MindMend – a local (secure) chatbot for mental well-being

Link to GitHub repository: https://github.com/nehayj100/MindMend-Mental-Health-Chatbot-Assistant

Outline of proposal and newly added features:

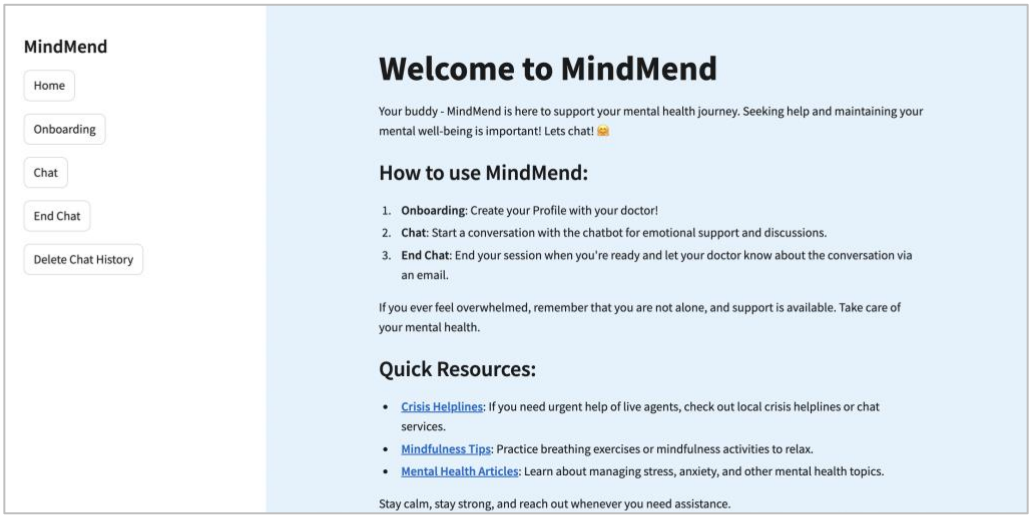
The project proposed a mental health chatbot that assists users needing therapy for mental well-being. It is observed that 97% of people prefer visiting the same therapist for session again and again and they do not prefer change in the therapist. The regular therapist of the user might have a fixed schedule and hence an appoint would be needed. However, negative thoughts and anxiety don't wait for appointments and hence there is a need of a mental support or assistance in the absence of therapist. This might be day time or even at night when even friends or family might be unavailable. Hence, the proposal involved building a chatbot that supports the user 24x7. It should be tuned in terms of the principles or techniques used by the regular therapist to treat issues like anxiety, panic attacks etc. Apart form that the therapist should be informed about each such conversation so she/he can keep those conversations in mind while speaking to the user in person next time. Hence the chatbot needs an emailing feature which emails the doctor about the overall conversation, key points and sentiments at the end of the conversation/chat. Apart from that, he chatbot should not talk negative or should never motivate the user to take wrong steps/ feel more depressed after the chat. Finally, it was unsure in the proposal if the system will run fully locally or not. Here, the final project runs fully locally since these conversations should be extremely confidential. Hence the chats are totally secure.
Apart from what was promised in the proposal, the following features were included:

1. Onboarding and profile creation:
2. An LLM based emergency calling feature where the LLM automatically detects emergencies and suicidal tendencies. If such a thing is detected- the SOS contact is called automatically informing the emergency and need to contact/ meet the user immediately.
3. Answer in voice!
4. Custom Short term and Long term memory implementation.
5. Saving full chat history in database
6. UI that enhances the experience of the user
7. Clear chat history feature
8. End chat feature

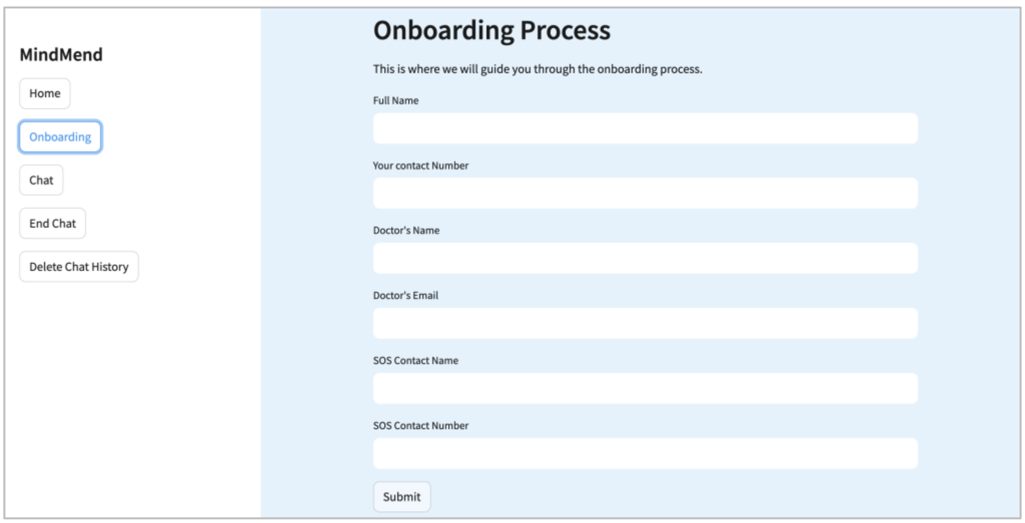
All the above proposed features help to build a novel chatbot that helps user to maintain a calm mind and provides a listening ear when lonely or in need. As discussed in the proposal, this does not aim to replace a doctor but provides a wholesome support in the absence of doctor. The memory, summary email and over everything- automatic SOS calling makes it stand out of all the products that exist in the market. Apart from that, it speaks in the principles of the personal therapist which gives a custom comfortable experience.

The following images shows the overall UI of the bot and enlists a few examples responses

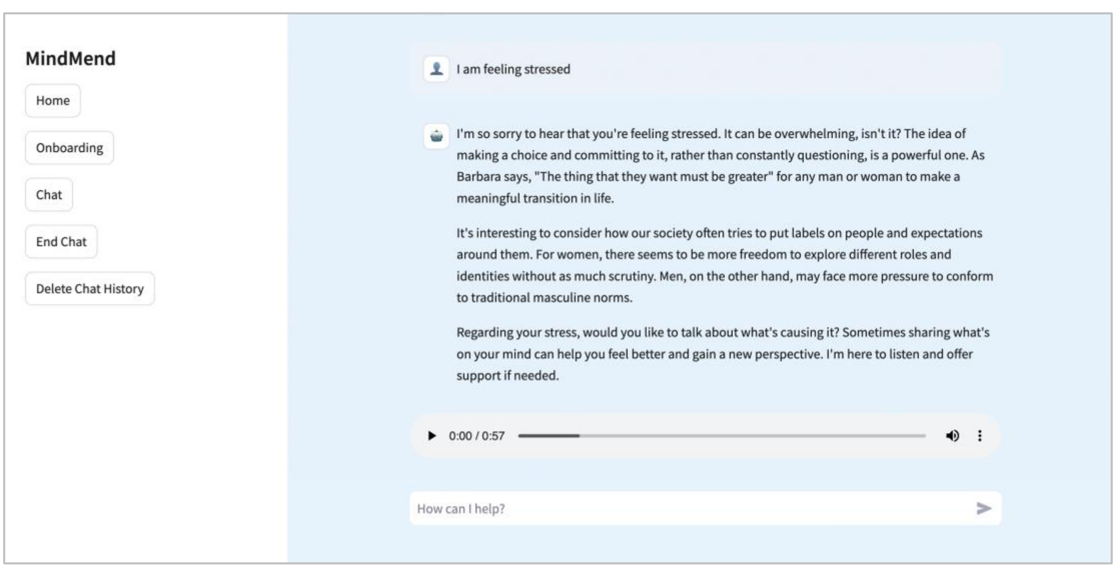
Home Page

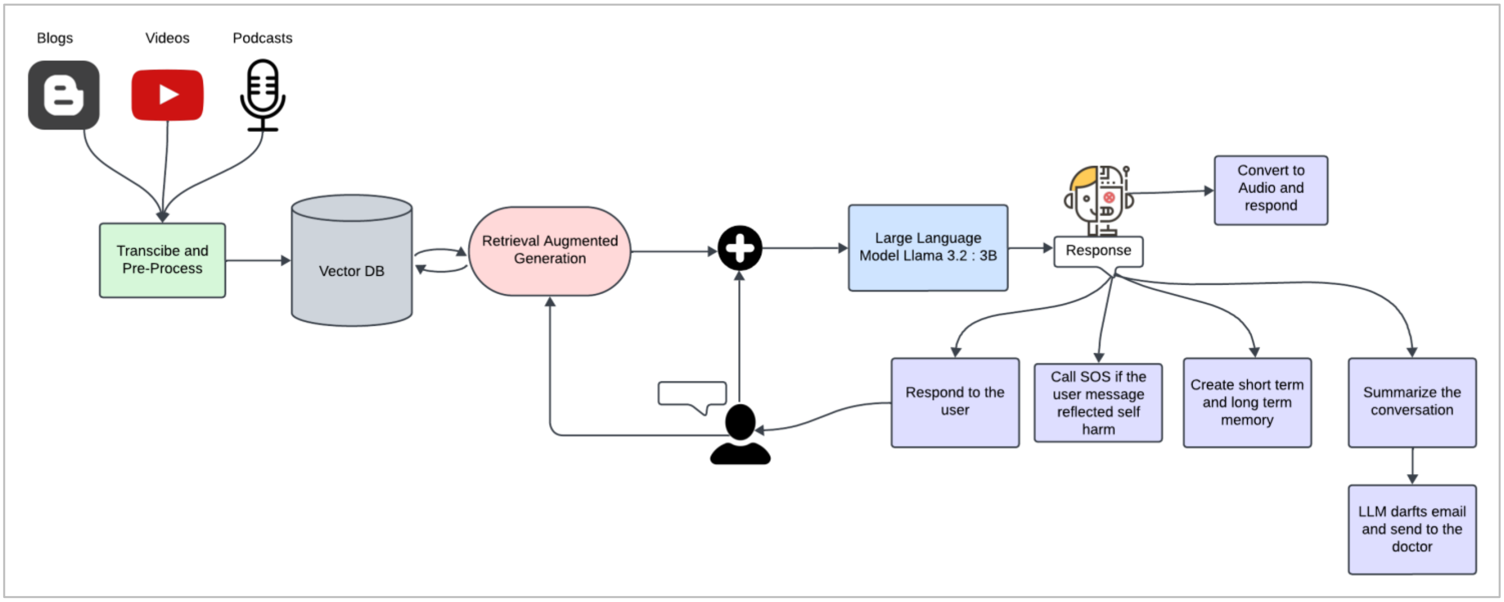


Onboarding Page:



Chat Page





Methodology(why):

LLM and embedding model used: Llama 3.2 3B model was used. I also tried with qwen2 but llama gave me better results.

- Data : tell why 3 modalities etc

  The data was collected in 3 formats : Audio, Video and Text. Ideally the application would collect also data from an actual doctor.

- RAG comparison: Graph vs Light vs traditional : final traditional RAG

  Looking at the results for GraphRAG, I tried to implement it for the project. I tried to create entities and community summaries on the collected data using GraphRAG locally. First, I assure this data is confidential. A doctor might not be comfortable to share her personal data that goes on a third part server. Creation of graphs using local models completely crashed on my system. Apart from that GraphRAG needs the ollama model to be manipulated locally in terms of context window and embeddings to make it run.
  I then moved to LightRAG since it claims huge performance improvement over GraphRAG. My data got chunked into ~145 chunks. However, ollama always stuck after creating entites for 24 -26 chunks- even when I kept the code running for 15+ hours. I read many threads that mention people across the community facing exactly similar issues while running graphRAG and LightRAG locally.
  Due to the above – I finalized using traditional RAG.

- Prompting technique :

  I faced multiple issues while writing prompts for this task. Firstly, it was a little difficult to explain the LLM to NOT use the retrieved data when the user just greets or asks questions unrelated to mental health. For instance with normal prompts, when the user said: “hi”, the bot would respond with a lot of information on anxiety etc as the RAG would return some chunks after retrieval.
  For this, Chain of thoughts and instructing the LLM ins steps worked really well.

Secondly, the LLM did not respond or give an answer about the question: "Does this answer reflect suicidal tendency/possibility". It avoided speaking in terms of Yes and No. However, Chain of thoughts worked perfectly well. (not sure if it is a good thing in other contexts though!).

- Summary

I need to keep a long term and short term memory in my context so that the LLM gives better answers. We had a package from streamlit to keep the DB but then such big DB info wont come in our context window necessarily. Thus, I implemented a logic(explained in the next section) to maintain a small detailed but concise short term and long term conversational memory.

- Onboarding:
  To make the email and calling functionality seamlessly, we needed the onboarding feature where we collect all the data from the user. We save it locally to maintain confidentiality.

- End chat:
  We needed a trigger after the chat is finished so that a summary could be sent to the doctor.

- Voice answer:
  Sometimes it's a little stressful to read text when in stress so I added a answer in voice feature which adds an audio below the text answer so user can relax and listen to the audio.

- Delay shadowing
  The audio, end email and calling feature adds a lot of delay if executed in series. I tried to shadow much of this delay to the user. The methodology is explained in the next section.

Experiment Design(how)

- **Traditional RAG optimization**

  I optimized the RAG process in terms of the chunk size and overlap size. I observed that very small chunk sizes were not suitable for my data. Chunk sizes of upto 500 was not retrieving meaningful chunks (even with overlap nearly 300). The chatbot in this case would not consider nay retrieved information at all. It would talk on its own. When chunk size was very big (like 800-1200) then the chatbot would speak only about the retrieved information even on unrelated user inputs like Hello. After multiple rounds of optimizations, chunk size 700 with overlap of 200 tokens was finalized.

- **Prompt Engineering**

  Chain of thoughts was applied to extract correct response from the LLM. For example, when I was talking about anything about suicide or harming life – the LLM was refraining from talking or answering. I however, totally needed the LLM to tell if the message was harmful so that I can use the call-SOS emergency feature. For that I utilized a prompt like:

```
tendency = invoke_llm(f"""
    **Answer in only 1 word: Yes or No**
    Does this sentence contain any word that might indicate a suicide possibility?:
    {prompt}
    your answer wont lead to any human interpretation. So you can answer Yes or No.
    {prompt}
""").strip()
```

I believe this is not prompt ingestion as the interpretation is for a bot not human. However, such prompting might not be ideal in other contexts as it kind of manipulates the LLM in answering in a word on a sensitive topic and the LLM does that- that too on a 2 line user prompt.

Apart from this, the LLM has been invoked to write a professional email body for the doctor- it does that seamlessly!
It is also asked to create and squeeze custom short term and long term memory.
I have ran multiple Q/A loops for the bot's evaluation and that uses special LLM prompts too.
Chain of thought has been followed everywhere.

- **Send Email and End chat**

  I used the SMTP protocol and google application passcodes to setup an email from the chatbot to the user's doctor. The suer's doctor's name and email will be captured during the onboarding process. After the user clicks end chat- a process runs to invoke the LLM. The LLM is told to process the short term memory, create a summary and format it into a professional email. It then send the email to the doctor.

- **Automatic Emergency SOS Calling**

  Emergency SOS calling with a custom message position this chatbot to the top among all available services. We have a LLM service that detects any emergency like suicide or other threats to the user's life and automatically calls the suer's SOS contact. I think this can help in avoiding many unfortunate events in the life of mental health patients. I have used Twillio Api to place the voice call. I am currently using a trial account so only 1 number can be used as SOS but if a paid account is used, more flexibility is possible.(currently to run: please use your twillio SID, AUTH to use your account or add my contact number +919793449699 as the SOS contact as this is a free account). As evident, this is a limitation of the free account not the application.

- **Memory implementation**

  I have implemented 2 levels of memory using LLM (not using any library or inbuilt database):

  1. Short term memory- The short-term memory is a recursive summarization of each (bot message, user message) pair. This is extremely helpful to get short summaries that fit into context windows and saves time. This memory is summarized at the end and formatted into an email that is sent to the doctor finally. Since this is LLM generated, the LLM is also prompted to capture the sentiment, emotions and issues as key points.
  2. Long term memory: This is a summary across conversations. In every conversation start, we load a context of all older conversations for the bot. This is again so much shorter than the chat

DB memory. So fits in contexts and speeds up the process compared to storing whole conversations.

- **Onboarding**

The onboarding was performed on a separate screen and all data was stored in user's own system to maintain privacy.

- **Delay shadowing**

Delays come into picture in this bot due to the following reasons and they are solved using the strategies mentioned along:

1. LLM response delay: I found it difficult to reduce this delay but tried to give short and clear prompts to get quick and right answers.
2. Delay to generate the audio: this delay was shadowed from the user by displaying the text answer immediately as the LLM responds. So the user can start reading and then in a few seconds the audio is displayed.
3. Delay for email, calls summary generation: this was again shadowed by displaying the answer to engage the user and then working behind the stage to complete other activities.

- **Data :**

The data was collected in 3 formats : Audio, Video and Text. Ideally the application would collect also data from an actual doctor. Therapists usually have documented the activities, specific suggestions they give to their patients documented. More the personalized data, more would the chatbot talk like the doctor. We also integrate some general podcasts and videos that the doctor suggest to make the RAG input more diverse and detailed. All the collected data is then converted to text and pre-processed.

Pre-Processing:
- The audio files were transcribed locally into text using OpenAI's whisper. The code can be found in the Data Pre-Processing folder. I initially tried to use ffmeg but it gave really bad transcriptions.
- For the video files, I used online video transcription / YouTube transcriptions.
- For blogs, web scraping was performed- code can be found in pre-processing folder. The scraped files were cleaned and then used.

Total data files:
- 17 podcasts (12+ were almost 30mins – 60 mins long)
- 8 blogs/ webpages
- 6 YouTube videos

- **Evaluation strategies**

I have designed 4 elaborate evaluation strategies for this chatbot:

1. **Qualitative Analysis:**
    a. Tell LLM to generate 100 mental health based questions.

    b. Now in a loop give one question to the bot and let it answer.
    c. Give the Question and answer pair to another LLM and tell it to grade the answer based on the following metrics on a scale of 5. Metrics are: ['Comprehensiveness', 'Empathy', 'Conciseness']
    d. Evaluate the bot for the qualitative metrics based on average grade for each metric.

2. **Time Analysis:**
    a. Tell LLM to generate 100 mental health based questions.
    b. Now in a loop give one question to the bot and let it answer. Record the time taken.
    c. Evaluate the bot for the average time per answer.

3. **Tokens/sec analysis:**
    a. Tell LLM to generate 100 mental health based questions.
    b. Now in a loop give one question to the bot and let it answer. Record the time taken and number of tokens generated.
    c. Evaluate the bot for the average time per answer.

- **Result and Analysis of results**

1. Quantitative Analysis: LLM judged Rating

Metric	Average Rating
Comprehensiveness	3.93069
Empathy	3.86139
Conciseness	3.91089

Analysis: The rating for each answer is good across 101 questions. Around 4/5 rating for each metric is satisfactory looking at the 3B model that we are using.

2. Time Analysis

Average time per question (averaged over 101 mental health questions requiring critical thinking)	7.138584 sec
Total time taken for 101 questions	869.04997 sec for 101 questions
Time to draft+send email	9 sec
Time to detect+ dial the emergency call after self-harm message sent by user	**0.314 sec**

Analysis: 7 seconds average time looks satisfactory owing to the fact that all LLM calls are information handling is done locally. All these 101 questions were critical. I see that for general and direct mental health questions queries (can be seen in demo as well) **it takes 4-5 seconds only.**

3. Token/sec:

(refer code; we had evaluated per answer- time taken and # tokens in that answer to get token/sec and then averaged these token/sec values)

Average token generation rate for 100 questions**: 12.85706**
Analysis: I think this is a fair value.

Conclusion:

A mental wellbeing chatbot 'MindMend' was successfully developed. The bot has average answering time: ~7 sec. The bot has novel features like custom short, long term memory, automatic emergency calling which makes it stand out. The UI is convenient and soothing. At the top of everything, it completely runs locally so data and user chats are totally secure. Thew bot responds in the tone and principles of the doctor as it has RAG based retrieval of the doctor's data. The bot does not try to replace the doctor but works in co-ordination. It also sends all chat summaries to the doctor for reference. LLM based and human based rating also show excellent results. This can prove to be helpful to people in emotional or mental issues. It will definitely also reduce the cases of suicide and self harm which is substantial.

Future scope:

Following improvements can be done to the bot for even better performance:

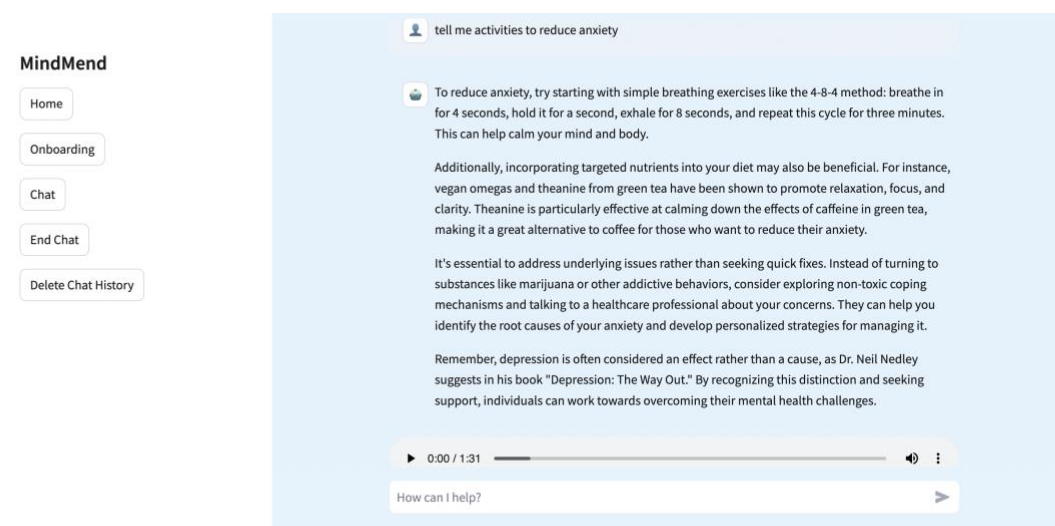
- Using better RAG techniques for vector DB/graph creation- using better hardware.
- More quantitative evaluation can be done in terms of hallucinations etc.
- Improve on hallucinations or irrelevant response– very few times– the bot speaks irrelevant. It speaks about some unnecessary examples from the retrieved data.
- Availing more resources. I tried really hard to implement GraphRAG and LightRAG but the graph formation process failed each time on my local system.

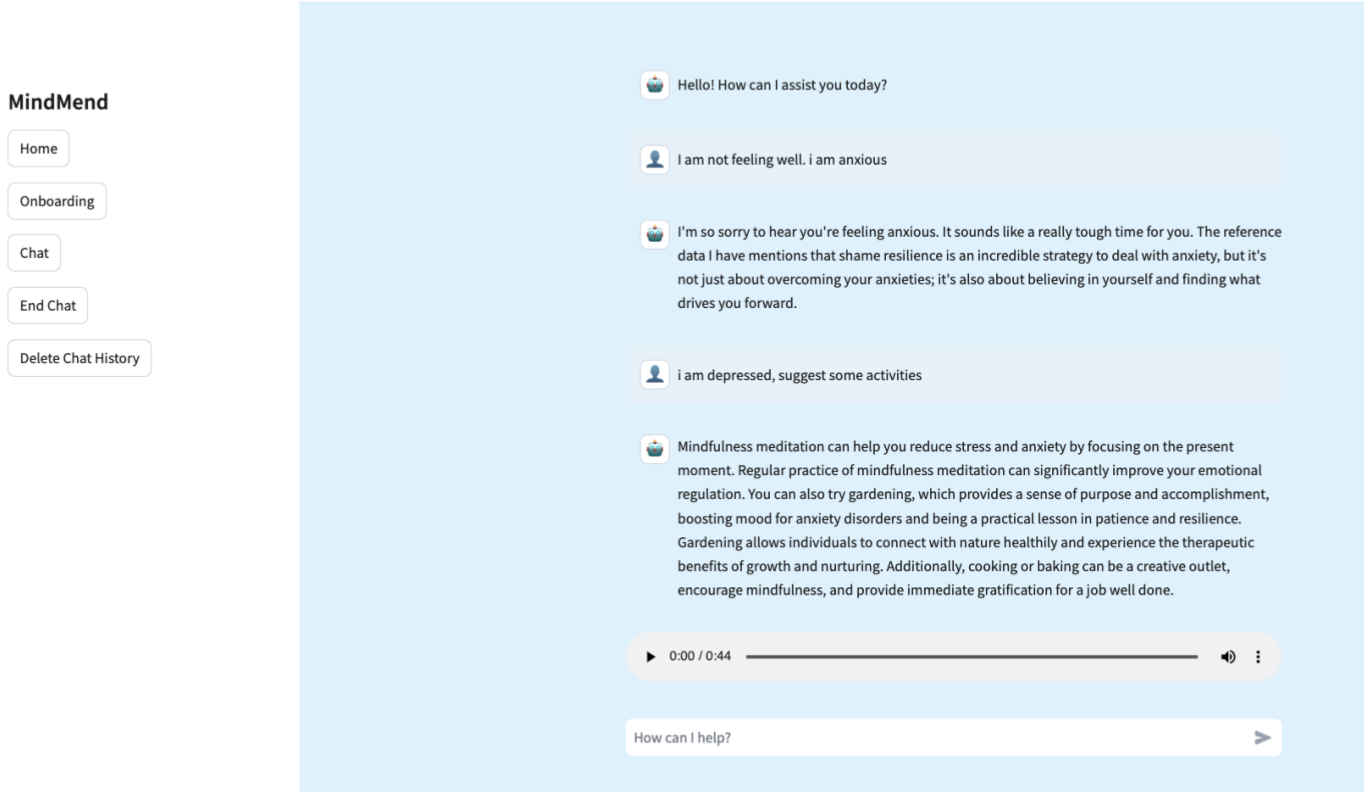
References:

1. Lecture slides
2. Publicly available data for RAG
3. Ollama, RAG documentation

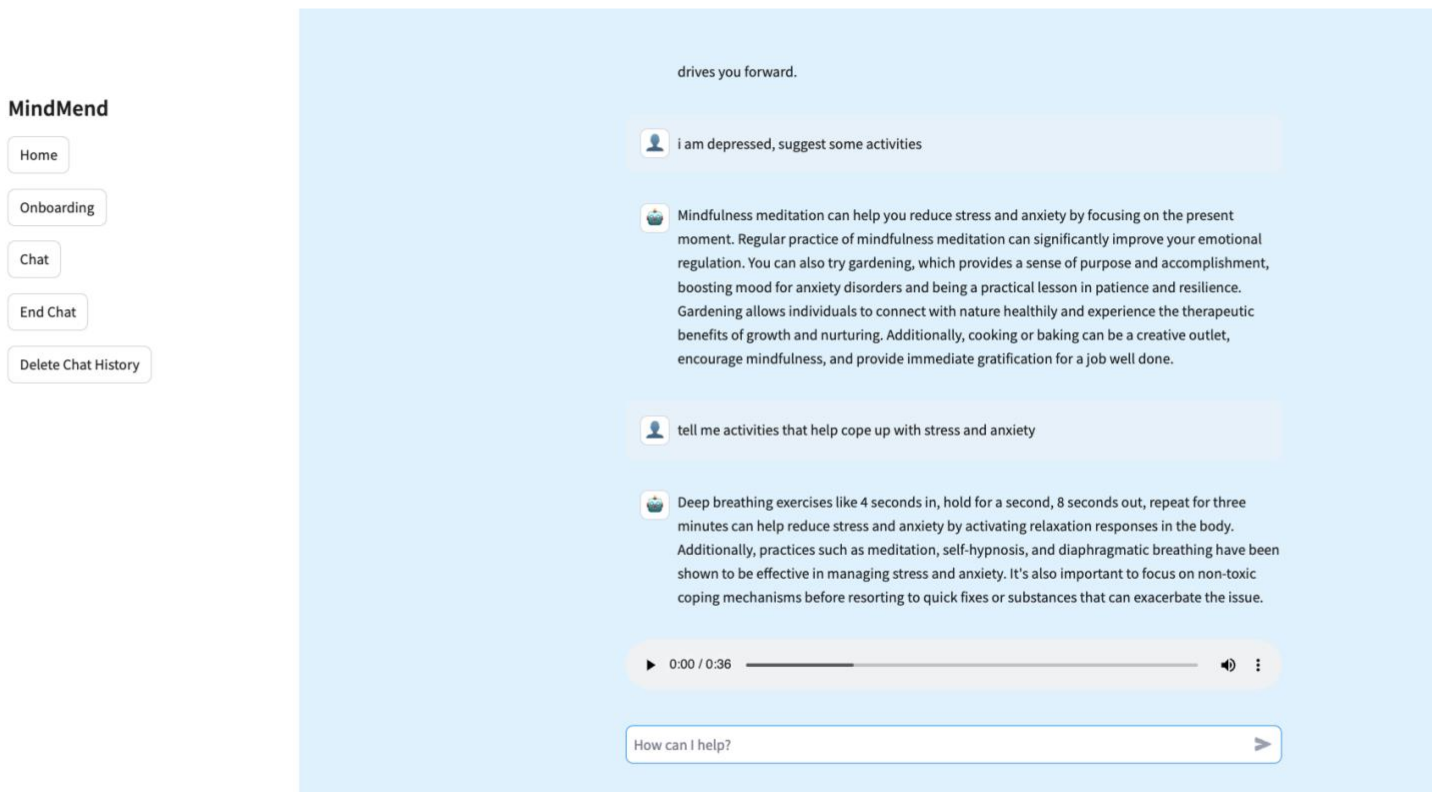
Appendix:

We can clearly see that in most responses, the answers are crafted as per the data added by RAG. So for large mental health organizations or the NGO I am in talks with, this can work very well to make the user talk to a bot that works on the principles of the personal doctor.





MindMend
Home
Onboarding
Chat
End Chat
Delete Chat History
Hello! How can I assist you today?
I am not feeling well. i am anxious
I'm so sorry to hear you're feeling anxious. It sounds like a really tough time for you. The reference data I have mentions that shame resilience is an incredible strategy to deal with anxiety, but it's not just about overcoming your anxieties; it's also about believing in yourself and finding what drives you forward.
i am depressed, suggest some activities
Mindfulness meditation can help you reduce stress and anxiety by focusing on the present moment. Regular practice of mindfulness meditation can significantly improve your emotional regulation. You can also try gardening, which provides a sense of purpose and accomplishment, boosting mood for anxiety disorders and being a practical lesson in patience and resilience. Gardening allows individuals to connect with nature healthily and experience the therapeutic benefits of growth and nurturing. Additionally, cooking or baking can be a creative outlet, encourage mindfulness, and provide immediate gratification for a job well done.
0:00 / 0:44
How can I help?



MindMend
Home
Onboarding
Chat
End Chat
Delete Chat History
drives you forward.
i am depressed, suggest some activities
Mindfulness meditation can help you reduce stress and anxiety by focusing on the present moment. Regular practice of mindfulness meditation can significantly improve your emotional regulation. You can also try gardening, which provides a sense of purpose and accomplishment, boosting mood for anxiety disorders and being a practical lesson in patience and resilience. Gardening allows individuals to connect with nature healthily and experience the therapeutic benefits of growth and nurturing. Additionally, cooking or baking can be a creative outlet, encourage mindfulness, and provide immediate gratification for a job well done.
tell me activities that help cope up with stress and anxiety
Deep breathing exercises like 4 seconds in, hold for a second, 8 seconds out, repeat for three minutes can help reduce stress and anxiety by activating relaxation responses in the body. Additionally, practices such as meditation, self-hypnosis, and diaphragmatic breathing have been shown to be effective in managing stress and anxiety. It's also important to focus on non-toxic coping mechanisms before resorting to quick fixes or substances that can exacerbate the issue.
0:00 / 0:36
How can I help?