LLMs For Good!

Design Document

sdmay25-35
Ahmed Nazar
Mohamed Selim

Assignment Manager: Emma Zatkalik
Stakeholder Liaison: Abrahim Toutoungi
Project Deliverables Manager: Brianna Norman
Meetings Coordinator: Halle Northway
Communications Manager: Gabriel Carlson
Timeline Coordinator: Ellery Sabado

sdmay25-35@iastate.edu
www.sdmay25-35.sd.ece.iastate.edu

Revised: 12/7/2024

Executive Summary

Our project aims to develop an empathetic chatbot-based website using a large language model (LLM) to assist people seeking humanitarian guidance during crises. The problem is critical because individuals in high-stress situations often struggle to find vital information quickly, whether dealing with natural disasters, medical emergencies, mental health issues, or displacement.

Technical Implementation

The LLM leverages open-source technologies and implements two key machine-learning techniques:

- Retrieval Augmented Generation (RAG) to optimize outputs using external data sources
- Fine-tuning to modify the model's parameters for specialized responses

The architecture includes a user-friendly frontend interface, a backend API running on ISU servers, and secure data transmission protocols. The system must provide responses within 10 seconds while maintaining user privacy by not storing personal data.

Current Progress

- Experimented with RAG and fine-tuning techniques on different pre-trained LLMs
- Started implementing fine-tuning scripts on our virtual machine
- Began developing the frontend interface
- Collected training performance metrics

Challenges and Solutions

- Virtual machine memory constraints for fine-tuning
- Dependency conflicts in implementation
- Limited availability of quality training datasets

To address these challenges, we focus on curating high-quality datasets, optimizing our virtual machine usage, and implementing comprehensive testing procedures. The system's design emphasizes accessibility standards, user privacy, and empathetic response generation to ensure it meets its core mission of providing reliable assistance to users in crises.

Learning Summary

Development Standards & Practices Used

Software Practices

- Agile development
- Data privacy
- Version control
- Accessibility standards
- Secure data transmission

Engineering standards

- ISO/IEC 27000: Information security management
- ISO/IEC 42001: Artificial Intelligence Management System
- IEEE Std 730-2014: Standard for Software Quality Assurance Processes
- IEEE Std 1028-2008: Standard for Software Reviews and Audits
- IEEE/ISO/IEC 26514-2021: Design and development of information for users

Summary of Requirements

- Create an LLM chatbot to assist users seeking guidance in the form of a website
- We will not store user information or train the LLM with it
- The website will have a user-friendly and easily navigable interface
- The website must be able to accept user input in the form of text

Applicable Courses from Iowa State University Curriculum

- Computer Science: 309, 319, 409, 472, 474
- Software Engineering: 317, 329, 339
- Cybersecurity Engineering: 230, 231

New Skills/Knowledge acquired that was not taught in courses

- Machine Learning
- Artificial Intelligence
- Developing a retrieval augmented generation (RAG) chatbot
- Developing a fine-tuned chatbot

Table of Contents

List of figures/tables/symbols/definitions

Figures

Tables

1. Introduction

1.1. Problem Statement

Our project seeks to create an LLM that can be interfaced with a chatbot-like website designed to provide people in need of humanitarian assistance and information with concise and empathetic answers. In high-stress situations, many people struggle to find important information in a timely manner. By training our model with various resources that display compassionate aid on various topics, we can develop a multimodal platform to direct them toward relevant resources.

Our main user base will be people in crisis struggling with issues like natural disasters, medical and mental health issues, and displaced living conditions. Due to our users' environment when using our chatbot, we need to provide an intuitive and easy-to-understand platform with multimodal input options. This will include things like text and image-based prompts, and no account or login will be required to use our platform. Disclaimers will inform users of the AI-generated nature of the responses and the privacy of their data not being used to train our models.

Currently, one of the main limiting factors of our project is the requirement for an internet connection and a device supporting a web browser. While we have discussed optimizing/minimizing our models to run on smaller devices, our initial implementation will require an internet connection. We also have the issue of user privacy and data. Our project will not store any session or user data, and our models will not be trained on any data supplied by our users.

1.2. Intended Users

Medical

Leah is an adult with access to her phone. She fell down outside and got a cut on her leg and wants to know quick ways to help treat her wound. She benefits from our product because she is in need of a quick and concise answer to a question. Our product aims to provide answers to people who are in need, which can be applied to Leah's situation.

Legal

Michael is getting divorced, but he isn't sure how assets get split in Missouri (his state of residence.) He asks our bot about asset division and is provided with valuable resources for divorce in the state of Missouri.

Disaster-Affected Individuals

Aziz is a teenager living in Gaza. He wants to know where he and his family can safely evacuate. He asks our chatbot, and it can provide him with location-specific information about evacuation and general safety measures that can be taken.

Mental Health
Simon is a college student and is feeling helpless. He doesn't want to talk to his friends about his feelings because he is afraid they will make fun of him. He can ask questions to our chatbot and get resources and direction.

Empathy Map:

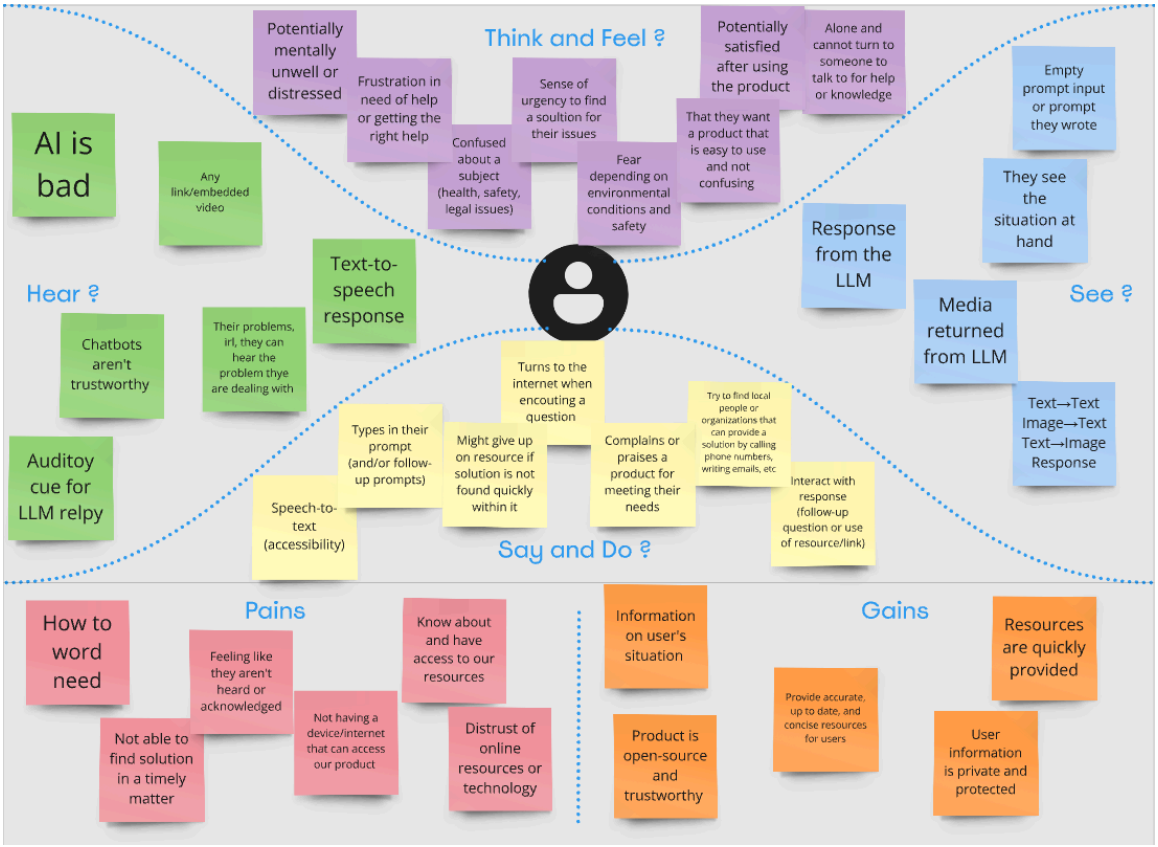


Figure 1.2.1: Empathy Map

2. Requirements, Constraints, And Standards

2.1. Requirements & Constraints

Application and Technical Requirements (functional)

User Experience

- The UI of the LLM shall have accessibility features
- Users shall be able to input text and image queries to the LLM

Technical

- The LLM shall answer user queries
- Users shall access the LLM through the internet and a web browser

Ethical

- The LLM shall not store users' private information
- The LLM shall be vetted for appropriate responses
- The datasets used to train the LLM shall contain appropriate and relevant information

Response Time (quantitative/constraint)

- LLM provides fast responses that should be returned to the user within 10 seconds.

Appropriateness (non-functional)

- LLM is vetted for appropriateness, and the datasets provided to the LLM contain relevant and appropriate information.

User Experience (non-functional)

- Application provides empathetic responses that establish a sense of support for users.
- Application will adhere to HIPAA compliance standards.

Accessibility (non-functional)

- Incorporates accessible user interface design principles.
    - High-contrast colors for easily readable content for users with visual impairments
    - Text alternatives for images, icons, and non-text content
    - Semantic HTML for logical structure to aid navigation for assistive technologies
    - Labels clearly denoting fields
    - Responsive design for compatibility across different devices

Security (non-functional)

- Application does not store users' private information.
- Encrypting data sent between client and server.
- Server only allows requests on designated application ports
- Update security patches for dependencies

2.2. Engineering Standards

Engineering standards are important because they exist in nearly every facet of life, including in everyday scenarios. They establish rules that are followed during product development to make sure that products or services provided by products are dependable and consistent. They also ensure that products are safe, efficient, compatible, and consistent. The standards provide a project with a foundation of well-defined specifications.

Some engineering standards that may apply to our project are:

- ISO/IEC 27000: Information Security Management
    - This standard provides guidelines on how to manage sensitive information. Important factors include integrity, availability, and confidentiality. The protection of information is important to our project.
- ISO/IEC 42001: Artificial Intelligence Management System
    - This standard focuses on the management of artificial intelligence systems, including establishing, implementing, and improving AI systems as a whole. Characteristics that this standard emphasizes are transparency, compliance, and continuous improvement.
- IEEE Std 730-2014: Standard for Software Quality Assurance Processes
    - Software quality assurance (SQA) ensures that software projects are maintained with high standards. This occurs throughout and beyond the development process. SQA establishes the requirements for initiating, planning, controlling, and executing a software project.

The three standards that were chosen have relevance to our project. Regarding ISO/IEC 27000, the confidentiality of user information is extremely important and we are aiming to not store any of it once the session with that user is completed. We want our users to feel like they can freely and safely use our product without any worry about their data and personal information being breached.

Our project will be using a large language model to generate responses to user queries. The use of artificial intelligence must be managed in an orderly and secure manner, so ISO/IEC 42001 directly applies to our project. This standard provides a guide on how to avoid certain risks that we may encounter throughout the development process, like our model displaying great bias.

For our project as a whole, IEEE Std 730-2014 is a relevant standard because it can be applied to any software project. Following this standard will help guide us to meet certain quality requirements for things such as performance, user experience, and security. Our product will properly address risks, be reliable to users, and be safe to use due to the adherence to all these standards.

Some of the other standards that were proposed are IEEE 2410-2021: Standard for Biometric Privacy and IEEE/ISO/IEC 26514-2021: Design and Development of Information for Users.

Regarding ISO/IEC 27000, we will assess within our project where it may be necessary to encrypt data, restrict data sharing, and verify compliance with legal standards for different data that may be passed to our application. With ISO/IEC 42001, we will include guidelines for mitigating bias and constraints we will impose on the LLM used within our project. Finally, for the IEEE Std 730-2014, we will plan code reviews, audits, and testing into our Gantt chart.

3. Project Plan

3.1 Project Management/Tracking Procedures

Our team will be adopting an agile-style project management method. This project management style works best for our team because several different chunks of our project can be completed simultaneously, whether that be front-end interface development, implementing security design choices, LLM setup and fine-tuning, and more. If we were to choose a waterfall approach, it would hinder progress in our project. We will be using Git and GitLab to create and track issues and feature development, the tasks of which will be based on our Gantt chart and our overall timeline for our project.

3.2 Task Decomposition

1. Researching LLMs
2. Collecting Datasets
3. Creating RAGS
4. Fine-tuning LLMs
5. Combining Fine-Tuning With RAG
6. Response Validation
7. Deploying LLMs to private cloud
8. UI/UX Research and Development
9. Frontend Deployment
10. Cybersecurity Risk Assessment

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

1. Students should understand the basics of how LLMs work and how they can be run locally.
2. LLM is to be running on our virtual machine and delivering satisfactory responses to inputted user prompts within approximately 10 seconds.
3. LLM will perform with 85% accuracy; the data fed to the model will be highly accurate and factual.
4. Datasets will be embedded into the vector store for the local RAG chain using HuggingFaceEmbeddings.
5. RAG Chains will return relevant responses based on information provided by embeddings.
6. Responses from fine-tuned LLM match provided template and training data.
7. Virtual machine setup and running deployed backend API.
8. Frontend web application can be accessed by anyone with a web browser on the Iowa State network.

The user interface is easy to use and accessible, with average new users able to ask a basic text question in under 15 seconds.

3.4 Project Timeline/Schedule

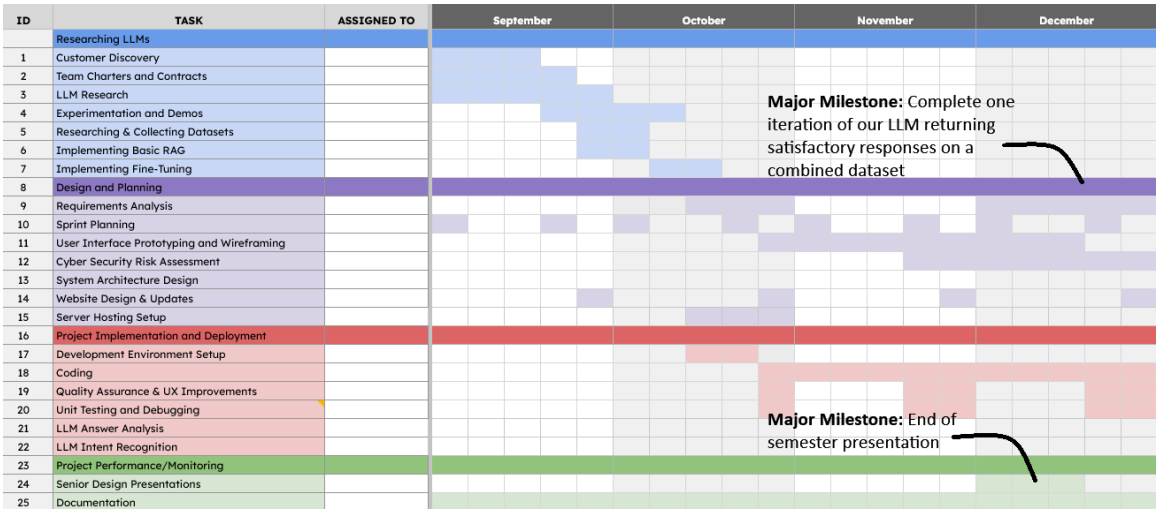


Figure 3.4.1: Gantt Chart for 491

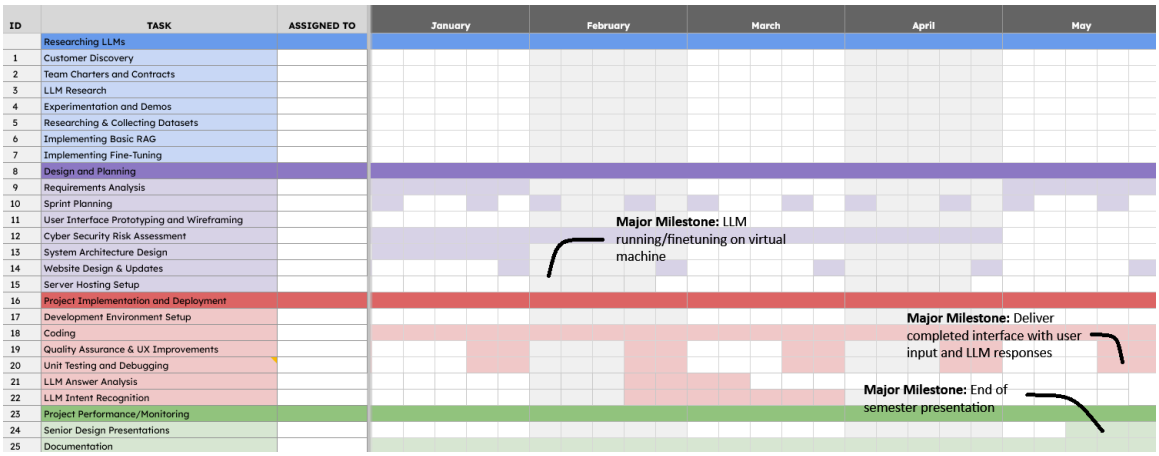


Figure 3.4.2: Gantt Chart for 492

We aim to have a functional prototype by the end of this semester. The functional prototype should be able to return satisfactory responses after being trained on a factual and trustworthy dataset. Another milestone that we aim to achieve by the end of this semester is to have documented the steps we took along the way so we can present them at the end of the semester.

For next semester, we plan on fine-tuning our LLM on the VM provided by ETG. Additionally, we expect to have our project complete. We will work for the rest of that semester on ironing out bugs and achieving stretch goals. At the end of the semester, we will deliver both our project and a presentation that documents the entire process we took.

3.5 Risks and Risk Management/Mitigation

Agile projects can associate risks and risk mitigation with each sprint.

Task 1: Researching LLMs - 0% risk of unusable models for our project.

Task 2: Collecting Datasets - 60% risk of being unable to find multiple comprehensive, appropriate datasets. For this risk, we have agreed that we will create our own datasets in the event that we cannot find any that suit the topics we want our LLM to be able to respond to.

Task 3: Creating RAGs - 40% risk of the model returning data incorrectly or not being able to return data.

Task 4: Fine-tuning LLMs - 50% risk of the model incorrectly replicating the style from the dataset if the dataset does not have consistency throughout.

Task 5: Combining Fine-Tuning With RAG - 45% risk of incompatible packages needed for fine-tuning versus RAGs.

Task 6: Response Validation - 0% risk of not being able to validate responses.

Task 7: Deploying LLMs to Private Cloud - 20% risk of complications facing paywalls and transferring work.

Task 8: UI/UX Research and Development - 80% risk of poorly secured user data and codebase. For this risk, we will make sure to incorporate more secure practices in our project development, including using up-to-date packages in our code and also performing security audits on our code.

Task 9: Frontend Development - 20% risk of failing to create a satisfactory user interface that uses all mediums of text, image, and voice for user input.

3.6 Personnel Effort Requirements

Task	**Projected Effort (in person-hours)**
Researching LLMs	140
Collecting Datasets	144
Creating RAGs	105
Fine-tuning LLMs	170
Combining Fine-Tuning With RAG	84
Response Validation	126
Deploying LLMs to Private Cloud	45
UI/UX Research and Development	100
Frontend Development	150

Table 3.6.1: Personnel Effort Requirements

In the beginning phase of this project, all team members conducted detailed research about LLMs, including how they are made and the different ways to customize them. We next did research about datasets we would use to train our LLM to specialize its knowledge base and stylize its responses. This required us to explore many different websites that have existing datasets and curate a unified dataset to use for our model that is in the desired format of question and answer. Making a RAG and fine-tuned LLM were both time-consuming tasks that each member was to complete on their own, and then we came together to create one unified RAG and fine-tuned LLM on our virtual machine.

The latter half of the above table are projected hours for tasks to be done next semester. Once our LLM is producing trustworthy responses, we will conduct response validation to ensure that the model is reliable and accurate. Deploying our model onto the private cloud will require time and effort from the team to complete. Then, the focus shifts to the front end of the project, which is extremely important to our project. We need a very navigable and accessible website for our users to interact with. A decent amount of time will need to be spent on the development of this to make sure we meet our requirements.

3.7 Other Resource Requirements

Since we are working on a web application that utilizes open-source materials, we do not need any other parts and materials other than the virtual machine provided by the university's ETG.

4. Design

4.1 Design Context

4.1.1 Broader Context

Our design problem is situated within the broader context of national humanitarian crises and everyday challenged faced by individuals facing times of stress and uncertainty. These situations can arise from medical emergencies, displacement, and natural disasters. The communities we are designing this for include crisis victims, healthcare seekers, displaced populations, and generally, people who are looking for support. Some communities that are affected by our design are aid organizations that our chatbot may direct people to and marginalized communities who encounter barriers when trying to get reliable information. Societal needs that our project addresses include improving quick and easy access to information, protecting users' privacy, and fostering inclusivity and equity.

Area	**Description**	**Examples**
Public health, safety, and welfare	Our project impacts the well-being of individuals facing crises by providing quick, accurate, and empathetic guidance.	Increasing safety by directing users to emergency resources. Improving mental health by offering empathetic responses and links to helpful information.
Global, cultural, and social	Our project must respect cultural diversity and reflect the values of the communities it serves while ensuring inclusivity and relevance.	The website is designed to aid marginalized groups with a trustworthy, secure, and easy-to-use platform.
Environmental	Our project may indirectly affect the environment through its computational energy demands.	Increase the efficiency of our chatbot and minimize unnecessary data processing.
Economic	Our project is meant to be financially accessible to our users; it will not require any type of payment for its services.	Designed to be free to users, so it will be affordable to anyone with access to a web browser and the internet.

Table 4.1.1.1: Relevant Area Considerations

4.1.2 Prior Work/Solutions

We read through several literature reviews that were relevant to our project. According to the research [1], there are existing chatbots for healthcare whose response style is not very human-like and natural sounding; it is important, especially in these healthcare chatbots, to put effort into making the model more empathetic and personal to users.

The research [2] explores the different chatbots that have been made and highlights certain attributes that are essential to making quality chatbots. Some important features include efficiency, reliability, and usability. This helped us identify which parts of our chatbot we should focus on the most while developing it.

Research [3] included background information about how chatbots have grown to be used by corporations as a form of customer service. It also details the many pros and cons that come along with the use of chatbots and the importance of curating quality datasets to train a model on.

Some similar products that already exist in the market include ChatGPT and Google BARD. ChatGPT provides answers on a wide range of topics, can generate code, and do technical computations. This product is very well known and trusted, but it does hold certain features behind a paywall, produces text that is easily recognized as AI-generated, and is slow when the conversation window is too large. Google BARD uses natural language processing and machine learning to generate human-like conversations. However, this product struggles with specific or niche topics and is still in the experimental phase, leading to gaps in knowledge and understanding.

Many of the products that are similar to ours are not open source. Our chatbot will be open source, designed with a focus on empathetic answering, accessibility, and the protection of user information. We will not be storing any user data or training our model on anything the user tells our chatbot. Our product will have the session information destroyed once the users have exited the website, be completely free to use for everyone, and will not require any type of account. The purpose of our product is to simply, quickly, and accurately answer user questions in a trustworthy manner.

4.1.3 Technical Complexity

This project has several components that require varying levels of complexity. A low-complexity component of our project is the construction of the user interface. The actual development of the frontend of our website is not very complicated, but will require time and effort to connect it with the backend that contains our trained model.

Some medium complexity parts of this project include the data security of the information the user provides in the chat session and making the user interface appealing and accessible. One of our top priorities is keeping users' privacy safe and session-based. Once the user has exited the chat session they were in, we will not store any of the information they disclosed in it and ensure that the data we use to train our model consists of datasets we purposely crafted for it. Designing what the user will see and interact with when using our product has medium complexity because we need to focus on certain aspects of interface design, such as contrast, navigability, and accessibility, that require proper research and time to decide on as a team.

A high complexity component of our project is the model having tone adjustment in its response to the user based on sentiment analysis. Depending on what text the user inputs, our model will respond with a different tone to personalize and reassure the user in a specific way. This task will require an accurate and time-efficient implementation of sentiment analysis, along with tailored instructions to our model when it produces its response. This will require more datasets, training, and time to implement.

Another high-complexity component of this project was the implementation of fine-tuning our model. This requires a great deal of debugging and errors, whether it be because we ran out of memory, have dependency issues, our virtual machine is unresponsive, or our training arguments were not ideal. Depending on the size of the dataset used to train the model, it would take multiple iterations of waiting for the model to finish training to test it and find another issue.

4.2 Design Exploration

4.2.1 Design Decisions

Key design decisions:

1. Implementation of multimodal input from the user
    - Importance: it gives the user more options to express their query and gives more context to the model; it makes it more accessible
2. Destruction of chat session data once the user is completed
    - Importance: user information is safe, secure, and will not be used by the model for training; ensures user trust in the product, keeping confidentiality
3. Implementation of sentiment analysis to tailor the model's tone to the user
    - Importance: the user may be in a worrisome situation, and our model's tone in its response is important in being empathetic to the user and their feelings

4.2.2 Ideation

For the design decision of implementing multimodal input (text, image, voice), there are five potential design options for multimodal implementation. We used brainstorming and structured analysis techniques to ideate potential solutions for multimodal input. Each team member contributed ideas about input modes and considered the specific need that the user was seeking, such as medical or emergency help. We used user Journey Mapping to visualize how people might interact with the application—and then used a pros and cons analysis for each idea, considering the complexity and user accessibility.

1. Text-Only Input
    a. Description: Start with text-only input and add some prompts(e.g., "How are you feeling?") to encourage users to provide context
    b. Pros: Simple to implement, computationally cheaper, and no added data handling or model changes
    c. Cons: Limits the user who prefers voice or image input and potentially inaccessible to those who struggle with typing
    d. Outcome: The baseline solution with minimal work
2. Text-Image Input
    a. Description: Allows the user to upload photos to the chatbot (e.g., a wound) for classification or analysis
    b. Pros: Adds valuable context for medical response and another accessibility option
    c. Cons: Requires resources for image processing and potential privacy concerns

    d. Outcome: High impact but requires more resources and conditional on data security and processing cost
3. Voice Dictation Input with possible emotion detection
    a. Description: Enable voice input with an optional emotion detection to determine emotional cues in the user's tone
    b. Pros: Accessible for users who find typing difficult and enhances empathic response
    c. Cons: Emotion detection can be inaccurate, and voice processing is resource-heavy. Privacy concerns with capturing user speech data.
    d. Outcome: potentially valuable for future stages with more advanced audio processing
4. Multi-Stage Input
    a. Description: Allows the user to choose their input mode (text, image, or voice) at the start of the interaction, with responses tailored to the input type
    b. Pros: Offers flexibility and accessibility
    c. Cons: More UI complexity and may delay interaction if the user is unfamiliar with each mode
    d. Outcome: A very flexible and user-friendly approach to enhance the accessibility for the user's needs
5. Hybrid Input (Dynamic mode based on context)
    a. Description: The application input modes based on context (e.g., text for stress, image for injury)
    b. Pros: Tailored to the user experience based on their needs
    c. Cons: Highly complex to implement, may confuse the user if suggestions are not intuitive
    d. Outcome: Flexible and streamlined approach allowing the most user-friendly but complex implementation.

4.2.3 Decision-Making and Trade-Off

We have created a weighted decision matrix to determine the input option for our multimodal application. We used a weighted decision matrix because it is a great tool to evaluate each option systematically. We scored it based on the criteria important to our project goal: Cost, time to implement, accessibility, value to the user, and privacy. Each criterion is weighted to reflect the importance, with time to implement and value given higher weights to ensure user-friendly and correctness to the user.

		Text-Only		**Text-Image**		**Voice Dictation**		**Multi-Stage**		**Hybrid Mode**	
Criteria	**Weighted Rating**	Rating	WR x Rating	Rating	WR x Rating	Rating	WR x Rating	Rating	WR x Rating	Rating	WR x Rating
Cost	3	10	30	5	15	4	12	4	12	5	15
Time to implement	10	10	100	7	70	2	20	2	20	1	10
Accessibility	8	9	72	7	56	8	64	10	80	10	80
Value to user	9	8	72	6	54	6	54	7	63	8	72
Privacy	7	9	63	4	36	5	35	8	56	7	49

Total value	46 / 337	29 / 231	25 / 185	31 / 231	31 / 226

Table 4.2.3.1: Weighted Rating Comparison

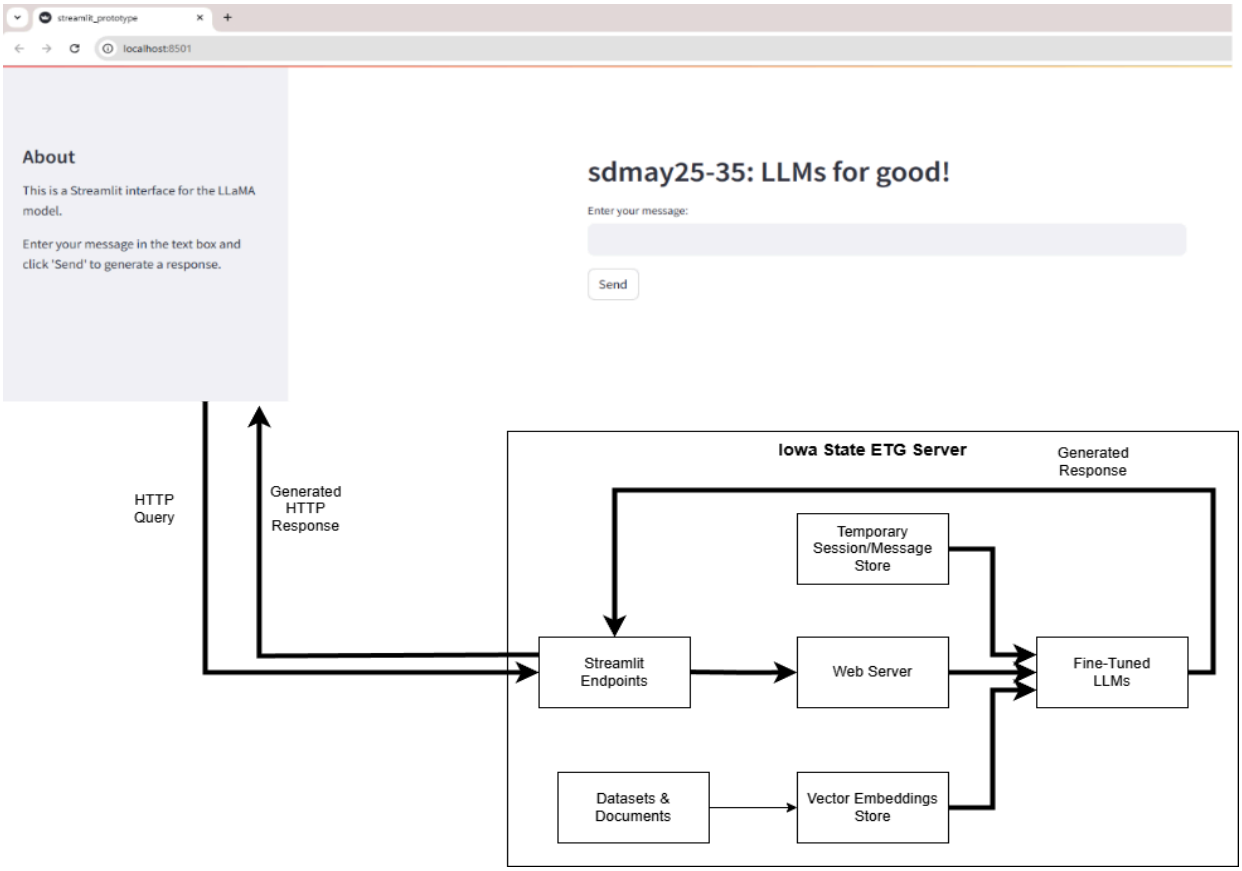
After calculating the weighted score, the Text-only with enhanced prompt option was the high-scoring idea. This approach was chosen because it met our project's core requirements for accessibility, privacy, and cost-effectiveness without adding significant implementation complexity. In addition, the text-only with enhanced prompts allows us to gather essential user context and help support an empathic response by guiding the user to share relevant details. This choice reflects a balanced approach by effectively addressing user needs while minimizing privacy risks and costs.

4.3 Proposed Design

4.3.1 Overview

Our project will be designed with an LLM, which is a type of artificial intelligence, to create responses to inquiries. The user can input text or images for their inquiry. Once they do so, the LLM will respond with empathetic and helpful responses to supply information that the user needs. The LLM will be trained on relevant datasets so that it can quickly and accurately respond to topics such as mental health and emergency aid. Our project will have an interactive experience that looks and feels similar to artificial intelligence chatbots like ChatGPT, which resemble a messaging format between the user and the LLM.

4.3.2 Detailed Design and Visual(s)



Section 4.3.2.1: Detailed Design Diagram

The user will send a query in text and/or image format to the model we built. This query will be sent to the ISU server our model is running on, assuming that the user had access to the internet when they sent their query. Our server will receive this query and feed it to our retrieval augmented generation LLM, which has been fine-tuned on handpicked high-quality datasets. These datasets get converted into vector embeddings when they are fed to the LLM. We also will have to temporarily store chat history for the specific chat session the user is having. This chat history will be destroyed at the end of the chat session. Once the LLM has processed the user query, a response will be generated in text form and displayed on the user's screen. The frontend design is much like that of other chatbot products, but we want to focus on it being easy to use and inviting for users.

Frontend: This will be what the user will see and interact with. They can enter a text query and receive a text response back, having everything displayed on the screen.

Iowa State ETG Server: This is the server that will host our model. This contains the whole backend of our project.

Streamlit Endpoints: These endpoints will be used to transmit the data to and from the back and front end of our application.

Web Server: The server that handles communication to our fine-tuned LLM.

Datasets & Documents: The datasets will be in JSON format, with the layout of questions and responses. These datasets will be used to fine-tune our LLM, so we will train our model with them. Any documents can be used to increase our model's external knowledge base for retrieval augmented generation.

Vector Embeddings Store: The datasets and documents must be turned into numerical representations so our model can process them. We must have a vector embedding store to do this.

Temporary Session/Message Store: Our model will keep information about the current chat session, but once the user has exited that session, the data will be sanitized, so we need a temporary message storage system.

Fine-Tuned LLM: This is our finalized and trained model. This will receive the user query, reference any relevant information from the chat session history, and use its knowledge from the database and documents to generate a reliable answer. This answer will get sent back to the front end for the user to see on the screen.

4.3.3 Functionality

Our design needs to be able to be used in situations that may be urgent or time-sensitive. A user may open the application on a website browser, input an inquiry, receive a quick response that they expect to be able to immediately take information away from, and then they will exit the application. The LLM response from the system should be virtually instantaneous and easy for the user to understand. The visual interface of the system should follow accessibility design principles, such as color contrast, readable text fonts, sizes, etc. so that the user does not have to struggle to interpret the information returned from the LLM.

Figure 4.3.3.1: Initial Design Storyboard

4.3.4 Areas of Concern and Development

Our current design meets the users' needs, which includes having an accessible, secure, empathetic, and trustworthy model to submit queries. The design we have has an emphasis on user navigability and care.

The primary concern our team has for our design is being able to provide comprehensive information through the LLM used in our project. We have discovered that there are not many datasets that provide useful and relevant data for the specific topics that we are interested in. If we are unable to find quality datasets to train the LLM on, we may fail to meet the users' needs and motivation for using our product because the LLM will be unable to respond to certain themes the user may be interested in.

Our immediate plan to resolve this concern is to curate our own datasets. It's a slower and more manual process than being able to import existing datasets. However, it may actually improve the quality of our product in general because it involves more verification and information checking from within our own team. Some questions we have for our faculty advisor are: how do we start to collect data from certain sources? What types of sources should we be gathering data from? What tools can we use that make data collection and organizing easier?

4.4 Technology Considerations

Some technologies we are using include Hugging Face, Langchain, FAISS, Virtual Machine, LLMs, and databases. We are using only open-source materials for our project, which makes it much more convenient for us to design our model because of how accessible they are. Some possible weaknesses include limited GPU use, paywalls for desired materials, and biased datasets. Solutions to these problems are utilizing our budget of $500 when necessary, attaining a powerful enough virtual machine that can handle our model, and cultivating a high-quality and unbiased set of data to train our model on.

A tradeoff we may encounter is if the open-source materials we have available do not exactly give what we need from them; in that situation, we could always modify the resource to fit our needs more specifically.

4.5 Design Analysis

So far, we have experimented with implementing two machine-learning techniques on different pre-trained LLMs. One technique is Retrieval Augmented Generation, which helps optimize the output of a model by feeding it data grabbed from external sources without modifying the model itself. Fine-tuning modifies the LLM itself by adjusting the weights and parameters it uses to produce a response. Both of these methods have worked in our experiments, but we are focusing on using fine-tuning for our purposes because it allows us to modify the model to cater to specific topics. We have begun implementing fine-tuning scripts on our team's virtual machine. We've been having issues with not having enough memory on our VM to run the fine-tuning scripts, as well as dependency and general troubleshooting conflicts. However, these problems do not complicate the feasibility of our design and are virtually all build issues, as well as issues that come with being beginners at implementing an LLM, especially within a virtual machine space.

5. Testing

5.1 Unit Testing

The units that are being tested for this project include code that fine-tunes our LLM and trains it with retrieval augmented generation, which sets up our LLM model's output. To test fine-tuning, we observe the output analytics and make adjustments as needed. Testing of the RAG training is done by running the model and asking it questions pertaining to the read document. To test the setup of model output, boundary testing is beneficial. Testing should be done as each of these steps is taken, not all of them together.

5.2 Interface Testing

Our design has a front end interface. This interface is what the user will be seeing and interacting with to communicate with our chatbot. The user will be able to enter text or images to submit their query. The model will then respond in text format for the user to read on the screen. This can be tested using any browser that has access to the internet. If we are able to develop an app for this project, that can be tested and written in Andriod Studio.

Another interface we will have is the backend API interface. This is the intermediary between our frontend interface and the trained LLM that will receive the user input and generate a response. A good tool to test this function is Postman. Using this will ensure that each endpoint is correctly processing requests.

5.3 Integration Testing

One of the most important critical integration paths in our design is the connection of the frontend interface with the backend LLM. This path is the entire functionality of our project: providing a user with a website where they can send questions and receive answers from our model. Incorrect transmission of the data to and from the model will disrupt the user's experience and make the chatbot inaccessible. As mentioned previously, Postman can be used to validate the API endpoint functionality, and for the front end, we can test that ourselves with a web browser.

Another critical integration path is the completion of training our model on the datasets we curated for it. We must ensure that the data we train our model on is up-to-date, unbiased, and factual. Without this, the model could generate responses with incomplete or outdated information, making it less useful to the user. This can be tested manually by asking specific questions to our model and ensuring that its responses are what we desire.

5.4 System Testing

System-level testing strategy is a type of testing that looks at the application as a whole, not just individual components like unit testing. During this testing phase, all the critical functionalities of the system are tested and must work together for successful completion.

For unit tests, we must ensure that our LLM is fully functional and trained enough to return reliable and truthful responses to various user queries. We also must verify that the data disclosed in the chat sessions is secure and that it gets properly sanitized once the user has ended the chat session.

Interfaces we will have to test include the front end of our application and the backend API. As mentioned earlier, we will use Postman to test the backend API functionality and ensure that our front end is desirable manually.

Our set of integration tests includes the connecting of the backend and frontend of our project and load testing to ensure our model can handle the maximum amount of concurrent users and confirm its scalability.

5.5 Regression Testing

New additions will be tested to the same methods that the original chunks were tested under in a local testing area before being pushed to our main model on the ISU network. We need to ensure to not break the basis of our model or the user's ability to communicate with it due to our functionality requirements.

5.6 Acceptance Testing

The design requirements will be demonstrated to be met by our documentation of the process, from identifying the client's needs to fulfilling them with a final project. The client will see multiple versions of the project as it progresses and was involved in the early planning stages for them to give feedback on what they liked or disliked.

5.7 Security Testing (if applicable)

Security testing for this project will ensure that users have proper permissions to interact with our server. This will include testing against the system as a potentially malicious user. Additionally, we need to test to ensure that session data is erased for each session created by any given user.

5.8 Results

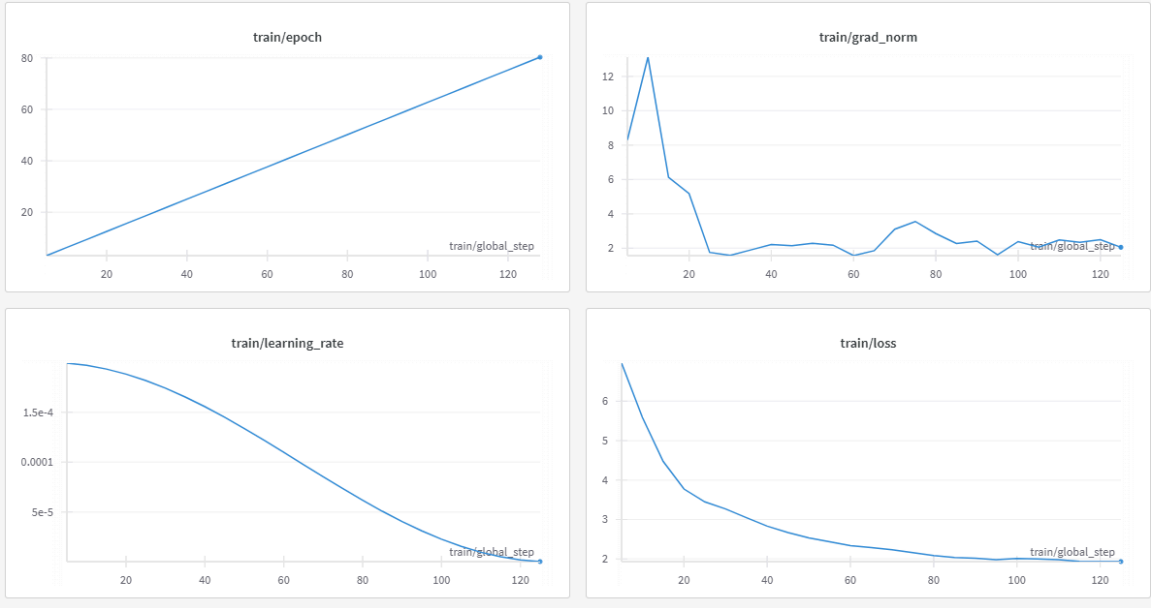


Figure 5.8.1: LLM Training Metrics

Training Progress

The epoch graph displays a steady linear increase from 0 to 80 over 120 global steps, indicating consistent training progression.

Model Optimization

The gradient norm graph shows initial volatility with a peak around 12, followed by stabilization below 4 after the first 20 steps, suggesting effective model optimization.

Learning Dynamics

The learning rate exhibits a controlled decay pattern, starting at approximately 1.5e-4 and gradually decreasing throughout the training process, which is typical for adaptive learning rate schedules.

Loss Convergence

The loss curve demonstrates expected training behavior, starting at around 6 and steadily decreasing to approximately 2 by the end of training, indicating successful model convergence.

These metrics collectively suggest a well-behaved training process with proper optimization and convergence characteristics.

6. Implementation

Retrieval Augmented Generation (RAG)

RAG has proven to be the most successful implementation thus far, offering several advantages:

- Optimizes model outputs using external data sources
- Maintains model stability while expanding the knowledge base
- Allows for dynamic information updates without retraining

Fine-Tuning Experiments

While fine-tuning was also attempted, it presented several technical challenges:

- Memory constraints on the virtual machine
- Dependency conflicts during implementation
- Limited availability of quality training datasets

Frontend Development

A preliminary frontend has been implemented using Streamlit, serving as a temporary solution until a custom interface is developed. The current implementation offers:

Core Features

- Text-based user input interface
- Response display functionality
- Basic chat history tracking

Technical Challenges

Several implementation hurdles have been encountered:

- Virtual machine memory limitations affecting model training
- Dependency conflicts requiring resolution
- Dataset quality and availability constraints

Next Steps

- The implementation roadmap includes:
    - Developing a custom frontend interface
    - Optimizing the RAG implementation
    - Expanding the knowledge base with curated datasets
    - Implementing security protocols for data handling

The focus remains on maintaining user privacy while delivering accurate, empathetic responses within the 15-second response time requirement.

7. Ethics and Professional Responsibility

7.1 Areas of Professional Responsibility/Codes of Ethics

Area of Responsibility	Definition	Relevant Item from IEEE Code of Ethics	Team Interaction/Adherence
Work Competence	Ensure tasks are performed to the best of your ability and stay updated with the knowledge.	"To improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies, including intelligent systems."	Our team operates with a focus on progressive elaboration. We try to produce high-quality results with the intention of improving upon those results as time continues.
Financial Responsibility	Managing resources effectively and avoiding unnecessary expenses	"To hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment."	The team will confidently not use the entirety of our senior design budget. We are hoping to avoid paying for things that are not absolutely necessary and find free alternatives.
Communication Honesty	Sharing truthful and transparent information with team members	"To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, to be honest, and realistic in stating claims or estimates based on available data."	The team holds importance in honesty with each other. When we have questions regarding another member's work, it is done so respectfully. If any members have scheduling conflicts, they are transparent and give status updates to explain what they worked on.
Health, Safety, Well-Being	Protecting the health and safety of the people involved	"To disclose promptly factors that might endanger the public or the environment."	We want to make sure that our users stay safe and are not harmed at all because of our product. We will put great effort into making disclosures about how LLMs cannot be 100% accurate with responses.
Property Ownership	Respecting intellectual and physical property	"To credit properly the contributions of others and to avoid injuring others, their property, reputation, or employment."	Our team always attributes the work of others correctly and adheres to licensing terms for open-source resources, ensuring respect for intellectual property.
Sustainability	Ensuring development meets present needs without composing future resource	"To strive to comply with ethical design and sustainable development practices."	Our team designs with sustainability in mind by minimizing the energy use of our model. We aim to optimize our LLM and increase the efficiency of our code to use as little energy as possible.
Social Responsibility	Considering the impact of work on society and ensuring benefits to the community	"To treat all persons fairly and with respect, and to not engage in harassment or discrimination."	We ensure our AI is designed with fairness and inclusivity in mind, actively avoiding bias and accuracy in the chatbot we develop.

Table 7.1.1: Areas of Professional Responsibility

Performing Well: Communication Honesty

- This is relevant to our project because we are providing a product that was proposed by our advisor, with whom we must communicate details about scope and deadlines.
- Our team's approach to this communication is by having weekly meetings with our advisor to give progress updates and learning sessions, and discuss future goals.
- This upholds our ethical and professional responsibilities because we are truthfully reporting our work to our stakeholder, which is our advisor.

Needs Improvement: Health, Safety, & Well-being

- This is relevant to our project because we want our product to do good for the public and our stakeholders.
- Our team's approach to this is to be empathetic to our stakeholders and prospective users by hopefully implementing sentiment analysis in our model and listening to our advisor/stakeholder concerns.
- We will change our approach by taking our advisor's worries more seriously and keeping close tabs on the progress he desires

7.2 Four Principles

	Beneficence	Nonmaleficence	Respect for Autonomy	Justice
Public health, safety, and welfare	Provide resources for safety and well-being	Avoid spreading misinformation	Allow users to control how they use health tools	Ensure equal access to safety resources
Global, cultural, and social	Address diverse cultural needs	Avoid harm to specific cultural groups	Respect cultural practices in platform design	Promote fair access across diverse groups
Environmental	Minimize unnecessary environmental impact	Avoid creating unnecessary environmental impact	Allow users to select resource-conscious options	Distribute benefits without causing harm to others
Economic	Reduce costs for users through efficient design	Avoid imposing financial burdens on vulnerable groups	Provide options for varying economic situations	Ensure affordability and accessibility for all

Table 7.2.1: Broader Context and Four Principles

Strength:

- Public health, safety, and welfare — Beneficence:

  This platform aims to improve well-being by providing safety and health resources. We ensure this by prioritizing accurate information and easy access for users.

Area to Improve:

- Environmental — Nonmaleficence:

This project may not significantly affect the environment, but its positive social impacts, including improved safety, global equity, and economic fairness, outweigh potential concerns. These benefits justify its resource use while the team works toward integrating more sustainable practices in the future.

7.3 Virtues

Empathy: This is important because it makes us understand the diverse needs of the user, especially in challenging situations. It ensures that the project is designed with care and consideration for the user's experience, providing the application is empathetic for those in need. By empathizing with the users, we can create effective solutions. We also must empathize with each other and ensure that we have a positive team environment.

Integrity: This virtue is important to our team because we as a team have never had to work on such a large project together before. It is important that we all remain honest with each other and communicate when issues are encountered. We all have to hold each other accountable for the work that we do or are not able to do.

Respect: This virtue is important to us because this team consists of six individual members, each bringing their own unique perspective and knowledge to this project, and we want to treat each other with respect, no matter what. We make sure to let everyone in the team speak their opinion and consider all ideas with seriousness.

Abrahim:

I have demonstrated the virtue of empathy so far in this project. I know that our project can be super helpful since I work as a Resident Assistant doing what our project should do but on a smaller scale. I know how it feels to be in a stressful situation where you need guidance, and you spend hours scouring the internet to find an answer. I applied this in our senior design project when I looked at datasets or trained the model and when I worked on the frontend.

A virtue that I have not demonstrated enough is patience. I was so eager to finish the project that I tried to cut corners during our research. However, it is not possible to do that so I had to go back and do the research anyway. I have been working on pacing myself and I think it has helped now that we have a strong outline and results from our research.

Brianna:

I have been able to work through this project with the virtue of purposefulness. It is important to me to know the purpose of your work in order to work efficiently towards it. I have used this within our senior design project by identifying our weekly goals and understanding how I can best work towards those, then contributing in those ways.

A virtue I value but have not been able to make use of during this project is organization. This concept is important, especially in larger projects to ensure things get done in a timely manner and done well. I have been working on recognizing the pieces needed to finish part of the project and focusing on them one at a time, which has helped.

Ellery:

A virtue that I have demonstrated in this senior design project is curiosity. It is important to me because it helps drive a deeper understanding of the project's complexities. It allows me to explore more solutions and look for areas of improvement for the project. I have demonstrated this by constantly engaging in LLM research and exploring new technology.

One virtue that I have not demonstrated in the project is confidence. This aspect is key to making decisive contributions and effectively communicating ideas. It helps in presenting work and giving updates in team discussions. I can demonstrate this virtue by actively sharing more ideas and taking more leadership roles.

Emma:

A virtue that I have demonstrated in this project is honesty. Throughout the entire project so far, I have been truthful in my progress updates delivered and try my hardest to follow through on tasks I am in charge of completing. This is an important virtue because we are working as a team towards a common goal, so if I were to lie about something, it could put the project timeline in jeopardy.

A virtue that I have not demonstrated in this project is public safety. This is an important virtue because our chatbot is meant to provide humanitarian assistance, whether it be for people who are displaced, need emergency aid, or need disaster information. I will demonstrate this virtue by helping with the quality dataset collection of resources that are more location-specific.

Halle:

A virtue I demonstrate in our project is diligence. I spend several hours a week studying technical resources, in addition to the many hours I contribute towards the development of the project itself. This determination to contribution is significant in that it makes progress towards materializing our project goals into a usable product.

One virtue that I feel I could demonstrate more of is communication. Communication is the backbone of a solid team, and I feel that I need to express my thoughts and ideas not only for the project, but also for the team, more frequently and more clearly. I will demonstrate this by prompting discussions, asking clarifying questions, and seeking feedback from my team.

Gabriel:

A virtue that I have demonstrated in our senior design project is adaptability. As we continue to research and prototype our project, we have constantly had to refine the training methods, datasets, and other aspects of our project. To create the best possible project and adapt to new requirements, it is crucial for me to remain flexible in overcoming roadblocks and adaptable in learning from and improving upon my teammates' work.

A virtue that I have not demonstrated enough in this project is organization. This virtue is important because the organization in our project will help us work more efficiently among team members and also keep track of what people are working on. I can work on maintaining a clearer project folder structure, using consistent naming conventions, documenting code, and tracking changes using version control.

8 Closing Material

8.1 Conclusion

Our project aims to develop an empathetic LLM chatbot-based website to assist people seeking humanitarian guidance during crises. Through our implementation efforts, we have successfully experimented with two key machine learning techniques: Retrieval Augmented Generation (RAG) and fine-tuning. RAG has proven more successful thus far, effectively responding to prompts.

The primary goal is to create a secure, accessible platform that provides quick, accurate, and empathetic responses to users in crises, with responses delivered within 15 seconds while maintaining user privacy. Our solution involves implementing a user-friendly frontend interface, a secure backend API running on ISU servers, and comprehensive data transmission protocols.

Several constraints have impacted our progress, including virtual machine memory limitations affecting model training, dependency conflicts during implementation, and the limited availability of quality training datasets. These technical challenges have affected our fine-tuning experiments, though we've maintained progress (regardless of how slow) through RAG implementation.

To move forward, we thought of a couple of things that we could do. First, creating our own higher-quality, specialized datasets to enhance the model's response accuracy and empathy. Second, optimizing the virtual machine usage through better resource allocation and management to address the memory constraints. Third, implementing a more robust testing procedure to ensure consistent response quality and security. Lastly, develop a custom frontend interface rather than relying on temporary solutions to improve user experience and accessibility.

8.2 References

[1] N. Bhirud, S. Tataale, S. Randive, and S. Nahar, "A Literature Review On Chatbots In Healthcare Domain," Jul. 2019. Available: https://www.ijstr.org/final-print/july2019/A-Literature-Review-On-Chatbots-In-Healthcare-Domain.pdf

[2] N. Johari and P. Ne, "QUALITY ATTRIBUTES FOR A GOOD CHATBOT: A LITERATURE REVIEW," *Article ID: IJEET_12_07_012 International Journal of Electrical Engineering and Technology (IJEET)*, vol. 12, no. 7, pp. 109–119, Jul. 2021, doi: https://doi.org/10.34218/IJEET.12.7.2021.012. Available: https://iaeme.com/MasterAdmin/Journal_uploads/IJEET/VOLUME_12_ISSUE_7/IJEET_12_07_012.pdf

[3] G. Caldarini, S. Jaf, and K. McGarry, "A Literature Survey of Recent Advances in Chatbots," *Information*, vol. 13, no. 1, p. 41, Jan. 2022, doi: https://doi.org/10.3390/info13010041. Available: https://www.mdpi.com/2078-2489/13/1/41

9 Team

9.1 Team Members

- Gabriel Carlson
- Brianna Norman
- Halle Northway
- Ellery Sabado
- Abrahim Toutoungi
- Emma Zatkalik

9.2 Required Skill Sets for Your Project

- Cybersecurity Standards Knowledge: Important for the security of the resource and the comfort of users.
- Website/Application Frontend Development: Used for users to be able to interact with our created model. It should support all of our functionality.
- Website/Application Backend Development: This project requires a secure and well-built backend in order to best support our model and user interaction with it.

9.3 Skill Sets covered by the Team

- Cybersecurity Standards Knowledge: Gabriel, Brianna
- Website/Application Frontend Development: Brianna, Emma, Ellery, Abrahim, Halle, Gabriel
- Website/Application Backend Development: Gabriel, Halle, Ellery

9.4 Project Management Style Adopted by the team

Agile is best for incremental steps and the ability to adapt to changes.

9.5 Initial Project Management Roles

- Assignment Manager: Emma Zatkalik
- Stakeholder Liaison: Abrahim Toutoungi
- Project Deliverables Manager: Brianna Norman
- Meetings Coordinator: Halle Northway
- Communications Manager: Gabriel Carlson
- Timeline Coordinator: Ellery Sabado

9.6 Team Contract

Team Members:

1) Abrahim Toutoungi 2) Gabriel Carlson

3) Brianna Norman 4) Halle Northway

5) Emma Zatkalik 6) Ellery Sabado

Team Procedures

Day, time, and location (face-to-face or virtual) for regular team meetings:

Regular team meetings will occur every Thursday from 16:15 to 17:15. These meetings will have hybrid attendance in the sdmay25-35 discord channel and Coover 2222.

2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-

mail, phone, app, face-to-face):

Our primary form of communication is the sdmay25-35 discord channel for updates, reminders, issues, scheduling, etc. Team members also have access to each other's emails and phone numbers for a backup secondary method of communication.

3. Decision-making policy (e.g., consensus, majority vote):

While consensus would be ideal for our decision-making, if we disagree, we will use a majority vote along with input from our advisor to determine what decision is best for the project.

4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be

shared/archived):

Students will take meeting minutes during our weekly meetings. These meeting minutes will be shared in the #meeting-minutes channel in our sdmay25-35 discord server.

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:

All students are expected to attend all team meetings unless they have an excused or unexpected absence. Team members should notify the team of any anticipated absences/tardiness at least 24 hours in advance when possible. Students should come prepared to participate in the weekly meetings with their reflections on the past week, any questions they have, items that need clarification, etc.

2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:

Everyone is expected to contribute to team assignments with equal levels of effort and meet the agreed-upon deadlines for completing things unless they have previously alerted the team that they need more time. As discussed in class, individual contributions may not be perfectly equal for all assignments, but equal participation is the overall goal.

3. Expected level of communication with other team members:

Team members are expected to regularly check our main form of communication: our discord server. During the school week, students are responsible for keeping up to date with project communications and should respond to tags in Discord within 24 hours or less if possible. If a student fails to respond to a message and the message is still relevant, they will be pinged again. Important things to communicate include things like problems with tasks, questions, behind deadlines, progress updates, etc.

4. Expected level of commitment to team decisions and tasks:

Each member of the team should accept and commit to team decisions. Despite the different pedagogical approach in 491 compared to our traditional courses, it is still expected that students treat their commitment and dedication as if it were any of their other classes/classwork.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction,

individual component design, testing, etc.):

Assignment Manager: Emma Zatkalik

Stakeholder Liaison: Abrahim Toutoungi

Project Deliverables Manager: Brianna Norman

Meetings Coordinator: Halle Northway

Communications Manager: Gabriel Carlson

Timeline Coordinator: Ellery Sabado

2. Strategies for supporting and guiding the work of all team members:

Creating and trying our best to follow a team calendar that lists out the tasks each person should be completing. Also, following along with deadlines made from the Gantt chart we made. Consistently updating the team on progress made regarding the project.

3. Strategies for recognizing the contributions of all team members:

Have a retrospective. Start the project with a set of tasks needed for successful completion of the project. Keep track of what has been completed.

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the

team.

Brianna - Familiar with cybersecurity standards and common vulnerabilities in web applications. Worked with user accounts and front-end development of an application. Strengths include problem-solving/identification, vulnerability assessment, and data analysis. Knowledgeable in C, Java, and moderately familiar with Python.

Emma - Experience with frontend mobile application development and is in the process of learning website development. Good with creating low-level designs and debugging. Knowledgeable in C, Java, and SQL.

Ellery - Experience with frontend development and backend development. Developed MERN stack web application with MongoDB and JSON files. Worked on the backend with Java and MySQL for an application(309). Right now learning a Mobile application for Android Studio that works with Java and XML. Some experience in Arduino Uno and C++. Experience with C coding with embedded systems.

Gabriel - Experience with Fullstack development using Ruby on Rails, Python, Javascript, React, Vue, and NodeJs. Worked on projects focusing on different user data format ingest, improving UI/UX, cloud deployment, information security, database management, session management, etc. I also have experience working on a team as the Systems Team Lead on ISU's MAVRIC rover project.

Abrahim - Experience with frontend development (from 309). Did a bunch of C stuff. Used rust on a personal project. Pretty familiar with Java since that's what they teach us here. I'm okay with Python since easy syntax and good documentation. I have worked on model-based stuff as well. I currently work part-time doing app development for J.B Hunt, mostly Java.

Halle - Experience with full-stack development using Java, Python, C, and Dart. Most familiar with projects developing graphical user interfaces, as well as data management, for Windows and mobile-based applications. Good with organization, facilitating communication, and debugging.

2. Strategies for encouraging and support contributions and ideas from all team members:

Take time for multiple ideas to be presented in brainstorming sessions. Ask each other about work status in meetings to encourage discussion and questions.

3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)

Encourage open and honest communication between group members. Allow structured time in meetings to listen about road-blocks in work and brainstorm solutions to them.

Goal-Setting, Planning, and Execution

1. Team goals for this semester:

Identify all tasks needed to complete the project successfully. Create an initial and flexible schedule for the tasks to be completed. Maintain solid communication and promote a positive team environment.

2. Strategies for planning and assigning individual and team work:

Understand the scope and end goal of the project, creating milestones and pieces needed for completion along the way. During weekly meetings, split these pieces between people based on areas of expertise, interest, and availability to complete milestones on time.

3. Strategies for keeping on task:

Our reflections and check-ins at our weekly meetings will be one of our strategies for keeping us on track. We will be keeping a list of tasks that must be completed and their ideal timelines using a Gantt chart. This chart will act as a guide for project progress and allow us to adjust tasks and their deadlines accordingly. We will also have an additional meeting among group members each week to update each other on our progress. We will also use the GitLab issue board.

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?

Depending on the severity of the infraction and the impact on the project, we may approach issues differently. If it is a small issue, we will first discuss it with the individual to see if we can get back on track or if we need to make accommodations. If it is a bigger issue with a large impact on the project and we do not feel comfortable discussing it individually first, we will speak with one of our course instructors and our advisor on how to proceed.

2. What will your team do if the infractions continue?

If the infractions continue and an individual resolution cannot be reached, we will contact the course instructor and our advisor. If this is a repeated infraction, we will inform the instructor/advisor about the repeated behavior and how to proceed.

****

a) I participated in formulating the standards, roles, and procedures as stated in this contract.

b) I understand that I am obligated to abide by these terms and conditions.

c) I understand that if I do not abide by these terms and conditions, I will suffer the

consequences as stated in this contract.

1) Abrahim Toutoungi DATE 7/12/24

2) Gabriel Carlson DATE 7/12/24

3) Ellery Sabado DATE 7/12/24

4) Emma Zatkalik DATE 7/12/24

5) Halle Northway DATE 7/12/24

6) Brianna Norman DATE 7/12/24