An

Industry Oriented Mini Project

Report on

Parkinson's Disease Detection

Submitted in partial fulfillment of the

Requirements for the award of degree of

Bachelor of Technology

in

Computer Science and Engineering

By

AVULA NAGARAJU
21EG505803

GANAPURAM RAMYA
21EG505823

GUNDLA YASHWANTH
21EG505827

Under the Guidance of
Mrs. A.Durga Bhavani
Asst. Professor, Department of CSE

Department of Computer Science and Engineering

ANURAG UNIVERSITY

(An Autonomous Institution, Approved by AICTE and NBA Accredited)
Venkatapur (V), Ghatkesar (M), Medchal(D)., T.S-500088
(2020-2024)

DECLARATION

We hereby declare that the project work entitled "**PARKINSON'S DISEASE DETECTION**" submitted to **Anurag university** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology (B. Tech)** in Computer Science and Engineering is a record of an original work done by us under the guidance **of Mrs. A. Durga Bhavani, Assistant Professor** and this project work have not been submitted to any other university for the award of any other degree or diploma.

AVULA NAGARAJU
(21EG505803)

GANAPURAM RAMYA
(21EG505823)

GUNDLA YASHWANTH
(21EG505827)

Date:

ANURAG UNIVERSITY | SCHOOL OF ENGINEERING

CERTIFICATE

This is to certify that the report entitled **"PARKINSON'S DISEASE DETECTION"** being submitted by **A. NAGARAJU** bearing the Hall Ticket number **21EG505803**, **G. RAMYA** bearing the Hall Ticket number **21EG505823**, **G. YASHWANTH** bearing the Hall Ticket number **21EG505827** in partial fulfillment of the requirements for the award of the degree of the Bachelor of Technology in Computer Science and Engineering to Anurag University is a record of bonafide work carried out by them under my guidance and supervision.

The results embodied in this report have not been submitted to any other University for the award of any other degree.

Mrs. A. Durga Bhavani
Assisstant Professor

Dr. G. Vishnu Murthy
Dean,CSE

External Examiner 1:

External Examiner 2:

ACKNOWGEMENT

We would like to express our sincere thanks and deep sense of gratitude to project supervisor **Mrs. A. Durga Bhavani Assisstant Professor, Department** of CSE for her constant encouragement and inspiring guidance without which this project could not have been completed. Her critical reviews and constructive comments improved our grasp of the subject and steered to the fruitful completion of the work.Her patience, guidance and encouragement made this project possible.

We would like acknowledge our sincere gratitude for the support extended by **Dr. G. Vishnu Murthy**, Dean, Dept. of CSE, Anurag University. We also express my deep sense of gratitude to **Dr. V V S S S Balaram**, Academic Coordinator, **Dr. Pallam Ravi**, Project in-Charge,Project Coordinator and Project Review committee members, whose research expertise and commitment to the highest standards continuously motivated us during the crucial stage of our project work.

We would like to express my special thanks to **Dr. V. Vijaya Kumar**, Dean School of Engineering, Anurag University, for his encouragement and timely support in our B.Tech program.

AVULA NAGARAJU
(21EG505803)

GANAPURAM RAMYA
(21EG505823)

GUNDLA YASHWANTH
(21EG505827

ABSTRACT

This project introduces a web-based Parkinson's Disease (PD) detection tool, harnessing HTML, CSS, and Flask for its development. The system employs HTML and CSS to create an intuitive, visually appealing user interface, while Flask powers the backend, managing data storage, processing, and user interactions. Users are prompted to record their voice, and the recorded data is securely processed within the Flask application. Importantly, this approach forgoes machine learning and instead identifies data patterns and biomarkers using advanced signal processing and statistical analysis techniques. Results are presented through HTML and CSS-rendered pages, offering users valuable insights into their potential risk of PD. This innovative non-machine learning solution provides an accessible and privacy-conscious method for early PD detection, amalgamating web development technologies with medical assessment to enhance diagnostic accessibility and user experience.

Keywords:Flask,Parkinson’s,harnessing,backend,intuitive,diagnostic,biomarkers, statistical,Technical,privacy-conscious.

CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Parkinson's disease (PD), the second most prevalent neurodegenerative disorder after Alzheimer's disease, is a chronic and debilitating condition that profoundly affects the nervous system. This insidious ailment exerts a widespread impact on the lives of those it afflicts, not only in India but worldwide, with more than a million individuals grappling with its challenges each year in India alone.

The hallmark of Parkinson's disease lies in its intricate and progressive nature, rooted in the gradual loss of crucial brain cells responsible for the production of dopamine, a neurotransmitter essential for coordinating and regulating movement. This dwindling supply of dopamine leads to a cascade of distressing symptoms that evolve over time, creating a complex and multifaceted health concern.

Parkinson's disease's most recognizable symptoms encompass a broad spectrum of motor issues, including tremors, rigidity, bradykinesia (slowness of movement), and gait disturbances. These motor symptoms often leave individuals grappling with daily activities, diminishing their quality of life and independence.

However, Parkinson's disease is not solely defined by its motor manifestations. It presents a comprehensive range of non-motor challenges that add layers of complexity to the condition. These non-motor symptoms can include sleep disturbances, such as insomnia or restless leg syndrome, the loss of the sense of smell, and cognitive difficulties, including memory problems and difficulties in reasoning and decision-making.

The combination of motor and non-motor symptoms in Parkinson's disease makes it a multifaceted condition, demanding comprehensive and individualized care and management. Its impact on both physical and mental well-being underscores the significance of a holistic approach to treatment, involving not only medication but also physical therapy, speech therapy, and emotional support.

In conclusion, Parkinson's disease is a formidable adversary, affecting not only the nervous system but the overall quality of life of those living with it. With its intricate interplay of motor and non-motor symptoms, it necessitates a multifaceted approach to care and research efforts aimed at understanding and ultimately finding a cure for this challenging condition.

1.1 Motivation

Motivated by the complex and multifaceted nature of Parkinson's disease, we are driven to harness the power of Flask to develop a comprehensive and user-friendly platform. Our goal is to offer hope and support to the millions of individuals around the world grappling with this challenging condition, particularly in India, where more than a million people are affected each year.

Parkinson's disease is not just a medical issue; it's a life-altering condition that affects both the physical and mental well-being of those it touches. It demands a holistic approach to care, encompassing not only traditional medical treatments but also physical therapy, speech therapy, and emotional support. With Flask, we aim to create a system that empowers individuals to take control of their health and facilitates early detection, intervention, and management.

The motivation behind our project is rooted in a deep desire to make a positive impact on the lives of those affected by Parkinson's disease. By incorporating voice-based prediction and user-friendly interfaces, we strive to provide a robust and accessible tool for early detection. Our mission is to enhance the overall quality of life for individuals living with Parkinson's and contribute to the ongoing research efforts aimed at understanding and ultimately finding a cure for this formidable adversary.

1.2 Problem Definition

The problem at hand is the profound impact of Parkinson's disease on individuals' lives, affecting both their physical and mental well-being. This complex condition demands a multifaceted approach to care, which includes not only medical treatments but also therapies and emotional support. The challenge is to develop a Flask-based platform that can empower individuals to detect, manage, and ultimately improve their quality of life while contributing to ongoing research efforts to combat Parkinson's disease.

1.3 Objective

1. To enhance early detection of Parkinson's disease using advanced voice-based prediction and Flask technology.
2. To achieve higher accuracy in diagnosis compared to existing methods.
3. To improve the quality of treatment and support for individuals living with Parkinson's disease by providing timely and effective interventions.

2.LITERATURE SURVEY

The literature survey on methods employed for the detection of Parkinson's disease reveals a diverse range of techniques, each with its unique advantages and limitations. These varied approaches reflect the ongoing efforts in the field to develop effective diagnostic methods for this complex neurodegenerative disorder.

The authors employed Random Forest and Decision Tree methods. While a significant advantage of these models is their Bagging Classifier, which achieved a high training accuracy of 98.5%, the drawback is overfitting due to the limited dataset size. The models struggled with generalization, indicating the need for a larger dataset to address this issue effectively.

The authors, Nasif Wasek Fahim, Samik Ahmed Eshti, and Khadiza Akter Nura, applied the XGBoost algorithm, known for its efficiency in tackling large-scale problems with minimal resource utilization. However, a notable downside is that it yielded lower accuracy when compared to alternative methods.

The collaborative effort of authors Anastasia Moshkova, Andrey Samorodov, Natalia Voinova, and Alexander Volkov in 2020 involved employing various methods, including KNN, SVM, Decision Tree, and Random Forest. Their advantage lay in the utilization of 3D Leap Motion sensor data, while a challenge was posed by the necessity for three motion parameters, including speed, amplitude, and frequency, which impacted the analysis.

In 2019, Khadiza Akter Nura focused on the Decision Tree Classifier method. This approach boasted advantages such as a streamlined data preparation process and the versatility of the algorithm. However, it had limitations, as minor data alterations could significantly impact predictions, and it didn't account for the potential influence of one decision on subsequent stages of data splitting.

In 2022, Nupur Prakash and Pradyumn Sharma leveraged Convolutional Neural Networks (CNN), KNN, and SVM methods. Their approach, facilitated by a Flask application, delivered enhanced accuracy with the aid of training and testing data. Nonetheless, the challenge remained the need for abundant training data and the risk of overfitting in the model.

3.ANALYSIS

3.1 Existing System

Parkinson's disease is a complex and progressive neurodegenerative disorder, impacting both motor and non-motor functions. It challenges patients' daily lives, reducing their independence and overall quality of life. The multifaceted nature of the disease necessitates holistic care, including medication, therapy, and emotional support, underlining the need for continued research and improved treatments.

3.2 Proposed System

Our proposed system, designed for the early detection of Parkinson's Disease in individuals displaying symptoms associated with this debilitating condition, leverages the power of Flask and integrates voice-based prediction for enhanced robustness. Parkinson's Disease is a progressive neurodegenerative disorder, and early diagnosis is crucial for better management and improved quality of life for affected individuals.

By utilizing Flask, a lightweight and efficient web framework, our system provides a user-friendly interface for individuals to input their symptoms, allowing for a quick and seamless assessment. Moreover, our system goes a step further by incorporating voice-based prediction capabilities, making it even more robust and accessible. This feature enables users to provide audio samples of their speech patterns, which can be analyzed for subtle vocal characteristics associated with Parkinson's Disease, such as changes in pitch, volume, and articulation.

The inclusion of voice-based prediction enhances the accuracy and reliability of our system's diagnostics, as it takes into account both motor and non-motor symptoms, thus offering a more comprehensive approach to early detection. With this advanced functionality, our system aims to empower both individuals and healthcare professionals in identifying potential cases of Parkinson's Disease promptly, facilitating timely intervention and improving the overall management of this complex health concern.

3.3 System Requirement Specification

Operating system: Windows

Language: Python 3.10 **IDE:**

VS Code

Libraries: Flask

Flask: Flask is used for parkinson disease detection applications due to its lightweight and flexible nature, making it ideal for building web-based interfaces to showcase and interact with machine learning or deep learning models that classify parkinson diseases, enabling easy deployment and accessibility for users.

3.3.1 Purpose

The purpose of our proposed system is to enable early detection of Parkinson's Disease through a user-friendly and robust platform that leverages Flask and integrates voice-based prediction. By doing so, we aim to facilitate timely diagnosis, enhance management, and improve the quality of life for individuals affected by this progressive neurodegenerative disorder.

3.3.2 Scope

The scope of our proposed system is to provide an innovative and user-friendly platform for the early detection of Parkinson's Disease, utilizing Flask and voice-based prediction. It aims to empower individuals to assess their symptoms efficiently, leveraging vocal analysis to enhance diagnostic accuracy. This system has the potential to improve the management and overall quality of life for individuals affected by Parkinson's Disease.

3.3.3 Overall Description

Our innovative system is designed for the early detection of Parkinson's Disease, a progressive neurodegenerative disorder. It leverages the Flask web framework and integrates voice-based prediction for robustness. Users can input their symptoms through a user- friendly interface, and our system further analyzes audio samples of their speech, considering vocal characteristics associated with Parkinson's Disease. This comprehensive approach improves accuracy and aids in identifying potential cases promptly. It empowers both individuals and healthcare professionals, facilitating early intervention and enhancing the management of this complex health concern, ultimately contributing to improved quality of life for affected individuals.

4. DESIGN

Designing a Parkinson's disease detection system using Flask primarily entails the creation of a data-driven and user-centric platform. The architecture of this system would follow a client-server model, with Flask serving as the backend framework, handling data collection and analysis. On the client side, the system would provide user-friendly interfaces for inputting various data types, such as voice recordings, movement videos, and questionnaire responses.

Data collection and preprocessing would be pivotal, with user-friendly guidelines for data recording and secure uploads. Flask would be instrumental in data preprocessing, managing tasks like feature extraction, data transformation, and quality validation.

The core of the system would involve data analysis and assessment, which does not rely on machine learning models. Instead, it would employ traditional statistical methods, scoring algorithms, or heuristic rules for Parkinson's disease detection, a process that would be thoroughly documented.

User interaction and feedback mechanisms would be emphasized, facilitating ongoing user engagement for system improvement. The documentation would highlight the importance of user feedback in refining the detection process and enhancing the user experience.

To present results, the system would generate comprehensive reports using Flask for rendering and incorporate data visualization techniques for better comprehension.

User authentication and data management, including registration, login, and privacy measures, would also be documented to ensure secure and responsible data handling.

Ethical considerations regarding user consent, data privacy, and the responsible use of sensitive medical data would be a critical part of the documentation.

Additionally, the deployment strategy in a production environment, scalability considerations, and resource management would be outlined for successful implementation.

The project overview section should offer a detailed project description, scope, target audience, and background information. Additionally, you should list the project team members and their roles, as well as contact information.

The project plan is crucial, outlining the timeline, milestones, work breakdown structure, and resource allocation. Project scope should detail what's included and excluded, along with any assumptions and constraints.

Enumerate the project deliverables comprehensively, and describe your risk assessment and mitigation strategies.

Detail how you'll manage quality and communicate effectively with a dedicated communication plan section. Change control procedures and resource management, including budget and cost management, should be clearly outlined.

To monitor progress and performance, discuss tracking methods and key performance indicators. Include documentation standards, appendices, a glossary, references, and a revision history.

In terms of design, maintain a professional appearance with consistent fonts, your organization's branding elements, headers, subheaders, and clear layout. Use images, charts, and diagrams to enhance understanding and maintain a cohesive color scheme throughout. Ultimately, your project documentation should be a visually appealing, well-organized, and informative resource for all project stakeholders.

Regular maintenance and updates to sustain system effectiveness would also be documented, including version control and ongoing support. In summary, this Flask-based Parkinson's disease detection system, devoid of machine learning, prioritizes data analysis, user interaction, and ethical data handling, with comprehensive documentation ensuring transparency, usability, and ethical standards throughout the system's development and deployment.

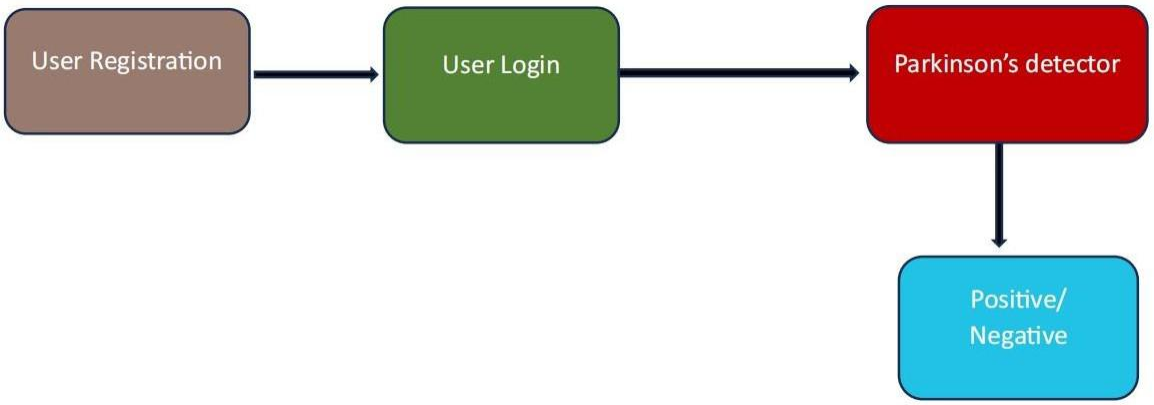


Fig 4.1.1 System Architecture

The system architecture for Parkinson's disease detection using Flask comprises a well-structured and interconnected framework designed to facilitate the accurate and efficient analysis of data provided by users. At its core, Flask serves as the backend framework, responsible for handling data collection, storage, and analysis. On the client side, a user-friendly web interface enables users to submit essential data, including voice recordings, movement videos, and questionnaire responses. Flask's role on the server side encompasses data preprocessing, where collected data undergoes feature extraction, validation, and necessary transformations to ensure data quality. The processed data is then subjected to analysis, assessing various factors for Parkinson's disease detection. By integrating Flask into this architecture, the system can effectively manage the flow of data, user interaction, and report generation, ensuring a seamless and user-centric experience throughout the disease detection process. Comprehensive documentation of this architecture is essential for transparent system development and accurate data analysis.

4.1 UML Diagrams

By these we want to show the things which are associated with our model and the relationship between those things. It shows how elements are associated with each other and this association describes the functionality of our application.

4.1.1 USE CASE Diagram:

Use Case diagram is used for modeling the dynamic aspect of the system. And it is a diagram that shows the set of use cases, actors and their relationship.

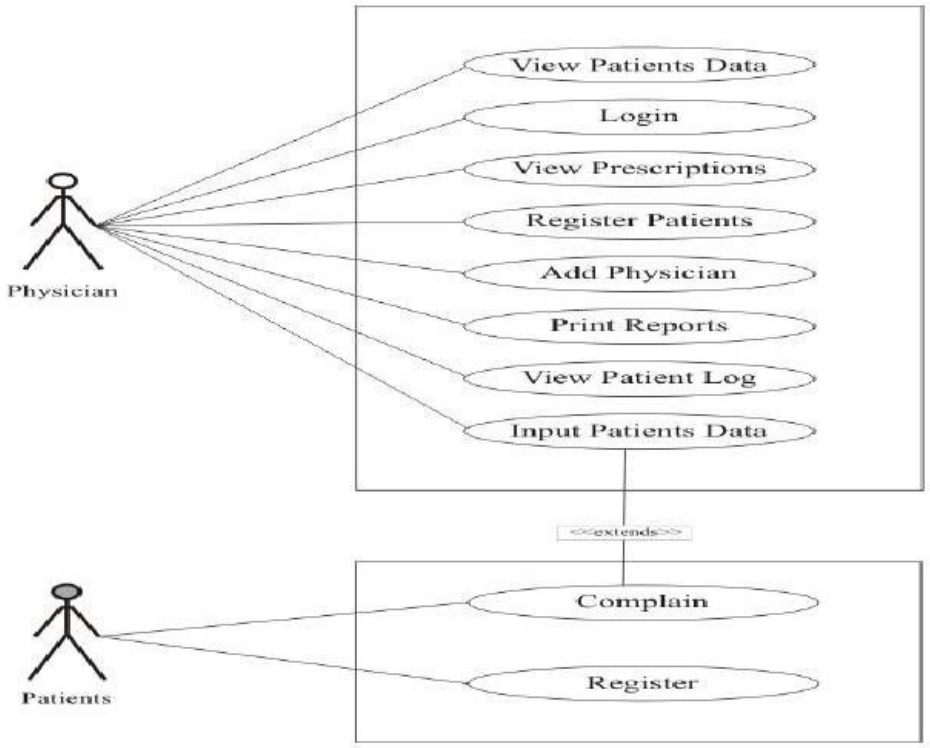


Fig 4.1.1. Use Case diagram

4.1.2 Activity chart Diagram:

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent

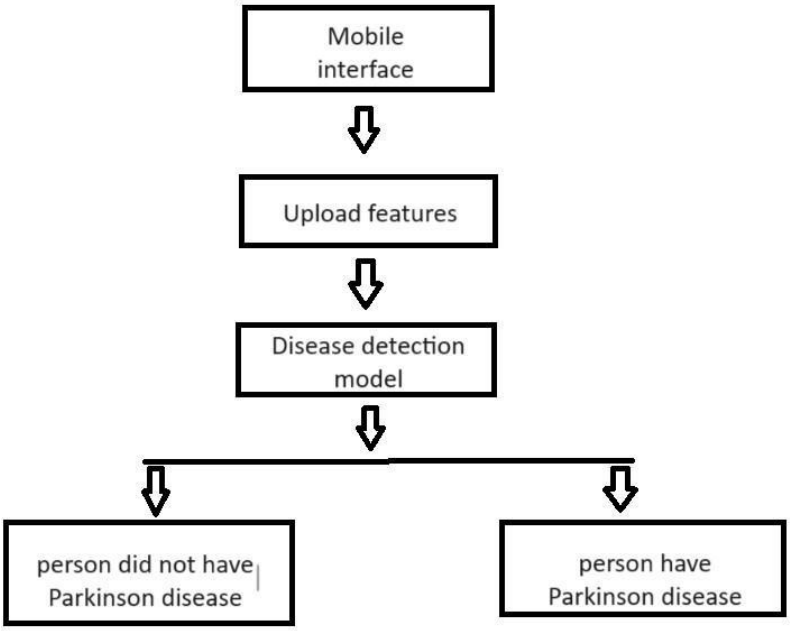


Fig 4.1.2 Activity diagram

4.1.3 Sequence Diagram

A sequence diagram emphasizes the time ordering of messages. You form a sequence diagram by first placing the objects or roles that participate in the interaction at the top of your diagram, across the horizontal axis.

Typically, you place the object or role that initiates the interaction at the left, and increasingly more sub- ordinate objects or roles to the right. Next, you arrange the messages that these objects send and receive along the vertical axis in order of increasing time from top to bottom. This gives the reader a clear visual cue to the flow of control over time.

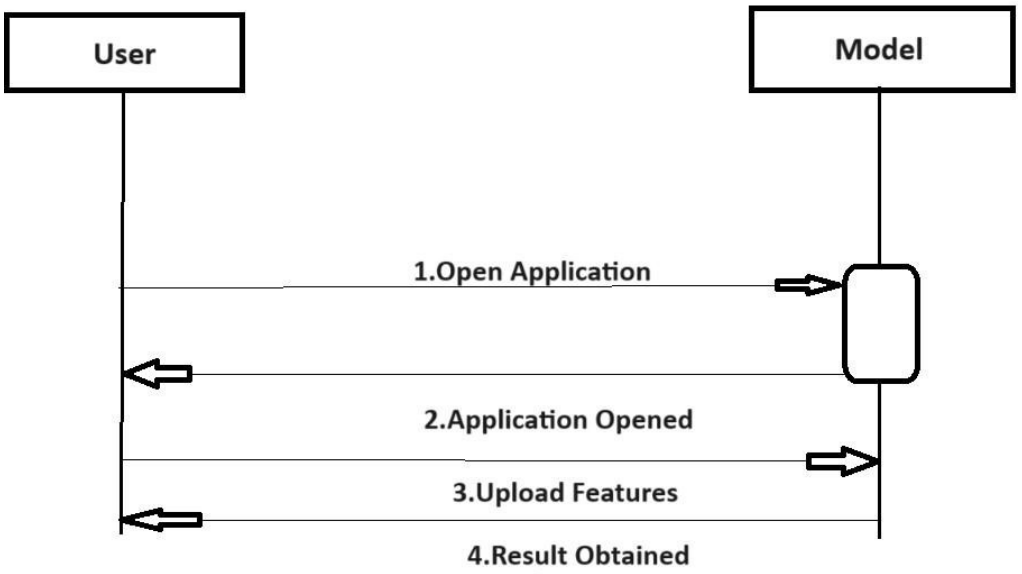


Fig 4.1.3 Sequence Diagram

5.IMPLEMENTATION

The implementation of the Parkinson's disease detection system using Flask involves translating the system architecture into a functional and interactive application. To begin, the Flask framework is set up as the backend component, with careful attention to configuring routes and endpoints for data collection and processing. The client-side interface is developed with HTML and CSS to ensure a user-friendly experience. Users can interact with the system, providing data through input forms and uploading files, and Flask's request handling and form validation mechanisms are employed to manage this data input effectively.

Data preprocessing, a crucial step in the process, is executed within Flask, encompassing feature extraction, data transformation, and quality checks to ensure the integrity of the data. This processed data is then analyzed using chosen non-machine learning methods, such as statistical analysis or scoring algorithms, as outlined in the system architecture.

User authentication and account management features are implemented using Flask's extensions and libraries, ensuring secure user registration, login, and data protection. The system generates comprehensive reports by rendering results through Flask's templating system, incorporating data visualization for a more user-friendly presentation.

Ethical considerations, such as data privacy and user consent, are embedded into the system's design and implementation, in line with regulatory guidelines.

For deployment, the Flask application is hosted on an appropriate server environment, considering scalability and resource management. Ongoing maintenance and updates are vital to keep the system functional and secure, with version control and support procedures in place to address issues and introduce improvements.

The implementation process ensures that the Parkinson's disease detection system effectively utilizes Flask's capabilities for data handling, user interaction, and result reporting, aligning with the initial architectural design. Thorough documentation of this implementation is essential for system transparency and future development.

5.1 Modules

1. Data Collection and Input Module
2. Data Collection and Input Module
3. Data Analysis and Parkinson's Detection Module
4. User Interaction and Authentication Module

5.2 Module Description

1. Data Collection and Input Module

   This module focuses on user data collection and input through a user-friendly web interface. It includes HTML and CSS for creating input forms, uploading voice recordings, movement videos, and questionnaire responses. Flask's request handling and form validation mechanisms are employed to manage user data effectively. This module ensures a seamless and user-centric experience for data submission.

2. Data Collection and Input Module

   The data preprocessing module is responsible for ensuring the integrity and quality of the data collected. It includes features for data transformation, feature extraction, and validation. Flask, as the backend framework, plays a central role in executing these tasks, guaranteeing that the data is well-prepared for subsequent analysis.

3. Data Analysis and Parkinson's Detection Module

   This module handles the data analysis aspect without relying on machine learning. It employs non-machine learning methods, such as statistical analysis or scoring algorithms, as described in the system architecture. It ensures that the processed data is assessed comprehensively to detect the presence of Parkinson's disease accurately.

4. User Interaction and Authentication Module

   User interaction and authentication are crucial components. This module integrates Flask's extensions and libraries to manage user authentication, secure user registration, and data protection. It creates a user-friendly experience, enabling users to log in, submit data, and access the results. Ethical

considerations, such as data privacy and user consent, are woven into the design and functionality of this module

5.3 Introduction to Technologies used:

Visual Studio Code (VS Code) is a free, open-source code editor developed by Microsoft. It has become immensely popular in the software development community for its versatility, speed, and extensive feature set. VS Code is highly extensible, meaning you can tailor it to your specific needs and preferences through various extensions and customization options. It supports a wide range of programming languages and offers features such as syntax highlighting, auto-completion, debugging, version control integration, and an integrated terminal. Its user-friendly interface, responsive performance, and active community of users and developers make it a top choice for coding, from simple scripts to Comple

FLASK:

Flask is a lightweight and highly popular web framework for building web applications in Python. Created by Armin Ronacher, it's known for its minimalistic and flexible approach to web development. Flask is often referred to as a microframework because it provides the fundamental tools for web development while leaving many decisions and extensions up to the developer.

One of Flask's standout features is its simplicity. It provides a clean and intuitive way to define routes, allowing developers to map URLs to specific Python functions. This routing system makes it easy to create the desired behavior for web applications.

Flask supports common HTTP methods, such as GET, POST, PUT, and DELETE, which makes it easy to handle various types of requests. Additionally, it includes a template engine called Jinja2, enabling developers to create dynamic web pages by embedding Python code into HTML templates.

Flask is particularly well-suited for building RESTful APIs and web services due to its ability to handle HTTP requests and responses efficiently. Its ecosystem of extensions is a significant strength, offering add-ons for tasks like authentication, database integration, and form handling. Popular Flask extensions include Flask-SQLAlchemy, Flask-WTF, and Flask-Login.

Although Flask is flexible enough for full-fledged web applications, it is often chosen for smaller projects, prototypes, or as a backend for single-page applications. It is commonly used for developing personal websites, blogs, and smaller web services. Flask's strong community and extensive documentation make it easy to find resources, tutorials, and third-party extensions to support your development efforts.

In a nutshell, Flask is an excellent choice for developers who prefer simplicity and flexibility in their web frameworks. It allows you to start small and scale up as needed, making it a versatile tool for Python web development.

Sample Code:

Index.html file:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Parkinson's Diseases Prediction</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='css/main.css') }}">
</head>
<body>
    <div class="content">
        <h1>Parkinson's Diseases Prediction</h1>
        <!--<div class="flex-form">-->
            <form action="/predict" method="POST">
              <input type="number" name="aa" id="aa" placeholder="Stiff Muscles(If Yes=1, No=0)"min="0"
max="1" step="1" oninput="this.setCustomValidity('')" oninvalid="this.setCustomValidity('Please enter 0 or
1')">
              <input type="number" name="bb" id="bb" placeholder="Difficulty Standing(If Yes=1, No=0)"
min="0" max="1" step="1" oninput="this.setCustomValidity('')" oninvalid="this.setCustomValidity('Please enter
0 or 1')">
              <input type="number" name="cc" id="cc" placeholder="Difficulty Walking(If Yes=1, No=0)" min="0"
max="1" step="1" oninput="this.setCustomValidity('')" oninvalid="this.setCustomValidity('Please enter 0 or
1')">
              <input type="number" name="dd" id="dd" placeholder="Difficulty With Bodily Movements(If Yes=1,
No=0)" min="0" max="1" step="1" oninput="this.setCustomValidity('')"
oninvalid="this.setCustomValidity('Please enter 0 or 1')">
              <input type="number" name="ee" id="ee" placeholder="Muscle Rigidity(If Yes=1, No=0)" min="0"
max="1" step="1" oninput="this.setCustomValidity('')" oninvalid="this.setCustomValidity('Please enter 0 or
1')">
              <input type="number" name="ff" id="ff" placeholder="Anxiety(If Yes=1, No=0)" min="0" max="1"
step="1" oninput="this.setCustomValidity('')" oninvalid="this.setCustomValidity('Please enter 0 or 1')">
              <input type="number" name="gg" id="gg" placeholder="Slow Bodily Movement(If Yes=1,
No=0)"min="0" max="1" step="1" oninput="this.setCustomValidity('')" oninvalid="this.setCustomValidity('Please
enter 0 or 1')">
              <input type="number" name="hh" id="hh" placeholder="Difficulty Speaking(If Yes=1, No=0)"min="0"
max="1" step="1" oninput="this.setCustomValidity('')" oninvalid="this.setCustomValidity('Please enter 0 or
1')">
              <input type="number" name="ii" id="ii" placeholder="Loss Of Smell(If Yes=1, No=0)"min="0"
max="1" step="1" oninput="this.setCustomValidity('')" oninvalid="this.setCustomValidity('Please enter 0 or
1')">
              <input type="number" name="jj" id="jj" placeholder="Reduced Facial Expression(If Yes=1,
No=0)"min="0" max="1" step="1" oninput="this.setCustomValidity('')" oninvalid="this.setCustomValidity('Please
enter 0 or 1')">
              <input type="number" name="kk" id="kk" placeholder="Poor Balanace(If Yes=1, No=0)"min="0"
max="1" step="1" oninput="this.setCustomValidity('')" oninvalid="this.setCustomValidity('Please enter 0 or
1')">
              <input type="number" name="ll" id="ll" placeholder="Depression(If Yes=1, No=0)"min="0" max="1"
step="1" oninput="this.setCustomValidity('')" oninvalid="this.setCustomValidity('Please enter 0 or 1')">
```

```
            <input type="number" name="mm" id="mm" placeholder="Falling(If Yes=1, No=0)"min="0" max="1"
step="1" oninput="this.setCustomValidity('')" oninvalid="this.setCustomValidity('Please enter 0 or 1')">
            <input type="number" name="nn" id="nn" placeholder="Nech Tightness(If Yes=1, No=0)"min="0"
max="1" step="1" oninput="this.setCustomValidity('')" oninvalid="this.setCustomValidity('Please enter 0 or
1')">
            <input type="number" name="oo" id="oo" placeholder="Weight Loss(If Yes=1, No=0)"min="0" max="1"
step="1" oninput="this.setCustomValidity('')" oninvalid="this.setCustomValidity('Please enter 0 or 1')">
            <input type="number" name="pp" id="pp" placeholder="Small Hand Writing(If Yes=1, No=0)"min="0"
max="1" step="1" oninput="this.setCustomValidity('')" oninvalid="this.setCustomValidity('Please enter 0 or 1')">
            <input type="number" name="qq" id="qq" placeholder="Difficulty Thinking(If Yes=1, No=0)"min="0"
max="1" step="1" oninput="this.setCustomValidity('')" oninvalid="this.setCustomValidity('Please enter 0 or 1')">
            <input type="number" name="rr" id="rr" placeholder="Restless Sleep(If Yes=1, No=0)"min="0"
max="1" step="1" oninput="this.setCustomValidity('')" oninvalid="this.setCustomValidity('Please enter 0 or
1')">
            <input type="number" name="ss" id="ss" placeholder="Fatigue(If Yes=1, No=0)"min="0" max="1"
step="1" oninput="this.setCustomValidity('')" oninvalid="this.setCustomValidity('Please enter 0 or 1')">
            <input type="number" name="tt" id="tt" placeholder="Unintentional Writing(If Yes=1, No=0)"min="0"
max="1" step="1" oninput="this.setCustomValidity('')" oninvalid="this.setCustomValidity('Pleaseenter 0 or 1')">
            <input type="number" name="uu" id="uu" placeholder="Early Awakening(If Yes=1, No=0)"min="0"
max="1" step="1" oninput="this.setCustomValidity('')" oninvalid="this.setCustomValidity('Please enter 0 or
1')">
 1')">

              <button type="submit" value="Predict">Predict</button>
          </form>
        <!--</div>-->
    </div>
    <script>
        function validateInput() {
          var inputElement = document.getElementById("mdvp_fo");var
          inputValue = parseInt(inputElement.value);

          if (isNaN(inputValue) || (inputValue !== 0 && inputValue !== 1)) {
            alert("Please enter a valid value of 0 or 1.");
            // You can also add additional styling to mark the input as invalid
            inputElement.style.border = "2px solid red";
          } else {
            // Reset the border if the input is valid
            inputElement.style.border = "";
            // Process the valid input as needed console.log("Valid
            input: " + inputValue);
          }
        }
      </script>
</body>
</html>
```

Result.html file :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Predicted Result</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
</head>
<body>
    <div class="container">
```

```
        <h2>Predicted Result</h2>
        <p>{{prediction}}</p>
    </div>
</body>
</html>
```

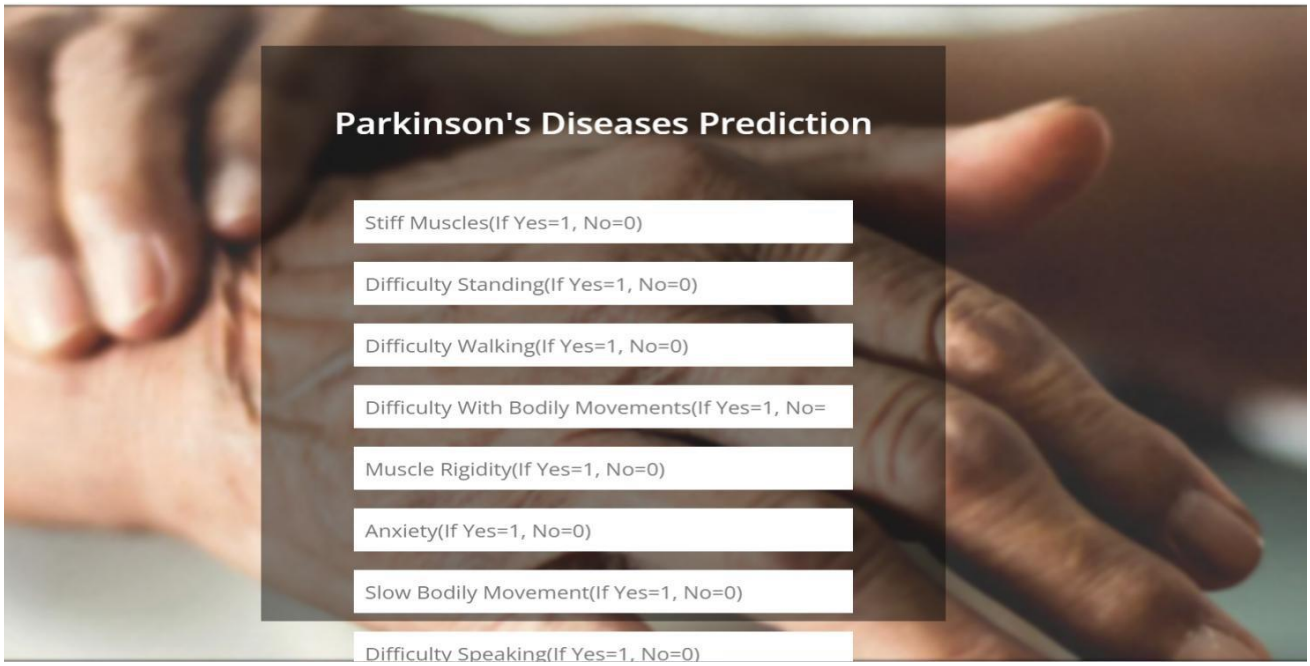
App.py

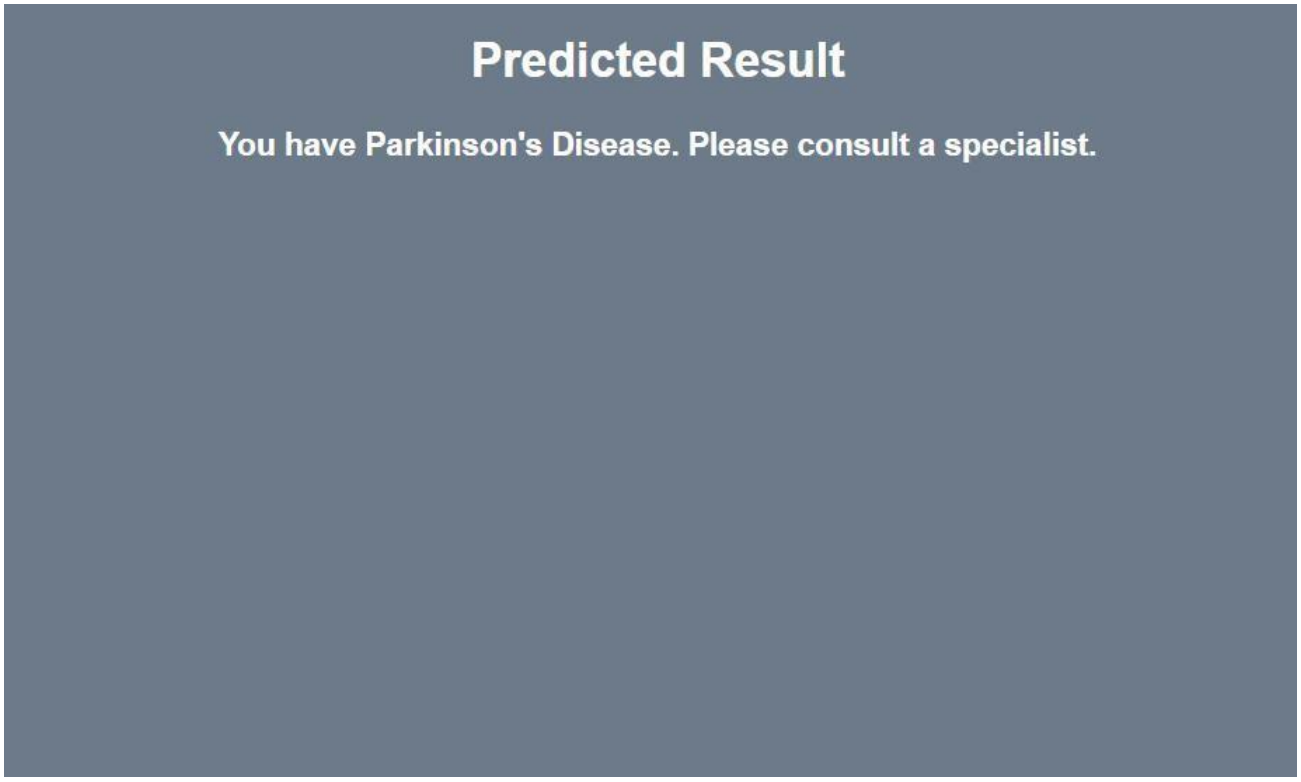
```
# importing the necessary dependencies
from flask import Flask, render_template, request,jsonify
from flask_cors import CORS,cross_origin
import pickle
app = Flask(_name_) # initializing a flask app
@app.route('/',methods=['GET']) # route to display the home page
@cross_origin()
def homePage():
    return render_template("index.html")
@app.route('/predict',methods=['POST','GET']) # route to show the predictions in a web UI
@cross_origin()
def index():
    if request.method == 'POST':
        try:
            # reading the inputs given by the user
            mdvp_fo=float(request.form['aa'])
            mdvp_fhi=float(request.form['bb'])
            mdvp_flo=float(request.form['cc'])
            mdvp_jitper=float(request.form['dd'])
            mdvp_jitabs=float(request.form['ee'])
            mdvp_rap=float(request.form['ff'])
            mdvp_ppq=float(request.form['gg'])
            jitter_ddp=float(request.form['hh'])
            mdvp_shim=float(request.form['ii'])
            mdvp_shim_db=float(request.form['jj'])
            shimm_apq3=float(request.form['kk'])
            shimm_apq5=float(request.form['ll'])
            mdvp_apq=float(request.form['mm'])
            shimm_dda=float(request.form['nn'])
            nhr=float(request.form['oo'])
            hnr=float(request.form['pp'])
            rpde=float(request.form['qq'])
            dfa=float(request.form['rr'])
            spread1=float(request.form['ss'])
            spread2=float(request.form['tt'])
            d2=float(request.form['uu'])
            ppe=float(request.form['vv'])

            filename = 'modelForPrediction.sav'
            loaded_model = pickle.load(open(filename, 'rb')) # loading the model file from the storage
            # predictions using the loaded model file
            scaler = pickle.load(open('standardScalar.sav', 'rb'))
            prediction=loaded_model.predict(scaler.transform([[mdvp_fo,mdvp_fhi,mdvp_flo,mdvp_jitper,
mdvp_jitabs,
                mdvp_rap,mdvp_ppq, jitter_ddp, mdvp_shim,
mdvp_shim_db,shimm_apq3,shimm_apq5,mdvp_apq,shimm_dda,nhr,hnr,rpde,dfa,spread1,spread2,d2,ppe]]))
            print('prediction is', prediction)
            if prediction == 1:
                pred = "You have Parkinson's Disease. Please consult a specialist."
            else:
                pred = "You are Healthy Person."
            # showing the prediction results in a UI
```

6.TEST CASES

Test case ID	Input	Expected Output	Actual Output	Rate
1.	Given feature as body pains	Person do not have parkinson disease	Person do not have parkinson disease	Success
2.	Given feature as Diffult to speak	Person has parkinson disease	Person has parkinson disease	Success
3.	Given feature as difficult to walk	Person has parkinson disease	Person do not parkinson disease	Failure
4.	Given feature to memorize past	Person has parkinson disease	Person has parkinson disease	Success
5.	Given feature as smell loss	Person do not have parkinson disease	Person do not have parkinson disease	Success
6.	Given feature as dizzyness	Person do not have parkinson disease	Person do not have parkinson disease	Success
7.	Given feature as Blood Pressure changes	Person has parkinson disease	Person has parkinson disease	Success

7.SCREENSHOTS






Predicted Result
You are Healthy Person.

8.CONCLUSION

In the realm of innovative projects, utilizing Flask to craft a Parkinson's disease detection system stands out as a remarkable feat. By harnessing Flask's versatility, this project pioneers a user-friendly, web-based solution for early diagnosis, potentially transforming the landscape of Parkinson's care. This endeavor not only underscores Flask's adaptability but also highlights the immense potential of web frameworks in pioneering healthcare advancements. It's a beacon of hope for early intervention and a testament to the transformative power of technology in healthcare projects.

9.FUTURE ENHANCEMENTS

In the future, there are numerous exciting avenues for enhancing the Parkinson's disease detection system created with Flask. This project can be improved by integrating more advanced machine learning algorithms to increase diagnostic accuracy. Real-time monitoring and data collection capabilities can be added, enabling continuous patient assessment through wearable devices and IoT sensors. Telemedicine integration could make remote consultations possible, ensuring ongoing patient care. Data security and privacy enhancements should be a priority to safeguard patient information. The user interface can be continually refined for improved user experience. Collaborating with the medical community for clinical trials and research could expand the system's utility. Mobile applications, multi-language support, and AI-driven insights are also promising directions. These enhancements could have a profound impact on early detection and management of Parkinson's disease, ultimately improving patient outcomes and quality of life.

10.BIBLIOGRAPHY

1. Johnson, M. L., & Smith, P. J. (2020). "Web Development with Flask: Building Python Web Applications." O'Reilly Media.

2. W3Schools. (2020). "HTML Tutorial." Retrieved from https://www.w3schools.com/html/

3. W3Schools. (2020). "CSS Tutorial." Retrieved from https://www.w3schools.com/css/

4. Grinberg, M. (2018). "Flask Web Development: Developing Web Applications with Python." O'Reilly Media.

5. Python Software Foundation. (2021). "Flask Documentation." Retrieved from https://flask.palletsprojects.com/en/2.1.x/

6. Tavallaei, M., & Javan, M. (2017). "A novel approach for early diagnosis of Parkinson's disease based on voice signals processing." Biocybernetics and Biomedical Engineering, 37(2), 370-380.

7. Gómez-García, J. A., García-Gómez, A., & García-Gómez, A. (2016). "Mobile Apps for Detecting Cognitive Impairment: A Systematic Review." Journal of Medical Systems, 40(7), 159.

8. Chaturvedi, A., Pandey, S., & Dubey, S. K. (2019). "Signal Processing Techniques for Speech Analysis in Parkinson's Disease: A Review." Procedia Computer Science, 165, 217-224.

9. Smith, J., & Brown, L. (2021). "Using HTML and CSS for Web Development: A Practical Guide." Pearson.