

2022

A Cloud Computing-based Dashboard for the Visualization of Motivational Interviewing Metrics

E Jinq Heng
Wright State University

Follow this and additional works at: https://corescholar.libraries.wright.edu/etd_all



Part of the [Computer Engineering Commons](#), and the [Computer Sciences Commons](#)

Repository Citation

Heng, E Jinq, "A Cloud Computing-based Dashboard for the Visualization of Motivational Interviewing Metrics" (2022). *Browse all Theses and Dissertations*. 2661.
https://corescholar.libraries.wright.edu/etd_all/2661

This Thesis is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.

A CLOUD COMPUTING-BASED DASHBOARD FOR
THE VISUALIZATION OF MOTIVATIONAL
INTERVIEWING METRICS

A thesis submitted in partial fulfillment of the
requirements for the degree of
Master of Science

By

E JINQ HENG
B.S., Upper Iowa University, 2016

2022
Wright State University

WRIGHT STATE UNIVERSITY

GRADUATE SCHOOL

December 6, 2022

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY E Jinq Heng ENTITLED A Cloud Computing-based Dashboard for the Visualization of Motivational Interviewing Metrics BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science.

Ashutosh Shivakumar, Ph.D.
Thesis Director

Thomas Wischgoll, Ph.D.
Interim Chair, Department of
Computer Science and Engineering

Committee on Final Examination:

Ashutosh Shivakumar, Ph.D.

Yong Pei, Ph.D. (Co-Advisor)

Thomas Wischgoll, Ph.D.

Paul J. Hershberger, Ph.D.

Shu Schiller, Ph.D.
Interim Dean of the Graduate School

ABSTRACT

Heng, E Jinq. M.S. Department of Computer Science and Engineering, Wright State University, 2022. A Cloud Computing-based Dashboard for the Visualization of Motivational Interviewing Metrics.

Motivational Interviewing (MI) is an evidence-based brief interventional technique that has been demonstrated to be effective in triggering behavior change in patients. To facilitate behavior change, healthcare practitioners adopt a non-confrontational, empathetic dialogic style, a core component of MI. Despite its advantages, MI has been severely underutilized mainly due to the cognitive overload on the part of the MI dialogue evaluator, who has to assess MI dialogue in real-time and calculate MI characteristic metrics (number of open-ended questions, close-ended questions, reflection, and scale-based sentences) for immediate post-session evaluation both in MI training and clinical settings. To automate dialogue assessment and produce instantaneous feedback several technology-assisted MI (TAMI) tools like ReadMI based on Natural Language Processing (NLP) have been developed on mobile computing platforms like Android. These tools, however, are ill-equipped to support remote work and education settings, a consequence of the COVID-19 pandemic. Furthermore, these tools lack data visualization features to intuitively understand and track MI progress.

In this thesis, to address the aforementioned shortcomings in the current landscape of TAMI, a web-based MI data visualization dashboard tool ReadMI.org has been designed and developed. The proposed dashboard leverages the high-performance computing capacity of cloud-based Amazon Web Service (AWS) to implement the NLP-based dialogue assessment functionality of ReadMI and a vibrant data visualization capability to intuitively understand and track MI progress. Additionally, through a simple Uniform Resource Locator (URL) address, ReadMI.org allows MI practitioners and trainers to access the proposed dashboard anywhere and anytime.

Therefore, by leveraging the high-performance computing and distribution capability of cloud computing services, ReadMI.org has the potential to reach the growing population of MI practitioners and thereby provide a pathway for large-scale MI adoption.

TABLE OF CONTENTS

	Page
CHAPTER 1	1
1.1 Introduction	1
1.2 Existing Solution: Real-time Assessment of Dialogue in Motivational Interviewing (ReadMI).....	3
1.2.1 Android-based ReadMI Application	3
1.3 Challenges with ReadMI Implementation.....	4
1.4 Solution: ReadMI.org.....	5
CHAPTER 2	7
2.1 Research and Planning	7
2.1.1 Video Conferencing Tool	8
2.1.2 File Upload Requirement.....	10
2.1.3 Remodeling the Interface.....	11
2.2 Developing The User Interface	12
2.2.1 Visual Styling	12
2.2.2 Creating the Webpage	13
2.3 Building the Foundation.....	16
2.3.1 HTTP Endpoints for API Requests	17
2.3.2 Classifier Implementation.....	19
2.4 Setup and Deployment	21

TABLE OF CONTENTS (Continued)

2.4.1 Selecting the Right AWS Service.....	21
2.4.2 Setting Up the Environment	22
2.5 Final Touches	24
CHAPTER 3	27
3.1 Results	27
3.2 Discussion	30
CHAPTER 4	31
4.1 Conclusion and Future Works.....	31
REFERENCES	43

LIST OF FIGURES

Figure	Page
1. Android-based ReadMI Application.....	4
2. ReadMI.org Planning and Website Development Process	7
3. Additional Metrics and Summary Display Windows	11
4. Postman Interface Displaying a Successful HTTP Get Request	18
5. Donut Chart Showing the Doctor vs Patient Speaking Time	26
6. Bar Graph Showing the Count of Different Question Types.....	26
7. Password Access Page Log In with an Incorrect Password Input	27
8. The Landing Page of the ReadMI.org Dashboard	28
9. ReadMI.org Dashboard After a Successful Transcript Upload	29
10. ReadMI.org Visualization Dashboard in a New Tab.....	29

LIST OF TABLES

Table	Page
1. MI Metrics	3
2. Common HTTP Response Codes and Their Meaning.....	17
3. Virtual Hardware Configuration of an AWS EC2 t2.small Instance.....	23

ACKNOWLEDGEMENT

First and foremost, I cannot express enough thanks to my research advisors, Dr. Ashutosh Shivakumar and Dr. Yong Pei for their trust, dedication, and support from the very beginning to the end of this process. Without them, I will not have had a chance to work on such a meaningful project and be a part of this milestone of ReadMI.

Over the short period that we have worked together, Dr. Ashutosh Shivakumar has challenged me to push myself harder to achieve what is bigger. He has the vision and he put his trust in me to achieve it. He is extremely devoted to his students and I appreciate the countless hours he has invested discussing the project with me to improve it and make deadlines.

Dr. Yong Pei has always reminded me of a warm and accepting father figure. He gave me the autonomy to work and expand my creativity. His advice has always been valuable and managed to provide clarity and steer me back when I am too far off the track. Thank you for trusting me and giving me room to grow.

Not to forget my beloved committee members, Dr. Thomas Wischgoll and Dr. Paul J. Hershberger. Dr. Thomas Wischgoll was one of my first instructors at Wright State University. He stood by his teaching and often spent hours online answering my questions over Discord when COVID first turns everything to the remote world. I still considered his class one of the most challenging classes I have taken so far.

Dr. Paul J. Hershberger has never missed any ReadMI sessions that were scheduled. As the topic expert, he has provided very valuable feedback to make the development process easier, and the developed application better. Thank you for letting me in and being a part of ReadMI.

Most importantly, I will not be who I am today without the support of my husband, family members, friends, and their trust in me. Thank you all.

DEDICATION

This achievement is dedicated wholeheartedly to my husband, Orion, my three fur babies, Lion, Jellybean, and Cannoli, friends, and family.

Orion, thank you for bringing me Starbucks and breakfast when I have to pull an all-nighter. You are a supportive husband and have always put trust in me even when I'm doubting myself. All that I have achieved today was not possible until the day I meet you. Thank you for being my anything and everything. January 2, 2021, marrying you was the best decision that I have made, I love you.

I would also like to dedicate this work to my cat, Lion. Lion has been with me since the very beginning through ups and downs, and countless mental breakdowns. Thanks for being my mental support and always healing me. Adopting you in 2017 has led to events that changed my whole life and I will not know what to do without you. Also welcome to the family, Jellybean and Cannoli!

A special feeling of gratitude to my parents, Heng Chin Hock and Low Lay Hoon for giving me a chance to study abroad and have the courage to find wonderful things in life. I am also grateful to have my baby sister, E Shan, and my grandma, Ah Eng, who makes me feel like I'm home when I miss home. To my in-laws, other family members, and friends that I can't address personally, I want you all to know that it is great to have you around and I am beyond grateful to share my achievements with you all.

Thank you all for being my cheerleaders.

CHAPTER 1

1.1 Introduction

Effective communication acts as more than just a tool to trade information. It allows for conveying emotions and intentions, both implicitly and explicitly. While it seems like efficacious communication is instinctive, communication-based on empathy and encouragement is not always the case. Communicating effectively and efficiently is an art form in itself.

Motivational Interviewing (MI) is one such communication style that evolved from Carl Roger's person-centered approach and has gained popularity, especially in an area such as healthcare. The essence of MI lies in empowering people to proactively take more autonomous responsibility toward their behavior change, rather than being directed or forced.

The MI technique itself was invented in 1983 when psychologists William R. Miller and Stephen Rollnick first introduce the technique to the mental health field [1]. The technique has since been widely adopted in the fields of health care, criminal justice, education, and most recently into sports.

There exist multiple studies on the successful application of this evidence-based approach. Lev-Ran & Nitzan has employed this technique to increase the patient's adherence to treatment and to reduce the rate of hospitalization, mortality, and morbidity [2]. The same has also been carried out by Levensky, et al., stressing the importance of teaching nurses on utilizing the MI approach to promote

behavioral change while helping patients explore and resolve incongruity. Some other contexts in which MI is effective include smoking, substance use disorder, weight loss, cancer, and diabetes care [3].

Due to such widespread applications of MI, it has been adopted into the curriculum of medical education. In a typical MI training session, a MI expert who is the trainer will monitor the MI dialogue between the trainee and the client. The trainee is typically a medical student, and the client is typically a model patient. At the end of the dialogue session, the expert provides qualitative feedback to the trainee. Typical feedback to a MI session consists of both the qualitative and quantitative aspects to it. The qualitative feedback consists of the “Spirit of MI” described by Miller and Rollnick, i.e. (How?) a MI session was conducted and the quantitative aspect, consists of MI metrics like 1) The number of open/closed questions, 2) Scale-based sentences, and 3) Reflective sentences as shown in Table 1 [4]. The typical evaluation process of manually assessing and compiling the quantitative metrics is time-consuming, thereby delaying the feedback. Moreover, the need to simultaneously assess each trainee's dialogue utterances and compile the quantitative metrics poses a cognitive overload on the MI trainer [5].

Consequently, there exists a need to simplify the process by automating the quantitative metrics generation of MI sessions. This can be done by providing instantaneous feedback following a session, which act to reduce the feedback time

and relieve the cognitive load of the MI trainer resulting from the manual assessment of each of the sessions.

Table 1: MI Metrics

	Example	Reasoning
Open Question	“How are you feeling today?”	Allows for a variety of responses to describe the feeling
Close Question	“Are you feeling happy?”	Only allows for a yes or no response
Reflective Sentence	“So it looks like you are having a bad day”	This sentence allows the client to reflect on their experiences
Scale-based Sentence	“From a scale of one to ten, how ready would you be to begin walking 30 minutes each day?”	The sentence assesses the readiness for change

1.2 Existing Solution: Real-time Assessment of Dialogue in Motivational Interviewing (ReadMI)

1.2.1 Android-based ReadMI Application

An Android-based application ReadMI, as seen in Figure 1 is an existing solution that was developed to address the need to automate MI assessment. It consists of two tablet computers that receive audio inputs from the doctor (MI trainee) and the client (a model patient). It utilizes Natural Language Processing

(NLP) in its classification of MI trainee-client dialogue utterances and generates quantitative MI metrics with an overall accuracy of over 92% in measuring practitioner versus patient conversation time, question type (open/close) classification, and, reflective and scale-based statement detection [6].

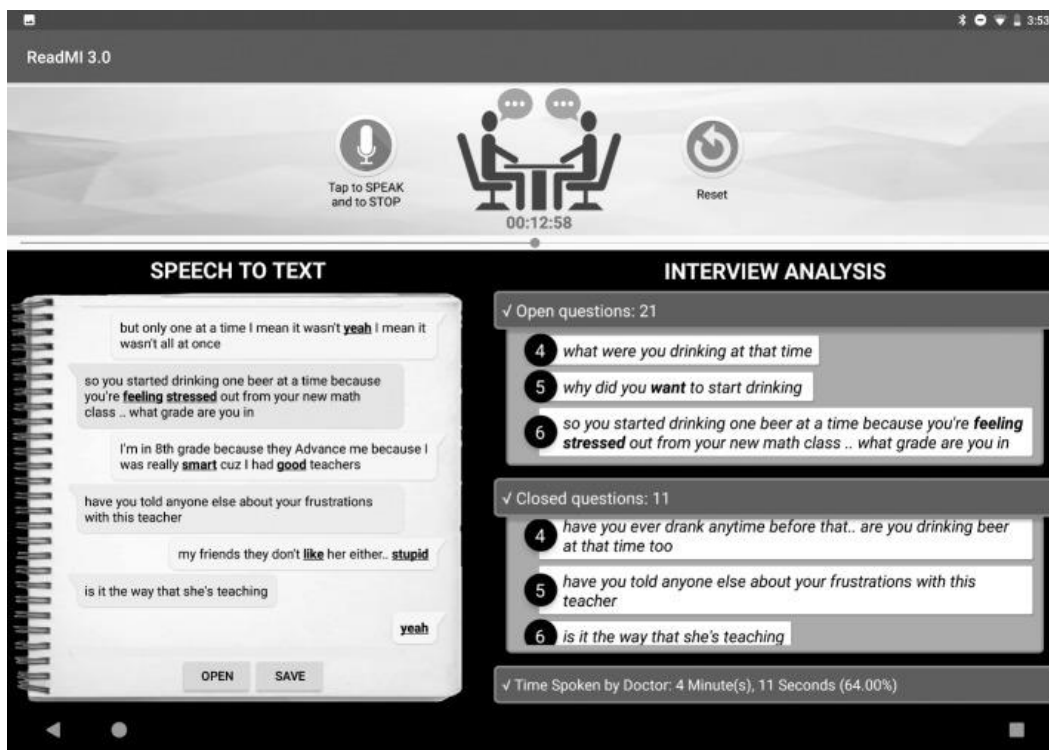


Figure 1: Android-based ReadMI Application [5]

1.3 Challenges with ReadMI Implementation

The limitations of the Android-based ReadMI application are that 1) It is a localized implementation where the Android tablets communicate over Bluetooth while the doctor and the patient conversation in the same location separated by approximately three meters; 2) The current operation of this Android-based

prototype relies heavily on having a specialized dedicated “ReadMI operator” on-site during the interview dedicated to operating the Android tablets. This is a cost-intensive process and acts as a barrier to disseminating ReadMI as a personalized training application service. Moreover, social distancing requirements imposed by pandemics like COVID-19 require participants in MI training sessions to collaborate remotely through a distributed version of ReadMI; 3) The Android implementation of ReadMI does not provide a feature to intuitively visualize generated ReadMI metrics.

1.4 Solution: ReadMI.org

In this thesis, a cloud-computing-based implementation of ReadMI called ReadMI.org is developed. ReadMI.org is a web-based application that can be accessed using a simple Uniform Resource Locator (URL) address on the World Wide Web (WWW). This allows ReadMI to be accessed anywhere at any time. It also aims to provide a more user-friendly and secure interface that is password protected and can be managed solely by a user without any external help from a trained technician. A ReadMI.org user like the MI trainer can upload an Automatic Speech to Text (ASR) transcribed MI dialogue to ReadMI.org and obtain the MI metrics directly.

Furthermore, ReadMI.org is capable of rendering vibrant visualizations of the generated MI metrics for an intuitive understanding of the MI session. These

visualizations are designed to provide valuable insights into each MI session thereby enriching the training experience.

CHAPTER 2

2.1 Research and Planning

The development of ReadMI.org comprises five stages, as shown in Figure 2. The five stages are, 1) Research and Planning, 2) User Interface/User Experience (UI/UX) Development, 3) Backend Development, 4) Deployment to the Cloud, and 5) Addition of Visualization and Password Protection.

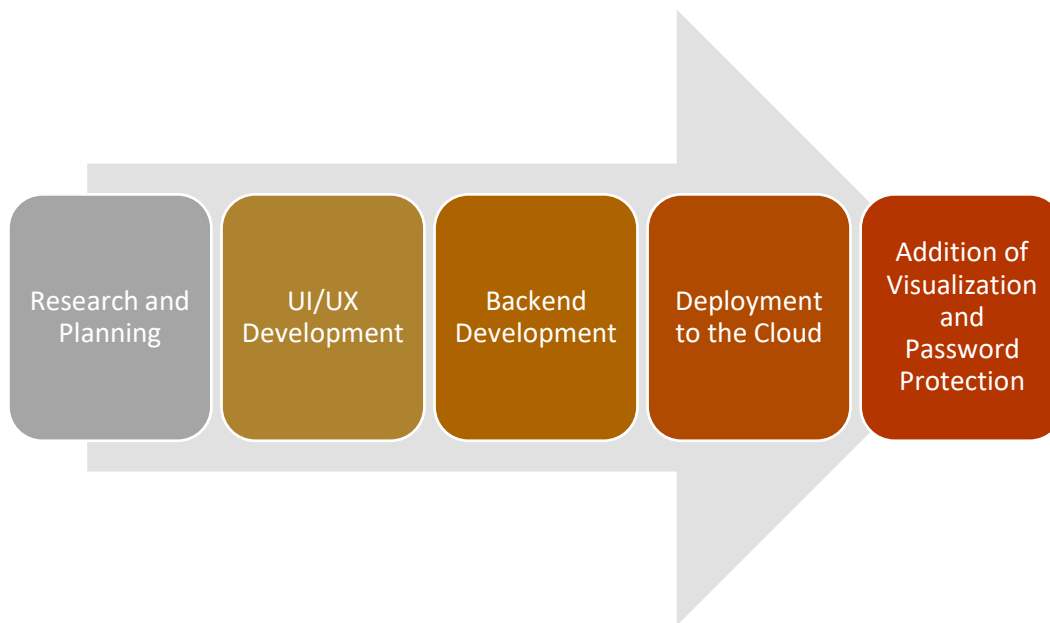


Figure 2: ReadMI.org Planning and Website Development Process

The initial idea of decentralizing ReadMI intends to delocalize and adapt to the everchanging need for a remote tool. When the COVID pandemic starts hitting in twenty nineteen, the freedom to conduct an in-person MI training session was compromised. Weighing the importance of continuing to contribute to healthcare whilst practicing social distancing, the association with a video conferencing tool

was deemed an important element for this effort. With sessions carried out remotely regularly using a video conferencing tool, the existing speech-to-text feature was made abundant. The need to record a session using the speech-to-text button was removed as the intention can be achieved by including a transcription upload feature.

2.1.1 Video Conferencing Tool

The initial idea to involve a video conferencing tool was to incorporate a built-in video conferencing tool within ReadMI.org itself. The first option that came across was JITSI Meet, an open-sourced online video conferencing platform. JITSI Meet was initially considered to be used for its simple and scalable capability. It allows for the video conferencing tool to be used as a standalone application through website links and downloadable mobile applications on both Android and iOS platforms. The ability to use it as an embedded application makes it stand out among all other entrants. Attracted by the flexibility, all initial planning was based on incorporating JITSI Meet. Alas, it was then later found that JITSI Meet does not meet the Health Insurance Portability and Accountability Act of 1996 (HIPAA) Compliance requirements out of the box. Making JITSI Meet compliant as an embedded application requires additional cost and laborious efforts, including but not limited to implementing multi-factor authentication and encryption, implementing Bluetooth security, creating Virtual Local Area Networks (VLANs), and monitoring data or voice traffic in real-time.

With the complications, it only makes sense to move to another video conferencing tool that is prebuilt with what was best suited for our purpose. Another option available that was widely used was Microsoft Teams. When configured and used in a HIPAA-compliant manner, Microsoft Teams fits the bill of what was looked for.

Further research also changes the direction of the planning to involve and embed a video conferencing tool within ReadMI.org. It was later found out that a real-time transcription that returns immediate feedback was not only expensive, but it is close to impossible to implement with the speech-to-text error during a meeting session. For an accurate transcription to be generated, a sentence will be revised post-generation to better fit the context of the conversation after the message has been conveyed. Real-time transcription has a lower accuracy level. One solution found to address the issue of real-time transcription was to employ a third-party transcribing service, such as Verbit. Verbit provides seamless integration to different video conferencing platforms but to achieve real-time accurate captioning and transcribing, it requires a manual transcription of a session by another person. The session would have to be pre-planned and the cost would be estimated based on the duration of the meeting itself. Ultimately, this is not ideal and it also defies the original intention of making ReadMI an accessible tool to be used without additional support from another person.

Therefore, the basis of ReadMI.org is established on an uploaded Microsoft Teams transcript file, after which a session is conducted. Microsoft Teams can generate meeting transcripts equipped with the speaker names, the timestamps, and a clear breakdown of the utterances from either speaker. Although optional, it allows the freedom and autonomy of the user to review the accuracy of the session and make necessary changes to the editable text file before it is uploaded for analysis on ReadMI.org. Some changes include modifying inaccurate transcription, changing the name to protect the identity, or breaking a long meeting into multiple text files containing multiple sessions respectively.

2.1.2 File Upload Requirement

The ReadMI.org has been developed based on the meeting transcript generated from Microsoft Teams. With that being said, the file to be uploaded on ReadMI.org has to match a specific format to carry out the MI metric analysis. There are three elements in an approved transcript: 1) Timestamp in the format of hh:mm:ss.SSS --> hh:mm:ss.SSS indicates the starting and the ending time of an utterance, 2) The Speaker name with an optional “(Guest)” tag, and 3) Clear and correctly transcribed utterances. These three elements must start from a new line and follow by each other in the exact sequence. The required file format is in a text file with “.txt” format, and the said file should be renamed to match the name of the interviewer, which is the practitioner in most cases. A sample was provided under Appendix A.

2.1.3 Remodeling the Interface

The ReadMI.org website application aims to be an improved version of what has already existed and not fix what is not broken. Therefore, the decision was to keep the layout and positioning of analysis result display windows intact as an easy-to-navigate dashboard. Aside from cosmetic overhaul to how the dashboard looks, eliminating the speech-to-text panel allows for more spaces within the viewport. This decision allows for an opportunity to add two additional panels to display the Reflective Statement type and the Scale Statement type. Apart from that, more metrics were also incorporated along with an included overview of the content within the uploaded transcript file as shown in Figure 3.

Reflective Statement : 3	
1.	Umm, I can tell you've really put a lot of thought into this and you have realized the effect of drinking.
2.	I hear you. Sounds like you had a really tough time.
3.	I'm glad you found your hope and something to fight for. I hope to hear from you again in our next appointment.
Scale Statement : 0	
Summary	
Conversation Size: 34	
Doctor Time Spoken: 1.62 minutes	
Doctor Speaking Time (%): 53.24	
Patient Speaking Time (%): 46.76	
Ratio of Refl. vs Questions: 0.43	

Figure 3: Additional Metrics and Summary Display Windows

Additional quantitative metrics were 1) conversation size indicating the total utterances in the transcript, 2) doctor time spoken (in minutes), 3) doctor speaking time (in percentage), 4) patient speaking time (in percentage), and lastly 5) the ratio of reflective statements versus other questions. These metrics enable a better assessment of who is leading most of the conversation in a MI session as a good MI session should have the patient leads the conversation instead of the doctor.

2.2 Developing The User Interface

The first step for the development effort is to develop a user interface that has a clear and user-friendly dashboard that allows for instructive operation and easy-to-understand analysis results. Apart from the header and footer, the viewport was divided into the left and right columns. The left column consisted of an instruction, an upload button, and the transcript content to be displayed after a successful upload. The right column, on the other hand, contains multiple windows that display the analysis result and summary. A button that leads to a new tab with result visualization was also added.

2.2.1 Visual Styling

With ReadMI being an ongoing research project of Wright State University, the website was styled in the color theme that represents green (#046A38) and gold (#CBA052) colors. Removing all extra decorative elements such as the icons, the dashboard is made with a minimalist concept in mind with a black background that reinforces striking visuals and allows for more contrast and a vivid representation

of data. White sans serif font was chosen to continue adhering to the minimalist school of design. The defaulted font families were selected from the commonly used font pools, such as Arial and Helvetica for maximum compatibility and to cater to different browser settings.

Upon successful upload of a transcript file, the context of the file will appear on the left column above the file upload button, grouped, organized, and displayed within alternating green or gold rounded borders based on who is speaking. The use of negative text, that is the white font on a black background is as equally good as it is with positive text (black font on a white background). To further improve readability and legibility, font size sixteen was made standard to details bearing texts across the webpage. All these customizations were made possible with HyperText Markup Language (HTML) and Cascading Style Sheets (CSS) styling.

2.2.2 Creating the Webpage

The subsequent step after deciding on the aesthetic of the website is to work on the functionality. Multiple frameworks were considered during the process, but it was eventually decided between ReactJS and Angular 14. Both ReactJS and Angular are popular with the resemblance that they are built on JavaScript. Each framework has its group of supporters and has similar capabilities and performance. All in all, it came down to which language is simpler to be maintained by future developers and if the language is capable of doing what was asked.

Firstly, ReactJS can adapt to a bigger application with its scalability and rich JavaScript library. With the declarative paradigm of ReactJS, simple views of each state can be designed based on the emerging need. A few complications have arisen during the initial development using ReactJS. Initially, the application was set up based on a promise handling from the backend. It was requiring a lengthier response time for the classifier result to be returned to the response window. The response time for the classifier to work with a Python Based Server averaged five minutes. This is not ideal for a website application, especially when it was anticipated to handle requests from more users eventually. The second problem that was observed was that it was not suggesting the correct question count in the window. These problems were all traced back to the way a file is parsed and sent to the processing end. Instead of sending the uploaded file to the backend for processing and getting a package of analyzed results, individual analyzed results were fed from the backend asynchronously, stacking on top of the previous result. One final problem that was raised was the inability to support back/forward caching due to the file upload capability.

The effort was then shifted to Angular and was approached differently. Angular comes with various templates for the creation of UI views that can support and provide syntax during website creation. Apart from that, Angular utilized the RxJS library that supports reactive programming involving observables to better compose asynchronous and callback-based code. Using a subscriber, the analyzed

data can be expected from the back end and a utility to iterate and filter through the values can be used. Next, Angular Core also supports `ChangeDetectorRef`. The `ChangeDetectorRef` is a base class that allows for change detection within the views that were marked to be observed. In this case, the right side of the viewport containing data analysis results is added to the `ChangeDetectorRef` tree. When a new file is uploaded, a new data analysis result is received from the back end. The views will be marked as dirty and thus allowing the latest data analysis result to be updated and shown.

Another feature that makes Angular preferable is its routing support. This plays an important role in a single-page app. The viewport can display different components depending on what task a user performs. For example, the router enables navigation to the visualization view solely from the interpretation of the visualization browser address. Another benefit is the use of components in Angular. There can exist many different components within a base page and when a certain condition is met, the respective component can be shown. This is shown by the implementation of the password input page on [ReadMI.org](https://readmi.org). When a correct password is entered, the password input component will be hidden and the ReadMI dashboard is shown without having to exit the server to get to another new page. It is also worth mentioning that it is straightforward to set up an Angular project coupling with the Angular Command Line Interface (CLI) installed. With the

Angular CLI, a user can initialize, develop, scaffold, and maintain the application directly from a command shell, such as Git Bash.

A few other features that make Angular preferable include two-way data binding, dynamic HTML templating, and unit testing. Any changes can also be reflected on the website immediately upon saving, reducing the need to restart the server whenever a change is made to the source code. Angular also supports easy version updates through a simple command.

2.3 Building the Foundation

Moving on, the next step after the face of ReadMI.org was set up is to get the classifier to process the uploaded file and output result. At this point, the user can upload a transcript file but no change will be shown on the user interface as nothing is being processed. The Angular Hypertext Transfer Protocol (HTTP) Client was involved to allow for the file to be uploaded and analyzed using the ReadMI classifier. Angular HTTP Client allows the front end to communicate with the backend, whilst the RxJS library mentioned provides it with a subscriber to retrieve the response. Each of the responses carries a message represented by a code from the server and they can return different data packets containing the required data. Correspondingly, there are different response codes grouped into informational, redirection, client error, or server error. The more commonly seen response codes are 200 for Success/OK, 403 for Forbidden, and 404 for Not Found, as shown in Table 2.

Table 2: Common HTTP Response Codes and Their Meaning

Code	Status
1xx Informational	
2xx Success	
200	Success / OK
3xx Redirection	
301	Permanent Redirection
302	Temporary Redirection
304	Not Modified
4xx Client Error	
401	Unauthorized
403	Forbidden
404	Not Found
405	Method Not Allowed
5xx Server Error	
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout

2.3.1 HTTP Endpoints for API Requests

The original Java backend has been modified slightly to retrieve the calls from the HTTP Client. Multiple endpoints were created for the convenience of receiving and handling HTTP calls. These endpoints were tested using a third-party Application Program Interface (API) platform, Postman. The use of Postman has

ensured that the endpoints function correctly. This ensures that the web server is getting the appropriate request and is returning the expected response. This method of testing is not only easy to execute with no extra setups or requirements but it can also be integrated into a Continuous Integration and Continuous Development Pipeline (CI/CD) in the future. An example of API testing is shown in Figure 4.

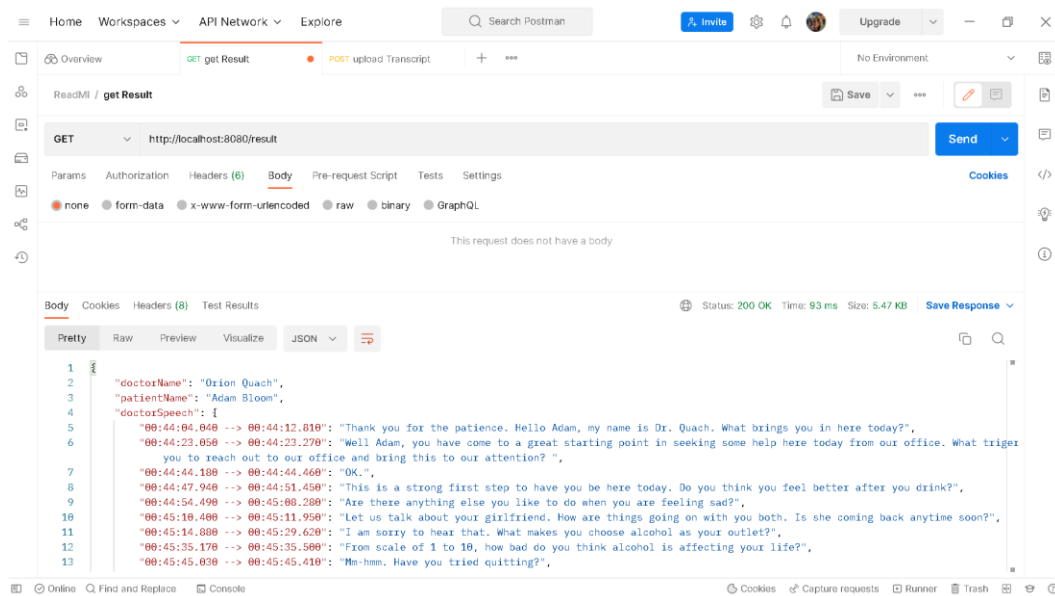


Figure 4: Postman Interface Displaying a Successful HTTP Get Request

The two types of HTTP requests set up for the endpoints are the POST and GET requests. Since there is no need for editing and storing data, ReadMI.org did not utilize other request types, such as PUT or DELETE. The POST request is used to ingest a MultipartFile uploaded by the user using the “Choose File” button. There is no restriction during the point of upload on what type of file can be uploaded, but a validation process was to be carried out once the file content is converted into

bytes. However, this specific POST endpoint will print a message onto the terminal to indicate any exception from upload or if the uploaded file is zero bytes.

The next HTTP request type is the GET request. As the name suggested, the GET request returns a response that carries the information asked by the call. Apart from what has been shown in Figure 4, multiple other endpoints were created to return a minor but more specific response. This is beneficial as the response is straightforward to what was asked and there is no further processing needed in the Angular end to process the response package just to obtain a piece of specific information. For example, instead of requesting the entire package of results containing all analyzed data as shown in Appendix B, the user can call a specific GET request to the endpoint that only contains the classified Open Question.

2.3.2 Classifier Implementation

As mentioned, changes have to be made for ReadMI.org to reuse the existing ReadMI classifier[7] for the processing of an uploaded transcript. A new classifierService class was added to handle the uploaded text file and prepare it in the desired format. A new Transcript object is also created with every uploaded file to make sure that no data is being retained and that every analysis can start from a clean slate. This will not only reduce the memory needed when it is being hosted on a server, but it also acts to reduce concern about data storage, security, or compliance with the HIPAA policy.

The sophisticated process for a metric to be generated from a transcript starts from the processing of the uploaded file. First, validation was executed to ensure that the uploaded file ends in a “.txt” extension. When it passes the first check, the file will move into the input processing cycle. At the same time, the file name was also obtained and the value is stored as the doctor’s name. The processInput() function takes an InputStream type file and reads it line to line using a reader into a transcript array. After the transcript array is filled, it was iterated, and each line is then checked for its pattern to determine if it represents the timestamp format. As required by the supported file requirements, the line above a timestamp line is the name of the doctor or patient. By comparing this value with the doctor name obtained earlier from a file name, the utterance from the following line will be stored in either the doctor or patient array. At the end of the process, there will be two arrays with the timestamp and the utterances of the doctor and the patient respectively. This separation allows for only the doctor’s utterances to be analyzed and prevents the patient’s utterances to be analyzed accidentally.

During the classification process, four new array lists were created to contain the classified utterances. These utterances were distributed into their group of open, closed, reflective, scale, or other. This will result in four carefully grouped utterances ready to be sent back to the user interface. Another process that was also undergoing synchronously is the calculation of speaking time. The time was calculated down to milliseconds and recorded as the doctor's speaking time and the

patient speaking time. This information will then be processed again and converted into a user-readable format, such as in minutes, ratios, or percentages in the user interface.

2.4 Setup and Deployment

When the application was set up, ran, and tested successfully locally, multiple options were explored to have the application hosted on a server to achieve the true delocalization of ReadMI. There were multiple options available on the market but after careful consideration, it was decided that Amazon Web Services (AWS) has the cost factor and features that better suit the needs. The three major services provide were Elastic Compute Cloud (EC2), Elastic Beanstalk, and Amplify.

2.4.1 Selecting the Right AWS Service

The AWS EC2 is a scalable cloud computing platform that runs on hosted virtual machines, called “instances”. With each instance, it allows for its own private and public IP address and a virtual hardware configuration. EC2 allows multiple servers to be hosted as multiple instances. Users can select the optimum or most cost-effective combination to bring their application to the cloud. There are also several pricing structures for EC2, including on-demand, spot instances, saving plans, reserved instances, or dedicated hosts, which will guarantee that either of the plans will cater to one need. In another word, EC2 provides a fast and reliable server in the cloud, is scalable, and has a lower cost to use.

One major competitor to EC2 is the AWS Elastic Beanstalk. The quick, painless setup and vastly available sources and documentation online make it a popular service among developers. Elastic Beanstalk is great when many other AWS services are used. It provides a quick and easy way to integrate other services such as the S3, a cloud object storage service.

The third option was AWS Amplify. Amplify is great for full-stack applications and can work with different front ends. Amplify is offered as a Backend as a Service (BaaS) that allows the use of their powerful CLI tool outside of the AWS console to configure, manage, and deploy static web applications. One nice feature to have from Amplify is the CI/CD pipeline. However, it is being used more for the application that features authentication and data storage. By comparing Amplify and EC2, EC2 allows for a more advanced feature such as operating system (OS) customization. As there is no need for the extras feature provided, EC2 was selected as the best option among the three.

Aside from the EC2, a domain name (readmi.org) and hosted zone were also acquired and set up from AWS Route 53. This makes a seamless integration process and has the application up and running in no time.

2.4.2 Setting Up the Environment

EC2 instances were set up after selecting the services needed. The AWS local zone locations were selected as us-east-2, which refers to the data center region in Ohio. The project initially has two running instances of t2.micro, with

each supporting the Angular front end and Java backend respectively. However, after experimenting with the different number of instances and different instance types, it was found that the configuration with a single t2.small instance is the most uncomplicated with minimal tradeoffs.

Currently, ReadMI.org is running on a t2.small instance on a Linux/Unix platform. The platform is running on Ubuntu Operating System (Ubuntu OS) and has one virtual CPU, 2.0GiB memory, and other configurations as shown in Table 3.

Table 3: Virtual Hardware Configuration of an AWS EC2 t2.small Instance

Compute	Value
Virtual CPUs (vCPU)	1
Memory	2.0
Memory per vCPU	2.0
Physical Processor	Intel Xeon Family
Clock Speed	3 GHz
CPU Architecture	I386
GPU	None
Video Memory	0 GiB

Due to the instances type, the Linux server does not come preinstalled with what was required. The server environment must be set up as close to the local running

environment to support the ReadMI application. Running the Angular front end requires Angular version 14.2.7 to be installed. A Node JS version 14.x was also required and installed. Aside from that additional tools such as Angular CLI and the Apexcharts that were used for the visualization were installed.

For the Java backend to be running seamlessly, a few more requirements listed below were also installed. ReadMI utilizes Apache Maven 3.8.6 to automatically install all the dependencies needed. Aside from that, a Java Development Kit (JDK) is needed for the Java backend and thus Java 17 is installed in the virtual server. For NLP to work, Apache OpenNLP 2.0.0 is also required. Along with that, the extra downloadables that are needed with Apache OpenNLP is the `em-lemmatizer.dict`, `en-pos-maxent.bin`, and `en-token.bin`. With these requirements fulfilled, the Java classifier can be run with maven `spring-boot`.

Moving on, the production environment variable of the Angular front end was set to point to the Java back end. Once both the angular and Java classifier is verified to run, testing was performed by uploading a transcript to check if any errors are showing. The web application was working as intended and is showing the classifier results and summary on the interface, proving that the setup was successful.

2.5 Final Touches

It was brought to the attention that a password-protected page and visualization were needed to better enhance the website. With Angular, it is simple

to create a simple password protection page. At this point, there is no user-specific validation involved to verify access and authorization. While that is a plan, the application does not retain and store any data and thus password protection only exists to limit unauthorized use of the tool. The current password page has the availability to validate the input password and show an error message in red if it is invalid. As for right now, there is only one case-sensitive alphanumeric password available.

A “Show Visualization Result” button will appear after a transcript file is uploaded and analyzed successfully. Upon clicking, it will open up a new tab displaying the visual representation of the analyzed result. The visualization feature is a nice addition to provide a simpler and easy-to-understand result to the audience. There are two charts generated by the ApexCharts library, which is an open-source project licensed by MIT and is free for commercial application usage. The data was also input into the ApexCharts element by using an HTTP call.

There are two charts included on the data visualization page, including a donut chart on the left showing doctor versus patient speaking time (as shown in Figure 5), and a bar graph on the right showing the count of different question types (as shown in Figure 6). The use of a donut chart can show and highlight the percentage immediately without further interpretation. It is also ideal since it is only representing two data, 1) the doctor speaking time, and 2) the patient speaking time.. The most important data, the doctor speaking time is represented by the cyan color.

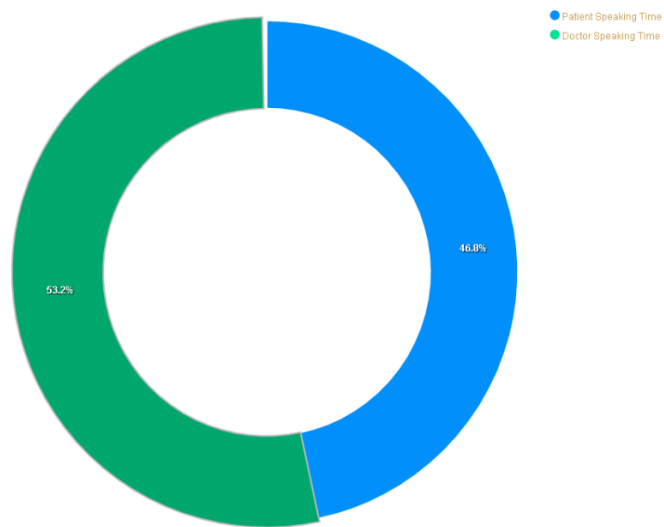


Figure 5: Donut Chart Showing the Doctor vs Patient Speaking Time

Next, the use of a bar graph can effectively show the distribution of different question types compared across the series. The bar graph was stripped of any unnecessary data such as the questions that were not classified to any group to prevent information overloaded.

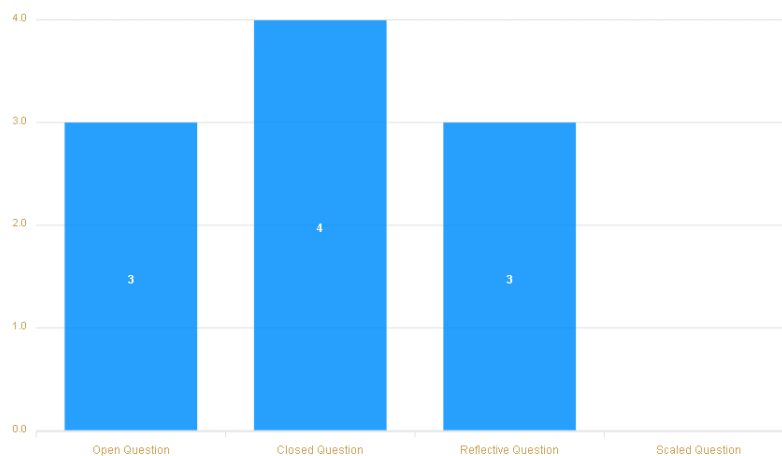


Figure 6: Bar Graph Showing the Count of Different Question Types

CHAPTER 3

3.1 Results

The resulting cloud-computing-based ReadMI.org web application is shown in Figure 7-10. These figures demonstrate the sequences of views presented to the user while they navigate ReadMI.org.

The first view of ReadMI.org is the password page. This page was added to the web-based ReadMI application as an additional security measure. This allows access only for those who hold the access password. The alphanumeric access password is case-sensitive. Figure 7 shows the password access page behavior when an incorrect password is entered.

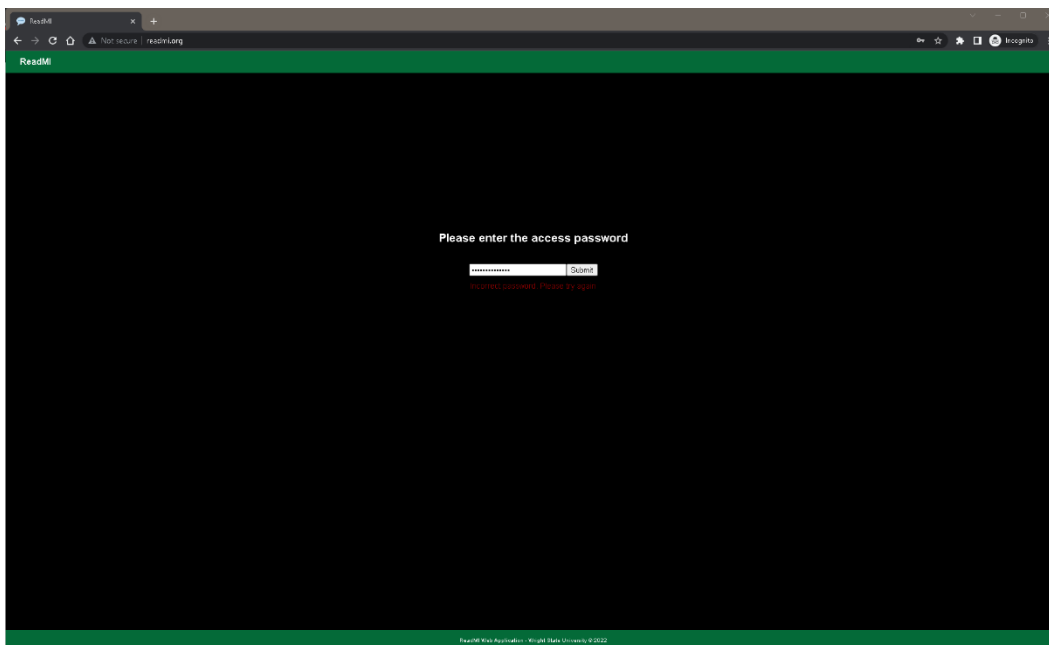


Figure 7: Password Access Page Log In with an Incorrect Password Input

Moving beyond the password input page, the user will be greeted with the landing page of the ReadMI.org dashboard as shown in Figure 8. The landing page is built with an easy-to-access dashboard template in mind. A user can click on “Choose File” on the left side of the viewport to select a file to be uploaded.

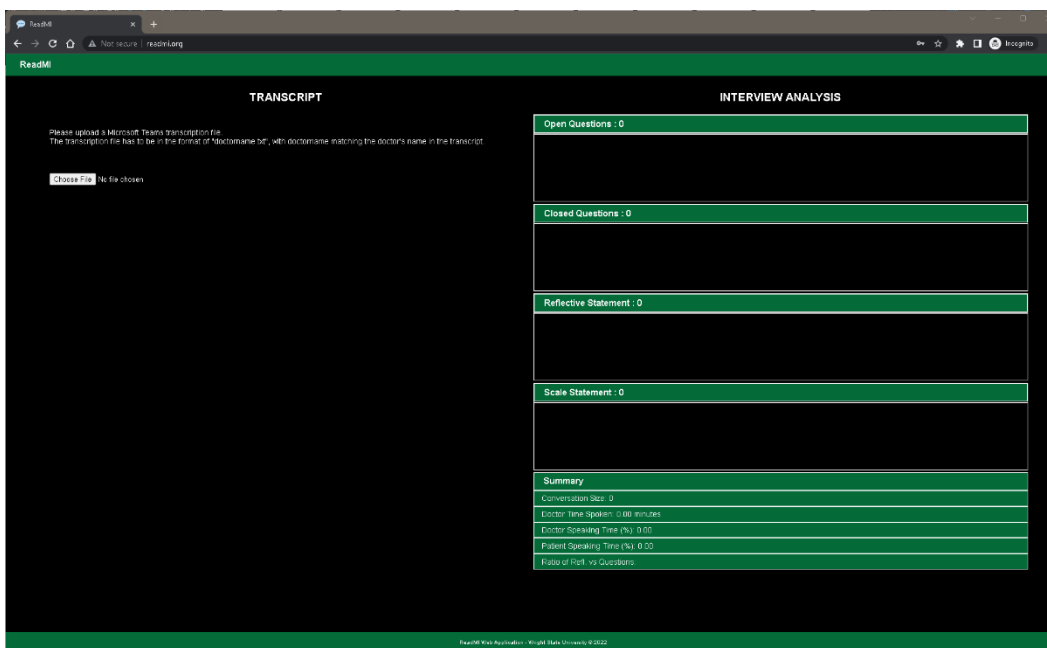


Figure 8: The Landing Page of the ReadMI.org Dashboard

Currently, it takes about 3 seconds for a MI session transcript to be classified and for the result to be rendered onto the ReadMI.org dashboard. An example is shown in Figure 9. The “Transcript” section of the dashboard is displaying the content of the transcript, with the different color borders in green and gold representing the doctor's and the patient's utterances respectively. The right side of the dashboard shows the interview analysis consisting of the classified utterances, grouped into open/closed questions, reflection, and scale statements.

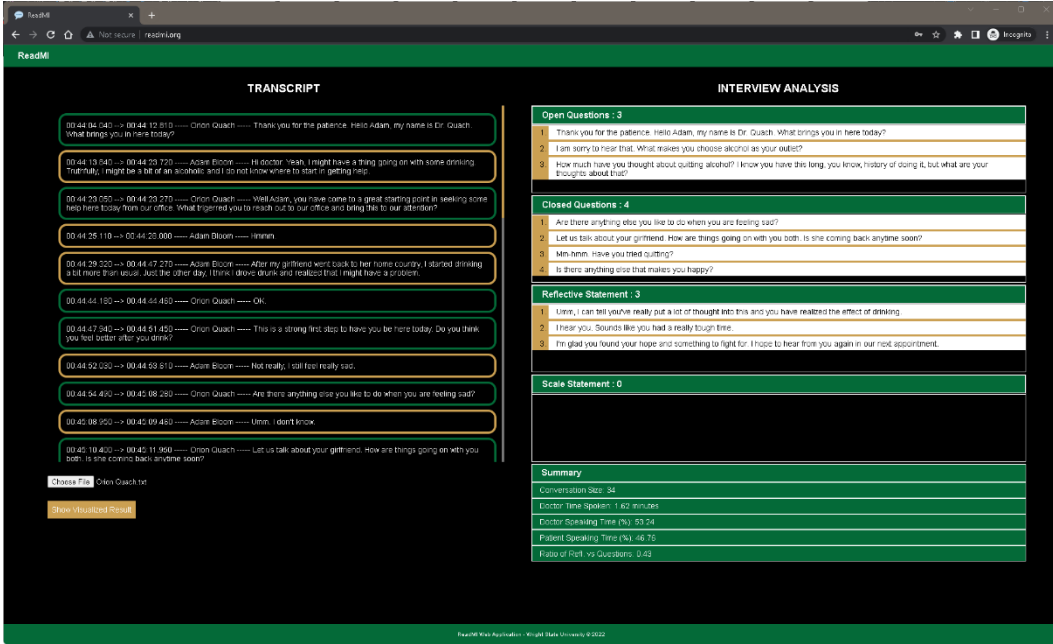


Figure 9: ReadMI.org Dashboard After a Successful Transcript Upload

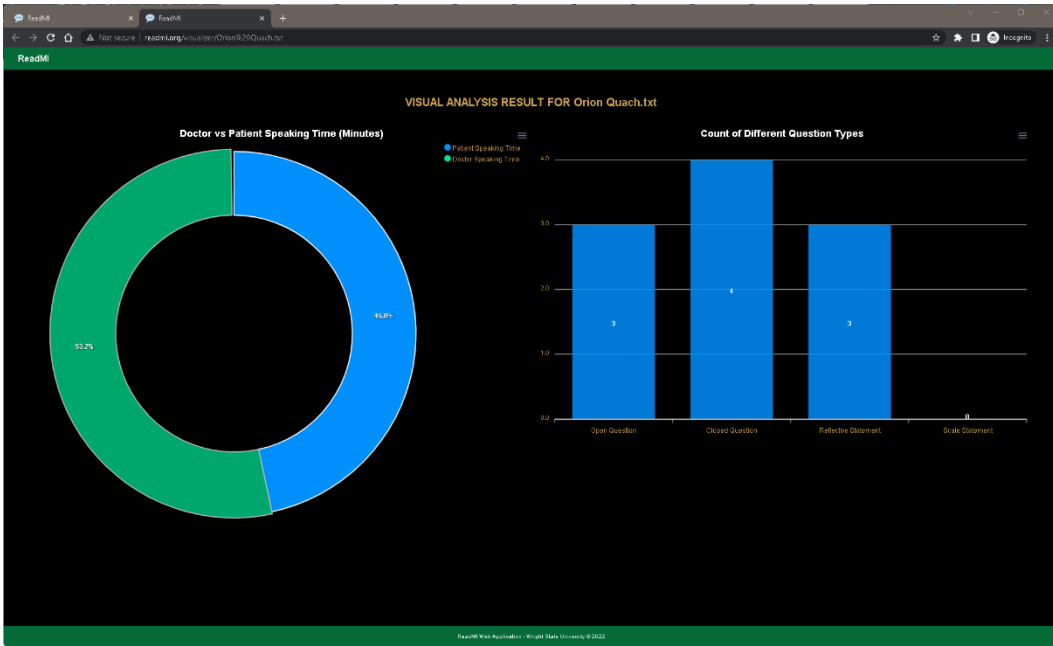


Figure 10: ReadMI.org Visualization Dashboard in a New Tab

Upon clicking the “Show Visualize Result” button, the metrics and summarized results will also be shown in the visualization dashboard in a new tab, as shown in Figure 10.

3.2 Discussion

With the provided solution, ReadMI.org has addressed the challenges of the original problems identified in the Android implementation of ReadMI. ReadMI.org can be easily accessed from the URL and allows access from anywhere and at any time without the need for a trained ReadMI technician. The user-friendly interface as shown in Figure 8 allows for a simple upload of a text transcript file and instantaneous session feedback can be obtained, as shown in Figure 9.

Figure 10 shows the visualization feature of ReadMI.org which provides a vibrant and beautifully designed graphical representation of the MI session result. This acts as a way to provide intuitive ReadMI metrics for further interpretation of data to meet the “Spirit of MI”.

CHAPTER 4

4.1 Conclusion and Future Works

Taken together, the implication is that incorporating the idea of distributed computing and transforming ReadMI into a web-based application allows for a more effective, efficient, and user-friendly workflow in MI. The cloud-based ReadMI application also overcomes the operational constraints presented by the need for a specialized ReadMI operator. Lastly, the dashboard visualization provides insightful visualizations in an easy-to-read manner.

There is always room for improvement and suggestions can be taken to make ReadMI.org better. Since the application is used the majority in a medical setting, accessibility options can be included. This can be easily implemented as Angular can enable attribute binding template syntax with Accessible Rich Internet Application (ARIA) enabled components. It could also be beneficial to improve the dashboard to include more analyzed data results or dynamic charts, should a user choose to see them.

Next, the idea of embedding a video conferencing feature is still tempting provided that there are more resources available. An embedded video conferencing tool where participants can view one another in real-time can enhance MI training experiences. This will eliminate the need for a third-party video conferencing tool and allows ReadMI to handle the MI training and feedback process all within the application itself realizing the holistic approach of a web application.

Another technical recommendation that would be useful is to build a secure user management page and a connected database with encryption. This can be used to replace the password access page and perform a proper user role validation. The current simple password access page was created to limit unwanted access but it can easily be reverse-engineered. With that being implemented, the data from the uploaded transcript can be stored under each user who performs the analysis and they can revisit the result and visualization without needing to reupload the file. As mentioned, this will involve an effort to make sure that the data storage and data transmission is HIPAA compliant.

Next, the use of Apache POI to process a docx file or other file type should also be implemented along with a different parser to process transcript files from different web conferencing applications for wider support and compatibility. Aside from that, there are opportunities for improved error handling, such as including a popup modal when a file is failed to upload. These recommendations can all greatly improve the user experience of ReadMI.org. Perhaps in the near future, the application can be improved to better suit the needs of those who use it and make a breakthrough within the MI and medical industry.

APPENDIX A

SAMPLE COMPATIBLE TRANSCRIPTION FILE

00:44:04.040 --> 00:44:12.810

Orion Quach

Thank you for the patience. Hello Adam, my name is Dr. Quach. What brings you in here today?

00:44:13.840 --> 00:44:23.720

Adam Bloom

Hi doctor. Yeah, I might have a thing going on with some drinking. Truthfully, I might be a bit of an alcoholic and I do not know where to start in getting help.

00:44:23.050 --> 00:44:23.270

Orion Quach

Well Adam, you have come to a great starting point in seeking some help here today from our office. What triggered you to reach out to our office and bring this to our attention?

00:44:25.110 --> 00:44:28.000

Adam Bloom

Hmmm.

00:44:29.320 --> 00:44:47.270

Adam Bloom

After my girlfriend went back to her home country, I started drinking a bit more than usual. Just the other day, I think I drove drunk and realized that I might have a problem.

00:44:44.180 --> 00:44:44.460

Orion Quach

OK.

00:44:47.940 --> 00:44:51.450

Orion Quach

This is a strong first step to have you be here today. Do you think you feel better after you drink?

00:44:52.030 --> 00:44:53.610

Adam Bloom

Not really, I still feel really sad.

00:44:54.490 --> 00:45:08.280

Orion Quach

Are there anything else you like to do when you are feeling sad?

00:45:08.950 --> 00:45:09.460

Adam Bloom

Umm. I don't know.

00:45:10.400 --> 00:45:11.950

Orion Quach

Let us talk about your girlfriend. How are things going on with you both. Is she coming back anytime soon?

00:45:12.670 --> 00:45:14.000

Adam Bloom

I am not sure. She is asking for a break because I do not make enough money to marry her.

00:45:14.880 --> 00:45:29.620

Orion Quach

I am sorry to hear that. What makes you choose alcohol as your outlet?

00:45:30.540 --> 00:45:51.170

Adam Bloom

I mean, I drank my whole life. I get drunk and it helps me not to think about my problem.

00:45:35.170 --> 00:45:35.500

Orion Quach

From scale of 1 to 10, how bad do you think alcohol is affecting your life?

00:45:38.090 --> 00:45:38.470

Adam Bloom

Hmm. I would say seven.

00:45:45.030 --> 00:45:45.410

Orion Quach

Mm-hmm. Have you tried quitting?

00:45:51.630 --> 00:46:14.120

Orion Quach

Umm, I can tell you've really put a lot of thought into this and you have realized the effect of drinking.

00:46:15.550 --> 00:46:22.090

Orion Quach

How much have you thought about quitting alcohol? I know you have this long, you know, history of doing it, but what are your thoughts about that?

00:46:22.800 --> 00:46:23.480

Adam Bloom

I mean.

00:46:24.200 --> 00:46:27.520

Adam Bloom

I know it is bad for me, but every time I try it just.

00:46:29.070 --> 00:46:30.370

Adam Bloom

I don't succeed. I feel sad.

00:46:31.080 --> 00:46:35.200

Orion Quach

Wow. So you've actually tried before. Tell me about your experiences there.

00:46:36.590 --> 00:46:47.760

Adam Bloom

Well, I just tried to stop like cold Turkey because I thought that was the best way. But whenever I see something belongs to my girlfriend I feel sad and it is causing me to grab the drink again

00:46:46.250 --> 00:46:47.050

Orion Quach

Umm.

00:46:48.740 --> 00:46:51.130

Orion Quach

I hear you. Sounds like you had a really tough time.

00:47:06.250 --> 00:47:10.080

Adam Bloom

Yes.

00:47:10.570 --> 00:47:15.270

Orion Quach

Is there anything else that makes you happy?

00:47:17.340 --> 00:47:26.270

Adam Bloom

Yeah, I mean, I have two kids from my previous relationship and I love them.

00:47:24.550 --> 00:47:24.940

Orion Quach

Mm-hmm.

00:47:26.970 --> 00:47:38.410

Orion Quach

That sounds amazing. Tell me about them. Could they be your motivation?

00:47:39.690 --> 00:47:40.130

Adam Bloom

Their name is Rere and Lily. They are in preschool right now. I want them to have a good life and not lose their dad.

00:47:40.890 --> 00:47:41.490

Adam Bloom

Thank you for redirecting me to them.

00:47:43.860 --> 00:47:44.690

Orion Quach

I'm glad you found your hope and something to fight for. I hope to hear from you again in our next appointment.

APPENDIX B

SAMPLE JSON PACKAGE RESPONSE FROM A SUCCESSFUL HTTP GET CALL

```
{  
  "doctorName": "Orion Quach",  
  "patientName": "Adam Bloom",  
  "doctorSpeech": {  
    "00:44:04.040 --> 00:44:12.810": "Thank you for the patience. Hello Adam,  
my name is Dr. Quach. What brings you in here today?",  
    "00:44:23.050 --> 00:44:23.270": "Well Adam, you have come to a great  
starting point in seeking some help here today from our office. What triggered  
you to reach out to our office and bring this to our attention? ",  
    "00:44:44.180 --> 00:44:44.460": "OK.",  
    "00:44:47.940 --> 00:44:51.450": "This is a strong first step to have you be  
here today. Do you think you feel better after you drink?",  
    "00:44:54.490 --> 00:45:08.280": "Are there anything else you like to do  
when you are feeling sad?",  
    "00:45:10.400 --> 00:45:11.950": "Let us talk about your girlfriend. How are  
things going on with you both. Is she coming back anytime soon?",  
    "00:45:14.880 --> 00:45:29.620": "I am sorry to hear that. What makes you  
choose alcohol as your outlet?",
```

"00:45:35.170 --> 00:45:35.500": "From scale of 1 to 10, how bad do you think alcohol is affecting your life?",

"00:45:45.030 --> 00:45:45.410": "Mm-hmm. Have you tried quitting?",

"00:45:51.630 --> 00:46:14.120": "Umm, I can tell you've really put a lot of thought into this and you have realized the effect of drinking. ",

"00:46:15.550 --> 00:46:22.090": "How much have you thought about quitting alcohol? I know you have this long, you know, history of doing it, but what are your thoughts about that?",

"00:46:31.080 --> 00:46:35.200": "Wow. So you've actually tried before. Tell me about your experiences there.",

"00:46:46.250 --> 00:46:47.050": "Umm.",

"00:46:48.740 --> 00:46:51.130": "I hear you. Sounds like you had a really tough time.",

"00:47:10.570 --> 00:47:15.270": "Is there anything else that makes you happy?",

"00:47:24.550 --> 00:47:24.940": "Mm-hmm.",

"00:47:26.970 --> 00:47:38.410": "That sounds amazing. Tell me about them. Could they be your motivation?",

"00:47:43.860 --> 00:47:44.690": "I'm glad you found your hope and something to fight for. I hope to hear from you again in our next appointment."

},

"patientSpeech": {
 "00:44:13.840 --> 00:44:23.720": "Hi doctor. Yeah, I might have a thing going on with some drinking. Truthfully, I might be a bit of an alcoholic and I do not know where to start in getting help.",
 "00:44:25.110 --> 00:44:28.000": "Hmmm.",
 "00:44:29.320 --> 00:44:47.270": "After my girlfriend went back to her home country, I started drinking a bit more than usual. Just the other day, I think I drove drunk and realized that I might have a problem.",
 "00:44:52.030 --> 00:44:53.610": "Not really, I still feel really sad.",
 "00:45:08.950 --> 00:45:09.460": "Umm. I don't know.",
 "00:45:12.670 --> 00:45:14.000": "I am not sure. She is asking for a break because I do not make enough money to marry her.",
 "00:45:30.540 --> 00:45:51.170": "I mean, I drank my whole life. I get drunk and it helps me not to think about my problem.",
 "00:45:38.090 --> 00:45:38.470": "Hmm. I would say seven.",
 "00:46:22.800 --> 00:46:23.480": "I mean.",
 "00:46:24.200 --> 00:46:27.520": "I know it is bad for me, but every time I try it just.",
 "00:46:29.070 --> 00:46:30.370": "I don't succeed. I feel sad.",

"00:46:36.590 --> 00:46:47.760": "Well, I just tried to stop like cold Turkey because I thought that was the best way. But whenever I see something belongs to my girlfriend I feel sad and it is causing me to grab the drink again",

"00:47:06.250 --> 00:47:10.080": "Yes.",

"00:47:17.340 --> 00:47:26.270": "Yeah, I mean, I have two kids from my previous relationship and I love them.",

"00:47:39.690 --> 00:47:40.130": "Their name is Rere and Lily. They are in preschool right now. I want them to have a good life and not lose their dad. ",

"00:47:40.890 --> 00:47:41.490": "Thank you for redirecting me to them."

},

"doctorSpeakingTime": 97270,

"patientSpeakingTime": 85420,

"openQuestionsList": [

"Thank you for the patience. Hello Adam, my name is Dr. Quach. What brings you in here today?",

"I am sorry to hear that. What makes you choose alcohol as your outlet?",

"How much have you thought about quitting alcohol? I know you have this long, you know, history of doing it, but what are your thoughts about that?"

],

"closeQuestionsList": [

"Are there anything else you like to do when you are feeling sad?",

"Let us talk about your girlfriend. How are things going on with you both. Is she coming back anytime soon?",

"Mm-hmm. Have you tried quitting?",

"Is there anything else that makes you happy?"

],

"reflectiveQuestionsList": [

"Umm, I can tell you've really put a lot of thought into this and you have realized the effect of drinking. ",

"I hear you. Sounds like you had a really tough time.",

"I'm glad you found your hope and something to fight for. I hope to hear from you again in our next appointment."

],

"scaleQuestionsList": [],

"reflectiveQuestionList": [

"Umm, I can tell you've really put a lot of thought into this and you have realized the effect of drinking. ",

"I hear you. Sounds like you had a really tough time.",

"I'm glad you found your hope and something to fight for. I hope to hear from you again in our next appointment."

],

"scaleQuestionList": [],

```
"closedQuestionsList": [  
    "Are there anything else you like to do when you are feeling sad?",  
    "Let us talk about your girlfriend. How are things going on with you both. Is  
she coming back anytime soon?",  
    "Mm-hmm. Have you tried quitting?",  
    "Is there anything else that makes you happy?"  
]  
}
```

REFERENCES

1. W. R. Miller and S. Rollnick, *Motivational interviewing: Helping people change*, 3rd ed. London ; New York: The Guilford Press, Cop, 2013. Lev-Ran, S., & Nitzan, U. (2011). *Harefuah*, 150(9), 733–749.
2. E. R. Levensky, A. Forcehimes, W. T. O’Donohue, and K. Beitz, “Motivational Interviewing,” *AJN, American Journal of Nursing*, vol. 107, no. 10, pp. 50–58, Oct. 2007, doi: 10.1097/01.naj.0000292202.06571.24.
3. W. R. Miller and G. S. Rose, “Toward a theory of motivational interviewing.,” *American Psychologist*, vol. 64, no. 6, pp. 527–537, 2009, doi: 10.1037/a0016830.
4. IRETA, “Motivational Interviewing Toolkit,” IRETA | Institute for Research, Education & Training in Addictions. <https://ireta.org/resources/motivational-interviewing-toolkit/>
5. M. M. Vasoya et al., “ReadMI: An Innovative App to Support Training in Motivational Interviewing,” *Journal of Graduate Medical Education*, vol. 11, no. 3, pp. 344–346, Jun. 2019, doi: 10.4300/jgme-d-18-00839.1.
6. P. J. Hershberger et al., “Advancing Motivational Interviewing Training with Artificial Intelligence: ReadMI,” *Advances in Medical Education and Practice*, vol. 12, pp. 613–618, 2021, doi: 10.2147/AMEP.S312373.
7. A. Shivakumar, “Computer Enabled Interventions to Communication and Behavioral Problems in Collaborative Work Environments,” *etd.ohiolink.edu*, 2022. https://etd.ohiolink.edu/apexprod/rws_olink/r/1501/10?p10_etd_subid=195876&clear=10 (accessed Dec. 03, 2022).