

PERSONAL EXPENSE TRACKER APPLICATION

IBM-Project-26494-1662612957

**NALAIYA THIRAN PROJECT BASED LEARNING ON
PROFESSIONAL READLINESS FOR INNOVATION,
EMPLOYNMENT AND ENTERPRENEURSHIP**

A PROJECT

REPORT BY

ABIYA FEBA MARY A (310619205003)

BHARGAVI B (310619202018)

KAVYA U (310619205047)

MANUSHRI (310619205058)

**BACHELOR OF TECHNOLOGY IN
INFORMATION TECHNOLOGY**

EASWARI ENGINEERING COLLEGE, CHENNAI - 600089

INDEX

1. INTRODUCTION

- a. Project Overview
- b. Purpose

2. LITERATURE SURVEY

- a. Existing problem
- b. References
- c. Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

- a. Empathy Map Canvas
- b. Ideation & Brainstorming
- c. Proposed Solution
- d. Problem Solution fit

4. REQUIREMENT ANALYSIS

- a. Functional requirement
- b. Non-Functional requirements

5. PROJECT DESIGN

- a. Data Flow Diagrams
- b. Solution & Technical Architecture
- c. User Stories

6. PROJECT PLANNING & SCHEDULING

- a. Sprint Planning & Estimation
- b. Sprint Delivery Schedule
- c. Reports from JIRA

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

- a. Feature 1
- b. Feature 2
- c. Database Schema (if Applicable)

8. TESTING

- a. Test Cases
- b. User Acceptance Testing

9. RESULTS

- a. Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub & Project Demo Link

1. INTRODUCTION

TEAM ID :

PNT2022TMID09607

INDUSTRY MENTOR : Prof Swetha

FACULTY MENTOR : Mrs. B Sandhiya

Skills Required:

IBM Cloud, HTML, Javascript, IBM Cloud Object Storage, Python- Flask, Kubernetes, Docker, IBM DB2, IBM Container Registry

a) Project Overview

- To understand the problem for to classify if it is a regression or a classification kind of problem.
- To pre-process the image by using different data pre-processing techniques.
- To implement the algorithm by using OpenCV framework and machine learning to automatically detect Parkinson's disease in hand-drawn images of spirals and waves.
- To know how to find the accuracy of the model.
- To build web application using the Flask framework that features the detection of Parkinson's Disease.

b) Purpose

You'll be able to understand the problem to classify if it is a regression or a classification kind of problem. You will be able to know how to pre-process the image by using different data pre-processing techniques. you will be able to learn how to use OpenCV and machine learning to automatically detect Parkinson's disease in hand-drawn images of spirals and waves You will be able to know how to find the accuracy of the model. You will be able to build web applications using the Flask framework.

2. LITERATURE SURVEY

a) Existing problem

Instead of going to hospital and taking MRI scan the existing problem helps the patient to detect the Parkinsons at home with the some basic values extracted from voice recording which is the simple and easiest way.

b) References

1. W. Wang, J. Lee, F. Harrou and Y. Sun, "Early Detection of Parkinson's Disease Using Deep Learning and Machine Learning," in *IEEE Access*, vol. 8, pp. 147635-147646, 2020, doi: 10.1109/ACCESS.2020.3016062.
2. Adams WR. High-accuracy detection of early Parkinson's Disease using multiple characteristics of finger movement while typing. *PLoS One*. 2017 Nov 30;12(11):e0188226. doi: 10.1371/journal.pone.0188226. PMID: 29190695; PMCID:PMC5708704.
3. L. Ali, C. Zhu, N. A. Golilarz, A. Javeed, M. Zhou and Y. Liu, "Reliable Parkinson's Disease Detection by Analyzing Handwritten Drawings: Construction of an Unbiased Cascaded Learning System Based on Feature Selection and Adaptive Boosting Model," in *IEEE Access*, vol. 7, pp. 116480-116489, 2019, doi: 10.1109/ACCESS.2019.2932037.
4. Chakraborty, Sabyasachi & Aich, Satyabrata & Jong-Seong-Sim, & Han, Eunyoung & Park, Jinse & Kim, Hee-Cheol. (2020). Parkinson's Disease Detection from Spiral and Wave Drawings using Convolutional Neural Networks: A Multistage Classifier Approach. 298-303. 10.23919/ICACT48636.2020.9061497.
5. Jahan, Nusrat & Nesa, Arifatun & Layek, Abu. (2021). Parkinson's Disease Detection Using ResNet50 with Transfer Learning. 11. 17-23.

d)Problem Statement Definition

Parkinson's disease (PD) is a common, neurodegenerative disorder, recognized by the motor symptoms of bradykinesia, tremor, rigidity, and postural impairment. At clinical onset, extensive amounts of dopaminergic neurons have already been lost. The duration of this prodromal phase is uncertain, and it is thought to include predominantly non-motor symptoms. The progressive nature and

the symptoms of PD are disabling and reduces the quality of life. Among patients affected in working

age, early cessation of employment is common, and such socioeconomic consequences of PD may contribute to an impaired quality of life.

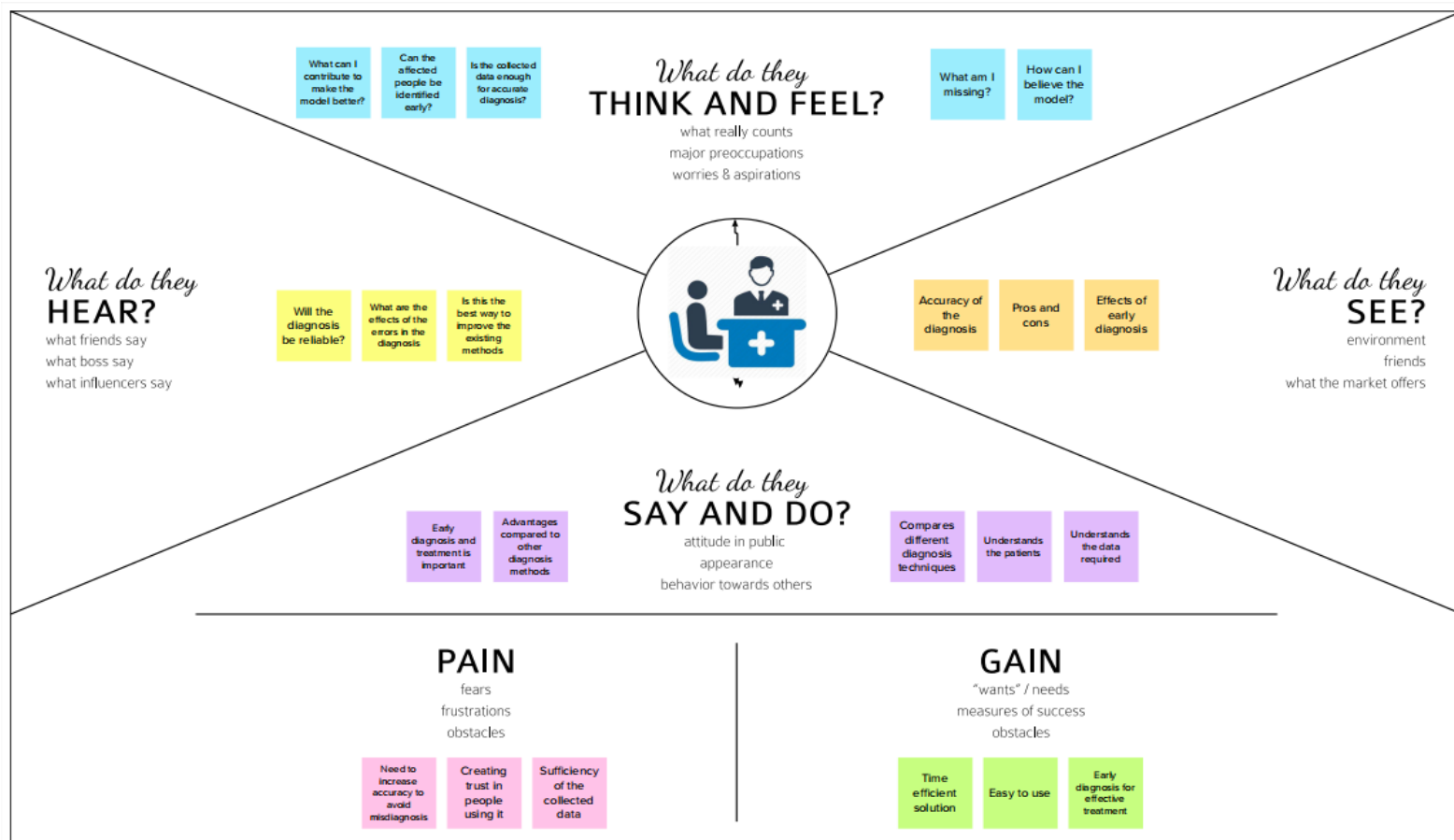
3. IDEATION & PROPOSED SOLUTION

a) Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

It is a useful tool to help teams better understand their users.


Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges



b) Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Advanced Data Science



Detecting Parkinson's Disease using Machine Learning

Brainstorm & idea prioritization

Ideation phase

10 minutes to prepare
1 hour to ideate
3-5 people recommended

Designed by:
Surya Perumal K (TL)
Sri Charan B J
Kabir S
Rohith Kumar S

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

To ideate

- Brain gathering**
Before you should participate in the session and read on both. Share relevant information or present about
- Know the goal**
Think about the problem you'd like to solve or solving in the brainstorming session
- Agree how to use the facilitation tools**
Clarify facilitation approaches to run things and produce results

Open an idea

Define your problem statement

What problem are you trying to solve? Frame your problem as a clear, tight statement. This will be the focus of your brainstorm.

To optimize

Exercise

To build a web application that improved with machine learning for detecting Parkinson's disease

Key rules of brainstorming

To run an ideation and production session

Highlighter	Encourage wild ideas
Vote papers	Listen actively
Clarify values	Respond to ideas



Brainstorm

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

10 points



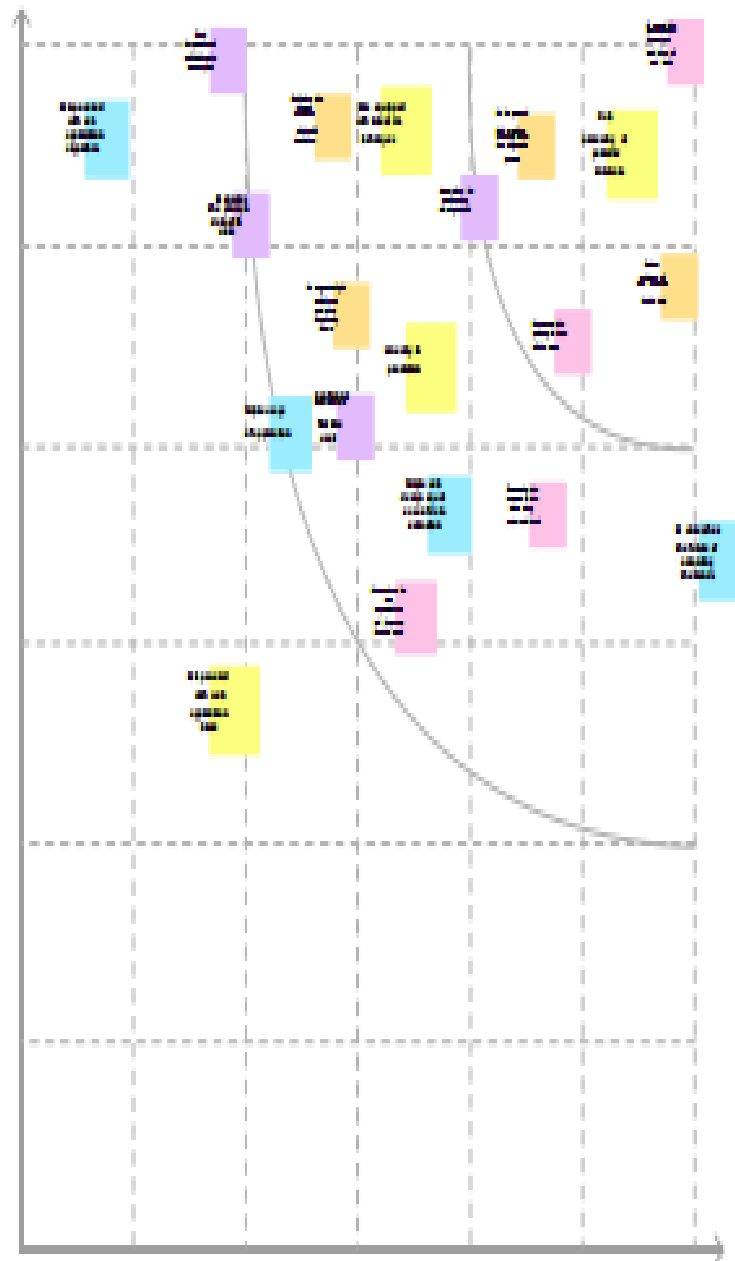
Group idea

Take turns sharing your ideas while clustering similar or related ideas as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than 10, brainstorm, try and see if you can break it up into smaller sub-groups.

20 points



Your team should all be on the same page about which important meeting is next. Place your ideas on this grid to determine which ideas are important and which are feasible.



You can export the model as an image or pdf to share with members of your company or to print if it's helpful.

- **Store the moral**
Store a reminder of the moral with a checklist, to keep them in the loop about the outcomes of the course.
- **Repeat the moral**
Repeat a copy of the moral as a PAGER MP task each week, include it in the, or save in portfolio.

Three small bar charts are displayed side-by-side. Each chart has a vertical axis labeled 'Percentage' ranging from 0 to 100. The first chart on the left has a horizontal axis with two categories: 'Yes' and 'No'. The second chart in the middle has a horizontal axis with three categories: 'Yes', 'No', and 'Don't know'. The third chart on the right has a horizontal axis with four categories: 'Yes', 'No', 'Don't know', and 'Refuse to answer'. In all three charts, the 'Yes' category shows the highest percentage, followed by 'No', 'Don't know', and 'Refuse to answer'.

1. **Introduction**
 2. **Background**
 3. **Methodology**
 4. **Results**
 5. **Conclusion**
 6. **References**

Figure 1: Schematic representation of the experimental design. The figure shows a timeline of the experiment. It starts with a 'Pretest' phase, followed by a 'Main Experiment' phase. The Main Experiment is divided into two parts: 'Part 1' and 'Part 2'. Part 1 involves a 'Pretest' and a 'Main Experiment' with 'Condition 1' and 'Condition 2'. Part 2 involves a 'Pretest' and a 'Main Experiment' with 'Condition 1' and 'Condition 2'. The timeline also indicates the duration of each phase and the number of participants in each group.

[Home](#)
[About Us](#)
[Contact Us](#)
[Privacy Policy](#)

Identity is a social construct, and it is not a fixed, essential trait. It is a process that is constantly being negotiated and re-negotiated. It is a process that is shaped by the social context in which it takes place.



c) proposed solution

Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Dopamine, a chemical that enables brain cells to connect with one another, is disrupted in Parkinson's disease-causing brain cells. It is a degenerative central nervous system illness that impairs movement and causes tremors and rigidity. The symptoms usually emerge slowly, and as the disease worsens, non-motor symptoms become more common. The most obvious early symptoms are tremor, rigidity, slowness of movement, and difficulty with walking.
2.	Idea/Solution description	The project's goal is to use the Python modules scikit-learn, numpy, pandas, and xgboost to offer a method for detecting Parkinson's illness. After loading the data, obtaining the features and labels, scaling the features, splitting the dataset, and creating an XGBClassifier, we will determine the model's correctness. The major goal of the implementation is to create a model using XGBoost that can be used to categorise a person as Healthy or having Parkinson's disease.
3.	Novelty/Uniqueness	A sparsity-aware split discovery technique is incorporated into the XGBoost algorithm for detecting Parkinson's disease in order to accommodate various sparsity patterns in the data. The XGBoost algorithm's out-of-core computing function maximises and optimises the use of the available disc space.

4.	SocialImpact/CustomerSatisfaction	The likelihood of the disease progressing, limiting the impact of PD on QoL, and possibly lowering long-term treatment costs are all dependent on the early diagnosis and treatment of PD. The suggested remedy tries to use a variety of indicators to predict early Parkinson disease in patients.
5.	BusinessModel(RevenueModel)	<p>Key partners:</p> <ul style="list-style-type: none"> • Distributors • Academia • Platforms <p>Key activities:</p> <ul style="list-style-type: none"> • Development of solutions • Data acquisition • Platform operation • Clinical <p>Key Resources:</p> <ul style="list-style-type: none"> • Data • People <p>Value Propositions:</p> <ul style="list-style-type: none"> • Uniqueness • Performance • Cost reduction <p>Customer Segments:</p> <ul style="list-style-type: none"> • Clinics, Hospitals • Software/platform developers
6.	Scalability of the Solution	With various calculations, XGBoost's exactness, correctness, review, and other qualities are quite good. In terms of performance, XGBoost not only keeps up with but outperforms all those other algorithms. Real-world scale issues can be resolved using XGBoost with the least amount of resources.

c) Problem Solution Fit

Define C.S., fit into	1. CUSTOMER SEGMENT(S) CS <ul style="list-style-type: none">• Parkinson's disease is a progressive disorder that affects the nervous system and the parts of the body controlled by the nerves.• Parkinson's patient have symptoms of Tremor, Slowed movement (bradykinesia), Rigid muscles, Writing changes, Impaired posture and balance, Loss of automatic movements, Speech changes.	6. CUSTOMER CC <ul style="list-style-type: none">• Accurate prediction of disease.• Early prediction of the disease.	5. AVAILABLE SOLUTIONS AS <ul style="list-style-type: none">• The physician takes a medical history and does a physical examination.• Performs a neurological examination, testing agility, muscle tone, gait and balance.• PET and MRI scan also used by doctor for classification.• In Machine learning field Several algorithms are proposed for classification.	Explore A.S.	
	2. JOBS-TO-BE-DONE / PROBLEMS J&P <ul style="list-style-type: none">• Parkinson patient have problem of rigid muscles and writing changes.• We have to collect the drawing of normal and parkinson patients.• Using the drawing, we have to detect the presence of parkinson disease by applying necessary algorithm.	3. PROBLEM ROOT CAUSE RC <ul style="list-style-type: none">• Lack of data• New to field of study	7. BEHAVIOUR BE <ul style="list-style-type: none">• Random forests are preferred over decision trees is that they are stable and are low variance models.• They also overcome the problem of overfitting present in decision trees. Since they use bootstrapped data and random set of features, they ensure diversity and robust performance.• They are immune to curse of dimensionality as they do not consider all the features at one time for individual trees.• The main disadvantage of random forests is their lack of interpretability.		Focus on J&P, tap into BE, understand
	3. TRIGGERS TR <ul style="list-style-type: none">• Parkinson disease are easy to treat if we detect in early stage.• Provide more efficient algorithm to detect the parkinson disease in its early stage.	10. YOUR SOLUTION SL <ul style="list-style-type: none">• The project aims at presenting a solution for parkinson's disease detection using suitable machine learning algorithms.• Algorithms such as random forest and decision trees are used for disease prediction.• We will load the dataset into dataframe and get the feature and label ,preprocess the data and classify it.	8. CHANNELS of BEHAVIOUR CH <p>8.1 Online:</p> <ul style="list-style-type: none">• If we use online channels, then the Customer can check their result with online comparison using our platform. <p>8.2 Offline:</p> <ul style="list-style-type: none">• offline channels• If the disease predicted then the customer need to go to Hospital for Treatment in offline mode.		
4. EMOTIONS: BEFORE / AFTER EM <p>Before:</p> <ul style="list-style-type: none">• Does not know about outcome of prediction. <p>After:</p> <ul style="list-style-type: none">• Got idea to detect parkinson disease in early stage.• Certain about prediction and to take necessary					
Identify strong TR & EM	Focus on J&P, tap into BE, understand				

4. REQUIREMENT ANALYSIS

a. Functional requirements

FR No.	Functional Requirement (Epic)	SubRequirement (Story/Sub-Task)
FR-1	Analyzing Symptoms	<p>Stiffness in muscles</p> <p>Rigidity and slowness in body movements</p> <p>Breaking of voice and shivering in tone</p> <p>Difficulty with walking</p> <p>Emotional and behavioral changes</p> <p>Dementia and depression</p>
FR-2	Collecting voice dataset	<p>Speech and voice recordings of the patient is collected.</p> <p>Various voice parameters are measured.</p>
FR-3	Working on dataset	<p>Voice recording is measured against the parameters.</p> <p>Data is preprocessed and dependent variables are found.</p> <p>Data is split into train and test data.</p> <p>Training and testing is done and the model is evaluated.</p>

FR-4	Applying SVM algorithm	<p>SVM finds a hyper-plane that creates a boundary between the types of data.</p> <p>We plot each data item in the dataset in an N-dimensional space.</p> <p>The algorithm tries to find the optimal hyperplane which can be used to classify dataset into healthy person or person suffering from Parkinson.</p>
------	------------------------	---

FR-5	Providing insights of dataset	<p>Raw data collection and sharing of data and systems are essential factors in hospital management.</p> <p>According to these data appropriate measures can be taken. Providing dataset without error.</p> <p>or.</p> <p>Providing treatment for the patients who are suffering from Parkinson.</p>
------	-------------------------------	--

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Usable systems are straightforward to use by as many people as possible, both in case of either end users or administrators to view the hospital records when needed.
NFR-2	Security	<p>Patient identification:</p> <p>To recognize and analyze the patient perfectly.</p>
NFR-3	Reliability	<p>Understanding the current trend and working on to it to solve the problem in an efficient manner.</p> <p>Being software as a service, HMS is highly resilient to any technology disruptions, downtime, or crashes experienced by other technology systems.</p>
NFR-4	Performance	<p>Response time:</p> <p>Providing acknowledgment in minimal time about the patient information.</p> <p>Comfortability:</p> <p>To ensure that the guidelines and accessibilities are followed.</p>

NFR-5	Availability	<p>Better coordination with the hospital management to provide all its resources accessible when needed.</p> <p>Accessibility of all medical facilities.</p>
NFR-6	Scalability	<p>Make sure that the work is done in more efficient way with the appropriate resources.</p> <p>Make complex decisions understandable with proper data.</p>

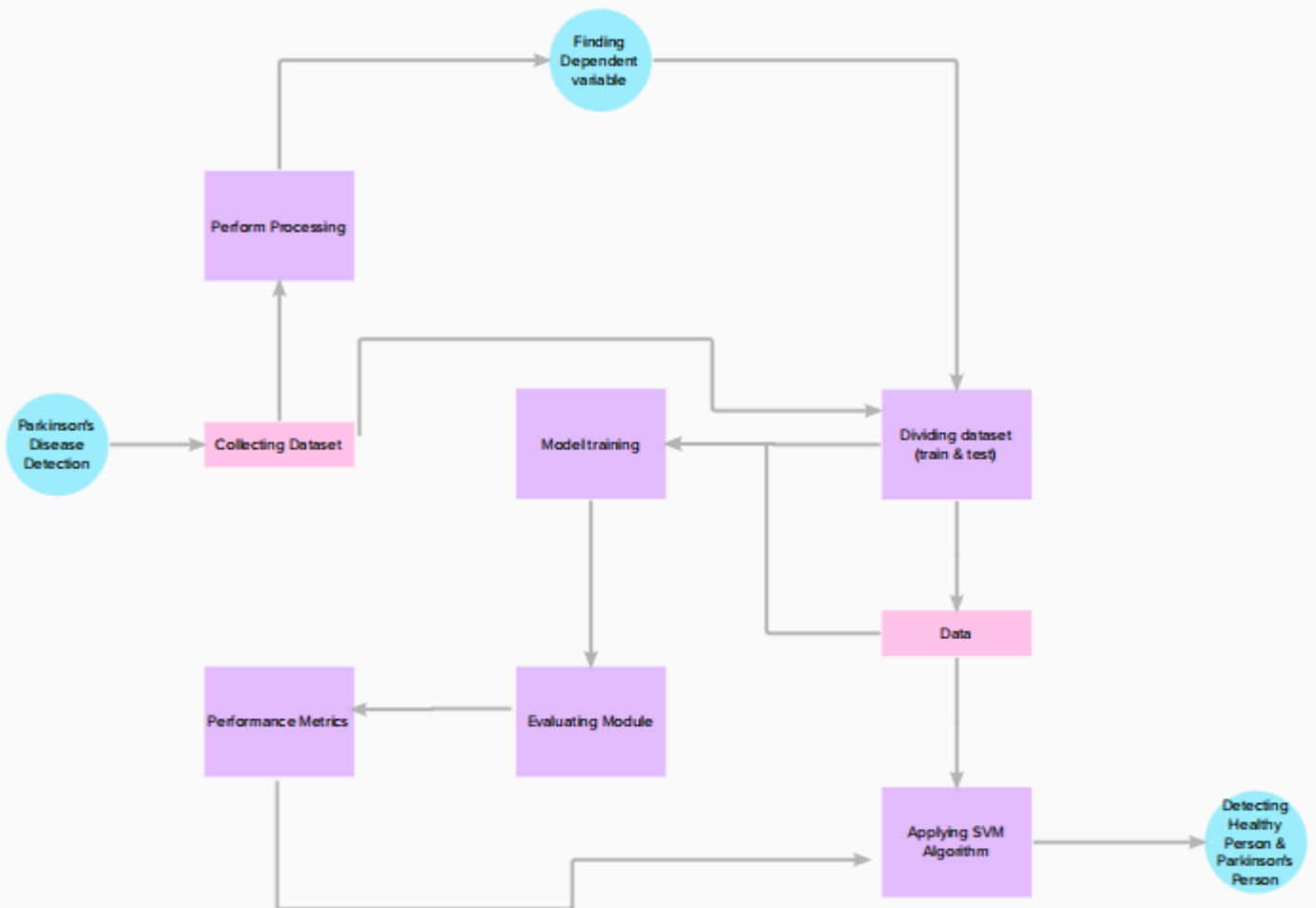
b. Non Functional Requirements

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Usable systems are straightforward to use by as many people as possible, both in case of either end users or administrators to view the hospital records when needed.
NFR-2	Security	Patient identification: To recognize and analyze the patient perfectly.
NFR-3	Reliability	Understanding the current trend and working on it to solve the problem in an efficient manner. Being software as a service, HMS is highly resilient to any technology disruptions, downtime, or crashes experienced by other technology systems.
NFR-4	Performance	Response time: Providing acknowledgment in minimal time about the patient information. Comfortability: To ensure that the guidelines and accessibilities are followed.
NFR-5	Availability	Better coordination with the hospital management to provide all its resources accessible when needed. Accessibility of all medical facilities.
NFR-6	Scalability	Make sure that the work is done in a more efficient way with the appropriate resources. Make complex decisions understandable with proper data.

5. PROJECT DESIGN

a) 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.



b) Solution & Technical Architecture

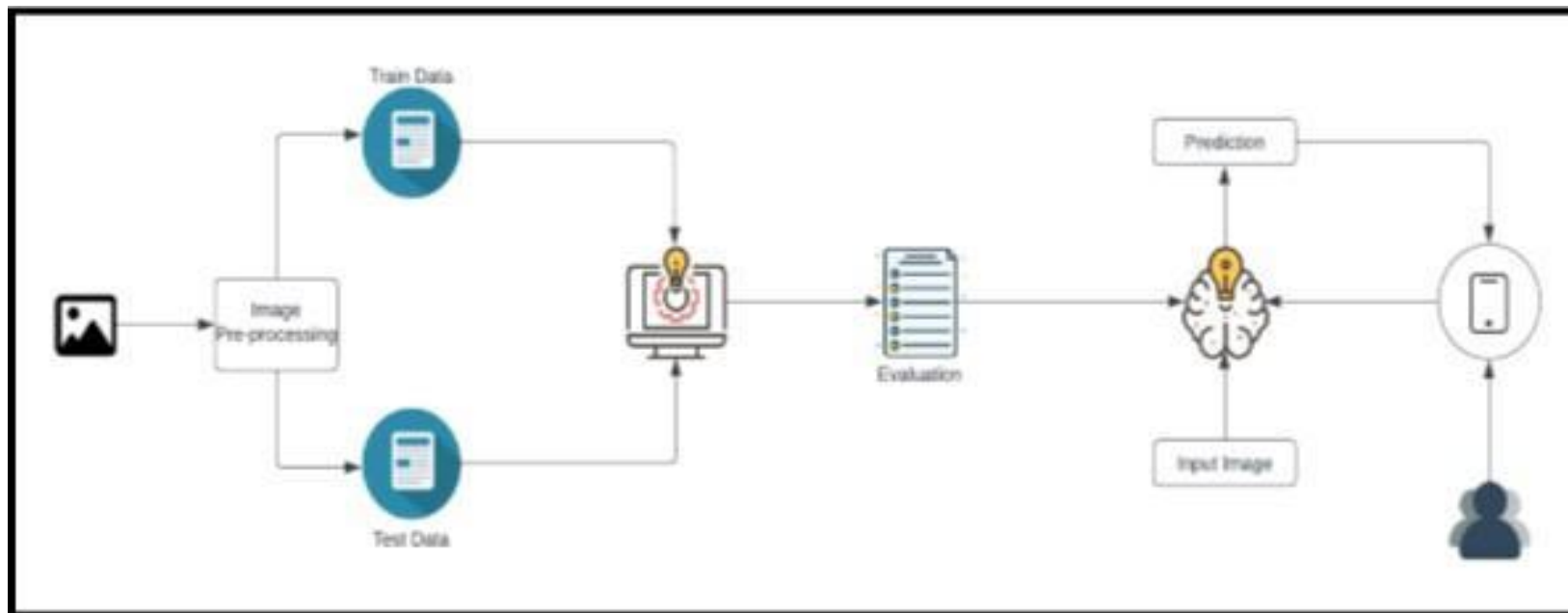
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

Find the best tech solution to solve existing business problems.

Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.

Define features, development phases, and solution requirements.

Provide specifications according to which the solution is defined, managed, and delivered.



Technical Architecture:

The deliverable shall include the technological stack as well as the required details in both the tables.

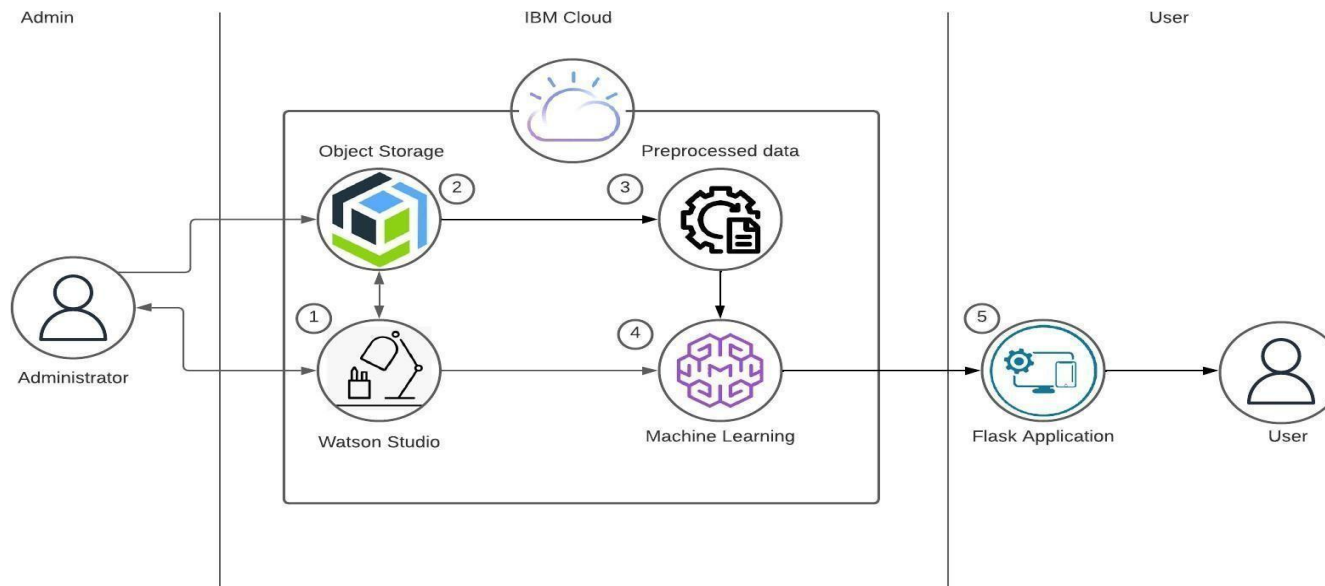


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How the user interacts with the application e.g. Web UI	HTML, CSS, Python flask.
2.	Application Logic-1	Register and Login page	HTML, CSS, Python flask.
3.	Application Logic-2	Home Page	HTML, CSS.
4.	Application Logic-3	Test vital page	HTML, CSS, Python flask.

5.	Database	Data Type, Configurations, etc.	MySQL.
6.	Cloud Database	Database Service on Cloud	IBM Database.
7.	File Storage	File Storage requirements	IBM Cloud Object Storage
8.	External API-1	Purpose of External API used in the application	IBM API Connect.
9.	External API-2	Purpose of External API used in the application	NIL
10.	Machine Learning Model	Train the classification model using the Random forest classification algorithm.	IBM Watson Studio.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud.	Local Server Configuration: Local System. Cloud Server Configuration:IBM Watson

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Python Flask, Jupyter Notebook, Tensorflow, and Python libraries.
2.	Security Implementations	List all the security/access controls implemented, use of firewalls, etc.	Through Password, Email Confirmation.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Python Libraries.
4.	Availability	Justify the availability of applications (e.g. use of load balancers, distributed servers, etc.)	IBM Watson Machine Learning.
5.	Performance	Design Considerations for the performance of the application (number of requests per sec, use of Cache, use of CDNs), etc.	Flask.

Table-2: Application Characteristics:

PNT2022TMID09607

c) User Stories

The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer. Note that "customers" don't have to be external end users in the traditional sense, they can also be internal customers or colleagues within your organization who depend on your team.

User stories are a few sentences in simple language that outline the desired outcome. They don't go into detail. Requirements are added later, once agreed upon by the team.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Login	USN-1	Entering Web page	Enter the application	High	Sprint 1
	Homepage	USN-2	Entering to the "Homepage" of the UI (Webpage)	Enter the homepage	High	Sprint 1
	About	USN-3	I can click on the "About" to details about the Application	Get the details about the application	Low	Sprint 2
	Begin	USN-4	As a user I can get my voice signal values from the computer.	Choose my voice Recording from my Device and extract the values	High	Sprint 2
	Predict	USN-5	As a user I can turn on the microphone or earphone to record my voice and extract needed values from it	Turn on the microphone or earphone to record the voice and extract values from the recording	High	Sprint 3
		USN-6	Predicting by using voice signal values	Can monitor change in voice or voice shaking and predict parkinsons disease	High	Sprint 3

6. PROJECT PLANNING & SCHEDULING

a. Sprint planning and estimation

Sprint planning and Estimation is done by the entire team during Sprint Planning Meeting. The objective of the Estimation would be to consider the User Stories for the Sprint by Priority and by the Ability of the team to deliver during the Time Box of the Sprint.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Login	USN-1	Entering Web page	Enter the application	High	Sprint 1
	Homepage	USN-2	Entering to the "Homepage" of the UI (Webpage)	Enter the homepage	High	Sprint 1
	About	USN-3	I can click on the "About" to details about the Application	Get the details about the application	Low	Sprint 2
	Begin	USN-4	As a user I can get my voice signal values from the computer.	Choose my voice Recording from my Device and extract the values	High	Sprint 2
	Predict	USN-5	As a user I can turn on the microphone or earphone to record my voice and extract needed values from it	Turn on the microphone or earphone to record the voice and extract values from the recording	High	Sprint 3
		USN-6	Predicting by using voice signal values	Can monitor change in voice or voice shaking and predict parkinsons disease	High	Sprint 3

6. PROJECT PLANNING & SCHEDULING

a . Sprint Planning & Estimation

Sprint planning and Estimation is done by the entire team during Sprint Planning Meeting. The objective of the Estimation would be to consider the User Stories for the Sprint by Priority and by the Ability of the team to deliver during the Time Box of the Sprint.

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my username, email, password, contact number and confirming my password.	5	High	TM-1 TM-2
Sprint-1	Login	USN-2	As a user, I can enter the username and password after registration for login	5	High	TM-1 TM-2
Sprint-2	Dashboard	USN-3	As a user, I can register for the application through Gmail and see the details in Dashboard	10	Low	TM-3 TM-4
Sprint-1	Details about	USN-4	As a user, I can register for the application through Gmail	5	Medium	TM-1 TM-2
Sprint-1	Login and repeated	USN-5	As a user, I can log into the application by entering email and password	5	High	TM-1 TM-2
Sprint-2	Web page details	USN-6	As a user I must extract certain values from the recorded voice and fill the form to detect Parkinsons Disease	10	High	TM-3 TM-4
Sprint-3	Upload the voice signal extracted details in the web application	USN-7	As a user I must receive a correct predicted output	20	High	TM-1 TM-2
Sprint-4	Provide efficient customer support	USN-8	As a user, I need to get support from developers in case of queries and failure of service provided	10	Medium	TM-3 TM-4
Sprint-4	Overview the entire process. Take all the responsibility and act bridge between users and developers	USN-9	We need to satisfy the customer needs in an efficient way and make sure any sort of errors are fixed	10	High	TM-3 TM-4





























b. Sprint Delivery Schedule

Sprint delivery schedule is a schedule prepare with timelines Within which a particular task should be completed.

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

ii. Board

iii. Road Map

 <p>To create a board Sketching, and asking a lot of questions</p>	 <p>Envision How do users currently interact with the process?</p>	 <p>Enter What do people experience and how they begin the process?</p>	 <p>Engage In the core moments in the process, what happens?</p>	 <p>Exit What do people typically experience in the process that does?</p>
 <p>Steps What does the process (or process) typically experience?</p>	   	 		
 <p>Interactions What interactions do they have at each step along the way?</p> <ol style="list-style-type: none"> 1. People who do they use or talk to? 2. Places where we use it? 3. Things that help or hinder the process? 	   			
 <p>Goals & motivations What are the goals, what do people want to achieve? (Why not...? or Why we want...?)</p>	 			
 <p>Positive moments What are the positive moments, what do people like, what do they like, or dislike?</p>				
 <p>Negative moments What are the negative moments, what do people dislike, what do they dislike, or dislike?</p>				
 <p>Areas of opportunity How might we make each step better? What ideas do we have? What is our vision for the future?</p>				

7. CODING & SOLUTIONING

i. Python App:

Base.html

```
<!DOCTYPE
html>
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Parkinson Disease Detection</title>
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.cs
s">
<link rel="stylesheet" href="{{ url_for('static', filename='css/styles1.css')}}">
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-light bg-light">
<div class="collapse navbar-collapse">
<ul class="navbar-nav ml-auto">
<li class="nav-item">
<a class="nav-link" href="{{url_for('home')}}">Home</a>
</li>
{% if not logged_in: %}
<li class="nav-item">
<a class="nav-link" href="{{url_for('login')}}">Login</a>
</li>
<li class="nav-item">
<a class="nav-link" href="{{url_for('register')}}">Register</a>
</li> {% endif %}
```

```

</li>
</ul>
</div>
</nav>
{% block content %}
{% endblock %}
</body>
</html>

```

Base1.html

```

<html
lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Parkinson Disease Detection</title>
<link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet">
<script
src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
<script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
<script
src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
<link href="{ { url_for('static', filename='css/main.css') } }" rel="stylesheet">
</head>
<body>
<nav class="navbar navbar-dark bg-dark">
<div class="container">
<a class="navbar-brand" href="#">Parkinson Disease Detection</a>
<!--<button class="btn btn-outline-secondary my-2 my-sm-0"
type="submit">Help</button-->
</div>
</nav>
<div class="container">
<div id="content" style="margin-top:2em">{% block content %}{%
endblock %}</div>

```

```

</div>
</body>
<footer>
<script src="{ { url_for('static', filename='js/main.js') } }"
type="text/javascript"></script>
</footer>
</html>

```

index.html

```

{% extends
"base1.html"
%} {% block
content %}
<center><h2>Parkinson Disease Detection</h2>
</center>
<center>
<div>
<form id="upload-file" method="post" enctype="multipart/form-data">
<label for="imageUpload" class="upload-label">
Upload image
</label>
<input type="file" name="file" id="imageUpload" accept=".png, .jpg, .jpeg">
</form>
<div class="image-section" style="display:none;">
<div class="img-preview">
<div id="imagePreview">
</div>
</div>
<div>
<button type="button" class="btn btn-primary btn-lg " id="btn
predict">Predict!</button>
</div>
</div>
<div class="loader" style="display:none;"></div>
<h3 id="result">
<span> </span>
</h3>

```

```
</div>
```

```
</center>
```

```
{% endblock %}
```

Index1.html

```
{% extends
```

```
"base.html"
```

```
%}
```

```
{% block content %}
```

```
<div class="box">
```

```
<h1>Parkinson Detection</h1><br><br>
```

```
<a href="{ {url_for('login')} }" class="btn btn-primary btn-block  
btn-large">Login</a>
```

```
<a href="{ {url_for('register')} }" class="btn btn-secondary btn-block btn  
large">Register</a>
```

```
</div>
```

```
{% endblock %} login.html
```

```
{% extends
```

```
"base.html"
```

```
%}
```

```
{% block content %}
```

```
<div class="box">
```

```
<h1>Login</h1>
```

```
{% with messages = get_flashed_messages() %}
```

```
{% if messages %}
```

```
{% for message in messages %}
```

```
<p>{{ message }}</p>
```

```
{% endfor %}
```

```
{% endif %}
```

```
{% endwith %}
```

```
<form action="{ { url_for('login')} }" method="post">
```

```
<input type="text" name="email" placeholder="Email" required="required"/>
```

```
<input type="password" name="password" placeholder="Password"  
required="required"/>
```

```
<button type="submit" class="btn btn-primary btn-block btn-large">Let me  
in.</button>
```

```
</form>
```

```
</div>
```

```
{% endblock %}
```

register.html

```
{% extends
```

```
"base.html"
```

```
%}
```

```
{% block content %}
```

```
<div class="box">
```

```
<h1>Register</h1>
```

```
<form action="{{ url_for('register') }}" method="post">
```

```
<input type="text" name="name" placeholder="Name" required="required"
```

```
/>
```

```
<input type="email" name="email" placeholder="Email" required="required"
```

```
/>
```

```
<input type="password" name="password" placeholder="Password"
```

```
required="required" />
```

```
<button type="submit" class="btn btn-primary btn-block btn-large">Sign me  
up.</button>
```

```
</form>
```

```
</div>
```

```
{% endblock %} Main.css
```

```
img
```

```
preview
```

```
{
```

```
width: 256px;
```

```
height: 256px;
```

```
position: relative;
```

```
border: 5px solid #F8F8F8;
```

```
box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);
```

```
margin-top: 1em;
```

```
margin-bottom: 1em;
```

```
}
```

```
.img-preview>div {
```

```
width: 100%;
```

```
height: 100%;
```

```
background-size: 256px 256px;
```

```
background-repeat: no-repeat;
background-position: center;
}
input[type="file"] {
display: none;
}
.upload-label{
display: inline-block;
padding: 12px 30px;
background: #541690;
color: #fff;
font-size: 1em;
transition: all .4s;
cursor: pointer;
}
.upload-label:hover{
background: #34495E;
color: #541690;
}
.loader {
border: 8px solid #34495E; /* Light grey */
border-top: 8px solid #34495E; /* Blue */
border-radius: 50%;
width: 50px;
height: 50px;
animation: spin 1s linear infinite;
}
@keyframes spin {
0% { transform: rotate(0deg); }
100% { transform: rotate(360deg); }
}
Styles.css
*,
*:before,
*:after {
box-sizing: border-box;
}
```



```
html {  
  font-size: 18px;  
  line-height: 1.5;  
  font-weight: 300;  
  color: #333;  
  font-family: "Nunito Sans", sans-serif;  
}  
body {  
  margin: 0;  
  padding: 0;  
  height: 100vh;  
  background-color: #ecf0f9;  
  background-attachment: fixed;  
}  
.large {  
  font-size: 3rem;  
}  
.content {  
  display: flex;  
  margin: 0 auto;  
  justify-content: center;  
  align-items: center;  
  flex-wrap: wrap;  
  max-width: 1500px;  
}  
p.overview {  
  font-size: 12px;  
  height: 200px;  
  width: 100%;  
  overflow: hidden;  
  text-overflow: ellipsis;  
}  
.heading {  
  width: 100%;  
  margin-left: 1rem;  
  font-weight: 900;
```

```
font-size: 1.618rem;
text-transform: uppercase;
letter-spacing: 0.1ch;
line-height: 1;
padding-bottom: 0.5em;
margin-bottom: 1rem;position: relative;
}
.heading:after {
display: block;
content: "";
position: absolute;
width: 60px;
height: 4px;
background: linear-gradient(135deg, #1a9be6, #1a57e6);
bottom: 0;
}
.description {
width: 100%;
margin-top: 0;
margin-left: 1rem;
margin-bottom: 3rem;
}
.card {
color: inherit;
cursor: pointer;
width: calc(33% - 3rem);
min-width: calc(33% - 3rem);
height: 400px;
min-height: 400px;
perspective: 1000px;
margin: 1rem auto;
position: relative;
}
@media screen and (max-width: 800px) {
.card {
width: calc(50% - 3rem);
```

```
}  
}  
}  
@media screen and (max-width: 500px) {  
.card {  
width: 100%;  
}  
}  
.front, .back {  
display: flex;  
border-radius: 6px;  
background-position: center;  
background-size: cover;  
text-align: center;  
justify-content: center;  
align-items: center;  
position: absolute;  
height: 100%;  
width: 100%;  
-webkit-backface-visibility: hidden;backface-visibility: hidden;  
transform-style: preserve-3d;  
transition: ease-in-out 600ms;  
}  
.front {  
background-size: cover;  
padding: 2rem;  
font-size: 1.618rem;  
font-weight: 600;  
color: #fff;  
overflow: hidden;  
font-family: Poppins, sans-serif;  
}  
.front:before {  
position: absolute;  
display: block;  
content: ";  
top: 0;
```

```
left: 0;
right: 0;
bottom: 0;
background: linear-gradient(135deg, #1a9be6, #1a57e6);
opacity: 0.25;
z-index: -1;
}
.card:hover .front {
transform: rotateY(180deg);
}
.card:nth-child(even):hover .front {
transform: rotateY(-180deg);
}
.back {
background: #fff;
transform: rotateY(-180deg);
padding: 0 2em;
}
.card:hover .back {
transform: rotateY(0deg);
}
.card:nth-child(even) .back {
transform: rotateY(180deg);
}
.card:nth-child(even):hover .back {
transform: rotateY(0deg);
}
.button {
transform: translateZ(40px);
cursor: pointer;
-webkit-backface-visibility: hidden;
backface-visibility: hidden;font-weight: bold;
color: #fff;
padding: 0.5em 1em;
border-radius: 100px;
font: inherit;
```

```
background: linear-gradient(135deg, #1a9be6, #1a57e6);
border: none;
position: relative;
transform-style: preserve-3d;
transition: 300ms ease;
}
.button:before {
transition: 300ms ease;
position: absolute;
display: block;
content: "";
transform: translateZ(-40px);
-webkit-backface-visibility: hidden;
backface-visibility: hidden;
height: calc(100% - 20px);
width: calc(100% - 20px);
border-radius: 100px;
left: 10px;
top: 16px;
box-shadow: 0 0 10px 10px rgba(26, 87, 230, 0.25);
background-color: rgba(26, 87, 230, 0.25);
}
.button.delete-button {
background-color: rgba(230, 87, 230, 0.25);
background: linear-gradient(135deg, #e61a46, #e61a1a);
}
.button.delete-button:before {
background-color: rgba(230, 87, 230, 0.25);
box-shadow: 0 0 10px 10px rgba(230, 87, 230, 0.25);
}
.button:hover {
transform: translateZ(55px);
}
.button:hover:before {
transform: translateZ(-55px);
}
```

```

.button:active {
transform: translateZ(20px);
}
.button:active:before {
transform: translateZ(-20px);
top: 12px;
top: 12px;
}
.container.add { margin-top: 40px;
margin-bottom: 20px;
}
.rating {
color: #E4BB23;
}
.review {
font-style: italic;
}
.movie_gens {
font-size: 11.5px;
}
.title {
font-weight: bold;
}
.release_date {
font-weight: normal;
}

```

Main.js

```

$(document).ready(function
() {
// Init
$('.image-section').hide();
$('.loader').hide();
$('#result').hide();
// Upload Preview
function readURL(input) {
if (input.files && input.files[0]) {

```

```

var reader = new FileReader();
reader.onload = function (e) {
$('#imagePreview').css('background-image', 'url(' +
e.target.result + ')');
$('#imagePreview').hide();
$('#imagePreview').fadeIn(650);
}
reader.readAsDataURL(input.files[0]);
}
}
$('#imageUpload').change(function () {
$('.image-section').show();
$('#btn-predict').show();
$('#result').text("");
$('#result').hide();
readURL(this);
});
// Predict
$('#btn-predict').click(function () {
var form_data = new FormData($('#upload-file')[0]); // Show loading
animation
$(this).hide();
$('.loader').show();
// Make prediction by calling api /predict
$.ajax({
type: 'POST',
url: '/predict',
data: form_data,
contentType: false,
cache: false,
processData: false,
async: true,
success: function (data) {
// Get and display the result
$('.loader').hide();
$('#result').fadeIn(600);

```

```

$('#result').text(' Result: ' + data);
console.log('Success!');
},
});
});
});
});

```

Main.py

```

import
pickle
import sklearn
from flask import Flask, render_template,request,redirect,url_for,flash
from flask_bootstrap import Bootstrap
from flask_sqlalchemy import SQLAlchemy
from sqlalchemy.orm import relationship
from flask_wtf import FlaskForm
from werkzeug.utils import secure_filename
from wtforms import StringField, SubmitField,FloatField,IntegerField
from wtforms.validators import DataRequired
from werkzeug.security import
generate_password_hash,check_password_hash
import os
import cv2
from skimage import feature
from flask_login import
login_user,logout_user,LoginManager,UserMixin,current_user,login_required
app = Flask(__name__)
app.config['SECRET_KEY'] = '8BYkEfBA6O6donzWlSihBXox7C0sKR6b'
app.config['SQLALCHEMY_DATABASE_URI']='sqlite:///database.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
db = SQLAlchemy(app)
Bootstrap(app)
login_manager = LoginManager()
login_manager.init_app(app)
class users(UserMixin,db.Model): id =
db.Column(db.Integer,primary_key=True)
email= db.Column(db.String(200),nullable=False)

```



```

password = db.Column(db.String(300),nullable=False)
name = db.Column(db.String(100),nullable=False)
@login_manager.user_loader
def user_load(id):
    return users.query.get(int(id))
@app.route("/")
def home():
    return render_template("index1.html")
@app.route("/register",methods=['GET','POST'])
def register():
    if request.method == 'POST':
        if users.query.filter_by(email=request.form['email']).first():
            flash('User already registered')
            return redirect(url_for('login'))
        else:
            password =
            generate_password_hash(request.form['password'],method="pbkdf2:sha256",
            salt_length=8)
            user = users(
            email = request.form['email'],
            password = password,
            name = request.form['name']
            )
            db.session.add(user)
            db.session.commit()
            return redirect(url_for('home'))
            return render_template('register.html')
@app.route("/login",methods=['GET','POST'])
def login():
    if request.method == 'POST':
        email= request.form['email']
        password = request.form['password']
        k=users.query.filter_by(email=email).first()
        if not k:
            flash('User not registered')
            return redirect(url_for('login'))

```

```

elif check_password_hash(k.password,password):
    login_user(k)
    return redirect(url_for('model'))
else:
    flash('Wrong password')
    return redirect(url_for('login'))
    return render_template('login.html')
@app.route("/logout")
def logout():
    logout_user()
    return redirect(url_for('home'))
@app.route("/parkinson")
def model(): return render_template('index.html')
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
@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':
        f = request.files['file'] # requesting the file
        basepath = os.path.dirname(os.path.realpath('__file__')) # storing the file
        directory
        filepath = os.path.join(basepath, "uploads", f.filename) # storing the file in
        uploads
        folder
        f.save(filepath)
        image = cv2.imread(filepath)
        image = cv2.cvtColor(image, cv2.COLOR_BGR2GR

```


10. ADVANTAGES & DISADVANTAGES

Advantages:

- Major advantage of this tool is that it helps to detect the Parkinsons disease from home.
- It is also easy to use and is quicker to detect Parkinsons disease.
- It can also be performed in any place and everywhere.

Disadvantages:

- The person's who doesn't able to speak cannot detect Parkinsons using this tool

1. CONCLUSION

Parkinson's disease has been plaguing humans for thousands of years and was described in detail

in ancient medical writings. Early sufferers from its effects were treated with varying results by a variety of plant-based treatments, some of which are still in use today. With the discovery of dopamine in the twentieth century and the subsequent development of dopamine replacement therapy, plus surgical techniques such as deep brain stimulation (DBS), many of the debilitating symptoms are now successfully treated—at least for a time. The hope is to find the cause of PD, along with treatments that stop the disease from progressing. Of particular interest, PD research is uncovering what may turn out to be a common pathophysiologic mechanism underlying dementia and PD. For now, healthcare providers must continue to educate themselves about currently available treatments and hope for better alternatives in the near future.

1. **FUTURE**

- The tool can be made more accurate by adding even more algorithms.
- The tool can be not only detected by voice also by image and also Gait detection.
- Can add and get more personal information and past medical information.
- Can add more security and authentication.

2. **APPENDIX**

Code Github Link : <https://github.com/IBM-EPBL/IBM-Project-35311-1660283318>
