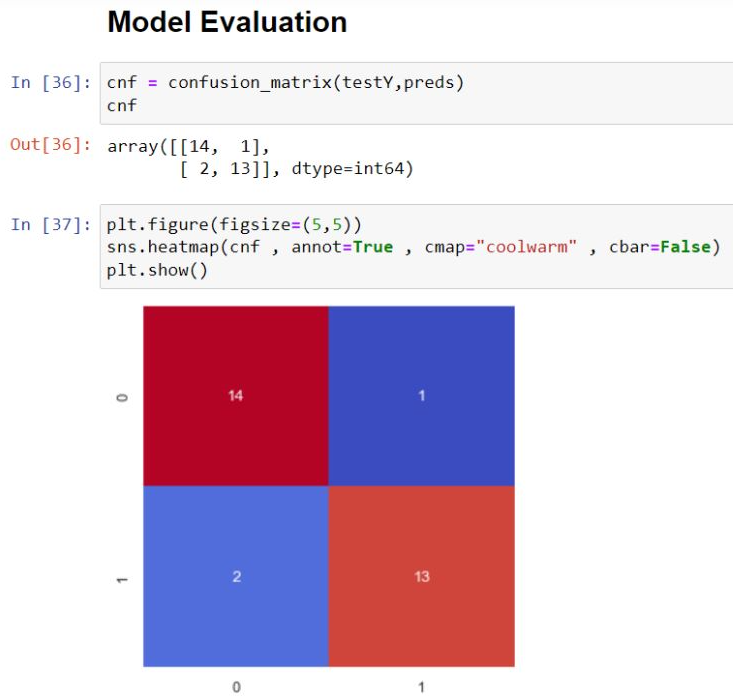


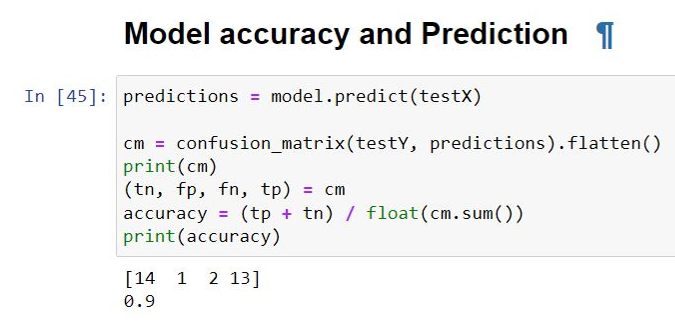






**9.RESULTS**

**9.1 Performance Metrics**



**10.ADVANTAGES &DISADVANTAGES :**

**•** Machine learning algorithm (MLA) can be used for early detection of disease to increase the chances of elderly people’s lifespan and improved lifestyle with Parkinson.

• Due to diseases diagnosis importance to mankind, several studies have been conducted on developing methods for the classification of Parkinson disease

• An efficient diagnosis system for Parkinson's disease using kernel-based extreme learning machine with subtractive clustering features weighting approach.

• Machine learning algorithms are able to support intelligent decisions by using different data types (demographic, laboratory, and image data) to make predictions of disease risk, diagnosis, prognosis, and appropriate treatments

• Early prediction of disease could be performed with the provided input

• In our research work, Parkinson’s disease is detected using image as input and we used machine learning technology for classification with higher accuracy

• The benefits of early prediction and management of PD would affect not only the individual (and their families) but also the wider society and research community.

• early detection of PD, ML models have been applied to multiple data modalities, including movement, neuroimaging, and voice and handwriting patterns

**DISADVANTAGES :**

• Predicted Parkinson’s are 31 on a heat map. Parkinson’s disease affects the CNS of the brain and has yet no treatment unless it’s detected early. Late detection leads to no treatment and loss of life.

• If input not given in the specified mentioned format results will not be displayed

• Prediction is made as yes or no ,the type will not be mentioned

• Parkinson’s disease prediction using machine learning requires massive data sets to train on and these should be inclusive/unbiased, and of good quality

• Parkinson’s disease prediction using machine learning needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you

• Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

.

**11.CONCLUSION :**

Parkinson’s disease is the second most dangerous neurodegenerative disease which has no cure till now and to make it reduce prediction is important. In this project, we have used three various prediction models to predict the Parkinson’s disease which are Machine Learning Techniques i.e. KNN, Naïve Bayes and Logistic Regression. The dataset is trained using these models and we also compared these different models built using different methods and identifies the best model that fits. The aim is to use various evaluation metrics such as Accuracy, Precision, Recall, Specificity, F1-score, LR+, LR- and Youden score that produce the predicts the disease efficiently. We have used the Speech dataset that contains voice features of the patients which is available in the Kaggle website. The dataset consists of more than 700 features and 750 patient details. The models are built using the five best features which were identified by feature selection. From this results, Naïve Bayes outstands from the other two machine learning algorithms with an accuracy of 81%. This system we designed can make the predictions of the Parkinson’s disease.

**12.FUTURE SCOPE :**

In future, these models can be trained with different datasets that have best features and can be predicted more accurately. If the accuracy rate increases, it can be used by the laboratories and hospitals so that it is easy to predict in early stages. This models can be also used with different medical and disease datasets. In future the work can be extended by building a hybrid model that can find more than one disease with an accurate dataset and that dataset has common features of two diseases. In future the work can extended to build a model that may extract more important features among all features in the dataset so that it produce more accuracy.

**13. APPENDIX**

**Source Code :**

**Importing required Libraries**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

import seaborn as sns

import zipfile as zf

import os

import random

import cv2

import pickle

from imutils import build\_montages

from imutils import paths

from sklearn.metrics import classification\_report,confusion\_matrix

from sklearn import metrics

from sklearn.preprocessing import LabelEncoder,LabelBinarizer

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier,GradientBoostingClassifier,ExtraTreesClassifier

from skimage import feature

*#from google.colab.patches import cv2\_imshow*

sns**.**set()

os**.**getcwd()

## Loading the dataset

handle\_spiral **=** zf**.**ZipFile(r'dataset.zip')

handle\_spiral**.**extractall('dataset')

handle\_spiral**.**close()

spiral\_train\_healthy **=** os**.**listdir('dataset/dataset/spiral/training/healthy/')

spiral\_train\_park **=** os**.**listdir('dataset/dataset/spiral/training/parkinson/')

fp\_spiral\_train\_healthy **=** 'dataset/dataset/spiral/training/healthy/'

fp\_spiral\_train\_park **=** 'dataset/dataset/spiral/training/parkinson/'

spiral\_test\_healthy **=** os**.**listdir('dataset/dataset/spiral/testing/healthy/')

spiral\_test\_park **=** os**.**listdir('dataset/dataset/spiral/testing/parkinson/')

fp\_spiral\_test\_healthy **=** 'dataset/dataset/spiral/testing/healthy/'

fp\_spiral\_test\_park **=** 'dataset/dataset/spiral/testing/parkinson/'

## Qunatifying the images

def quantify\_image(image):

features **=** feature**.**hog(image,orientations**=**9,

pixels\_per\_cell**=**(10,10),cells\_per\_block**=**(2,2),transform\_sqrt**=**True,block\_norm**=**"L1")

return features

## Splitting of Training and Testing Data

trainX **=** []

testX **=** []

outputs **=** []

trainY **=** []

testY **=** []

for i in spiral\_train\_healthy:

image **=** cv2**.**imread(fp\_spiral\_train\_healthy**+**i)

image **=** cv2**.**cvtColor(image , cv2**.**COLOR\_BGR2GRAY)

image **=** cv2**.**resize(image , (200,200))

image **=**cv2**.**threshold(image, 0, 255,cv2**.**THRESH\_BINARY\_INV **|** cv2**.**THRESH\_OTSU)[1]

features **=** quantify\_image(image)

trainX**.**append(features)

trainY**.**append('healthy')

for i in spiral\_train\_park:

image **=** cv2**.**imread(fp\_spiral\_train\_park**+**i)

image **=** cv2**.**cvtColor(image , cv2**.**COLOR\_BGR2GRAY)

image **=** cv2**.**resize(image , (200,200))

image **=** cv2**.**threshold(image ,0,255,cv2**.**THRESH\_BINARY\_INV **|** cv2**.**THRESH\_OTSU)[1]

features **=** quantify\_image(image)

trainX**.**append(features)

trainY**.**append('parkinson')

for i in spiral\_test\_healthy:

image **=** cv2**.**imread(fp\_spiral\_test\_healthy**+**i)

outputs**.**append(image)

image **=** cv2**.**cvtColor(image , cv2**.**COLOR\_BGR2GRAY)

image **=** cv2**.**resize(image , (200,200))

image **=** cv2**.**threshold(image ,0,255,cv2**.**THRESH\_BINARY\_INV **|** cv2**.**THRESH\_OTSU)[1]

features **=** quantify\_image(image)

testX**.**append(features)

testY**.**append('healthy')

for i in spiral\_test\_park:

image **=** cv2**.**imread(fp\_spiral\_test\_park**+**i)

outputs**.**append(image)

image **=** cv2**.**cvtColor(image , cv2**.**COLOR\_BGR2GRAY)

image **=** cv2**.**resize(image , (200,200))

image **=** cv2**.**threshold(image ,0,255,cv2**.**THRESH\_BINARY\_INV **|** cv2**.**THRESH\_OTSU)[1]

features **=** quantify\_image(image)

testX**.**append(features)

testY**.**append('parkinson')

trainX **=** np**.**array(trainX)

testX **=** np**.**array(testX)

trainY **=** np**.**array(trainY)

testY **=** np**.**array(testY)

trainX

trainY

testX

testY

## Label Encoding

le **=** LabelEncoder()

trainY **=** le**.**fit\_transform(trainY)

testY **=** le**.**transform(testY)

print(trainX**.**shape,trainY**.**shape)

trainY

testY

# **Model Building**

## Train the model

print("Training model....")

model **=** RandomForestClassifier(n\_estimators**=**100)

model**.**fit(trainX,trainY)

preds **=** model**.**predict(testX)

preds

**Model Evaluation**

cnf **=** confusion\_matrix(testY,preds)

cnf

plt**.**figure(figsize**=**(5,5))

sns**.**heatmap(cnf , annot**=**True , cmap**=**"coolwarm" , cbar**=**False)

plt**.**show()

acc **=** metrics**.**accuracy\_score(testY,preds)

acc

indexes **=** np**.**random**.**randint(0,30,25)

indexes

**Test the model**

testpath**=**list(paths**.**list\_images(fp\_spiral\_train\_healthy))

idxs**=**np**.**arange(0,len(testpath))

idxs**=**np**.**random**.**choice(idxs,size**=**(25,),replace**=**False)

images**=**[]

for i in idxs:

image**=**cv2**.**imread(testpath[i])

output**=**image**.**copy()

output**=**cv2**.**resize(output,(128,128))

image**=**cv2**.**cvtColor(image,cv2**.**COLOR\_BGR2GRAY)

image**=**cv2**.**resize(image,(200,200))

image**=**cv2**.**threshold(image,0,255,cv2**.**THRESH\_BINARY\_INV **|** cv2**.**THRESH\_OTSU)[1]

features**=** quantify\_image(image)

preds**=**model**.**predict([features])

label**=**le**.**inverse\_transform(preds)[0]

if label**==**"healthy":

color**=**(0,255,0)

else:

(0,0,255)

cv2**.**putText(output,label, (3,20),cv2**.**FONT\_HERSHEY\_SIMPLEX,0.5,color,2)

images**.**append(output)

montage**=**build\_montages(images,(128,128),(5,5))[0]

cv2**.**imshow('',montage)

cv2**.**waitKey(0)

**Model accuracy and Prediction**

predictions **=** model**.**predict(testX)

cm **=** confusion\_matrix(testY, predictions)**.**flatten()

print(cm)

(tn, fp, fn, tp) **=** cm

accuracy **=** (tp **+** tn) **/** float(cm**.**sum())

print(accuracy)

**Save the model**

pickle**.**dump(model,open('parkinson.pkl','wb'))

**Application Source code using flas**k:

from flask import Flask, flash, redirect, render\_template, request, session, url\_for

from flask\_mail import Mail, Message

from itsdangerous import URLSafeTimedSerializer, SignatureExpired

import pickle

import cv2

from skimage import feature

import os.path

import sqlite3

app=Flask(\_\_name\_\_)

app.secret\_key="#@universityflaskapp@#"

# email verification

app.config.from\_pyfile('config.cfg')

mail=Mail(app)

s = URLSafeTimedSerializer(app.config['SECRET\_KEY'])

# database creation

con=sqlite3.connect("database.db")

print("Opened database successfully")

con.execute("create table if not exists customer(pid integer primary key, name text, email text, password text,status BOOLEAN)")

print("Table created successfully")

con.close()

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

def signup():

  if request.method == 'POST':

        try:

            name=request.form['name']

            email=request.form['email']

            password=request.form['password']

            con=sqlite3.connect("database.db")

            cur=con.cursor()

            cur.execute("INSERT INTO customer(name,email,password) VALUES (?,?,?)",(name,email,password))

            con.commit()

            flash("Registered successfully","success")

        except:

            con.rollback()

            flash("Problem in Registration, Please try again","danger")

        finally:

            return redirect(url\_for("index"))

            con.close()

  else:

        return render\_template('signup.html')

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

def login():

  if request.method =='POST':

        email = request.form['email']

        password = request.form['password']

        con=sqlite3.connect("database.db")

        con.row\_factory=sqlite3.Row

        cur=con.cursor()

        cur.execute("SELECT \* FROM customer where email=? and password=?",(email,password))

        data=cur.fetchone()

        if data:

            session["email"]=data["email"]

            print("sent to home")

            return redirect(url\_for("home"))

        else:

            flash("Username or Password is incorrect","danger")

            print("not sent to home")

  return redirect(url\_for("home"))

@app.route('/')

def index():

  return render\_template("index.html")

@app.route('/home')

def home():

  return render\_template("homepage.html")

@app.route('/prediction')

def prediction():

  return render\_template("prediction.html")

@app.route('/images')

def images():

  return render\_template("image.html")

@app.route('/contact')

def contact():

  return render\_template("contact.html")

@app.route('/explore')

def explore():

  return render\_template("explore.html")

@app.route('/logout')

def logout():

    session.clear()

    return render\_template('index.html')

@app.route('/healthy')

def healthy():

    return render\_template('healthy.html')

@app.route('/parkinson')

def parkinson():

    return render\_template('parkinson.html')

@app.route('/check')

def check():

    email = session["email"]

    con=sqlite3.connect("database.db")

    cur=con.cursor()

    cur.execute("SELECT status FROM customer where email=?",[email])

    data=cur.fetchone()

    con.commit()

    print(data)

    if data[0]==1:

        return render\_template('prediction.html')

    else:

        return render\_template('verify.html')

@app.route('/verify')

def verify():

    email = session["email"]

    token = s.dumps(email, salt='email-confirm')

    msg=Message('Confirm Email', sender='ibmproject2023@gmail.com.', recipients=[email])

    link=url\_for('confirm\_email', token=token, \_external=True)

    msg.body= 'Please click the link to verify your account to continue  : {} '.format(link)

    mail.send(msg)

    return render\_template("homepage.html")

@app.route('/confirm\_email/<token>')

def confirm\_email(token):

  try:

    email=s.loads(token, salt='email-confirm' , max\_age=3600\*5)

  except SignatureExpired:

    return render\_template("verify.html")

  con=sqlite3.connect("database.db")

  con.row\_factory=sqlite3.Row

  cur=con.cursor()

  cur.execute("UPDATE customer SET status = 1 WHERE email = ?",(email,))

  con.commit()

  con.close()

  return redirect(url\_for("prediction"))

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

def upload():

    if request.method == 'POST':

        f = request.files['file2']  # requesting the file

        #filename\_secure = secure\_filename(f.filename)

        basepath = os.path.dirname(

            '\_\_file\_\_')  # storing the file directory

        # storing the file in uploads folder

        filepath = os.path.join(basepath, "uploads", f.filename)

        f.save(filepath)  # saving the file

        # Loading the saved model

        print("[INFO] loading model...")

        model = pickle.loads(open('parkinson.pkl', "rb").read())

        ''' local\_filename = "./uploads/"

        local\_filename += filename\_secure

        print(local\_filename)'''

        # Pre-process the image in the same manner we did earlier

        image = cv2.imread(filepath)

        output = image.copy()

        # Load the input image, convert it to grayscale, and resize

        output = cv2.resize(output, (128, 128))

        image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

        image = cv2.resize(image, (200, 200))

        image = cv2.threshold(image, 0, 255,

                              cv2.THRESH\_BINARY\_INV | cv2.THRESH\_OTSU)[1]

        # Quantify the image and make predictions based on the extracted features using the last trained Random Forest

        features = feature.hog(image, orientations=9,

                               pixels\_per\_cell=(10, 10), cells\_per\_block=(2, 2),

                               transform\_sqrt=True, block\_norm="L1")

        preds = model.predict([features])

        print(preds)

        ls = ["healthy", "parkinson"]

        result = ls[preds[0]]

        '''color = (0, 255, 0) if result == "healthy" else (0, 0, 255)

        cv2.putText(output, result, (3, 20),

                    cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, color, 2)

        cv2.imshow("Output", output)

        cv2.waitKey(0)'''

        if result =="healthy":

          return redirect("healthy")

        else:

          return redirect("parkinson")

    return None

if \_\_name\_\_=='\_\_main\_\_':

  app.run(Debug=True)

**GitHub & Project Demo Link :**

https://github.com/IBM-EPBL/IBM-Project-54753-1662453739