**VOICE ENABLED PATIENT DOCUMENTATION AND**

**ASSISTANCE SYSTEM**

*Submitted in partial fulfillment of the requirements for the degree of*

Bachelor of Technology

In

Computer Science and Engineering

*By*

Saraswathi Bavadharini S 20BCE0339

Under the guidance of Prof. Manoov R

School of Computer Science and Engineering VIT, Vellore.



April 2024

**DECLARATION**

I hereby declare that the thesis entitled “VOICE ENABLED PATIENT DOCUMENTATION AND

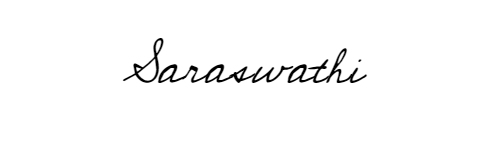
ASSISTANCE SYSTEM” submitted by me, for the award of the degree of *Bachelor of Technology in Programme* to VIT is a record of bonafide work carried out by me under the supervision of Prof. / Dr.

Manoov R.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Vellore

Date : 8/5/2024

****

**Signature of the Candidate**

**CERTIFICATE**

This is to certify that the thesis entitled “VOICE ENABLED PATIENT DOCUMENTATION AND

ASSISTANCE SYSTEM” submitted by

Saraswathi Bavadharini S; 20BCE0339, School of Computer Science and Engineering, VIT, for

the award of the degree of Bachelor of Technology in Programme, is a record of

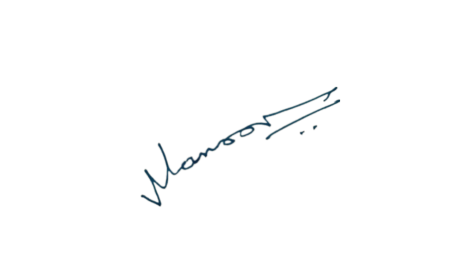
bonafide work carried out by her under my supervision during the period, 01.

12. 2023 to 30.04.2024, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted

either in part or in full, for the award of any other degree or diploma in this institute or

any other institute or university. The thesis fulfills the requirements and regulations of

the University and in my opinion meets the necessary standards for submission.

**Place : Vellore**

**Date : 8/5/24 Signature of the Guide**

**Internal Examiner External Examiner**

**Head of the Department**

**Programme**

**ACKNOWLEDGEMENTS**

I would like to take the opportunity to thank all the people who helped me. Firstly, I would like to express my sincere gratitude to our professor Manoov R, Assistant Professor (Senior) at Vellore Institute of Technology for their valuable guidance, continuous support, and understanding throughout the duration of the project. I am highly indebted for their constant supervision and their knowledge in regards to the field. Working with them was proved to be a very good opportunity for me.

I would also like to thank the teaching and non-teaching staff of Vellore Institute of Technology for their selfless enthusiasm and the environment they provided, which further prompted me to complete the project successfully.

Finally, I would also like to thank my family and friends for their constant support and help for the successful completion of the project.

Saraswathi Bavadharini S

**Executive Summary**

The project, titled "Doctor's Assistant (Nurse)," is a web-based application designed to streamline the management of patient information and documents for healthcare professionals. Leveraging technologies such as speech recognition, text summarization, and AI-powered response generation, the application offers efficient retrieval of patient details through spoken input and summarization of uploaded patient documents. Developed under the guidance of Dr. S. Anto and Associate Professor Professor Manoov R from Vellore Institute of Technology, the project aims to enhance the efficiency and productivity of healthcare professionals by providing quick access to patient information and concise summaries of medical records. The application's user-friendly interface, accessibility from any internet-enabled device, and integration of advanced AI models contribute to improved patient care by facilitating informed decision-making and reducing the time and effort required for data retrieval and analysis. Acknowledgments are extended to the project mentors, teaching and non-teaching staff of Vellore Institute of Technology, as well as family and friends for their invaluable support throughout the project's successful completion.

|  |  |  |
| --- | --- | --- |
| S.NO | CONTENTS | PAGE NO. |
|  | Acknowledgement | 4 |
|  | Executive Summary | 5 |
|  | Table of Contents | 6 |
|  | List of Figures | 9 |
|  | List of Tables | 10 |
|  | Abbreviations | 11 |
|  | Symbols and Notations | 11 |
| 1 | INTRODUCTION | 12 |
|  | 1.1 Objectives | 12 |
|  | 1.2 Motivation | 12 |
|  | 1.3 Background | 13 |
| 2 | PROJECT DESCRIPTION AND GOALS | 14 |
|  | 2.1 Survey on Existing System | 14 |
|  | 2.2 Research Gap | 19 |
|  | 2.3 Problem Statement | 19 |
| 3 | TECHNICAL SPECIFICATION | 20 |
|  | 3.1 Requirements  3.1.1 Functional  3.1.2 Non- Fuctional | 20  20  21 |
|  | 3.2 Feasibility Study  3.2.1 Technical Feasibility  3.2.2 Economic Feasibility  3.2.3 Social Feasibility | 21  21  22  22 |
|  | 3.3 System Specification  3.3.1 Hardware Specification  3.3.2 Software Specification  3.3.3 Standards and Policies | 23  23  23  23 |
| 4 | DESIGN APPROACH AND DETAILS | 24 |
|  | 4.1 System Architecture | 24 |
|  | 4.2 Design  4.2.1 Data Flow Diagram  4.2.2 Use Case Diagram  4.2.3 Class Diagram  4.2.4 Sequence Diagram | 24  25  25  26  27 |
|  | 4.3 Constraints,Alternatives and Tradeoffs | 27 |
| 5 | SCHEDULE,TASKS AND MILESTONES | 29 |
|  | 5.1 Gnatt Chart | 29 |
|  | 5.2 Module Description  5.2.1 User Interaction and Retriving Details  5.2.2 Uploading Details  5.2.3 Speech Recognition and OpenAI Engine  5.2.4 Document Summarization  5.2.5 Text to Speech Mode | 29  29  30  30  30  31 |
| 6 | PROJECT DEMONSTRATION | 32 |
| 7 | COST ANALYSIS/RESULT & DISCUSSION | 35 |
| 8 | SUMMARY | 37 |
| 9 | REFERENCES | 39 |
|  | APPENDIX A – SAMPLE CODE | 41 |

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **S.NO** | **FIGURE DESCRIPTION** | **PAGE NO.** |
| 1 | Workflow of the Project | 18 |
| 2 | Dataflow Diagram | 19 |
| 3 | Use Case Diagram | 19 |
| 4 | Class Diagram | 20 |
| 5 | Sequence Diagram | 21 |
| 6 | Gnatt Chart | 29 |
| 7 | Landing page of the Voice Assistant | 32 |
| 8 | Upload Patient Documents | 32 |
| 9 | Document Summary of Patient (Single file) | 33 |
| 10 | Document Summary of Patient (Multiple files) | 33 |
| 11 | Combined Summary of Patient (Multiple files) | 34 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TABLE DESCRIPTION** | **PAGE NO.** |
| 1 | Literature Survey | 9 |
| 2 | List of Abbrevations | 11 |

**LIST OF ABBREVATIONS**

|  |  |
| --- | --- |
| **GPT** | Generative Pre-trained Transformer |
| **NLP** | Natural Language processing |
| **OPENAI** | Open Artificial Intelligence |
| **VSCODE** | Visual Studio Code |

**SYMBOLS AND NOTATIONS**

**None**

1. **INTRODUCTION**

**1.1 OBJECTIVES**

The primary objective of this project is to develop a comprehensive voice-enabled application tailored to the needs of medical professionals, particularly nurses, aimed at streamlining the documentation process in healthcare settings. Through the integration of advanced technologies such as natural language processing (NLP) and speech recognition, the system aims to facilitate efficient access to patient information and medical document summaries. By enabling users to interact with the application using intuitive voice commands, the objective is to reduce the time and effort required for data entry and retrieval tasks, ultimately improving workflow efficiency and enhancing the quality of patient care.

Additionally, the system aims to leverage AI-driven approaches, such as OpenAI's GPT-3 model, to generate accurate patient details and concise summaries of medical documents, further enhancing the usability and effectiveness of the application. Overall, the objective is to provide a user-friendly and efficient solution that addresses the challenges faced by medical professionals in managing patient information and documentation, thereby contributing to the advancement of healthcare technology and patient care delivery.

**1.2** **MOTIVATIONS**

This project addresses several critical needs within the healthcare sector. Firstly, it streamlines the process of accessing patient information by integrating speech recognition technology, enabling healthcare professionals to quickly retrieve patient details through natural language input. This functionality significantly reduces the time spent on administrative tasks, allowing medical staff to allocate more time and resources towards direct patient care.Secondly, the project facilitates the analysis of patient documents by incorporating text summarization techniques. By automatically generating concise summaries from uploaded documents, healthcare professionals can quickly grasp the key information within medical records, reports, or other pertinent documents. This not only enhances efficiency but also improves decision-making processes, as medical staff can more readily identify relevant information and trends.

Moreover, the integration of speech synthesis capabilities enables the system to provide auditory feedback, making it accessible to users with visual impairments or those who prefer auditory information. This inclusivity fosters better communication and collaboration among healthcare team members, ultimately leading to improved patient outcomes and satisfaction.Overall, the project's motivation lies in its ability to optimize workflow processes, enhance information accessibility, and promote inclusivity within healthcare settings. By leveraging advanced technologies to address these challenges, the project aims to empower healthcare professionals to deliver more efficient, effective, and patient-centered care.

**1.3 BACKGROUND**

The genesis of this project stems from the recognition of key challenges within the healthcare sector, particularly regarding the management and retrieval of patient information. Historically, healthcare professionals have grappled with cumbersome processes for accessing and analyzing patient data, often leading to inefficiencies and potential errors in care delivery.To address these challenges, the project draws inspiration from recent advancements in natural language processing (NLP), speech recognition, and text summarization technologies. These advancements offer unprecedented opportunities to automate and streamline various aspects of healthcare operations, ultimately improving patient care outcomes.

**2. PROJECT DESCRIPTION AND GOALS**

**2.1 SURVEY OF EXISTING SYSTEMS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ref.** | **Paper Title** | **Author**  **and Year** | **Research**  **Question** | **Methodology/Approach** | **Limitation** |
| 1 | Benefits, Limits, and Risks of GPT-4 as an AI  Chatbot for Medicine | Lee *et al.*  (2023) | Using GPT 4 and similar Generative AI tools such as Google LaMDA and GPT 3.5 in Medical conversational ChatBOTs | ChatBOTs use the GPT 4 LLM to retrieve ansers for user queries from web and this model has been tested out and found to have an accuracy of over 90% | Authenticity of Data obtained from the web by GPT models. |
| 2 | Speech emotion recognition using machine learning - A systematic review | Madanian  *et al*.  (2023) | Properties, methodology and working of SER model and analysing its efficiency . | Training a speech recognition (SR) system, including language  corpus, nursing  activities, clinical conversations, and accents. It compared documentation time and error rates between SR- generated records and keyboard entry, | The paper may overlook non-ML approaches and interdisciplinary perspectives in SER, and while it discusses challenges and solutions, it may not encompass all potential obstacles or emerging  trends. |
| 3 | Development of the Speech- to-Text  Chatbot | Shakhovska  *et al.*  (2019) | Utilizing the Google Speech- to-Text  API.data | The proposed method involves employing prefix functions and | The study may potentially overlook |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Interface Based on Google API |  | from social networks to  focused on remote and local storage  processes. | hashing algorithms for keyword searching and verb ending  identification in chatbot conversations | alternative methods and their effectiveness in real-world applications. |
| 4 | Machine learning-based speech recognition system for nursing documentation  – A pilot study | Lee *et al.*  (2023) | Machine learning-based speech recognition (SR) system's effectiveness in reducing nursing documentation workload in a psychiatry ward. | The study collected language corpus, nursing activities, clinical conversations, and accent data for SR system training in four sessions and achieved model had an accuracy score of 87.06% to 95.07% across sessions. | The study's findings are based on a pilot implementation in a psychiatry ward, potentially limiting generalizability to other nursing specialties or healthcare settings. |
| 5 | Intelligent speech technologies for transcription, disease diagnosis, and medical equipment interactive control in smart hospitals: A  review | Zhang *et al.*  (2023) | To explore the application and potential of intelligent speech technology (IST) in  addressing medical resource shortages and improving healthcare  efficiency amid | The paper introduces IST's procedure and system architecture, reviews its applications in smart hospitals, and presents a case study on stroke patient care.  Additionally, it proposes a novel medical voice analysis system architecture. | Challenges include noise interference and pronunciation differences, which may hinder the widespread application of IST in hospitals. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | challenges like noise interference and pronunciation  differences. |  |  |
| 6 | The Capability of ChatGPT in Predicting and Explaining Common Drug- Drug Interactions | Juhi A *et al.*  (2023) | To assess the effectiveness of ChatGPT in predicting and explaining common drug- drug interactions (DDIs) | Utilized 40 DDI lists from literature to converse with ChatGPT using two-stage  questions, assessing responses' correctness with pharmacologists' consensus. | ChatGPT provided incomplete guidance at times, necessitating further improvement for patient use regarding DDI  awareness. |
| 7 | Deep Cross- Corpus Speech Emotion Recognition: Recent Advances and Perspectives | Zhang *et al.*  (2021) | To comprehensivel y survey the state- of-the-art techniques in cross-corpus speech emotion recognition (SER),  particularly focusing on deep learning methods associated with supervised, unsupervised,  and semi- | The paper reviews existing literature on speech emotion databases, traditional methods for cross-corpus SER, recent advances in deep learning techniques, and discusses challenges and future directions in the field. | Challenges such as natural data scarcity, multimodal integration, and limitations of deep learning techniques, potentially affecting the comprehensiveness of its findings are discussed. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | supervised  learning. |  |  |
| 8 | Natural  Language | Wang *et al.* | Propose  Contrastive | Utilize two  innovative encoders | Limited  evaluation on |
|  | Supervision | (2023) | Language- | for audio and text, | tasks with true |
|  | for General- |  | Audio | trained with | Zero-Shot |
|  | Purpose |  | Pretraining | Contrastive | setup. |
|  | Audio |  | (CLAP) for | Learning to create | Increased |
|  | Representati |  | joint audio- | multimodal | training pair |
|  | ons |  | text | representations. | diversity |
|  |  |  | representation | Train audio encoder | affects |
|  |  |  | learning, | (HTSAT- 22) on 22 | performance |
|  |  |  | enabling | tasks and adapt | variably across |
|  |  |  | Zero- Shot | GPT2 for text | domains. |
|  |  |  | inference | encoding, enabling |  |
|  |  |  | across 26 | joint learning of |  |
|  |  |  | downstream | representations in a |  |
|  |  |  | tasks, | multimodal space |  |
|  |  |  | surpassing | for Zero-Shot |  |
|  |  |  | state-of-the- | inference. |  |
|  |  |  | art models for |  |  |
|  |  |  | general- |  |  |
|  |  |  | purpose audio |  |  |
|  |  |  | representation |  |  |

# 2.2 GAPS IDENTIFIED

In the field of voice assistant solutions for healthcare, several significant gaps have been identified, posing challenges to their widespread adoption and effectiveness. One significant gap lies in the system's accuracy and reliability, particularly concerning speech recognition and natural language processing. Despite advancements in these technologies, instances of inaccurate transcription or irrelevant responses may occur, impacting the system's usability in real-world healthcare settings. Additionally, the system must adapt to diverse healthcare environments, accommodating variations in workflows, terminology, and documentation requirements across different healthcare facilities. Ensuring robust data security and privacy measures is another critical gap, as the system handles sensitive patient information and must comply with regulations such as HIPAA. Integration with existing healthcare systems presents another challenge, requiring seamless interoperability to avoid duplication of efforts and streamline workflow integration. Moreover, providing comprehensive user training and support is essential to empower healthcare professionals in utilizing the system effectively, particularly for those less familiar with technology. Ethical considerations, including biases in AI-driven algorithms and transparency in decision-making processes, represent another gap that must be addressed to maintain trust and integrity in the system. Finally, scalability and performance are vital considerations, with the system needing to handle increasing volumes of data and user interactions while maintaining optimal performance levels. By identifying and addressing these gaps proactively, the voice-enabled patient documentation and assistance system can fulfill its potential to revolutionize healthcare documentation processes and improve patient care outcomes.

# 2.3 PROBLEM STATEMENT

In modern healthcare settings, the documentation of patient information and medical records is a critical aspect of delivering quality care and ensuring patient safety. However, traditional methods of data entry and retrieval often prove to be inefficient, error-prone, and time-consuming. Healthcare professionals, particularly nurses, face challenges in accessing patient details and summarizing medical documents quickly and accurately, leading to potential delays in care delivery and compromised workflow efficiency. Additionally, the increasing volume of patient data and the complexity of medical documentation further exacerbate these challenges, highlighting the need for innovative solutions to streamline the documentation process and enhance accessibility to critical

information.

The problem at hand revolves around the inefficiencies and limitations of current documentation practices in healthcare settings, which hinder the timely and accurate retrieval of patient information and medical document summaries. There is a pressing need for a comprehensive solution that leverages advanced technologies, such as natural language processing (NLP), speech recognition, and artificial intelligence (AI), to enable healthcare professionals to access patient details and summarize medical documents efficiently and accurately. Furthermore, the solution must address concerns related to data security, privacy, interoperability with existing healthcare systems, user training, ethical considerations, and scalability to ensure its viability and effectiveness in real-world healthcare environments. Thus, the problem statement encapsulates the imperative to develop a voice-enabled patient documentation and assistance system that overcomes the limitations of current practices and empowers healthcare professionals to deliver optimal care through streamlined documentation processes.

**3. TECHNICAL SPECIFICATIONS**

**3.1 REQUIREMENT ANALYSIS**

# 3.1.1 FUNCTIONAL REQUIREMENTS

* Speech Recognition: The system should be able to transcribe speech input from the user accurately to retrieve patient details.
* Retrieve Patient Details: Upon receiving the patient's name or ID through speech input, the system should retrieve the corresponding details from the database.
* Text-to-Speech Conversion: The system should convert retrieved patient details and combined document summaries into speech for user interaction.
* Summarize Documents: When uploading patient documents, the system should summarize each document individually and combine the summaries.
* Combined Summary Display: The system should display the combined summary of uploaded documents upon user request.
* Error Handling: Proper error handling should be implemented for scenarios such as unrecognized speech, failure to retrieve patient details, or document summarization errors

# 3.1.2 NON- FUNCTIONAL REQUIREMENTS

* Accuracy: The system's speech recognition and document summarization functionalities should have high accuracy to ensure reliable results.
* Performance: The system should respond promptly to user inputs, with minimal latency in speech recognition, response generation, and document summarization.
* Security: Patient data should be handled securely, ensuring confidentiality and compliance with relevant privacy regulations.
* Scalability: The system should be able to handle a growing number of patient records and documents without compromising performance.
* Usability: The user interface should be intuitive and easy to use, with clear instructions provided to the user for each action.
* Robustness: The system should be robust enough to handle unexpected inputs or errors gracefully, providing informative error messages to the user.
* Accessibility: The system should be accessible to users with disabilities, with support for alternative input methods and assistive technologies.
* Portability: The application should be deployable on various platforms and environments, ensuring compatibility with different devices and operating systems.
* Maintainability: The codebase should be well-structured and documented to facilitate future updates, maintenance, and enhancements.
* Compliance: The system should comply with relevant standards and regulations governing healthcare software, including data protection and patient rights.

# 3.2 FEASIBILITY STUDY

# 3.2.1Technical Feasibility:

1. Existing Technologies: Both Visual Studio Code (VS Code) and OpenAI's GPT model are well-established technologies with extensive documentation and community support. Integrating them into a single project is technically feasible.
2. Hardware and Software Requirements: The hardware and software requirements for running the integrated AI code assistant are within reasonable bounds and can be met by most modern computers and operating systems.
3. Development Expertise: Availability of developers with expertise in JavaScript/TypeScript, VS Code extension development, natural language processing (NLP), and machine learning (ML) is essential. The project feasibility relies on having a skilled team capable of implementing and maintaining the solution.

# 

# 3.2.2 Economic Feasibility:

1. Cost of Development: The primary costs include developer salaries, infrastructure costs (e.g., cloud computing for running the AI model), and any licensing fees associated with using OpenAI's services. These costs must be weighed against the potential benefits and revenue streams of the project.
2. Market Potential: Assessing the market demand for an AI-powered code assistant within the VS Code ecosystem is crucial. Conducting market research to identify potential users, competitors, and pricing models can help gauge the economic viability of the project.
3. Return on Investment (ROI): Calculating the expected ROI based on projected revenue streams (e.g., subscription fees, advertising, premium features) and cost estimates can provide insights into the economic feasibility of the project.

# 3.2.3 Social Feasibility:

Intellectual Property Rights: Ensuring compliance with intellectual property rights, licensing agreements, and terms of service for using VS Code, OpenAI's GPT model, and any other third-party libraries or services is essential to avoid legal issues.

Data Privacy and Security: Adhering to data privacy regulations (e.g., GDPR, CCPA) and implementing robust security measures to protect user data and ensure secure communication with external APIs is imperative.

# 3.3 SYSTEM SPECIFICATION

**3.3.1 HARDWARE SPECIFICATION**

Processor: Intel Core i3 or AMD equivalent

Storage: 100MB available disk space

RAM: 4GB

# 3.3.2 SOFTWARE SPECIFICATION

Operating System:

* Windows 10 (64-bit)
* macOS 10.12 or later
* Linux (Ubuntu 18.04 LTS or later, Fedora 28 Development Environment:
* Visual Studio Code (latestversion) Programming Languages: Python

# 3.3.3 STANDARDS AND SPECIFICATION

* + Coding Standards: Adherence to industry-standard coding practices such as clean code principles, naming conventions, and code readability guidelines (e.g., Google's JavaScript Style Guide for JavaScript/TypeScript code).
  + Security Standards: Compliance with industry-standard security practices such as OWASP (Open Web Application Security Project) guidelines for secure coding, encryption standards for data transmission, and vulnerability assessment protocols.
  + Natural Language Processing (NLP): Specialization in NLP techniques and algorithms to

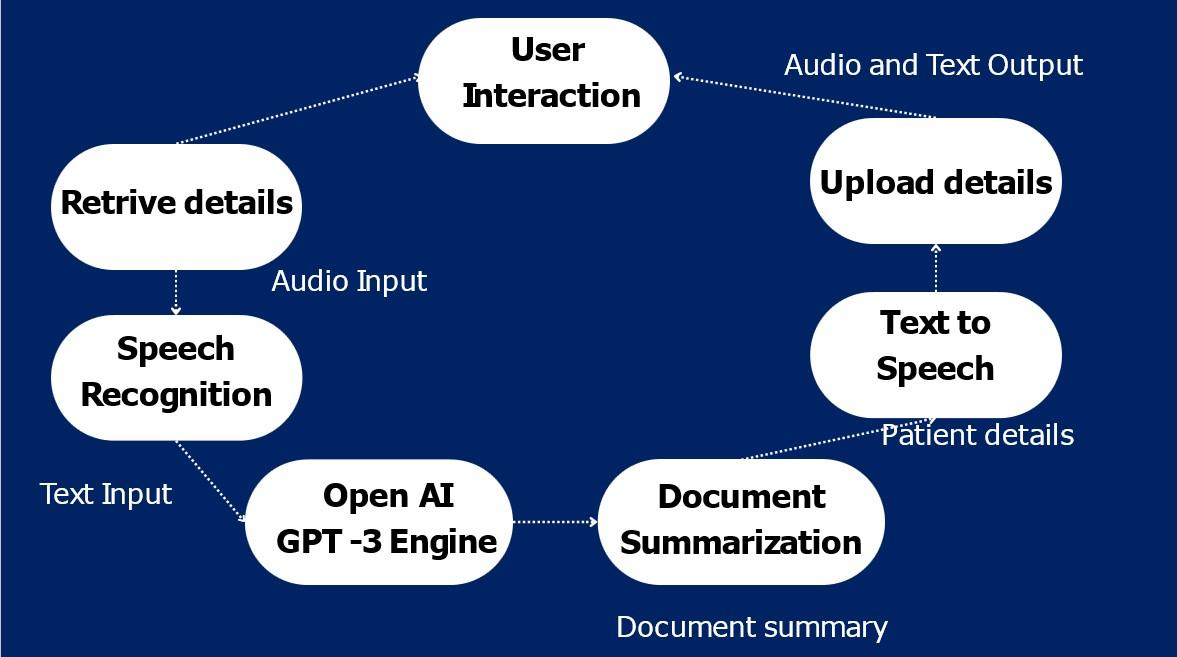
effectively utilize the OpenAI GPT model for code suggestions, completions, and contextual

guidance.

* + Machine Learning and AI: Expertise in machine learning techniques, particularly in training and fine-tuning language models like GPT for specific tasks, as well as continuous improvement of AI models based on user feedback.

**4 . DESIGN APPROACH AND DETAILS**

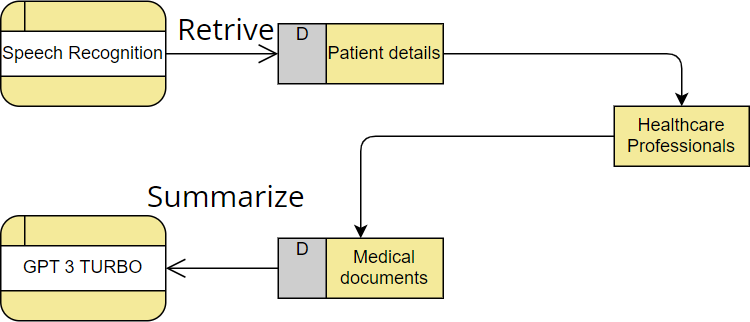
# 4.1 SYSTEM ARCHITECTURE



# Figure 1 :Workflow of project

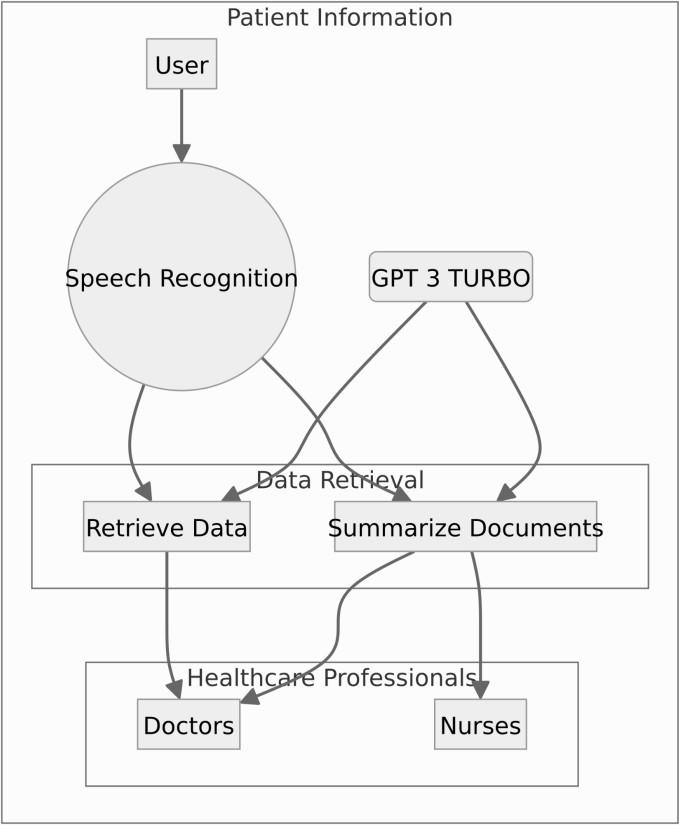
# 4.2 DESIGN

# 4.2.1 DATA FLOW DIAGRAM



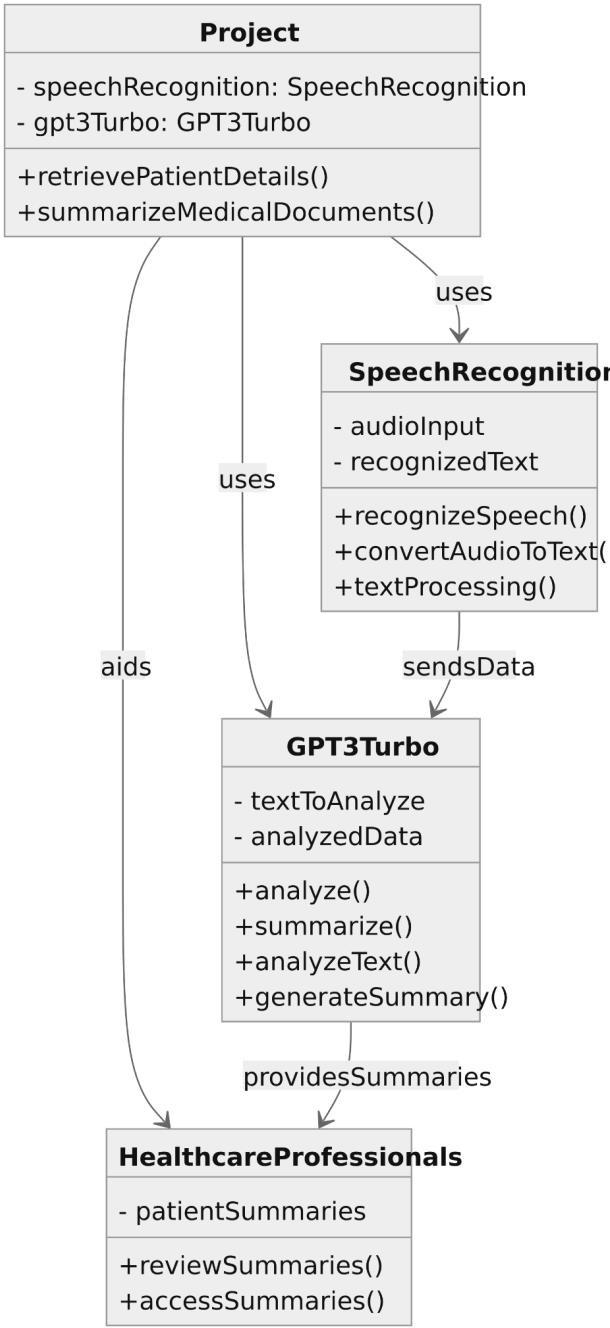
**Figure 2 : Level 2 Dataflow Diagram**

# 4.2.2 USE CASE DIAGRAM



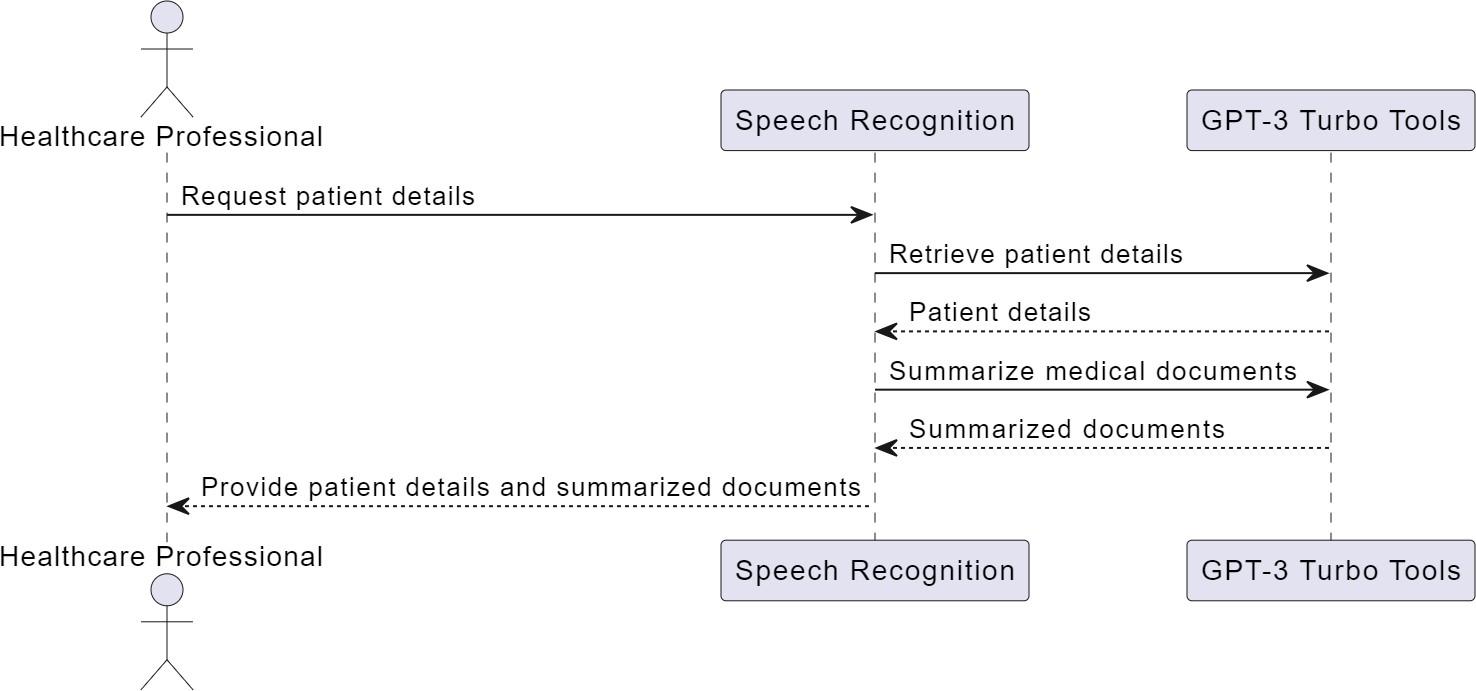
**Figure 3 : Use case diagram**

# 4.2.3 CLASS DIAGRAM



**Figure 4 : Class Diagram**

# 4.2.4 SEQUENCE DIAGRAM



**Figure 5 : Sequence diagram**

**4.3 CONSTRAINTS ,ALTERNATIVES AND TRADEOFFS**

**Constraints:**

* Accuracy of Speech Recognition: The accuracy of the speech recognition system may be affected by factors such as background noise, accents, and speech impediments, leading to potential errors in transcribing audio inputs.
* Summarization Quality: The quality of text summarization may vary depending on the complexity and structure of the input documents. Ensuring accurate and informative summaries for diverse document types can be challenging.
* Integration Complexity: Integrating multiple technologies (speech recognition, text summarization, speech synthesis) into a cohesive system requires careful planning and implementation to ensure seamless functionality and compatibility.

**Alternatives:**

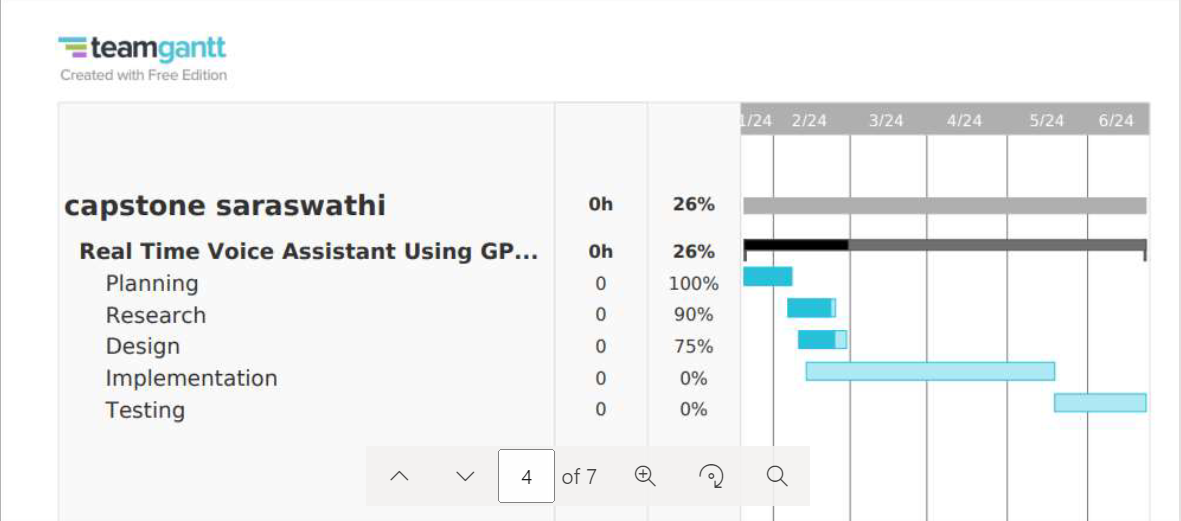
* Alternative Speech Recognition Systems: Exploring alternative speech recognition APIs or libraries could provide different levels of accuracy and support for various languages and accents, allowing for better customization based on user needs.
* Different Summarization Models: Experimenting with different summarization models or algorithms beyond the one used (e.g., BERT-based models) may yield improved summarization results for specific document types or languages.
* Hybrid Approaches: Combining both automatic summarization techniques with manual review by healthcare professionals could ensure the accuracy and relevance of generated summaries, albeit at the cost of increased time and effort.

**Tradeoffs:**

* Accuracy vs. Speed: Balancing the need for accurate speech recognition and text summarization with the desire for fast response times may require compromising one aspect over the other. Increasing accuracy often involves more extensive processing, which can slow down system performance.
* Complexity vs. Usability: While integrating advanced technologies enhances system capabilities, it may also increase complexity for end-users. Striking a balance between functionality and user-friendliness is crucial to ensure adoption and usability among healthcare professionals.
* Resource Consumption vs. Scalability: Resource-intensive processes like text summarization can strain system resources, especially with a growing number of users or documents. Implementing scalable solutions, such as distributed computing or cloud-based services, involves tradeoffs in terms of cost and infrastructure management.

**5 . SCHEDULE TASKS AND MILESTONE**

**5.1 GNATT CHART**

****

**Fig 6 : Gnatt Chart**

**5.2 MODULE DESCRIPTION**

**User Interaction - Retrieving Details:**

* This module handles the interaction with the user for retrieving patient details.
* It prompts the user to state the patient's name or ID using the Streamlit framework.
* Utilizes the speech\_recognition library to transcribe the user's speech input.
* Once the transcription is obtained, it passes the information to the OpenAI engine to generate a response.
* The response containing patient details is displayed to the user through the Streamlit interface.
* Additionally, the response is converted to speech using the Text-to-Speech module and played to the user.

# Uploading Details:

* This module facilitates the user in uploading patient documents.
* It prompts the user to select a patient and upload one or more documents (PDF, DOCX, TXT formats).
* Utilizes the Streamlit framework for user interface and file upload functionality.
* After successful document upload, it summarizes each document using the Summarizer module.
* Individual document summaries are displayed to the user and combined to generate a comprehensive summary.
* The combined summary is displayed to the user and also converted to speech for auditory feedback.

# Speech Recognition and OpenAI Engine:

* The Speech Recognition module uses the speech\_recognition library to transcribe audio input from the user.
* It captures audio input from the user's microphone and transcribes it into text.
* The transcribed text is then used as input to the OpenAI engine for generating responses.
* The generated responses contain patient details when retrieving details or other relevant information based on the user's query.
* The OpenAI engine uses the GPT-3.5 model (gpt-3.5-turbo-instruct) to generate natural

language responses based on the provided prompts.

# Document Summarization:

* This module summarizes the content of uploaded patient documents.
* It uses the Summarizer library, presumably based on Transformer models like GPT, to generate concise summaries of each document.
* The summarization process involves extracting key information from the document while preserving its meaning.
* Summarized content is displayed to the user and combined to create an overall summary of all uploaded documents.

# Text-to-Speech Module:

* The Text-to-Speech module converts textual information into audible speech.
* It utilizes the pyttsx3 library to initialize a text-to-speech engine.
* The engine processes text input and generates corresponding speech output.
* The generated speech is played to the user, providing auditory feedback for retrieved patient details and document summaries.

**5.3 TESTING**

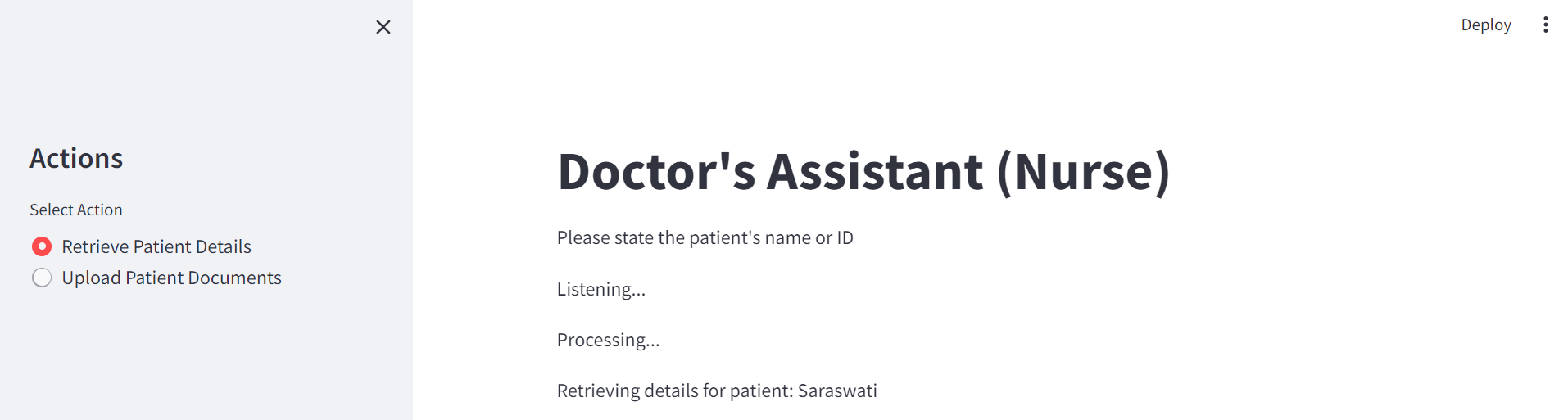
**5.3.1 UNIT TESTING**

* The audio from user is accurately transcribed and understood by the program.
* Various errors such as Request Error , Unknown value error are managed.
* Accurately handles tests with different patient names and summaries to verify correct responses.
* Patient details not found case handled .
* Tested to ensure that text is correctly converted to speech.

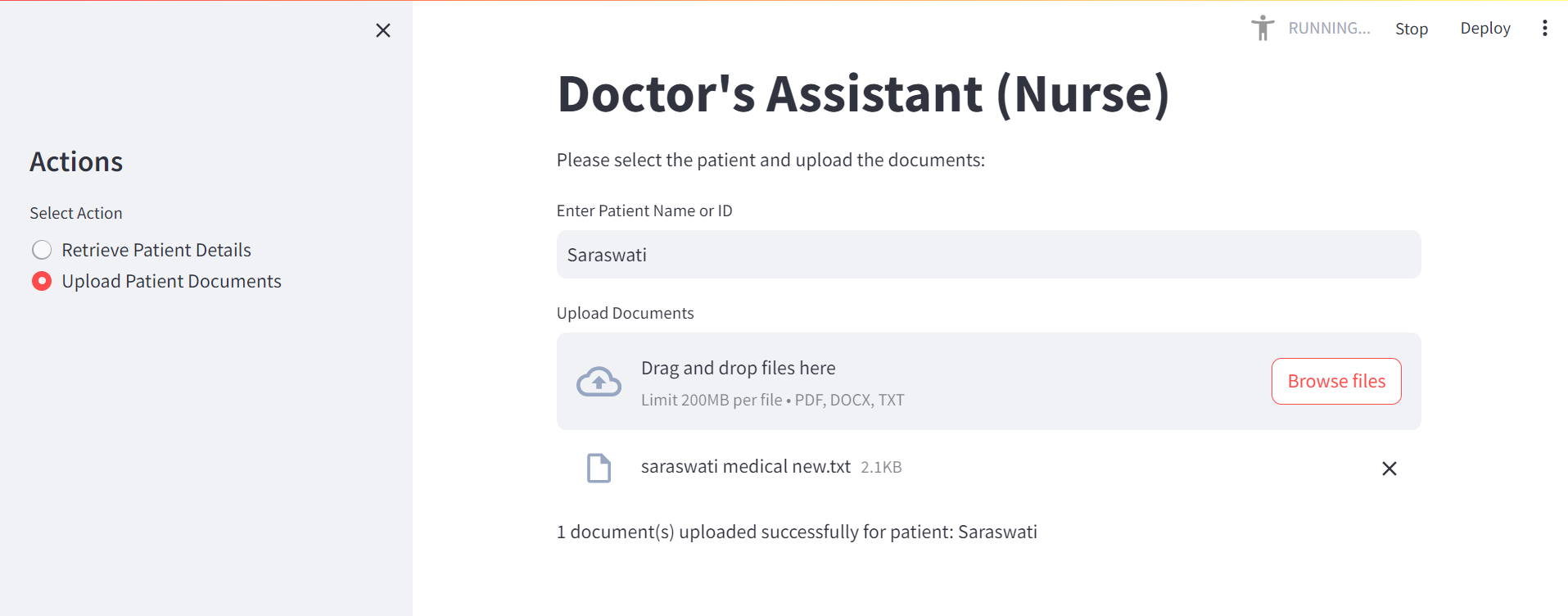
**5.3.2 INTEGRATION TESTING**

* Simulated audio inputs and verify that the speech recognition module correctly transcribes them into text.
* Provided different text inputs and ensure that the summarization module generates accurate summaries.
* Tested the entire workflow and ensured smooth transcribing audio to retrieving patient details or summarizing uploaded documents, to ensure seamless integration and functionality.
* Verified that the synthesized speech matches the expected output based on the provided text inputs.

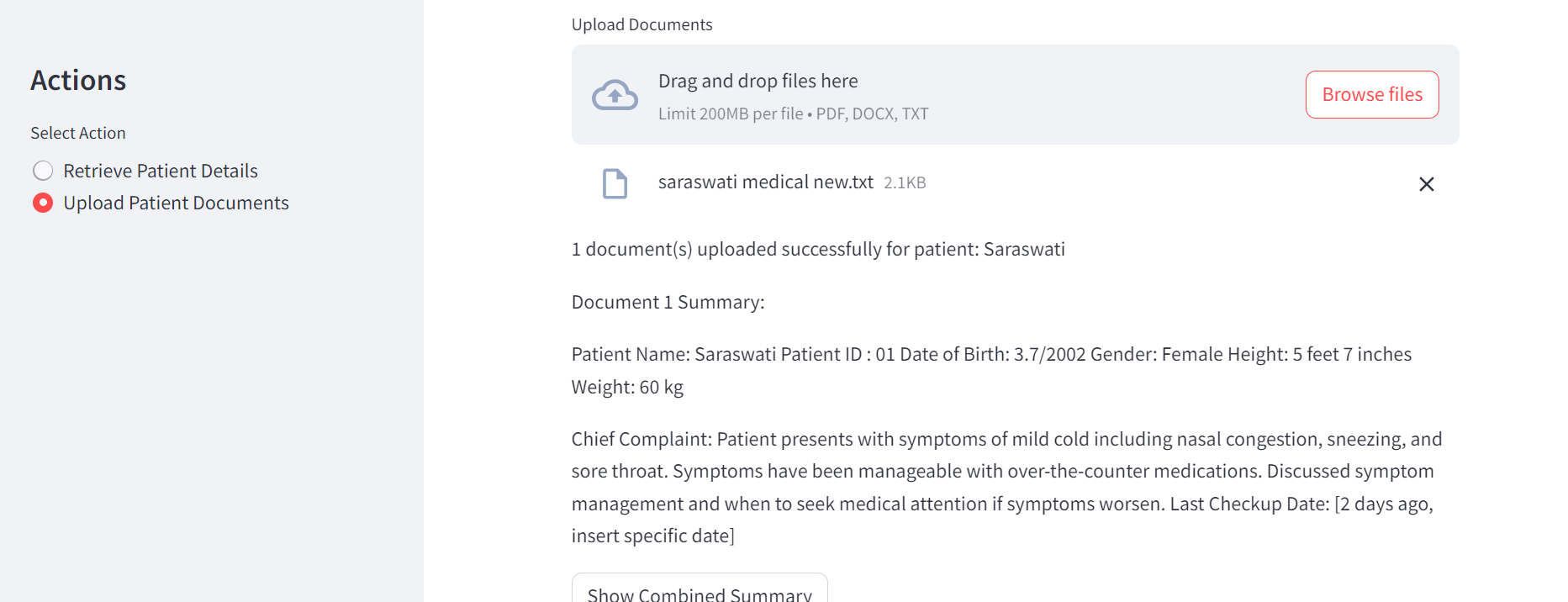
**6. PROJECT DEMONSTRATION**



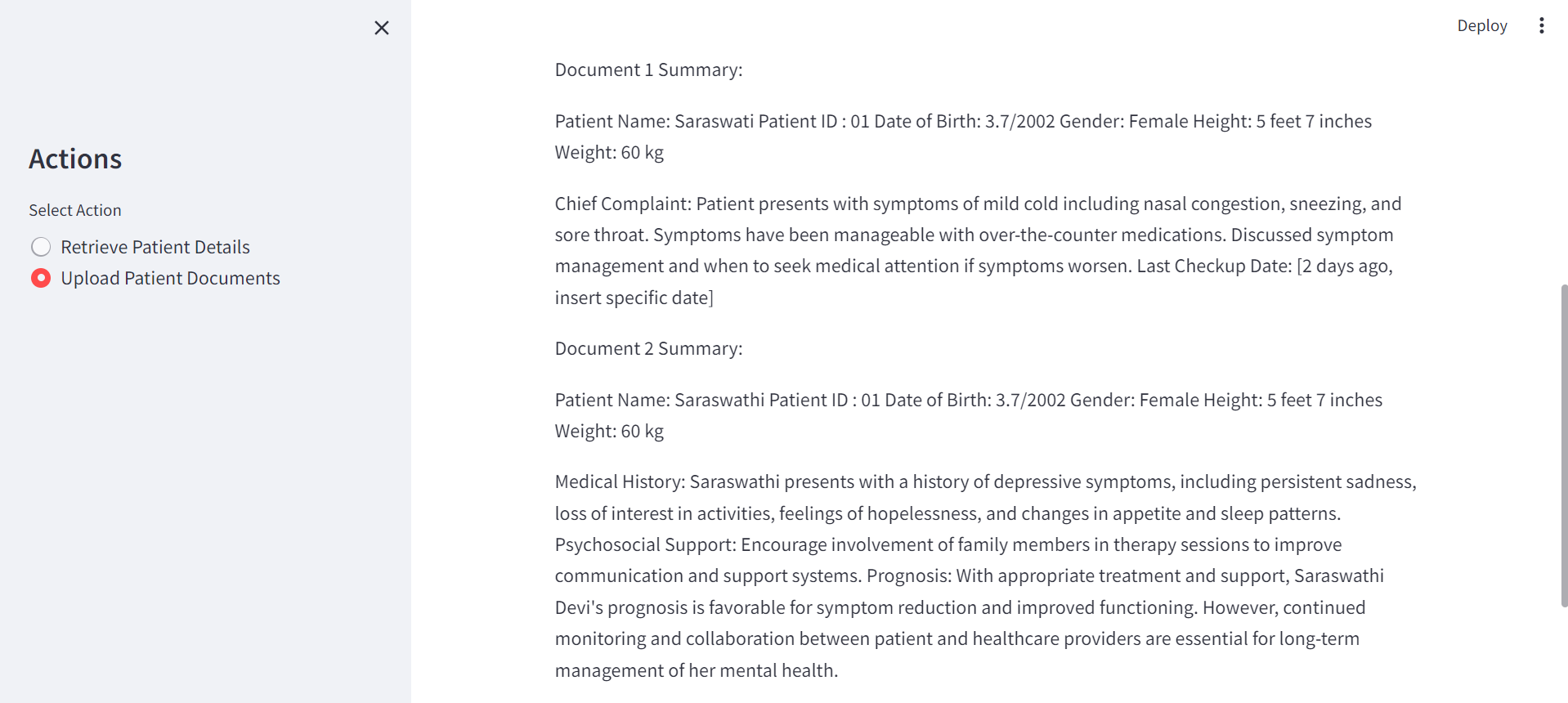
**Fig 7: Landing page of the Voice Assistant**



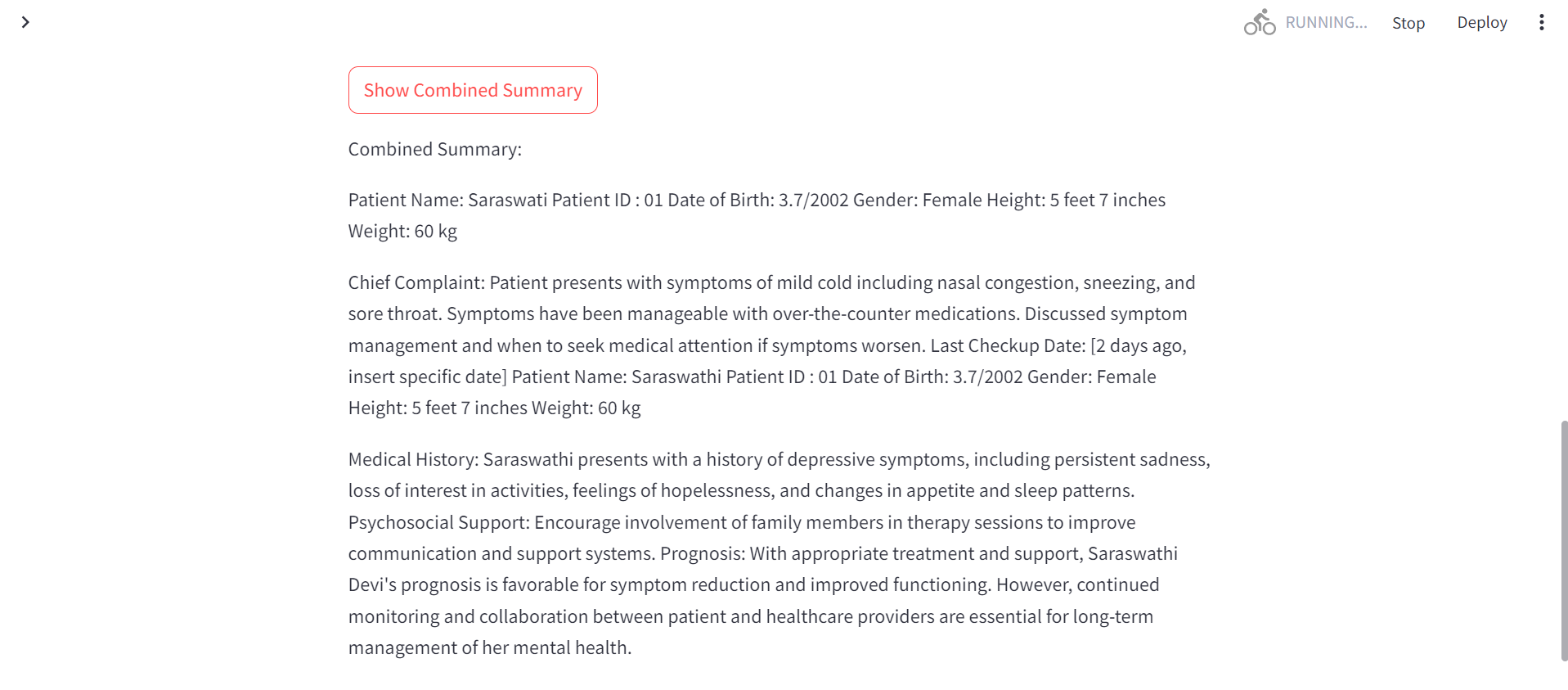
**Fig 8: Upload Patient Documents**



**Fig 9: Document Summary of Patient (Single file)**



**Fig 10: Document Summary of Patient (Multiple files)**



**Fig 10: Combined Summary of Patient (Multiple files)**

**7. RESULT AND DISCUSSION**

The implemented system successfully enables medical professionals to:

1. Retrieve patient details by voice command.
2. Upload medical documents for summarization.
3. Obtain summarized document content in both text and audio formats.
4. Experience streamlined workflow and enhanced efficiency in accessing patient information.

The Doctor's Assistant (Nurse) project combines various technologies to enhance the efficiency of healthcare professionals in managing patient information. Utilizing Streamlit for user interface, the application offers two main functionalities: retrieving patient details and uploading patient documents. For retrieving patient details, users can simply speak the patient's name or ID, which is transcribed using speech recognition. This transcription is then used to prompt an AI model, powered by OpenAI's GPT-3.5 engine, to generate detailed patient information. The application also supports uploading patient documents in PDF, DOCX, or TXT formats. Upon document upload, the system summarizes each document using the Summarizer library, providing concise summaries. Additionally, it combines these individual summaries into a comprehensive overview, which can be displayed to the user and converted into speech for auditory feedback. Overall, this project aims to streamline the process of accessing patient information and summarizing medical documents, ultimately improving the workflow and productivity of healthcare professionals.

Despite the project's notable accomplishments, several areas warrant attention for further refinement and enhancement. Firstly, continuous improvement in speech recognition accuracy and robustness is imperative to ensure reliable performance across diverse environments and accents. Fine-tuning the speech recognition model or exploring alternative solutions could address potential challenges in transcription accuracy, thereby enhancing user satisfaction.Additionally, augmenting the system's natural language understanding capabilities can enhance response generation accuracy and relevance. Incorporating domain-specific knowledge or context awareness may enable more tailored and informative responses, thereby improving the overall user experience.

Overall, by addressing these considerations and leveraging technological advancements, the Doctor's Assistant (Nurse) project can further enhance its effectiveness and usability, thereby contributing to improved patient care delivery in real-world healthcare environments.

**8. SUMMARY**

The Doctor's Assistant (Nurse) project represents a significant advancement in leveraging technology to streamline healthcare workflows and enhance patient care delivery. By integrating cutting-edge technologies such as speech recognition, natural language processing (NLP), text summarization, and speech synthesis, the project aims to empower healthcare professionals with efficient tools for accessing patient information and analyzing medical documents.

At its core, the project offers two primary functionalities: retrieving patient details and summarizing uploaded patient documents. The speech recognition module enables users to verbally provide patient names or IDs, facilitating seamless interaction with the system. Upon transcription, the system leverages OpenAI's powerful GPT-3.5 model to generate detailed responses, encompassing relevant patient information. This feature enables healthcare professionals to quickly access patient details, improving workflow efficiency and enabling more informed decision-making.

Additionally, the text summarization module plays a crucial role in enhancing document analysis capabilities. By generating concise summaries of uploaded patient documents, the system enables healthcare professionals to grasp key information swiftly and efficiently. The summaries encapsulate essential details within the documents, aiding in diagnosing conditions, formulating treatment plans, and monitoring patient progress. Moreover, the integration of speech synthesis allows users to listen to the summaries, enhancing accessibility for individuals with visual impairments and promoting inclusivity within healthcare settings.

Throughout the project, usability and user experience have been prioritized to ensure seamless interaction and adoption by healthcare professionals. The user interface is designed to be intuitive and user-friendly, with features such as error handling and feedback mechanisms enhancing usability. Continuous refinement based on user feedback further enhances the system's effectiveness and user satisfaction.

While the project demonstrates commendable performance across its key functionalities, ongoing refinement and optimization are essential for further enhancing its efficacy. Areas such as speech recognition accuracy, natural language understanding, and text summarization quality present opportunities for improvement. By addressing these considerations and leveraging technological advancements, the Doctor's Assistant (Nurse) project aims to continue driving innovation in healthcare delivery, ultimately contributing to improved patient outcomes and enhanced efficiency within healthcare settings.

**9.REFERENCES**

1. Juhi A, Pipil N, Santra S, Mondal S, Behera JK, Mondal H. The Capability of ChatGPT in Predicting and Explaining Common Drug-Drug Interactions. Cureus. 2023 Mar 17;15(3):e36272. doi: 10.7759/cureus.36272. PMID: 37073184; PMCID: PMC10105894.
2. Lee P, Bubeck S, Petro J. Benefits, Limits, and Risks of GPT-4 as an AI Chatbot for Medicine. N Engl J Med. 2023 Mar 30;388(13):1233-1239. doi: 10.1056/NEJMsr2214184. PMID: 36988602.
3. Chi-Chun Lee, Emily Mower, Carlos Busso, Sungbok Lee, Shrikanth Narayanan,Emotion recognition using a hierarchical binary decision tree approach,Speech Communication,Volume 53, Issues 9–10,2011,Pages 1162- 1171,ISSN 0167-6393.
4. Imani, Maryam & Montazer, Gholam Ali. (2019). A survey of emotion recognition methods with emphasis on E-Learning environments. Journal of Network and Computer Applications. 147. 102423. 10.1016/j.jnca.2019.102423.
5. Anagnostopoulos, Christos-Nikolaos & Iliou, Theodoros & Giannoukos, Ioannis. (2012). Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011. Artificial Intelligence Review. 43. 10.1007/s10462-012-9368-5.
6. Liu, Zhen-Tao & Wu, Min & Cao, Weihua & Mao, Jun-Wei & Xu, Jian-Ping & Tan, Guan-Zheng. (2017). Speech emotion recognition based on feature selection and extreme learning machine decision tree. Neurocomputing. 273. 10.1016/j.neucom.2017.07.050.
7. Ghosh, Sayan, Eugene Laksana, Louis-Philippe Morency, and Stefan Scherer. "Representation learning for speech emotion recognition." In Interspeech, pp. 3603-3607. 2016.
8. Jahangir, Rashid, Ying Wah Teh, Faiqa Hanif, and Ghulam Mujtaba. "Deep learning approaches for speech emotion recognition: State of the art and research challenges." Multimedia Tools and Applications (2021): 1-68.
9. Fahad, Md Shah, Ashish Ranjan, Jainath Yadav, and Akshay Deepak. "A survey of speech emotion recognition in natural environment." Digital signal processing 110 (2021): 102951.
10. Lee, Tso-Ying, Chin-Ching Li, Kuei-Ru Chou, Min-Huey Chung, Shu-Tai Hsiao, Shu- Liu Guo, Lung-Yun Hung, and Hao-Ting Wu. "Machine learning-based speech recognition system for nursing documentation–A pilot study." International Journal of Medical Informatics 178 (2023): 105213.
11. Singh, Aashdeep, P. Murugeswari, SD Prabu Ragavendiran, Amanpreet Kaur, Gurpreet Singh, and Sathiyamoorthy Margabandu. "AI-based Chatbot for Physically Challenged People." In 2022 International Conference on Edge Computing and Applications (ICECAA), pp. 1039-1044. IEEE, 2022.
12. Shakhovska, Nataliya, Oleh Basystiuk, and Khrystyna Shakhovska. "Development of the Speech-to-Text Chatbot Interface Based on Google API." In MoMLeT, pp. 212-221. 2019.
13. Madanian, Samaneh, Talen Chen, Olayinka Adeleye, John Michael Templeton, Christian Poellabauer, Dave Parry, and Sandra L. Schneider. "Speech emotion recognition using machine learning—A systematic review." Intelligent systems with applications (2023): 200266.
14. Zhang, Jun, Jingyue Wu, Yiyi Qiu, Aiguo Song, Weifeng Li, Xin Li, and Yecheng Liu. "Intelligent speech technologies for transcription, disease diagnosis, and medical equipment interactive control in smart hospitals: A review." Computers in Biology and Medicine (2023): 106517.
15. Choudhary, Nurendra, and Chandan K. Reddy. "Complex Logical Reasoning over Knowledge Graphs using Large Language Models." arXiv preprint arXiv:2305.01157 (2023).
16. Elizalde, Benjamin, Soham Deshmukh, and Huaming Wang. "Natural language supervision for general-purpose audio representations." arXiv preprint arXiv:2309.05767 (2023).

**APPENDIX A**

**SAMPLE CODE**

import streamlit as st

import openai

import pyttsx3

import speech\_recognition as sr

from summarizer import Summarizer

import time

openai.api\_key = "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"

engine = pyttsx3.init()

def transcribe\_audio\_to\_text(audio):

recognizer = sr.Recognizer()

try:

return recognizer.recognize\_google(audio)

except sr.UnknownValueError:

st.error("Could not understand audio")

except sr.RequestError as e:

st.error(f"Could not request results from Google Speech Recognition service; {e}")

except Exception as e:

st.error(f"Error occurred during speech recognition: {e}")

return None

def generate\_response(prompt):

try:

response = openai.Completion.create(

engine="gpt-3.5-turbo-instruct",

prompt=prompt,

max\_tokens=150,

n=1,

stop=None,

temperature=0.7,

)

return response["choices"][0]["text"]

except Exception as e:

st.error(f"Error occurred during response generation: {e}")

return None

def speak\_text(text):

global engine

try:

while engine.isBusy():

time.sleep(0.1) # Wait for 0.1 seconds before checking again

engine.say(text)

engine.runAndWait()

print("Text spoken successfully")

except Exception as e:

st.error(f"Error occurred during text-to-speech conversion: {e}")

def summarize\_text(text):

model = Summarizer()

summary = model(text, min\_length=50)

return ''.join(summary)

def main():

st.title("Doctor's Assistant (Nurse)")

st.sidebar.title("Actions")

action = st.sidebar.radio("Select Action", ("Retrieve Patient Details", "Upload Patient Documents"))

if action == "Retrieve Patient Details":

st.write("Please state the patient's name or ID")

with sr.Microphone() as source:

recognizer = sr.Recognizer()

st.write("Listening...")

audio = recognizer.listen(source)

st.write("Processing...")

transcription = transcribe\_audio\_to\_text(audio)

if transcription:

st.write(f"Retrieving details for patient: {transcription}")

prompt = f"Retrieve details for patient {transcription}"

response = generate\_response(prompt)

if response:

st.write("Patient Details:")

st.write(response)

speak\_text(response)

elif action == "Upload Patient Documents":

st.write("Please select the patient and upload the documents:")

patient\_name = st.text\_input("Enter Patient Name or ID")

uploaded\_files = st.file\_uploader("Upload Documents", type=['pdf', 'docx', 'txt'], accept\_multiple\_files=True)

if uploaded\_files:

st.write(f"{len(uploaded\_files)} document(s) uploaded successfully for patient: {patient\_name}")

combined\_summary = "" # Variable to store combined summaries

for i, uploaded\_file in enumerate(uploaded\_files):

text = uploaded\_file.read().decode("utf-8") # Read the file content

summary = summarize\_text(text)

st.write(f"Document {i+1} Summary:")

st.write(summary)

combined\_summary += summary + "\n" # Append individual summary to combined summary

if st.button("Show Combined Summary"):

st.write("Combined Summary:")

st.write(combined\_summary)

speak\_text(combined\_summary) # Convert combined summary to speech and output

if \_\_name\_\_ == "\_\_main\_\_":

main()