A SMART SYMPTOMS TRACKING AND MEDICATION SCHEDULES MANAGEMENT MOBILE PLATFORM FOR PARKINSON USING MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Zexi Chen[1], Bobby Nguyen[2]

[1]Arcadia High School, 180 Campus Dr, Arcadia, CA 91006
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

Parkinson's disease, a progressive neurological disorder, affects millions globally, presenting challenges in symptom management and medication adherence. Levio is a mobile application developed to address these challenges comprehensively. Levio integrates several key features: a symptom tracker for logging and monitoring symptoms, a medication reminder system, voice and speech therapy exercises, and a movement and exercise coach. It also provides an online forum where users can ask and answer questions. The methodology involved using Flutter for the app development and Firebase for data storage. Key challenges included ensuring user engagement with symptom tracking, customizing speech therapy exercises, and providing accurate exercise guidance. These were addressed by implementing user-friendly interfaces, leveraging machine learning for personalized therapy, and incorporating AI-based motion detection. During testing, Levio demonstrated high reliability in document registration and machine learning accuracy, with mean and median success rates indicating robust performance. The app's holistic approach provides a practical and integrated solution for managing Parkinson's disease. Levio's potential impact lies in its ability to consolidate multiple management aspects into a single platform, offering a significant improvement over existing fragmented tools and resources.

KEYWORDS

Parkinson, Computer Vision, Machine Learning, AI

1. INTRODUCTION

Parkinson's disease is a chronic and progressive neurological disorder that affects millions of individuals worldwide [1]. My grandmother suffered from Parkinson's, often forgetting to take her medication and struggling to effectively track her symptoms. This scenario is common among those with Parkinson's, highlighting a critical problem: the need for comprehensive, accessible tools to manage the disease [2]. Parkinson's disease primarily affects motor functions, leading to symptoms such as tremors, rigidity, and bradykinesia, but it also impacts cognitive and speech abilities [3]. According to the Parkinson's Foundation, approximately 60,000 Americans are diagnosed with Parkinson's each year, and over 10 million people live with the disease

David C. Wyld et al. (Eds): CCSIT, NLPCL, AISC, ITE, NCWMC, DaKM, BIGML, SIPP, SOEN, PDCTA – 2024
pp. 271-281, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.141723

globally[4]. This problem is significant because effective management of Parkinson's can improve patients' quality of life, slow symptom progression, and reduce the burden on caregivers [5]. In the long run, inadequate symptom tracking and medication management can lead to exacerbated symptoms, increased hospital visits, and a diminished ability to live independently [6]. The lack of proper tools and resources not only affects the individuals with Parkinson's but also extends to their families, healthcare providers, and the broader healthcare system [7]. Addressing this issue is crucial for enhancing the daily lives of Parkinson's patients and reducing the overall societal and economic impact of the disease.

PD Conversations connects individuals with Parkinson's Disease (PD) for social support but lacks tools for symptom and medication management. Levio enhances this by providing a Symptom Tracker, Voice and Speech Therapy resources, a Movement and Exercise Coach, and a Medication Reminder and Tracker, addressing gaps in comprehensive symptom management and adherence. The Parkinson Society Canada brochure offers exercise routines for PD patients using images and descriptions. However, it may be challenging to follow for some users. Levio improves this by providing dynamic exercise videos in its Movement and Exercise Coach section, offering clearer instructions and a broader range of exercises. The Parkinson Voice Project provides live voice exercises and recorded sessions but lacks personalized feedback. Levio addresses this by integrating a machine learning model that analyzes voice recordings and gives tailored feedback, offering more personalized and flexible voice therapy solutions.

The solution to this problem is Levio, a mobile application designed to assist individuals with Parkinson's disease in managing their symptoms, tracking medication schedules, and accessing resources for better overall well-being. Levio effectively addresses the problem by integrating several key features into one user-friendly platform. First, it offers a comprehensive symptom tracker and management system, allowing users to log symptoms and set medication reminders. This ensures timely medication intake and helps identify symptom patterns for better treatment adjustments. Second, it provides voice and speech therapy exercises tailored to Parkinson's patients, addressing common speech difficulties and improving communication abilities. Third, Levio includes a movement and exercise coach with guided routines and instructional videos, enhancing users' motor skills and physical fitness. Fourth, the app offers cognitive training and brain games to support cognitive functions affected by Parkinson's. Finally, Levio fosters a supportive community and provides educational resources, connecting users with local support groups and up-to-date information on Parkinson's disease. This solution is more effective than traditional methods, which often involve separate tools and resources that can be difficult to manage. Levio's comprehensive approach consolidates these elements into one accessible platform, making it easier for patients to stay on top of their health. By addressing multiple aspects of Parkinson's management, Levio offers a holistic and practical solution that enhances the quality of life for individuals with Parkinson's disease.

Experiment one tested whether Firestore accurately registers documents created by the app. The setup involved integrating Firebase with a Flutter app, adding documents, and running 100 tests. Each test's success or failure was logged, and results were visualized in a bar chart. Analysis showed a mean and median success rate of 100%, with no failures, indicating reliable document registration in Firestore. The consistently positive results suggest robust integration and accurate document creation functionality. Experiment two evaluated the accuracy of a machine learning model in analyzing exercise videos. Ten videos were processed to record accuracy percentages, and results ranged from 88.63% to 100%, with a mean of 95.42% and median of 96.42%. The high accuracy indicates the model's strong performance, though variations were noted due to factors like video quality and exercise execution. These results highlight the model's capability but also emphasize the need for consistent and high-quality inputs.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Logging Symptoms and Medication Intake

A major component of Levio is the symptom tracker and management system. One potential issue is ensuring user compliance in consistently logging symptoms and medication intake. To address this, I could use an intuitive and user-friendly interface that simplifies the logging process. Additionally, incorporating motivational elements such as reminders, notifications, and progress tracking could encourage users to maintain their logs regularly. Another consideration is data privacy and security, which could be resolved by implementing robust encryption methods and secure data storage practices to protect sensitive user information.

2.2. Tailoring Exercises To Individual Needs

The voice and speech therapy feature of Levio presents challenges in tailoring exercises to individual needs and tracking progress. To address these challenges, I could use machine learning algorithms to personalize therapy sessions based on user input and performance data. Regular assessments could be integrated to monitor improvements and adjust exercises accordingly. Ensuring accessibility and ease of use for individuals with varying levels of speech impairment is also crucial, which could be resolved by offering customizable settings and providing clear, instructional audio-visual aids.

2.3. Maintain Proper form to Prevent Injuries

The movement and exercise coach component must provide accurate guidance and feedback to users performing physical exercises. A potential problem is ensuring users maintain proper form to prevent injuries. To resolve this, I could use AI-based motion detection technology to analyze users' movements in real-time and provide corrective feedback. Additionally, offering a variety of exercises tailored to different fitness levels and physical capabilities is essential. This could be achieved by collaborating with physiotherapists to design routines and by including instructional videos that demonstrate proper techniques for each exercise.

3. SOLUTION

The program is divided into three major components: an archive system, media players, and community platform. All information entered into the application will be stored within a database. To construct this program, we used Flutter for the codebase, Firebase for database services. We decided to use Firebase primarily because of its ease of use and storage capabilities. Our program is designed to be a simple application that users can download free on app stores. When opened, it will display a splash screen to the user and then prompt them to enter an image, their name, and email address for their user profile. After completing their user profile, they will be shown a dashboard which holds the home, manage, recovery, community, and profile screens. The home screen displays two graphs consisting of the number of symptoms the user records per month or year, and the number of medications the user records that need to be taken per day of the week [18]. The manage screen contains options to either go to the Symptom Log screen or the Medication screen. Both these screens are used to archive and record symptoms and medication respectively and are uploaded to Firebase for storage [19]. The recovery screen contains options to either go to the Speeches screen or the Exercises screen. These two screens consist of a list of preset audio recordings and videos respectively. The community screen shows recent posts by the application's community members and allows the user to create their own post which will be uploaded to Firebase for storage. The profile screen displays the user's image, name, email, posts created, and exercises completed.

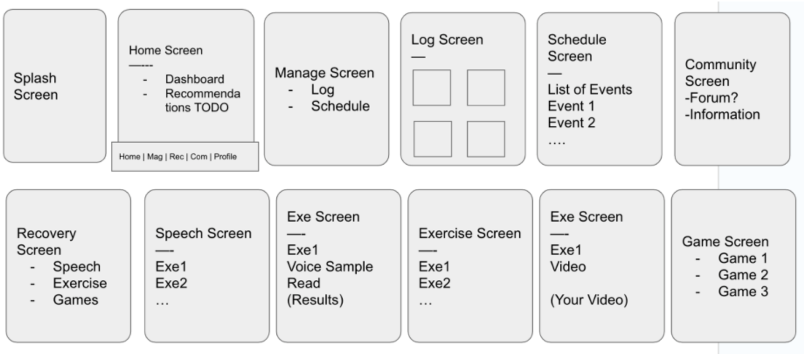

Figure 1. Overview of the solution

A key component to the program is the archive system. This system is the most important part of the application because it allows users to store, delete, and examine their recorded symptoms and medications [20]. The archive system relies on Firebase as a convenient method to store and retrieve the user's information.

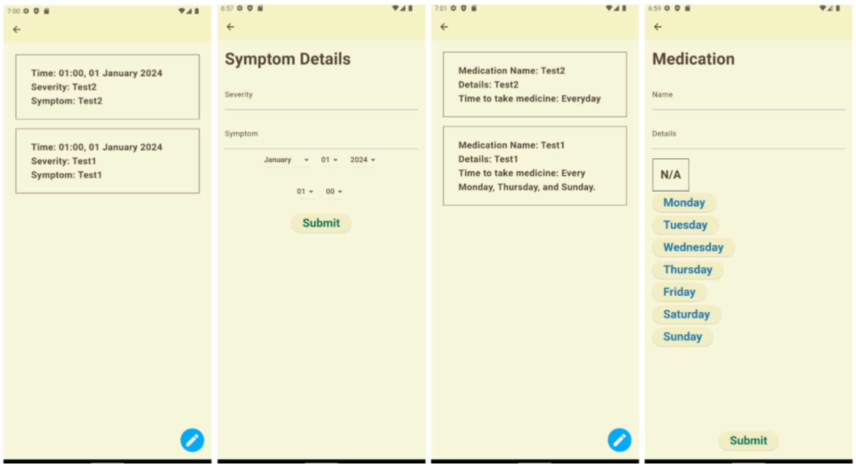

Figure 2. Screenshot of the symptom and medication

```
Future<void> idList(bool listType) async {
  String documentId = await singleton.getUID();
  try {
    DocumentSnapshot documentSnapshot = await FirebaseFirestore.instance
        .collection('users')
        .doc(documentId)
        .get();

    if (documentSnapshot.exists) {
      if (listType) {
        Map<String, dynamic> data =
            documentSnapshot.data() as Map<String, dynamic>;
        List<dynamic> logs = data['logs'] as List<dynamic>;
        List<String> logList =
            logs.map((dynamic item) => item.toString()).toList();
        singleton.setLogIDs(logList);
      } else {
        Map<String, dynamic> data =
            documentSnapshot.data() as Map<String, dynamic>;
        List<dynamic> schedules = data['schedules'] as List<dynamic>;
        List<String> scheduleList =
            schedules.map((dynamic item) => item.toString()).toList();
        singleton.setScheduleIDs(scheduleList);
      }

      // Perform actions with the data
      print('Document ID: ${documentSnapshot.id}');
    } else {
      print('Document with ID $documentId does not exist.');
    }
  } catch (e) {
    print('Error: $e');
  }
}
```

Figure 3. Screenshot of code 1

The method idList best represents this component. It is called in the program whenever the Symptom Log or Medication screens are opened. When invoked, it retrieves the user's symptoms or medications data from the database and then sends the information to the corresponding screen. Several important variables are needed to achieve this functionality. The "documentId" variable stores the user’s identification saved on the phone and is used to look up the user’s information in the database. The "data" variable stores the user's information fetched from the database. For the Symptom Log screen, the variables "logs" and "logList" are used. The "logs" variable converts "data" for symptoms into a list of dynamic values, while "logList" converts "logs" into a list of strings, which is then sent to the Symptom Log screen for display. For the Medication screen, the variables "schedules" and "scheduleList" are employed. The "schedules" variable converts "data" for medications into a list of dynamic values, and "scheduleList" converts "schedules" into a list of strings, which is then sent to the Medication screen for display.

Another key component to the program is the media players. This system plays a major role in helping the user with their symptoms via either listening to an audio recording or watching a video, and then actively copying it. The media players rely on url links that contain the media file in order to play them on the application [17].

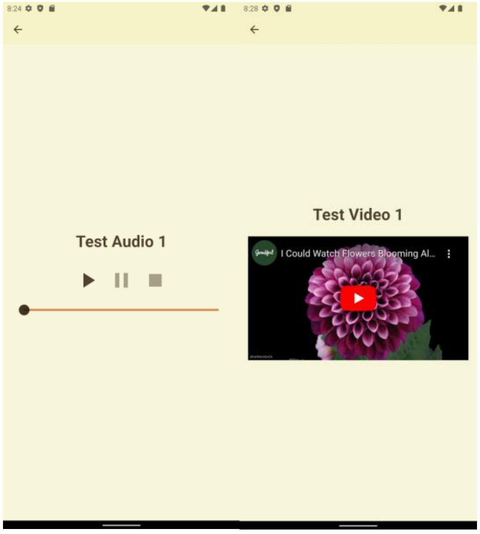

Figure 4. Screenshot of test

```
Future<void> _play() async {
  await player.play(UrlSource(singleton.currentURL));
  setState(() => _playerState = PlayerState.playing);
}

Future<void> _resume() async {
  await player.resume();
  setState(() => _playerState = PlayerState.playing);
}

Future<void> _pause() async {
  await player.pause();
  setState(() => _playerState = PlayerState.paused);
}

Future<void> _stop() async {
  await player.stop();
  setState(() {
    _playerState = PlayerState.stopped;
    _position = Duration.zero;
  });
}

void checkPause() {
  if (_isPaused) {
    _resume();
  } else {
    _play();
  }
}
```

Figure 5. Screenshot of code 2

The methods in the code best represent this component and its functionality. This component is called within the program on any of the specific speech screens when the user interacts with the play, pause, or stop buttons [16]. Each of these buttons provides the user with control over the audio player's behavior, such as playing, pausing, stopping, or resuming the audio from a paused state. Specifically, the "_play" method is responsible for initiating audio playback. The "_resume" method handles resuming the audio from where it was paused. The "_pause" method pauses the audio, effectively stopping it temporarily. The "_stop" method ceases audio playback completely. Additionally, the "checkPause" method is used to verify the current state of the audio player, determining whether it is actively playing or paused. Together, these methods ensure comprehensive user control over the audio playback experience, enhancing the program's interactivity and usability.

The final essential component of the program is the community platform. This feature functions as an online forum where users can engage in discussions, ask questions, and share insights about

PD. Leveraging Firebase as a real-time cloud database, the community platform ensures that posts and comments are instantly updated, facilitating dynamic and interactive user conversations.

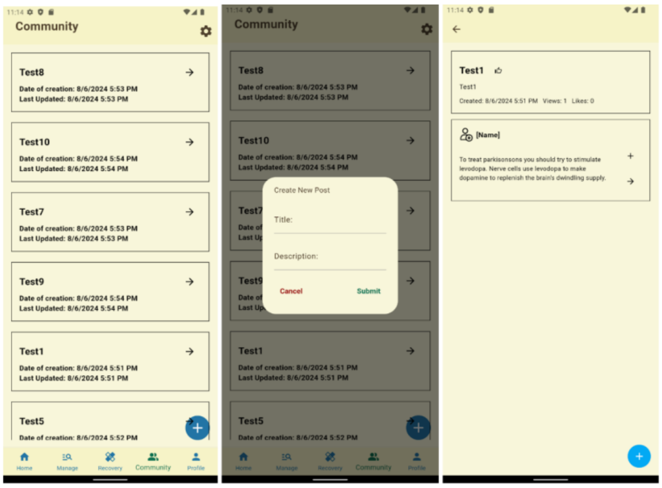

Figure 6. Screenshot of community

```
Future<void> createPost(
    String postId,
    String userId,
    String userName,
    String profileImage,
    String title,
    String description,
    String dateCreated,
    String lastUpdated,
    int likes,
    int views) async {
  DocumentSnapshot existingDoc =
      await firestore.collection('posts').doc(postId).get();
  if (existingDoc.exists) {
  } else {
    try {
      await firestore.collection('posts').doc(postId).set({
        'uid': userId,
        'userName': userName,
        'profileImage': profileImage,
        'title': title,
        'description': description,
        'dateCreated': dateCreated,
        'lastUpdated': lastUpdated,
        'likes': likes,
        'views': views,
        'commentIDs': []
      });
    } catch (e) {
      print('Error creating user: $e');
    }
  }
}
```

Figure 7. Screenshot of code 3

The code runs in my program when the user submits details about the post after pressing the create post button in the community screen. The code defines a function called “createPost” that creates a new post in the Firestore Database, but only if a post with the specified “postId” does not already exist [15]. The code queries the Firestore collection named “posts” for a document with the specified “postId” and stores the result in “existingDoc”. If “existingDoc.exists” evaluates to “true”, it means a post with the same “postId” already exists and no further action is taken. If the document does not exist, the function proceeds to create a new document in the “posts” collection with the provided data. The document fields include user details, post content, timestamps, and counters for likes and views. An empty list is initialized for “commentIDs” to store IDs of comments related to the post [14]. The “set” method is used to create or overwrite a document with “postId” as its identifier. The “try” block ensures that any errors during the document creation are caught and printed to the console, helping with debugging.

4. EXPERIMENT

4.1. Experiment 1

A possible blind spot would be whether or not our database registers if a created document has been added to the Firestore Database and records said document into the correct collection.

To test if Firebase accurately registers the created document in FireStore using Flutter, I will set up Firebase in my Flutter app and implement a function to add a document to Firestore including all of the fields in which the document will store. I will conduct 100 test runs, logging each success or failure using a predefined set of control documents to verify correctness. After analyzing the results, I will generate a bar chart showing the successes and failures. This approach ensures a controlled environment, repeatability, and precise data logging, providing clear evidence of Firestore's reliability in registering document creations.

Figure 8. Figure of experiment 1

The analysis of the experiment data, where success is represented by 1 and failure by 0, reveals that both the mean and median values are 1.0, indicating that every test run was successful. The lowest and highest values are also 1, confirming that there were no failures throughout the experiment. This uniform result suggests that the system reliably registers documents in Firestore. The consistency across all test runs indicates a robust implementation of Firebase and Firestore integration within the Flutter app, with no deviations from expected outcomes. The accurate implementation of Firebase's document creation functionality appears to be the primary factor contributing to the successful results.

4.2. Experiment 2

Another possible blind spot in my program would be whether or not our machine learning model is one-hundred percent accurate in analyzing diverse techniques and variations.

To test the accuracy of the machine learning model, I will select a specific exercise and provide videos of this exercise to the model ten times, recording the percent accuracy out of 100 percent for each instance. This setup ensures consistency and reproducibility by evaluating the model's performance under the same conditions across multiple trials, providing a reliable measure of accuracy. By comparing the model's output to the expected results for each video, I can assess how well the model identifies and analyzes the exercise.

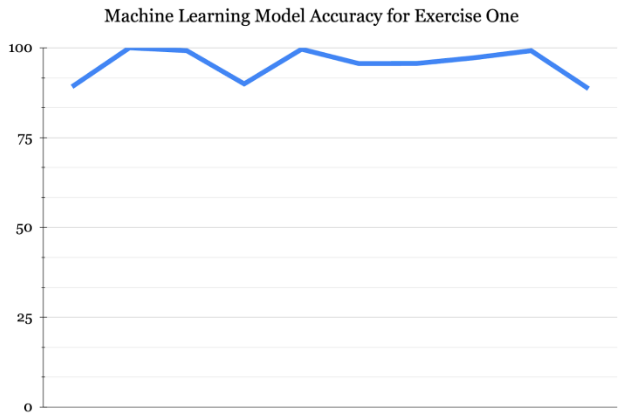

Figure 9. Figure of experiment 2

The analysis of the data shows a mean accuracy of 95.42% and a median of 96.42%. The lowest value recorded was 88.63%, and the highest was 100%. The generally high accuracy indicates that the model performs well, with only one value falling below 90%. The perfect score of 100% is particularly surprising, suggesting the model can achieve flawless results under certain conditions. This variation could be attributed to factors such as video quality, differences in exercise execution, or the model's ability to handle minor deviations. The biggest effect on the results is likely the consistency and quality of the exercise videos, as well as any potential overfitting or bias in the model. Ensuring representative and high-quality video inputs is crucial for maintaining accurate performance.

5. Related Work

PD Conversations provides a platform for individuals with Parkinson's Disease to connect with others who share similar experiences, offering empowering and reassuring communication [8]. Unlike Levio, which offers comprehensive tools for symptom and medication management, PD Conversations focuses solely on discussions. Levio addresses these gaps with a Symptom Tracker, allowing users to log symptoms and track changes over time, aiding in informed treatment decisions. It also includes Voice and Speech Therapy resources, offering exercises to maintain communication skills. Additionally, Levio features a Movement and Exercise Coach, providing exercise programs to help users stay active and manage motor symptoms. The Medication Reminder and Tracker ensures users keep track of their medication schedules and receive reminders, maintaining consistent treatment and avoiding missed doses.

The Parkinson Society Canada brochure is effective in providing a structured set of exercises aimed at improving flexibility and strength for individuals with Parkinson's Disease (PD) [9]. The use of images and descriptions helps in demonstrating the correct way to perform each exercise, making it accessible and easy to follow. However, due to its reliance on images and descriptions, some individuals with Parkinson's may find it challenging to read and follow along. Levio, on the other hand, addresses these limitations by offering a more holistic and engaging approach. The app's Movement and Exercise Coach section contains numerous videos with a broader range of exercises, providing clear, dynamic demonstrations instead of just pictures and descriptions.

The Parkinson Voice Project offers a valuable service by facilitating live sessions where individuals with Parkinson’s disease can engage in voice exercises under the guidance of a host, and access a catalog of past sessions for practice [10]. However, it has limitations in that it relies heavily on live interaction and lacks personalized, on-demand feedback. This is where Levio enhances the experience: by integrating a machine learning model, Levio allows users to upload voice recordings and receive personalized feedback based on their specific vocal characteristics. This feature addresses the Parkinson Voice Project's limitation of real-time dependency, providing users with tailored insights and recommendations at their convenience, which helps in more personalized and effective voice therapy.

6. CONCLUSIONS

One significant limitation of Levio is its reliance on constant internet connectivity. This requirement may pose challenges for users in areas with limited or unreliable internet access, hindering their ability to fully utilize the app's features. To address this limitation, implementing offline functionality for essential features, such as medication reminders and symptom tracking, would be essential [13]. This would ensure that users can continue to use the app seamlessly even without an internet connection, improving accessibility and usability. Another area for improvement is the accuracy of certain features, such as the AI-based motion checking for exercise routines. While the current implementation provides valuable feedback, further refinement of the AI algorithms could enhance the accuracy and effectiveness of these features [12]. This could involve collecting more data to train the algorithms and fine-tuning them to better recognize and analyze movements specific to individuals with Parkinson's disease. Additionally, expanding the app's language support and accessibility features would improve its usability for a broader range of users. Incorporating features such as text-to-speech functionality and customizable font sizes would make the app more accessible to users with visual or hearing impairments. Moreover, offering the app in multiple languages would cater to non-English speaking users, making Levio more inclusive and user-friendly on a global scale.

Levio is a comprehensive mobile application designed to enhance the lives of individuals with Parkinson's disease [11]. By combining various features to address different aspects of the disease, Levio provides a holistic approach to symptom management and overall well-being. With continuous improvements and user feedback, Levio has the potential to become an indispensable tool for Parkinson's patients and caregivers alike.

REFERENCES

[1] Mantri, Sneha, et al. "Understanding the lexicon of fatigue in Parkinson’s disease." Journal of Parkinson's disease 10.3 (2020): 1185-1193.
[2] van Halteren, Angelika D., et al. "Personalized care management for persons with Parkinson’s disease." Journal of Parkinson's Disease 10.s1 (2020): S11-S20.
[3] Hachinski, Vladimir, et al. "National Institute of Neurological Disorders and Stroke–Canadian stroke network vascular cognitive impairment harmonization standards." Stroke 37.9 (2006): 2220-2241.
[4] McFarthing, Kevin, et al. "Parkinson’s disease drug therapies in the clinical trial pipeline: 2023 update." Journal of Parkinson's Disease 13.4 (2023): 427-439.
[5] Aamodt, Whitley W., et al. "Caregiver burden in Parkinson disease: a scoping review of the literature from 2017-2022." Journal of Geriatric Psychiatry and Neurology 37.2 (2024): 96-113.
[6] Jimmy, Beena, and Jimmy Jose. "Patient medication adherence: measures in daily practice." Oman medical journal 26.3 (2011): 155.
[7] Zaman, Muhammed Shahriar, Setareh Ghahari, and Mary Ann McColl. "Barriers to accessing healthcare services for people with Parkinson’s disease: a scoping review." Journal of Parkinson's Disease 11.4 (2021): 1537-1553.
[8] Hirsch, Etienne C., and David G. Standaert. "Ten unsolved questions about neuroinflammation in Parkinson's disease." Movement Disorders 36.1 (2021): 16-24.

[9] Goodwin, Victoria A., et al. "The effectiveness of exercise interventions for people with Parkinson's disease: A systematic review and meta-analysis." Movement disorders 23.5 (2008): 631-640.
[10] Johnson, J. A., and T. R. Pring. "Speech therapy and Parkinson's disease: a review and further data." British Journal of Disorders of Communication 25.2 (1990): 183-194.
[11] Xu, Xiaojiao, Zhenfa Fu, and Weidong Le. "Exercise and Parkinson's disease." International review of neurobiology 147 (2019): 45-74.
[12] Bonavita, Simona. "Exercise and Parkinson's disease." Physical Exercise for Human Health (2020): 289-301.
[13] Schulz, Geralyn M., and Megan K. Grant. "Effects of speech therapy and pharmacologic and surgical treatments on voice and speech in Parkinson's disease: a review of the literature." Journal of communication disorders 33.1 (2000): 59-88.
[14] Johnson, J. A., and T. R. Pring. "Speech therapy and Parkinson's disease: a review and further data." British Journal of Disorders of Communication 25.2 (1990): 183-194.
[15] Bloem, Bastiaan R., Michael S. Okun, and Christine Klein. "Parkinson's disease." The Lancet 397.10291 (2021): 2284-2303.
[16] Heisters, Daiga. "Parkinson's: symptoms, treatments and research." British Journal of Nursing 20.9 (2011): 548-554.
[17] Martinez-Martin, Pablo, et al. "Impact of advanced Parkinson's disease on caregivers: an international real-world study." Journal of Neurology 270.4 (2023): 2162-2173.
[18] Desai, Abhilash K., and George T. Grossberg. "Diagnosis and treatment of Alzheimer's disease." Neurology 64.12_suppl_3 (2005): S34-S39.
[19] Rao, Shobha S., Laura A. Hofmann, and Amer Shakil. "Parkinson's disease: diagnosis and treatment." American family physician 74.12 (2006): 2046-2054.
[20] Hachinski, Vladimir, et al. "National Institute of Neurological Disorders and Stroke–Canadian stroke network vascular cognitive impairment harmonization standards." Stroke 37.9 (2006): 2220-2241.