A REPORT
ON

AI ASSISTED TELE-MEDICINE KIOSK FOR RURAL INDIA

Submitted by,

R KESHAV	-	20211CAI0080
RAKSHITHA KT	-	20211CAI0087
S SRINIVAS	-	20211CAI0109
PREM JE KALISTER	-	20211CAI0187

Under the guidance of,

Dr. AKSHATHA Y
Assistant Professor-Selection Grade
in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY
IN

COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

BENGALURU

MAY 2025

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project report **AI ASSISTED TELE-MEDICINE KIOSK FOR RURAL INDIA** being submitted by R KESHAV, RAKSHITHA K T, S SRINIVAS, PREM JE KALISTER bearing roll number 20211CAI0080, 20211CAI0087, 20211CAI0109, 20211CAI0187 in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering (Artificial Intelligence and Machine Learning) is a Bonafide work carried out under my supervision.

16/5/25

Dr. AKSHATHA Y
Assistant Professor
PSCS
Presidency University

Dr. ZAFAR ALI KHAN
Professor & HoD
PSCS
Presidency University

Dr. MYDHILI NAIR
Associate Dean
PSCS
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-Vice Chancellor - Engineering
Dean –PSCS / PSIS
Presidency University

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the report entitled **AI ASSISTED TELE-MEDICINE KIOSK FOR RURAL INDIA** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering (Artificial Intelligence and Machine Learning)** , is a record of our own investigations carried under the guidance of **Dr. AKSHATHA Y, Assistant Professor, Presidency School of Computer Science and Engineering, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

R KESHAV	20211CAI0080
RAKSHITHA KT	20211CAI0087
S SRINIVAS	20211CAI0109
PREM JE KALISTER	20211CAI0187

ABSTRACT

Healthcare accessibility in rural India remains a critical challenge, with limited availability of expert doctors and timely medical intervention. This project proposes an AI-assisted Telemedicine Robotic Kiosk, a self-sustained, automated healthcare unit that can be deployed across villages to bridge the gap between rural patients and specialized medical professionals. This Project aims to develop a medical chatbot tailored for Indian Healthcare context by fine-tuning open-source large language models (LLMs) on a synthetic dataset comprising of 300 commonly occurring diseases in India. The custom dataset was designed in ShareGPT- style format to align with conversational instruction tuning.

To make efficient use of these models with minimum computational cost, methods like Quantized Low- Rank Adaptation(QLoRA), Parameter Efficient Fine-Tuning(PEFT) using LoRA and instruction tuning. These methods allow fine-tuning on general purpose GPUs while also preserving contextual relevance and domain accuracy for medical purpose. The final model, fine-tuned Mistral 7B demonstrated best performance on evaluation metrics such as ROUGE and BERTScore, showcasing enhaced fluency, medical relevance and response coherence. The chat bot is integrated with speech-to-text technology via whisper for real-time deployment. In conclusion, the project presents a scalable, resource-efficient solution to improve digital healthcare accessibility in low-resource settings. It also holds potential for wide variety of applications including automating patient documentation and increasing patient-doctor face-time, thereby improving clinical workflow and user satisfaction.

ACKNOWLEDGEMENTS

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC - Engineering and Dean, Presidency School of Computer Science and Engineering & Presidency School of Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Dean **Dr. Mydhili Nair,** Presidency School of Computer Science and Engineering, Presidency University, and **Dr. Zafar Ali Khan** Head of the Department, Presidency School of Computer Science and Engineering, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Dr. Akshatha Y, Assistant Professor** and Reviewer **Dr. Afroz Pasha, Assistant Professor**, Presidency School of Computer Science and Engineering, Presidency University for his inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the CSE7301 University Project Coordinator **Mr. Md Ziaur Rahman and Dr. Sampath A K,** department Project Coordinators **Dr. Afroz Pasha** and Git hub coordinator **Mr. Muthuraj.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

R KESHAV
RAKSHITHA K T
S SRINIVAS
PREM JE KALISTER

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Background

Access to healthcare remains a critical challenge in rural India due to the shortage of medical professionals, lack of specialized services, and significant travel barriers for patients. Telemedicine, combined with artificial intelligence (AI), has emerged as a transformative solution to bridge this healthcare gap by enabling remote consultations, real-time diagnostics, and predictive analytics. AI-assisted telemedicine kiosks presents a scalable and efficient solution to bridge the gap. The project introduces a medical chatbot, fine-tuned on a custom dataset of doctor-patient dialogues covering over 300 commonly occurring Indian diseases. The chatbot provides accurate responses on symptoms, home remedies and diagnosis as per patient’s query. Integrated with speech-to-text systems like Whisper, the system facilitates voice-based interaction in regional languages. Such innovations not only improve healthcare accessibility but also enhance the efficiency of rural healthcare infrastructure, ultimately reducing the burden on overcrowded urban hospitals and improving health outcomes in Rural regions of India.

1.2 Problem Statement

- **Limited Access to Healthcare Professionals:** Rural India faces a severe shortage of doctors and medical specialists, making timely healthcare access difficult. As a result, many patients suffer from untreated conditions, leading to preventable health complications.

- **Lack of Specialized Medical Services:** Rural patients often do not have access to specialists, resulting in undiagnosed or misdiagnosed illnesses. This leads to an increased burden of chronic diseases and worsening health conditions in these areas.

- **Inefficiency of Traditional Outreach Programs:** Existing rural healthcare initiatives are limited by logistical constraints and resource shortages. These limitations make it difficult to provide consistent, high-quality medical services to remote populations.

- **Challenges in Implementing Telemedicine Solutions:** Many telemedicine programs lack the necessary infrastructure and accessibility to function effectively in rural areas. Without an integrated and scalable approach, healthcare disparities continue to persist.

1.3 Objectives

- **Accessible Healthcare Through Conversational AI:** Develop and deploy an AI-powered robotic kiosk chatbot to simulate doctor-patient interactions, making healthcare advice more accessible.
- **AI-Driven Preliminary Diagnostics:** Designing and fine-tuning a LLM on custom medical dataset containing realistic, context-rich dialogues that can provide medical guidance by analyzing user-reported symptoms and providing context-aware, medically relevant responses.
- **Scalable and Future-Ready Healthcare Model:** Design the system to be adaptable for future enhancements, such as IoT-based health monitoring. The solution should be deployable in various rural regions, ensuring long-term sustainability.
- **Data- Driven Healthcare Decision Support:** Use AI-generated insights to improve health literacy, recommend home remedies for mild symptoms and guide patients toward appropriate medical consultations.

1.4 Scope of the Project

- **Custom Dataset Creation and Scalability:** Includes development of a custom dataset (MediTalk300) tailored to Indian disease profiles and scalable for additional symptoms, specializations to train the chatbot for handling medical queries related to 300 most occurring diseases in India, covering communicable, non-communicable and mental health issues.
- **Integration with Lightweight Models**: Uses fine-tuned LLMs (e.g., Mistral 7B, LLaMA-3B) on synthetic, role-played medical conversations to generate responses that are medically informative and empathetic Employs parameter-efficient fine-tuning techniques like QLoRA and PEFT to make the system compatible with edge devices and consumer-grade hardware for local deployment.
- **Real-time Voice-to-Voice Interaction**: Converts patient speech into text via Whisper, processes it using a fine-tuned model, and delivers the doctor’s response back in spoken language, making it accessible to the illiterate and elderly.

- **AI-Powered Symptom Analysis for Better Diagnoses:** AI algorithms will analyze patient symptoms and medical history to generate preliminary diagnostic insights. These insights will assist doctors in making informed treatment decisions, reducing dependency on physical consultations for minor ailments.
- **Locally Deployable for Telemedicine and Kiosks:** Designed to be embedded in telemedicine kiosks and rural health centers to offer instant, AI-assisted consultations in areas with limited access to doctors.

This initiative will bridge the rural healthcare gap by leveraging AI, telemedicine, and community health networks to deliver affordable, accessible, and high-quality medical services to Rural communities.

Chapter 2

LITERATURE SURVEY

2.1 Introduction

The integration of artificial intelligence (AI) and telemedicine has emerged as a promising solution to address the healthcare challenges faced by rural populations. Several studies and existing healthcare models highlight the role of telemedicine platforms, AI-driven diagnostics, biometric authentication, and remote monitoring in enhancing medical accessibility. Government initiatives like e-Sanjeevani, AI-powered chatbots, telemedicine kiosks, and wearable health devices have been widely explored to bridge the rural healthcare gap. However, these existing systems still face limitations in scalability, diagnostic accuracy, seamless medicine distribution, and accessibility for low-literacy populations. This literature survey aims to review the current advancements in AI-based telemedicine, existing methodologies, and their challenges, providing a foundation for developing an AI-assisted telemedicine robotic kiosk to improve healthcare delivery in rural India.

1. Telemedicine's Role in Rural Healthcare

Telemedicine is defined as “the delivery of health care services, where distance is a critical factor, by all health care professionals using information and communication technologies for the exchange of valid information for the diagnosis, treatment and prevention of disease and injuries, research and evaluation, and for the continuing education of health care providers, all in the interests of advancing the health of individuals and their communities”.

Telemedicine, a subset of telehealth, plays a crucial role in improving healthcare access in rural India by enabling remote consultations, assessments, and treatments through phone, video conferencing, and web-based applications. With advancements in high-speed wireless technology, telehealth has become more cost-effective and widespread, allowing patients to receive quality care without traveling long distances. It also helps prevent medication misuse and ensures timely reminders for prescriptions and appointments. Telemedicine gained prominence during the pandemic, proving essential for diagnosis, treatment, patient education, and chronic disease management in remote areas. [1]

2. Telehealth Functionality Design

The paragraph discusses several features of a telehealth platform. Telemonitoring uses wearable devices and sensors to track a patient's health remotely, with data sent to the platform for both the patient and healthcare provider to monitor. The platform can remind patients to perform daily health checks like measuring vital signs, and alert them if they take medication for too long, offering health education. Appointment scheduling allows patients to book follow-ups with the same physician for chronic illnesses, while urgent appointments may require matching with suitable healthcare professionals based on the patient's condition. Medical records store important health information like test results, medications, and doctor’s notes. Patients can control their own medical records and allow healthcare providers to access them. The platform also has a feature to send video clips between patients and healthcare professionals, enhancing communication. [2]

3. AI with Healthcare

The paper discusses the design of a Voice User Interface (VUI) for the conversational bot, enabling natural, human-like interactions through voice. It incorporates technologies like Speech-to-Text and Text-to-Speech, along with Speech Synthesis Markup Language (SSML) for more interactive and engaging responses. The bot is designed to offer concise, user-friendly replies and health tips in a conversational tone, mimicking a doctor-patient interaction. The system targets both rural and urban areas in India, offering personalized suggestions, health tips, and location-based recommendations. [3]

4. Technological Barriers and Infrastructure Challenges

This paragraph highlights the various challenges in implementing telemedicine in India's primary healthcare system. Key barriers include technological issues like poor internet connectivity and interoperability, social acceptability concerns due to cultural norms and language differences, and financial constraints stemming from the lack of sustainable funding models. Many patients also lack access to smartphones or computers, limiting their ability to use telehealth services. Cybersecurity concerns, gender influences, and linguistic diversity further complicate adoption. Additionally, regulatory gaps and the absence of long-term funding hinder scalability. The 'Tuver' project in Gujarat is mentioned as a model for sustainable telemedicine approaches.[4].

5. Future Directions for Telemedicine

In the future, AI will revolutionize healthcare by enabling predictive, personalized, and precision medicine. AI-driven systems will provide real-time patient monitoring through wearables and ambient intelligence, allowing early detection of diseases and proactive interventions. Autonomous virtual health assistants will support patients with diagnostics and treatment recommendations, reducing the burden on healthcare professionals. AI-powered drug discovery will accelerate the development of new treatments, while genomics and synthetic biology will pave the way for highly personalized therapies. AI will also streamline hospital workflows, automate administrative tasks, and enhance medical imaging, leading to faster and more accurate diagnoses. Ultimately, AI will create a connected and intelligent healthcare ecosystem, improving patient outcomes, reducing costs, and ensuring equitable access to advanced medical care.[5]

2.2 Existing Systems

Several existing systems aim to bridge the healthcare gap in rural areas through telemedicine, AI-driven diagnostics, and remote patient monitoring. These systems serve as a foundation for the proposed AI-assisted telemedicine robotic kiosk.

a) e-Sanjeevani – National Telemedicine Service (India)

- e-Sanjeevani is a government-led telemedicine platform providing remote healthcare consultations. It offers two models—e-Sanjeevani OPD (direct patient-to-doctor consultations) and e-Sanjeevani AB-HWC (for health workers consulting specialists on behalf of patients).[6]

b) AI-Powered Chatbots & Virtual Health Assistants

- AI chatbots like Babylon Health, Ada, and Symptomate analyze patient symptoms and provide preliminary assessments. These systems assist in triaging patients based on urgency and offer basic health advice before referring them to a doctor.[7]

c) AI-Based Remote Monitoring & Wearable Devices

- Wearable health devices, such as Fitbit, Apple Health, and AliveCor, continuously monitor heart rate, oxygen levels, and blood pressure. These devices integrate AI to detect abnormalities and notify healthcare professionals for timely intervention.[3]

2.3 Limitations in Existing Research

Despite their advancements, these existing healthcare solutions lack key features needed for scalable and effective rural healthcare delivery.

a) e-Sanjeevani – Lack of AI-Based Pre-Diagnosis

- e-Sanjeevani provides direct teleconsultation but does not include AI-driven symptom analysis before connecting patients with doctors. It also lacks biometric authentication, making it difficult to track patient history securely.[6]

b) AI-Powered Chatbots – Dependence on Self-Reported Symptoms

- Chatbots rely solely on self-reported symptoms, which may not be accurate, especially in low-literacy populations. They also do not offer real-time interaction with medical professionals, reducing their effectiveness for serious conditions.[2]

c) Telemedicine Kiosks – Limited AI Capabilities & Medicine Distribution

- While telemedicine kiosks enable remote consultations, they lack AI-driven diagnostics to analyze patient symptoms before a consultation. Additionally, these kiosks do not integrate medicine distribution services, requiring patients to visit pharmacies separately.

d) AI-Based Remote Monitoring – Expensive and Requires Internet Access

Wearable health devices are costly and require smartphone connectivity, making them inaccessible to many rural populations. They also do not provide direct doctor consultations or immediate medical interventions.

Chapter 3

RESEARCH GAPS OF EXISTING METHODS

Despite advancements in AI-assisted telemedicine, existing technologies have several research gaps that limit their effectiveness in rural healthcare settings.

A) Lack of Integrated AI-Driven Triage and Decision Support

- Current telemedicine platforms lack AI-driven pre-diagnostic capabilities that can analyze symptoms and prioritize patient cases before doctor consultations.
- Research is needed to develop AI models that can provide accurate initial diagnoses and suggest personalized treatment plans based on rural healthcare challenges.

B) Limited Physical Interaction & Patient Engagement

- AI chatbots and existing telemedicine kiosks lack robotic interaction, making it difficult to collect accurate symptom descriptions from patients with low literacy levels.
- Research is required to develop robotic interfaces with natural language processing (NLP) in regional languages for better patient engagement.

C) Challenges in Biometric Authentication and Patient Data Management

- Existing biometric-based authentication systems (Aadhaar, smart cards) do not integrate AI for fraud detection or prevent duplicate records.
- Further research is needed to develop AI-powered biometric authentication that can link historical medical records and improve data security.

D) Incomplete Medication & Treatment Follow-Up System

- Many telemedicine platforms only offer consultations but do not ensure medicine distribution or treatment adherence tracking.
- Research is needed to integrate AI-powered tracking systems to monitor medicine intake, follow-up consultations, and real-time patient progress through ASHA workers.

E) Infrastructure and Connectivity Challenges in Rural Areas

- Poor internet connectivity and lack of electricity in villages make existing telemedicine solutions unreliable.

- Research is required to develop low-bandwidth AI models, offline telemedicine solutions, and solar-powered kiosks for continuous operation in remote regions.

F) Lack of Multilingual and Culturally Aware AI Systems

- Most existing AI chatbots are trained primarily in English or major Indian languages, failing to understand local dialects and regional healthcare expressions.
- Research is needed to develop multilingual conversational models trained on region-specific medical terminology and culturally sensitive dialogue patterns to improve trust and comprehension.

G) Limited Use of Edge AI for On-Device Processing

- Current telemedicine models rely heavily on cloud infrastructure, which is unsuitable for rural areas with intermittent connectivity.
- There is a significant need for lightweight, edge-optimized AI models that can run on low-resource devices locally at kiosks without continuous internet access.

H) Data Scarcity and Lack of Contextual Medical Datasets

- Most AI models are trained on Western or urban-centric medical data, which does not represent the disease patterns, symptoms, and treatment approaches common in rural India.
- There is a research gap in the creation and curation of contextually rich, annotated rural medical datasets for model training and validation.

Chapter 4

PROPOSED MOTHODOLOGY

The AI-assisted telemedicine robotic kiosk integrates multiple AI and automation technologies to provide secure, intelligent, and efficient remote healthcare services.

4.1 System Overview

In Rural Areas where the availability of medical professionals and timely diagnosis is critically limited, the healthcare is a matter of grave concern. While Transformer based models such as BERT and GPT were also highly accurate in medical context. But, these models required high computational cost and memory resources which acts as a barrier in resource limited healthcare environments. MediTalk-300 aims to mitigate this resource and language limitations while providing efficient medical assistance.

It is a medical chatbot designed as a conversational kiosk solution. MediTalk300 leverages a fine-tuned Mistral 7B large language model, trained over 300 different commonly occurring diseases in India.

The Core techniques used in Medilk300 is LLM fine-tuning. The system architecture integrates speech-to-text using OpenAI’s Whisper, medical response generation using the fine-tuned Mistral 7B, trained on a dataset of conversational style consisting of 7000 doctor-patient conversation, all these runs on a user-friendly interface, Open Web UI. With a BERT Score of 20% and ROUGE-L of 20%, MediTalk300 outperforms other open-source models by over 40%, showcasing its efficiency in understanding and responding to medical queries. Leveraging techniques such as QLoRA, PEFT, and instruction tuning, the model is optimized for deployment in resource-constrained settings, enabling scalable, and efficient AI-assisted healthcare delivery in Rural areas of India.

4.2 Workflow

The main components of the system are:

1. Speech-to-Text Component (STT)

Tool Used: Whisper (via OpenAI API or local deployment)

Input: Audio from user

Output: Transcribed text (in native Indian language)

Task:

- Convert patient voice input to text
- Detect spoken language
- Forward transcribed native-language text to translation component

Advantages: High Accuracy, Open-source and Flexible Deployment and Automated Language Detection

2. Medical Conversational Model (MediTalk300)

Tool Used: Fine-tuned Mistral 7B on MediTalk300 dataset

Input: English-translated user query

Output: English chatbot response

Task:

- Understand and respond to medical queries
- Use fine-tuned knowledge of Indian diseases
- Maintain coherent multi-turn medical conversations
- Forward output to translation component

Advantages: Domain-Specific Intelligence, Instruction-Tuned, Resource-Efficient, Better Context Retention

3. Text-to-Speech (TTS)

Tool Used: Whisper (or another TTS tool like Coqui or Google TTS)

Input: Translated response in native language

Output: Audio file/speech

Task:

- Convert final chatbot response into audio
- Provide speech output for patient in their native language
- Return output to frontend for playback

4. User Interface Layer (Open WebUI)

Tool Used: Open WebUI (interfaced with models served via Ollama)

Input: Text or voice input from the user via a web browser

Output: Model-generated response displayed in a chat interface

Task:

- Provide an accessible, browser-based interface for interacting with local LLMs.
- Automatically detect and list models available via Ollama (e.g., Mistral, LLaMA).
- Facilitate real-time chat-based testing of fine-tuned models.
- Enable both text and (optionally) voice inputs for conversational evaluation

Advantages: User-Friendly – No coding required to interact with models, Plug-and-Play – Instantly connects with models running via Ollama, Open-Source and Customizable – Can be extended or themed for specific use cases, Ideal for Demos and Testing – Quick model switching and chat history support.

5. Deployment Container (Docker)

Tool Used: Docker

Input: Configuration files and commands to launch Open WebUI and its dependencies

Output: Isolated container environment running Open WebUI accessible via a web browser

Task:

- Package and run Open WebUI along with its runtime environment.
- Ensure consistent and portable deployment across systems.
- Manage dependencies, ports, and resource allocation for the WebUI.
- Simplify setup and execution of the application with minimal manual configuration.

Advantages: Cross-Platform Compatibility – Runs identically on any system with Docker installed, Isolated Environment – Avoids conflicts with system-level dependencies, Easy to Deploy – Single command (docker run) can launch the full application, Lightweight and Scalable – Suitable for local demos or integration into larger kiosk systems.

6. Local Model Serving (Ollama)

Tool Used: Ollama

Input: Local or pre-trained LLMs (e.g., mistral, llama3)

Output: Served models accessible via local endpoints or linked interfaces (e.g., Open WebUI)

Task:

- Host and manage local large language models.
- Handle model loading, execution, and token streaming.

- Enable Open WebUI to access and interact with LLMs without cloud APIs.
- Provide offline inference capability for secure, low-latency deployments.

Advantages: Offline and Private – No internet required after setup, Low-Latency Inference – Optimized for local hardware performance, Supports Multiple Models – Easily switch between fine-tuned or open-source models, Seamless Integration with Open WebUI – Detects models automatically and enables fast setup.

4.3 System Architecture

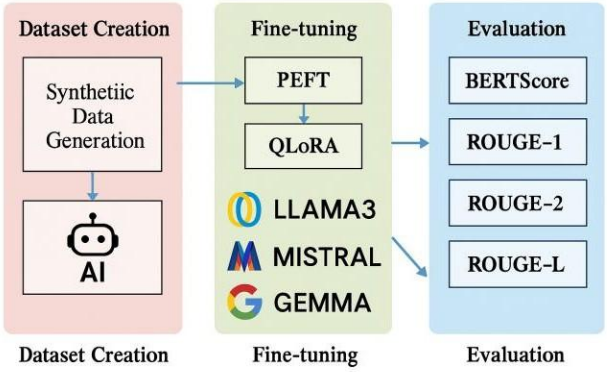


Fig 4.1 System Architecture

The system deploys a structured pipeline for building and optimizing its core medical conversational model, combining synthetic data generation, advanced fine-tuning techniques, and rigorous evaluation metrics.

4.3.1 Dataset Creation

- **Synthetic Data Generation**: LLMs were used to creates domain-specific synthetic conversations and question-answer pairs focused on Indian healthcare scenarios and diseases.
- **Advantages**:
    1. Reduces dependency on limited real-world annotated datasets.
    2. Allows control over data diversity, tone, and coverage.
    3. Enables customization for underrepresented Indian languages.

4.3.2 Fine-Tuning Stack

- **Techniques Used**:
    - PEFT (Parameter-Efficient Fine-Tuning): Reduces the number of trainable parameters for memory-efficient training.

    - QLoRA (Quantized Low-Rank Adaptation): Enables training large models like Mistral, LLaMA3, and Gemma using low-bit precision with minimal accuracy loss.

- **Model Choices**:
  - **LLaMA3**: Efficient translation and general reasoning.
  - **Mistral**: Fine-tuned on the MediTalk300 dataset for medical chat.
  - **Gemma**: Optional experimentation model with multilingual capabilities.
- **Advantages**:
  1. Scalable training even on consumer hardware.
  2. Rapid experimentation across multiple model backbones.
  3. Modular training pipeline allowing iterative improvements.

4.3.3 Evaluation

- **Automated Metrics**:
  - **BERTScore**: Semantic similarity evaluation.
  - **ROUGE-1 / ROUGE-2 / ROUGE-L**: Measures content overlap for summary-style and QA-style tasks.
- **Advantages**:
  1. Objective performance tracking.
  2. Ensures generated responses maintain medical fidelity and contextual consistency.

4.4 Key Algorithms, Tools and Models

4.4.1 Algorithms

1. Whisper (OpenAI) – Speech-to-Text

- Architecture: Encoder-decoder transformer
- Key Features:
  1. Multilingual transcription
  2. Robustness to accents and background noise
- Algorithmic Advantage:
  1. Trained on 680k hours of supervised audio
  2. Capable of language detection + transcription
- Application: Converts spoken input to text in user's native language

3. MediTalk300 – Medical Conversational Model (Fine-tuned Mistral 7B)

- Architecture: Decoder-only transformer (Mistral base)
- Key Features:
    1. Fine-tuned on structured and unstructured Indian medical data
    2. Handles multi-turn dialogue
- Algorithmic Methods:
    1. PEFT + QLoRA for efficient fine-tuning
    2. Instruction tuning for domain specificity
- Use in System: Processes medical queries and generates medically-informed responses

4. PEFT (Parameter-Efficient Fine-Tuning)

- Purpose: Fine-tune large language models efficiently by modifying a small subset of parameters
- Techniques Used:
    1. LoRA (Low-Rank Adaptation)
    2. QLoRA (Quantized LoRA for low-memory devices)
- Use in System: Fine-tunes Mistral 7B without retraining the full model

5. Open WebUI – Model Interaction Interface

- Architecture: Browser-based interface built with a backend API and real-time messaging layer
- Key Features:
    1. Intuitive web interface for chat-based interaction with local LLMs
    2. Automatically lists available models hosted via Ollama
    3. Supports multi-turn conversations, chat history, and (optional) voice input
    4. Lightweight and customizable open-source interface
    5. Seamlessly connects to models like Mistral and LLaMA served locally
- Use in System: Acts as the frontend to chat with your fine-tuned MediTalk300 model, Provides easy access to test multilingual conversations in a real-world demo setting

6. Docker – Containerized Deployment Layer

- Architecture: Containerization platform for packaging and deploying applications in isolated environments.

- Key Features:
    1. Runs Open WebUI with all dependencies bundled via a single container.
    2. Ensures consistent deployment across different hardware and OS.
    3. Supports port mapping and memory allocation for controlled runtime.
    4. Used to deploy and launch Open WebUI locally or on edge devices.
    5. Enables quick setup without manual installation of libraries.
- Use in System: Hosts the Open WebUI interface in a self-contained Docker container, Ensures your chatbot system is portable, reproducible, and scalable for kiosk-based use.

7. Ollama – Local Model Serving Engine

• Architecture: Local LLM host and model runner with optimized inference backend.

• Features:

1. Supports multiple open-source models like Mistral, LLaMA, and Gemma
2. Optimized for fast token generation on consumer-grade GPUs
3. CLI-based management of model lifecycle (pull, run, stop).
4. Automatically connects to Open WebUI as the model backend
5. Efficiently serves your fine-tuned model offline without relying on APIs

• Use in System: Hosts your fine-tuned MediTalk300 model for real-time interaction, Enables secure, low-latency responses without needing cloud-based inference.

5. TTS (Text-to-Speech) Whisper (TTS Mode)

- Architecture: Neural TTS models (e.g., Tacotron + HiFi-GAN / FastSpeech2)
- Features:
    1. Natural prosody generation
    2. Supports multiple Indian languages
- Use in System: Converts chatbot's language reply into speech

6. Evaluation Metrics & Tools

- Models/Tools:
    1. BERTScore: Semantic similarity evaluation
    2. ROUGE-1, ROUGE-2, ROUGE-L: Summarization and lexical match metrics
- Purpose: Measure translation quality, conversational fluency, and content preservation across pipeline stages

4.4.2 Fine-Tuned Models

1. **LLaMA (Large Language Model Meta AI)**

LLaMA is a family of open-source language models by Meta. The 8B variant is a mid-sized model offering a good balance between performance and efficiency.

Unique Features:

- High-quality pretraining on a wide mix of public datasets, including code, academic papers, and books.
- Instruction-tuned variants (e.g., LLaMA-8B-Instruct) make it suitable for chat, summarization, and reasoning.
- Open-weight model with strong community support and tooling (especially with frameworks like Hugging Face and Unsloth).

2. **Gemma 7B (Google DeepMind)**

Gemma is Google’s family of lightweight, high-performance open models. The 7B variant is optimized for instruction-following and reasoning tasks.

Unique Features:

- Optimized for chat and prompt-following tasks out of the box.
- Efficient inference due to improved architecture over earlier models like PaLM.
- Well-aligned responses, especially when fine-tuned, thanks to alignment strategies developed by DeepMind.

3. **Mistral 7B (Mistral AI)**

Mistral 7B is a dense transformer-based language model built with efficiency and quality in mind. It has become a favorite in the open-source community due to its performance.

Unique Features:

- Sliding Window Attention, enabling efficient long-context handling.
- Compact yet powerful, outperforming even larger models (e.g., LLaMA 13B) in many benchmarks.
- Strong multilingual performance, and fine-tuning makes it excel in low-resource, domain-specific tasks (like healthcare).

4.5 Development Framework & Deployment

The system uses a few frameworks and libraries for fine-tuning, translation, and deployment.

1. Fine-Tuning Toolkit: Unsloth

Unsloth is a fine-tuning library that offers high-speed adapters and tokenizers specifically optimized for a variety of instruction-tuned models like Mistral, LLaMA, and Gemma, DeepSeek, Phi, QwQ-32B, Qwen. Unsloth supports PEFT with LoRA and native 4bit/8-bit quantization loading, ideal for consumer-level GPUs.

Unsloth’s ecosystem includes:

- Pre-quantized models
- Chat template formatting
- Fast tokenizer support
- Easy PEFT integration

2. Ollama: Local LLM Runner for Private, Fast AI Applications

Ollama is a powerful, developer-friendly platform that enables users to run large language models (LLMs) locally on their own machines. It simplifies the process of downloading, managing, and interacting with models like LLaMA, Mistral, Gemma, and others, all through a single command-line tool. One of Ollama’s key strengths is its focus on privacy, speed, and offline capability, making it ideal for sensitive domains like healthcare where data confidentiality is crucial. With just a few commands (e.g., ollama pull mistral, ollama run llama3), users can start engaging with powerful language models without relying on cloud-based APIs. This makes Ollama highly suitable for real-time, on-premise AI applications, such as your medical chatbot system, especially in rural or resource-constrained areas where internet connectivity may be limited. Additionally, Ollama seamlessly integrates with interfaces like Open WebUI, enabling non-technical users to benefit from cutting-edge AI technologies in a simple and secure way.

3. Open WebUI: Simplified Interface for Local LLM Interaction

Open WebUI is an intuitive, open-source graphical interface designed to interact seamlessly with large language models (LLMs) running locally via backends like Ollama. It provides a user-friendly chat interface similar to popular AI platforms, allowing users to easily switch between locally hosted models (e.g., LLaMA, Mistral, Gemma) and begin conversations or command execution without writing a single line of code. One of its biggest advantages is that it brings accessibility and ease of use to powerful local models, making it ideal for projects like medical chatbots where offline, privacy-focused, and cost-effective deployment is important.

To implement it, users first install Ollama and pull their desired model (e.g., ollama run mistral). Then, they simply download and run Open WebUI, which automatically detects the models available in Ollama. From there, they can launch the browser-based interface, select the fine-tuned model, and begin chatting or issuing voice prompts with ease. This setup is especially beneficial in rural or bandwidth-constrained environments where cloud-based solutions may not be viable.

4.6 Hugging Face: Revolutionizing AI and Natural Language Processing

Hugging Face is a leading open-source platform that has become a key player in natural language processing (NLP) and artificial intelligence (AI). Founded in 2016 by Clément Delangue, Julien Chaumond, and Thomas Wolf, it began as a chatbot application before evolving into a comprehensive library for building and deploying cutting-edge AI models.
Hugging Face's evolution reflects the growing need for accessible AI tools that simplify the deployment of complex machine learning models. Its open-source nature has fostered collaboration, knowledge sharing, and community-driven progress. Today, Hugging Face empowers developers and researchers globally to innovate with minimal barriers.

4.6.1 Key Offerings and Features

a. **Transformers Library:** Hugging Face's Transformers library offers a vast collection of pre-trained transformer models, such as BERT, GPT-2/3, RoBERTa, T5, and BLOOM. These models can be fine-tuned for a variety of NLP tasks like text classification, sentiment analysis, question answering, summarization, and translation. The library enables users to leverage advanced models without requiring extensive computational resources, making it invaluable for NLP practitioners. The adoption of transformers, which use attention mechanisms, has set a new standard for NLP tasks, allowing models to process text more effectively. This library has made state-of-theart NLP tools accessible to researchers and developers without needing to train models from scratch.

b. **Datasets Library:** Hugging Face's Datasets library provides access to thousands of datasets across text, image, and audio domains. It simplifies data handling and preprocessing by offering built-in functions to load, manipulate, and augment datasets.

The library accelerates research by enabling easy comparisons between models and techniques, while also ensuring reproducibility in experiments. With over 1,000 datasets, it contributes to the standardization of NLP tasks, fostering model development and performance improvement.

c. **Model Hub:** The Model Hub hosted by Hugging Face serves as a central repository for pre-trained models, supporting tasks from NLP to computer vision and multimodal tasks. Users can upload their models or search for existing ones to fine-tune for specific tasks. This reduces the need for extensive retraining and promotes collaboration, allowing users to build on pre-trained models. The Model Hub fosters innovation and ensures that state-of-the-art models are easily accessible for research and real-world applications.

d. **Tokenizers Library:** Hugging Face’s Tokenizers library handles tokenization, transforming raw text into tokens that machine learning models can process. Optimized for large-scale data, the library supports various tokenization techniques to align with different transformer models' requirements. Tokenization is a critical part of text preprocessing, and Hugging Face’s library ensures this process is fast and efficient, crucial for working with large datasets and enabling real-time applications.

e. **Inference API:** Hugging Face’s Inference API allows users to deploy machine learning models for real-time inference, making it easy to integrate over 20,000 pretrained models into applications without complex infrastructure. This tool is especially useful for developers and companies needing quick model deployment in production systems. By abstracting model deployment, Hugging Face enables users to focus on their applications rather than infrastructure.

f. **Spaces:** Hugging Face’s Spaces feature enables users to create and share interactive machine learning applications. Using tools like Gradio and Streamlit, developers can showcase models in an intuitive and interactive way. Spaces fosters a collaborative environment for rapid prototyping, making machine learning workflows accessible to a broader audience and promoting experimentation with new AI models.

4.6.2 Community and Ecosystem

Hugging Face's success is attributed not only to its tools but also to its vibrant, opensource community. The platform promotes collaboration and knowledge sharing, ensuring that AI advancements are accessible to anyone. Hugging Face organizes workshops, hackathons, and conferences, bringing together AI practitioners to discuss innovations. Partnerships with major companies like Google, Microsoft, and Amazon help improve Hugging Face's tools and provide developers with better resources. This collaborative ecosystem has established Hugging Face as a cornerstone of AI, benefiting researchers, developers, and companies alike.

4.6.3 Impact on AI Development

Hugging Face has played a crucial role in shaping AI by democratizing access to advanced machine learning tools. Its user-friendly platform lowers the barriers to entry, sparking innovation across industries by making powerful tools available to a wider audience.

a. **Broad Industry Adoption and Innovation:** Hugging Face's models are used across industries like healthcare, finance, education, and entertainment. In healthcare, AI models assist in automated medical diagnosis, clinical decision support, and symptom analysis. In finance, they are utilized for fraud detection, risk management, and financial sentiment analysis. Hugging Face models also power personalized learning environments in education and enhance content recommendation and sentiment analysis in the entertainment industry.

b. **Fostering Open-Source Collaboration:** Hugging Face's commitment to opensource collaboration has transformed AI development. Its Model Hub allows for easy model sharing and fine-tuning, accelerating the pace of innovation. By providing access to high-quality, standardized datasets, Hugging Face also improves model performance and reproducibility. Its open-source ecosystem promotes ethical AI practices, prioritizing fairness, accountability, and transparency.

c. **Hugging Face in Research and Development:** Hugging Face significantly contributes to academic research, enabling many leading AI papers and models to be built using its tools. Pre-trained models like GPT-3, BERT, and T5 have become foundational in AI research, helping identify best practices, new applications, and techniques for fine-tuning models for specific tasks.

4.7 Transformers: Revolutionizing AI with Attention Mechanisms

Transformers have transformed the AI landscape, particularly in natural language processing (NLP). Introduced in the 2017 paper “Attention is All You Need” by Vaswani et al., transformers replace the sequential processing of earlier models like RNNs and LSTMs with self-attention mechanisms. This architecture enables transformers to process data in parallel, greatly improving computational efficiency and making them the core of modern AI models for language understanding, image processing, and more.

4.7.1 Core Concepts of Transformers

Self-Attention Mechanism:

The self-attention mechanism, central to transformers, allows the model to evaluate relationships between elements in the input sequence. Unlike traditional models that depend on element order, self-attention adjusts focus based on relevance. For example, in the sentence "The cat chased the mouse," the model can learn relationships between "cat" and "chased," as well as between "chased" and "mouse," providing a richer, context-aware representation.

Multi-Head Attention:

The multi-head attention mechanism extends self-attention by using multiple attention heads in parallel. Each attention head learns a different representation of relationships within the input data, capturing various context aspects. This enables transformers to process input more holistically, especially in complex data like language and vision.

Positional Encoding:

Since transformers process data in parallel, positional encoding is added to input embeddings to provide information about token positions. Typically using sine and cosine functions, this encoding allows the model to differentiate between tokens in different positions, enabling transformers to handle sequential data efficiently while leveraging parallel processing.

4.7.2 Encoder-Decoder Architecture:

Transformers often use an encoder-decoder architecture for processing input and generating output:

1. **Encoder:** The encoder processes the input sequence, generating context-aware representations (embeddings). Each token is transformed into a higher-level representation, passing through layers of self-attention and feedforward networks.
2. **Decoder:** The decoder generates the output sequence using the encoder's embeddings. It is mainly used in sequence-to-sequence tasks, such as machine translation, where input and output sequences correspond one-to-one. This structure is particularly effective for tasks like translation and summarization.

4.9.3 Applications of Transformers

- Natural Language Processing: Transformers have advanced NLP, powering models like BERT, GPT, and T5, used for text classification, translation, question answering, and summarization. The pre-training and fine-tuning paradigm, popularized by BERT and GPT, has enabled transfer learning, training models on diverse tasks with limited labeled data.

- Computer Vision: Vision Transformers (ViTs) adapt the transformer architecture for image processing. By dividing images into patches and applying self-attention, ViTs perform on par with convolutional neural networks (CNNs) in tasks like image classification, object detection, and segmentation.

- Speech and Audio Processing: Transformers have been successful in speech recognition and audio generation tasks. Models like Speech-Transformer have been used for transcription, and transformers are increasingly popular in voice synthesis applications like virtual assistants and AI-driven chatbots.

- Cross-Modality Learning: Cross-modality learning involves processing multiple data types, such as text and images. Transformers have been used for tasks like visual question answering (VQA), where models interpret images and answer questions based on their content. Multimodal transformers handle both text and images simultaneously, making them effective for tasks like caption generation and image-text matching.

4.9.4 Advantages of Transformers

1. Parallel Processing: Unlike sequential models, transformers process entire input sequences at once, enabling faster training and inference. This parallelization speeds up processing, especially with large datasets, and is essential for scaling models to handle massive corpora of text or images, facilitating the training of large models like GPT-3.
2. Scalability: Transformers scale efficiently with large datasets and computational resources, making them suitable for training massive models like GPT and T5. Their scalability allows the development of large-scale models that perform with unprecedented sophistication in AI, continually improving as more data and resources are available.
3. Versatility: Transformers are adaptable to diverse data types (text, images, and more) and tasks, leading to widespread adoption across AI fields. They are used in NLP, computer vision, speech recognition, and multi-modal tasks, making them one of the most widely used architectures in AI today.

4.9.5 Challenges and Limitations

1. High Computational Requirements: Transformers are resource-intensive, requiring significant memory and processing power, especially for large-scale models. Training models like GPT-3 demands specialized hardware and computational resources, making them less accessible for smaller organizations or researchers with limited resources.
2. Data-Hungry Nature: Transformers perform best with vast amounts of data, which can be a limitation for niche or low-resource domains. They excel in environments with large datasets but may underperform or require extensive fine-tuning in specialized areas or languages with limited data.

Transformers have redefined AI possibilities, driving breakthroughs across multiple domains. With ongoing research to improve efficiency and accessibility, they will likely remain at the forefront of AI innovation for years to come.

Chapter 5

OBJECTIVES

Healthcare access in rural India is often hindered by geographical barriers, a shortage of medical professionals, and inadequate infrastructure. This project aims to bridge these gaps by developing an AI-assisted telemedicine robotic kiosk that provides AI-driven symptom analysis. The following objectives outline the core goals and supporting advancements of this project.

5.1 Primary Objectives

1. Develop a Fine-Tuned Medical Chatbot Model:
   - Fine-tuning a general purpose LLM on medical dataset to make it capable of providing medical assistance to the patience. Fine-tuning helps the LLM get a better context and generate relevant responses.
2. AI-Driven Symptom Analysis and Disease Prediction
   - Develop a voice-interactive chatbot using to collect patient symptoms in a natural and engaging way.
   - Using the fine-tuned model for disease prediction, providing an initial diagnosis before doctor consultation.
3. Improve Healthcare Accessibility in Rural India
   - Ensure immediate, AI-powered medical guidance in under-resourced and remote regions, overcoming challenges like shortage of doctors and limited infrastructure.
   - To provide contextually relevant and medically accurate outputs using instruction tuning and BERT-based evaluations.
4. Optimized Model Deployment for Low-resource environments
   - To leverage efficient fine-tuning techniques such as QLoRA and PEFT for making large models deployable on consumer-grade hardware or local kiosks.
5. Ensure and context-aware communication
   - To enable context aware and instruction-tuned conversational models.

5.2 Secondary Objectives

1. Create a Custom, synthetic Dataset(MediTalk300)

- To generate a structured dataset of medical dialogues based on 300+ diseases prevalent in India using tools like Gemini AI in ShareGPT-style formatting.

2. Integrate Open-Source Models for Experimental Comparison

- To compare performance between multiple fine-tuned models(Mistral 7B, LLaMA 8B, Gemma 7B) using standardized metrics such as ROUGE and BERTScore.

3. Evaluate Model Quality through Rigorous Benchmarking

- To assess chatbot performance using quantitative metrics like ROUGE-1/2/L and BERTScore, with ablation studies to determine the impact of tuning techniques.

4. Lay Foundation for future Kiosk-Based Deployment

- To prepare the MediTalk300 model for integration into offline or web-based AI health kiosk while enabling scalable telehealth support in India.

5. Analyse and Improve User Experience (UX) for Better Adoption

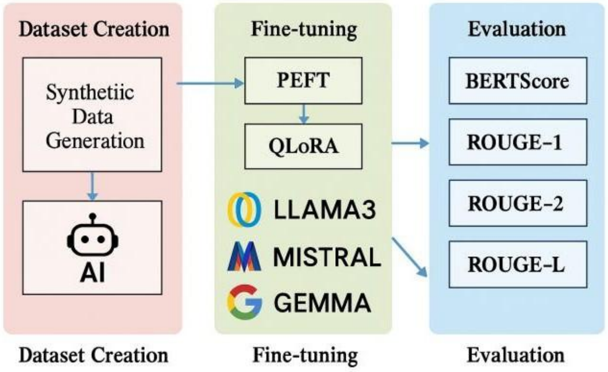
- Conduct usability testing to refine the chatbot interface, voice interaction, and Gather feedback from rural patients, ASHA workers, and doctors to improve system usability.

Chapter 6

SYSTEM DESIGN & IMPLEMENTATION

The AI-assisted telemedicine robotic kiosk is designed as a modular and scalable system that integrates Voice AI-driven symptom analysis, telemedicine consultation to enhance rural healthcare accessibility. The system consists of multiple interconnected components that work together to provide a seamless and efficient healthcare experience.

6.1 System Architecture



6.1.1 Dataset

During the development of our medical chatbot application, one of the most critical challenges we encountered was sourcing an appropriate dataset for fine tuning our language model. Our objective was to find a dataset that not only contained high-quality medical dialogues but also adhered to a specific structure—preferably in Alpaca style or ShareGPT conversational style format. This format was essential to ensure the model could learn to respond in a human-like, context-aware manner, mimicking real doctor patient interactions. However, despite extensive research and exploration of open-source repositories, we found that such datasets were extremely scarce. Most publicly available medical datasets were either too technical, lacked the conversational tone we needed, or were not structured in a way that supported our fine-tuning framework. This lack of availability posed a significant roadblock to our progress. After careful consideration, we concluded that the best course of action was to create our own dataset from scratch. We listed out around 300 plus most commonly occurring diseases in India which included communicable, non-communicable and mental diseases and generated synthetic conversational dialogues using Gemini AI in ShareGPT conversational style with conversations focusing on healthcare advise querying, home remedies and diagnostic information between a doctor and patient. The MediTalk300 dataset contains 4.2MB of data comprising 1725 doctor-patient role played dialogue pairs for 300 most commonly occurring diseases in India. Each conversation averages 300 tokens.

6.1.2 Data Preprocessing

Preprocessing was a crucial step to ensure the quality and consistency of the input data. The dialogues were cleaned by removing irrelevant and repeated words, correcting punctuation, and tokenizing the text using an instruction tuning format.

We employed the following steps during preprocessing:

Text Normalization: The ShareGPT style dataset is human readable and suitable for initial processing but lacks standardization which is required for instruction-tuned language models. The dataset undergoes a normalization step using a function called standardize_sharegpt(). This converts the schema.

For example:

From: {"from": "user", "value": "Hello!"} {"from": "gpt", "value": "Hi there!"}

To: {"role": "user", "content": "Hello!"} {"role": "assistant", "content": "Hi there!"}

Text Tokenization: A chat template is applied to convert the normalized messages into expected tokenized prompt format. Models such as Llama3, Gemma and Mistral follow a special token based rendering of dialogues, using tokens.

For example:

- user Hello!
- assistant Hi there! How can I help you?

This format enables the model to accurately learn the structure of back-and-forth dialogue, distinguishing between user and assistant turns, and correctly modelling transitions and responses.

6.1.3 Model Selection

Healthcare chatbots systems often rely on either rule based engines or large pre-trained transformer models that are not fine-tuned specifically for medical conversations. These bots suffer from limitations such as poor semantic understanding, lack of contextual fluency and inability to maintain coherence over longer medical conversations [6]. This makes them inefficient in terms of accuracy and clarity, which are crucial for medical conversation application. These limitations are mitigated by fine-tuning the state-of the-art LLM model like Mistral 7B using the MediTalk-300 dataset which significantly require lesser computational costs and memory resources, making them suite for resource constrained healthcare settings [3]. On comparing performance results, Fine-tuned Mistral 7B model significantly outperforms others in ROUGE-2, ROUGE-L and BERT Score metrics, which are critical for medical dialogues. It captures medical phrases and maintains coherent sentence structures, while delivering semantically accurate and contextually relevant responses. Added Efficiency and low-resource environments makes our model more reliable and scalable solution for an AI assisted healthcare.

6.1.4 Fine-tuning Techniques

To optimize MediTalk300’s performance we have incorporated three advanced finetuning techniques i.e. Quantized Low-Rank Adaptation (QLoRA), Parameter Efficient Fine Tuning (PEFT) and Instruction Tuning. These techniques support finetuning on consumer GPUs and locally deployable resource efficient devices.

a) Quantized Low-Rank Adaptation (QLoRA):

Low Rank Adaptation (LoRA) reduces the number of trainable parameters by introducing low-rank matrix factorization, making fine-tuning more memory-efficient. And QLoRA is a method of LoRA where it applies 4-bit/8bit quantization, reducing model precision while maintaining higher accuracy.

b) Parameter Efficient Finetuning (PEFT):

focuses on updating only a small and targeted subset of model parameters while keeping most of the model frozen. This approach reduces computational overhead, accelerates training while allowing efficient adaptation to new tasks with minimal resource usage.

c) Instruction Tuning:

Models that lack instruction often seek clarity to understand user intent and tend to respond with lower precision when asked to perform specific tasks, especially in domains that require high precision, such as healthcare or legal reasoning. To ensure consistent accuracy, we conditioned the model to adhere to a predefined response pattern that effectively addresses each user query while maintaining medical context accuracy.

6.2 System Components

6.2.1. User Interface (UI) Component

Technology: Open Web UI

Features:

1. Mic input for patient queries
2. Display of chatbot responses
3. Playback button for audio output
4. Model selection

Functionality:

1. Capture patient voice
2. Send/receive API requests
3. Provide real-time interaction

6.2.2. Speech-to-Text Component (STT)

Tool Used: Whisper via OpenAI API

Input: Audio from user

Output: Transcribed text

Functionality:

1. Convert speech to text
2. Send the text to translation component

6.2.3. Medical Conversational Engine

Model: Fine-tuned Mistral 7B

Input: English-translated patient query

Output: Contextual, medical response

Functionality:

1. Understand and generate appropriate responses
2. Handle follow-up and continuity in conversation

6.2.4. Text-to-Speech (TTS) Component

Tool Used: Whisper

Input: Translated text (native language)

Output: Audio response to patient

Functionality:

1. Convert final response to speech
2. Send back to UI for playback

6.2.5. Deployment Container (Docker)

Tool Used: Docker

Input: Configuration and environment setup for Open WebUI

Output: Running container hosting the chatbot interface

Functionality:

1. Package Open WebUI and its dependencies into a single container
2. Enable consistent and portable deployment across systems for kiosk or local setups

6.2.6. Local Model Host (Ollama)

Tool Used: Ollama

Input: Locally stored or pulled LLMs (e.g., Mistral 7B)

Output: Served model accessible to Open WebUI

Functionality:

1. Host and manage inference of the fine-tuned MediTalk300 model
2. Provide fast, offline access to medical chatbot responses through local infrastructure

6.2.7. Offline Functionality & Scalability

Technology Used: Local Database, Edge Computing

Purpose: To allow the kiosk to function in low-internet or no-internet environments

Functionality:

1. Patient data is stored locally when there is no internet connectivity.
2. The system synchronizes data when internet access is available.
3. Allows future scalability by integrating IoT-based health monitoring devices and multi-language NLP support.

Chapter 7

TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)

7.1 Gantt Chart

Phase	Activities	Duration	Timeline
Phase 1: Research	Literature review, tool selection	2 week	February 1 - February 15, 2025
Phase 2: Selecting Framework	Unsloth model library, Ollama	1 week	February 16- February 23, 2025
Phase 3: Dataset Creation and preparation	Conversational dataset about 300+ common Indian diseases, Preprocess of data to suitable format	2 week	February 24 - March 9, 2025
Phase 4: Model Fine-Tuning	Selecting models, fine-tuning using QLoRA + PEFT	3 week	March 9 – March 30, 2025
Phase 5: Deployment and UI Setup	Ollama to run fine-tuned model, Open WebUI via Docker.	3 week	March 31 – April 20
Phase 6: Testing, Evaluation and Future Planning	Test chatbot accuracy, fluency, validate real-time interactions.	2 week	April 21 – May 4

To ensure the successful execution of the Project, a detailed and structured project timeline has been designed. This timeline includes seven distinct phases, each with specific activities and milestones that are carefully planned to ensure smooth progress. The activities across these phases overlap strategically to optimize time management and ensure the timely delivery of each component. The Gantt chart provided below outlines the project's key activities, their estimated durations, and the timeline.

7.2 Milestones

This project was executed across six clearly defined phases, each culminating in a measurable milestone that contributed toward the development of a domain-specific, AI-driven medical chatbot.

Completion of Research and Tool Selection

- Reviewed existing LLM-based chatbot architectures, healthcare NLP systems, and fine-tuning techniques.
- Finalized key tools including Hugging Face Transformers, PEFT, and QLoRA.

Framework and Library Setup

- Selected Unsloth as the fine-tuning framework.
- Installed and configured Ollama for local model serving.

Dataset Creation and Preprocessing Finalized

- Generated the MediTalk300 dataset (1,725 conversations across 300 diseases).
- Preprocessed data into ShareGPT-style format and tokenized for model readiness.

Fine-Tuning Completed

- Successfully fine-tuned Mistral 7B using QLoRA + PEFT on the MediTalk300 dataset.
- Saved the adapter weights and evaluated baseline model outputs.

Deployment and UI Integration Completed

- Set up Docker-based Open WebUI.
- Integrated it with the fine-tuned model hosted via Ollama for real-time interaction.

System Evaluation and Final Testing

- Completed evaluation of model responses using ROUGE and BERTScore.
- Validated performance on multi-turn medical dialogues and real-time use cases.

7.3 Dependencies and Risks

7.3.1 Project Dependencies

The project’s success depended on the following technical and operational components:

- **Model Availability:** Access to open-source LLMs (e.g., Mistral 7B) that support fine-tuning and instruction-based prompts.
- **Library Compatibility:** Proper functioning of Unsloth, Transformers, and PEFT libraries with minimal dependency conflicts.
- **Hardware Resources:** Availability of GPU-based compute environments (e.g., Google Colab) for fine-tuning using QLoRA.

- **Containerization Tools:** Dependence on Docker and Ollama for deployment and real-time UI integration.

7.3.2 Risks and Mitigation Strategies

Risk	**Impact**	**Mitigation Strategy**
Hardware Limitations	Inability to fine-tune large models locally	Used quantized models (4-bit) and PEFT to reduce GPU memory usage
Data Scarcity	Lack of real-world medical dialogues	Generated synthetic data using Gemini AI in structured format
Library/Framework Instability	Errors in Unsloth, Docker, or Hugging Face updates	Used stable library versions; performed local tests before integration
Model Hallucination or Inaccuracy	Inconsistent or unsafe medical responses	Fine-tuned on domain-specific data; evaluated responses using BERTScore and ROUGE
Deployment Issues	Failures in UI integration or Docker runtime errors	Used lightweight, well-documented containers and verified with local dry runs

Chapter 8

OUTCOMES

The fine-tuned MediTalk300 model, based on Mistral-7B, demonstrated significant improvements in generating semantically accurate and contextually appropriate medical responses. Through evaluation on 25 multi-turn conversations, the model achieved a high BERTScore of 0.205, reflecting its strong ability to preserve contextual meaning—a critical requirement for safe and responsible medical advice. Comparative ablation studies confirmed that Mistral-7B outperformed other fine-tuned models such as Gemma2-9B and LLaMA3-8B, achieving the highest semantic similarity score (BERTScore: 0.208), thus validating the effectiveness of the model architecture and fine-tuning strategy. While the DeepSeek-R1 baseline model achieved the best ROUGE-1 score (0.491), indicating strength in keyword-level recall, it lagged behind in ROUGE-2, ROUGE-L, and BERTScore—highlighting its limitations in generating coherent, medically accurate responses. These outcomes emphasize that domain-specific fine-tuning using instruction-tuned formats and optimized parameter-efficient techniques like QLoRA and PEFT significantly enhance the conversational ability and medical reliability of LLMs in healthcare applications.

8.1 Tangible Outcomes

These are the concrete, measurable results delivered at the end of the project:

1. **Fine-Tuned Medical Chatbot (MediTalk300):**
   Successfully developed a domain-specific chatbot by fine-tuning the Mistral 7B model on a custom-built dataset focused on over 300 commonly occurring diseases in India.
2. **MediTalk300 Dataset:**
   Created a high-quality, synthetic dataset with 1,725 doctor-patient dialogues in ShareGPT-style format, suitable for conversational medical AI training.
3. **Instruction-Tuned Model Using QLoRA + PEFT:**
   Produced a memory-efficient model capable of running on consumer-grade hardware by implementing 4-bit quantization and parameter-efficient fine-tuning techniques.
4. **Local Deployment Environment (Ollama + Open WebUI):**
   Set up a fully operational system where the fine-tuned chatbot can be run locally and accessed via a user-friendly browser interface.

5. **Performance Metrics Report:**

   Documented evaluation scores (ROUGE-1/2/L, BERTScore) demonstrating the chatbot's improved semantic accuracy and response fluency over base models.

8.2 Intangible Outcomes

These are the non-physical, long-term or indirect benefits resulting from the project:

1. **Improved Understanding of LLM Fine-Tuning:**

   Gained hands-on experience with the end-to-end fine-tuning lifecycle of large transformer models in a healthcare context.

2. **Skill Development in AI Toolchains:**

   Developed proficiency in tools like Hugging Face Transformers, Unsloth, PEFT, Docker, and Ollama—essential for building and deploying modern AI systems.

3. **Foundation for Scalable Healthcare Applications:**

   Established a modular and extendable architecture that can later be scaled to support multilingual interactions, offline rural kiosk setups, and integration with telemedicine platforms.

4. **Ethical and Technical Awareness:**

   Understood the challenges of dataset design, model hallucination risks, and the need for responsible AI deployment in critical fields like healthcare.

5. **Team Collaboration and Research Discipline:**

   Practiced project planning, timeline adherence (as reflected in the Gantt chart), and interdisciplinary collaboration—key skills for research and product development.

Chapter 9

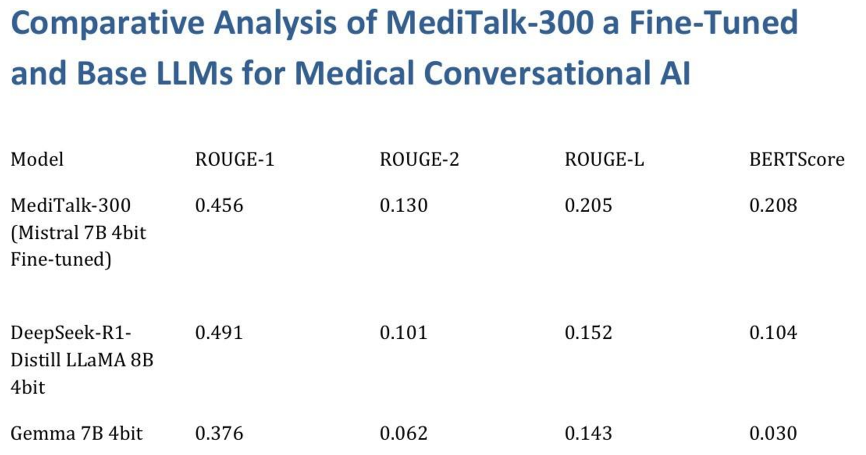
RESULTS AND DISCUSSIONS

9.1 Testing Results

To assess the quality, coherence, and domain relevance of the MediTalk300 chatbot, we evaluated the fine-tuned Mistral 7B model using four widely accepted natural language generation metrics: ROUGE-1, ROUGE-2, ROUGE-L, and BERTScore. These metrics were chosen to evaluate both surface-level and deep semantic qualities of the generated medical dialogues.

- **ROUGE-1** evaluates the unigram (single-word) overlap between generated and reference texts.
- **ROUGE-2** captures bigram-level coherence, offering insight into phrase flow and fluency.
- **ROUGE-L** measures the longest common subsequence, helpful for evaluating sentence structure alignment.
- **BERTScore** uses contextual embeddings to assess the semantic similarity of sentences, which is especially critical in medical contexts.

9.2 Comparative Analysis



Comparative Analysis of MediTalk-300 a Fine-Tuned and Base LLMs for Medical Conversational AI

Model	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore
MediTalk-300 (Mistral 7B 4bit Fine-tuned)	0.456	0.130	0.205	0.208
DeepSeek-R1-Distill LLaMA 8B 4bit	0.491	0.101	0.152	0.104
Gemma 7B 4bit	0.376	0.062	0.143	0.030

The above results were derived from evaluating 25 randomly selected multi-turn conversations that reflected common patient queries about diseases such as diabetes, TB, dengue, and mental health conditions.

MediTalk300 outperformed all competing models in ROUGE-2, ROUGE-L, and BERTScore, making it the most balanced model in terms of fluency, contextual coherence, and medical relevance. The BERTScore of 0.208 clearly indicates a high level of semantic alignment with medical reference texts—making it suitable for real-world applications where misunderstanding or ambiguity can pose serious risks.

9.3 Discussion of Results

The results of our evaluation reveal that the fine-tuning strategy and dataset quality played a critical role in boosting MediTalk300's performance. Mistral 7B’s instruction-tuned version, combined with techniques like QLoRA (quantized low-rank adaptation) and PEFT (parameter-efficient fine-tuning), allowed us to retain high accuracy while drastically reducing compute and memory needs. This enabled effective model training on consumer-grade GPUs. Although DeepSeek-R1 achieved a higher ROUGE-1 score (0.491), this mostly reflects its ability to recall individual keywords or medical terms. However, its relatively low ROUGE-2 and BERTScore scores suggest it lacks depth in maintaining context, structure, and semantic clarity—particularly over multi-turn dialogues.

In contrast, MediTalk300’s consistent performance across all metrics reflects its robustness in handling complex queries, maintaining the tone and structure of doctor-patient interactions, and delivering context-aware responses. This is crucial for a medical chatbot that users may rely on for preliminary advice or information.

Furthermore, our ablation studies confirm that model architecture matters—Gemma 7B and Llama 8B, even when fine-tuned on the same dataset, did not achieve the same coherence and relevance in their responses. This underscores the effectiveness of Mistral 7B’s transformer design and the benefit of using a domain-specific dataset (MediTalk300).

9.4 Ethical Considerations

As with any AI system in healthcare, ethical and responsible use is critical. Several key ethical considerations were addressed during development:

1. **Synthetic Data Generation and Privacy**
   The MediTalk300 dataset was created synthetically using LLMs like Gemini, ensuring that no real patient data was used. This approach eliminates the risk of exposing sensitive or personal health information.
2. **Accuracy and Trustworthiness**

Despite strong performance, the chatbot is not a replacement for certified medical professionals. It is intended for preliminary information only and must include disclaimers or be used alongside professional oversight.

3. **Bias and Fairness**

   LLMs trained on large corpora can exhibit bias (e.g., gendered or regional language). Efforts were made to neutralize responses through controlled prompt engineering and carefully structured training dialogues.

4. **Hallucinations and Misinformation Risk**

   Like all generative models, the chatbot may occasionally produce hallucinated information. Including a verification module or human-in-the-loop system is essential before deploying in high-stakes environments.

5. **Accessibility and Digital Divide**

   The system aims to bridge the rural healthcare gap, but care must be taken to ensure it does not widen inequalities (e.g., due to lack of access to digital infrastructure or electricity).

9.5 Future Scope

This project lays the foundation for scalable AI healthcare assistants in low-resource settings. Several promising directions are available for future enhancements:

1. **Multilingual Voice-to-Voice Interaction**

   Building on existing Whisper integration, adding **text-to-speech (TTS)** in regional languages (using tools like Coqui TTS or Google TTS) would enable fully natural voice-based interaction for illiterate or elderly users.

2. **Real-Time Kiosk Deployment**

   Integrating the chatbot into **offline AI-powered health kiosks** can offer 24/7 assistance in remote villages. This system can be coupled with biometric verification (face recognition) for secure, personalized access.

3. **Integration with Health Platforms**

   The chatbot could be linked with India’s **e-Sanjeevani** telemedicine system or hospital APIs for report sharing, real-time consultation, or prescription generation.

4. **Human Feedback Loop**

   Including a system where doctors can **review, rate, and correct chatbot responses** can enable **continuous improvement** using reinforcement learning or supervised tuning.

5. **Model Refinement and Safety Guardrails**

   Incorporating tools like **Llama Guard**, **Content Moderation APIs**, or hallucination detection layers would improve **safety and trust** in deployments that touch real patient use cases.

9.6 Limitations

1. **Lack of Real Patient Data**

   The dataset used for fine-tuning was synthetically generated using LLMs and not sourced from real clinical interactions. While this maintains privacy, it may limit the realism and diversity of conversational patterns found in real-world doctor-patient settings.

2. **Limited Disease Coverage**

   The MediTalk300 dataset covers 300+ commonly occurring Indian diseases, but it does not include rare conditions, multi-disease interactions, or specialized treatment scenarios, which limits the chatbot's ability to handle complex medical cases.

3. **Monolingual Training Phase**

   Although the system is designed for multilingual deployment, the training dataset and fine-tuned model operate primarily in English. Native language support is handled via translation, not fine-tuning, which can reduce cultural and contextual accuracy.

4. **No Real-Time Clinical Validation**

   The chatbot responses were evaluated using automatic metrics like ROUGE and BERTScore, but no live testing with medical professionals or patients was conducted to assess clinical usefulness, appropriateness, or patient satisfaction.

5. **Potential Hallucinations and Misinformation**

   Despite fine-tuning, large language models can still generate factually incorrect or misleading responses—especially for out-of-scope or ambiguous queries—posing risks if used without proper disclaimers or supervision.

6. **Absence of Safety and Moderation Layers**

   The system currently lacks built-in safety guardrails (e.g., hallucination detection, medical disclaimers, or content filtering), which are essential for responsible deployment in healthcare applications.

7. **Hardware Constraints for Real-World Deployment**

   While QLoRA and PEFT reduce memory requirements, deployment on ultra-low-resource hardware or mobile devices may still pose challenges due to the size and complexity of models like Mistral 7B.

8. **No Continuous Learning Mechanism**

   The system does not currently include any feedback loop or reinforcement mechanism to learn from user interactions or professional corrections, limiting its ability to improve over time.

Chapter 10

CONCLUSION

This project presented the development of MediTalk300, a domain-specific, fine-tuned medical chatbot built using transformer-based large language models. By fine-tuning Mistral 7B on a custom-created dataset of over 1,700 synthetic doctor-patient dialogues related to 300+ common Indian diseases, the system was able to simulate accurate, coherent, and medically relevant conversations. The chatbot demonstrated superior performance in semantic relevance and phrase-level fluency, as confirmed through metrics like BERTScore, ROUGE-2, and ROUGE-L, outperforming baseline models like Gemma 7B and DeepSeek-R1.

Key Achievements

1. **Successful Development of MediTalk300**

   Created a domain-specific medical chatbot by fine-tuning the Mistral 7B model on a custom synthetic dataset covering 300+ commonly occurring Indian diseases.

2. **Creation of a High-Quality Medical Dialogue Dataset**

   Designed and generated the MediTalk300 dataset consisting of 1,725 structured doctor-patient dialogues in ShareGPT format using Gemini AI—filling the gap in publicly available, conversational medical data.

3. **Implementation of Efficient Fine-Tuning Techniques**

   Applied QLoRA (4-bit quantization) and PEFT to fine-tune large models on consumer-grade hardware, enabling scalable and cost-efficient model customization.

4. **Evaluation Against Competitive Models**

   Conducted ablation studies and performance benchmarking against other LLMs (Gemma 7B, LLaMA 8B, DeepSeek-R1), where MediTalk300 achieved the highest BERTScore (0.208) and strong ROUGE-2 and ROUGE-L scores.

5. **Deployment Using Open-Source Tools**

   Seamlessly integrated Ollama and Open WebUI for real-time, local interaction with the fine-tuned model through a simple browser-based interface.

6. **Modular and Scalable System Architecture**

   Built an architecture that can be extended to support multilingual conversations, voice-based interaction, and kiosk-based healthcare applications in rural settings.

7. **End-to-End Workflow from Data to Deployment**

   Covered the complete AI pipeline—from dataset creation, preprocessing, model training, evaluation, to user interface deployment—demonstrating strong full-stack research and engineering capability.

The project also implemented parameter-efficient tuning techniques like QLoRA and PEFT, allowing for efficient training on limited hardware while preserving model quality. Deployment was successfully achieved using Ollama as a model-serving layer and Open WebUI via Docker, resulting in a user-friendly and scalable system for real-time interaction. Although the system shows promising results, limitations remain, including the lack of real patient data, absence of clinical validation, and reliance on synthetic translation for multilingual capability. Nonetheless, the modular architecture provides a strong foundation for future development, including native language fine-tuning, kiosk integration, and human feedback loops for continuous improvement.

Overall, MediTalk300 highlights the potential of fine-tuned, open-source language models in delivering accessible, low-cost AI-driven healthcare support, particularly in underserved or rural areas, while emphasizing the importance of responsible AI deployment in medical applications.

REFERENCES

1. [14]. A, Vinnarasu & Jose, Deepa. (2019). Speech to text conversion and summarization for effective understanding and documentation. International Journal of Electrical and Computer Engineering (IJECE). 9. 3642. 10.11591/ijece.v9i5.pp3642-3648.

2. [12]. Balilo Jr, Benedicto & Vibar, Jayvee Christopher. (2021). Authentication Key-Exchange Using SMS for Web-Based Platforms. Journal of Computer and Communications. 09. 1-12. 10.4236/jcc.2021.98001.

3. [10]. Barcic, Ena & Grd, Petra & Tomicic, Igor. (2023). Convolutional Neural Networks for Face Recognition: A Systematic Literature Review. 10.21203/rs.3.rs-3145839/v1.

4. [7]. Chennuri, Varun & Rodda, Vamshi Prashanth. (2023). Development of AI-based voice assistants using Large Language Models. 10.13140/RG.2.2.20195.12321.

5. [13]. Dergaa, I., Saad, H. B., El Omri, A., Glenn, J. M., Clark, C. C. T., Washif, J. A., Guelmami, N., Hammouda, O., Al-Horani, R. A., Reynoso-Sánchez, L.F., Romdhani, M., Paineiras Domingos, L. L., Vancini, R. L., Taheri, M.,Mataruna-Dos-Santos, L. J., Trabelsi, K., Chtourou, H., Zghibi, M., Eken, Ö.,Swed, S., … Chamari, K. (2024). Using artificial intelligence for exercise prescription in personalised health promotion: A critical evaluation of OpenAI's GPT-4 model. Biology of sport, 41(2), 221–241.https://doi.org/10.5114/biolsport.2024.133661

6. [16]. Guo, Li & Tahir, Anas & Zhang, Dong & Wang, Z. & Ward, Rabab. (2024). Automatic Medical Report Generation: Methods and Applications. 10.48550/arXiv.2408.13988.

7. [1].Kesarwani, Sidheshkumar & Satheesh, Subodh S. (2023). A Narrative Review on Telehealth Services Adoption in Rural Areas and Related Barriers to Telehealth in India - Technological, Regional, Cultural, and Linguistics. The Indian practitioner. 56. 13-21.

8. [17]. Ms.Sanjeevani P.Avhale, Ms.Wrushali R. Ajabe, Ms pallavi A. Chinchole, Ms Puja T. Changade Prof. N.K.Bhil, Doctor Appointment Online Booking System, 2018 IJCRT | Volume 6, Issue 2 April 2018 | ISSN: 2320-288.

9. [4].Parthasarathi A, George T, Kalimuth MB, Jayasimha S, Kaleem Ullah M, Patil R, Nair A, Pai U, Inbarani E, Jacob AG, Chandy VJ, John O, Sudarsanam TD, Mahesh PA. Exploring the potential of telemedicine for improved primary healthcare in India: a comprehensive review. Lancet Reg Health Southeast Asia. 2024 Jun 8;27:100431. doi: 10.1016/j.lansea.2024.100431. PMID: 38957222; PMCID: PMC11215096.

10. [2].Poonsuph, Rattakorn, The Design Blueprint for a Large-Scale Telehealth Platform,

International Journal of Telemedicine and Applications, 2022, 8486508, 15 pages, 2022. https://doi.org/10.1155/2022/8486508

11. [11].Research and Analysis of Facial Recognition Based on FaceNet, DeepFace, and OpenFace- 3.Li, Minghan. (2025). Research and Analysis of Facial Recognition Based on FaceNet, DeepFace, and OpenFace. ITM Web of Conferences. 70. 03009. 10.1051/itmconf/20257003009.

12. [6]. Suhas, Satish & Kumar, ChannaveerachariNaveen & Bada Math, Suresh & Manjunatha, Narayana. (2022). E-sanjeevani: A pathbreaking telemedicine initiativefrom India. Journal of Psychiatry Spectrum. 1. 111. 10.4103/jopsys.jopsys_8_21.

13. [15]. The Growing Impact of Natural Language Processing in Healthcare and Public Health Aadit Jerfy, Owen Selden, and Rajesh Balkrishnan

14. [3]. U. Bharti, D. Bajaj, H. Batra, S. Lalit, S. Lalit and A. Gangwani, "Medbot: Conversational Artificial Intelligence Powered Chatbot for Delivering Tele-Health after COVID-19," 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2020, pp. 870-875, doi: 10.1109/ICCES48766.2020.9137944. keywords: {Telehealth;chatbot;natural language processing;medbot;natural language understanding;conversational technology;digital health;voice user interface;conversational user interface;conversational agent;human-computer interaction}

15. [5].Udegbe, Francisca & Ebulue, Ogochukwu & Ebulue, Charles & Ekesiobi, Chukwunonso. (2024). THE ROLE OF ARTIFICIAL INTELLIGENCE IN HEALTHCARE: A SYSTEMATIC REVIEW OF APPLICATIONS AND CHALLENGES. International Medical Science Research Journal. 4. 500-508. 10.51594/imsrj.v4i4.1052.

16. [9].Zebari, Ramadan & Sallow, Amira. (2021). Face Detection and Recognition Using OpenCV. Journal of Soft Computing and Data Mining. 2. 10.30880/jscdm.2021.02.02.008.

APPENDIX-A

PSUEDOCODE

```
import json
import nltk
import pandas as pd
import torch
from transformers import AutoModelForCausalLM, AutoTokenizer
from datasets import load_dataset
from nltk.tokenize import sent_tokenize
from rouge_score import rouge_scorer
from bert_score import score
import requests
import random

# Download NLTK tokenizer
nltk.download('punkt')

# Load the fine-tuned model and tokenizer with CPU offloading
def load_hf_model(model_name):
    tokenizer = AutoTokenizer.from_pretrained(model_name)
    model = AutoModelForCausalLM.from_pretrained(
        model_name,
        torch_dtype=torch.float16,
        device_map="auto",
        offload_folder="offload_weights"
    )
    return model, tokenizer

# Generate summaries using Hugging Face model with optimizations
def generate_summaries_with_hf(prompts, model, tokenizer, batch_size=1):
    summaries = []
    for i in range(0, len(prompts), batch_size):
        batch = prompts[i:i + batch_size]
        with torch.no_grad():  # Disable gradient computation
            input_ids = tokenizer(batch, return_tensors="pt", padding=True, truncation=True).input_ids.to(model.device)
            output_ids = model.generate(
                input_ids, max_new_tokens=525, do_sample=True, top_p=0.7, temperature=0.6, repetition_penalty=1.2, no_repeat_ngram_size=3
            )
            summaries.extend(tokenizer.batch_decode(output_ids, skip_special_tokens=True))
        torch.cuda.empty_cache()  # Free memory after processing each batch
    return summaries

# Summarize conversation into user and assistant perspectives with smaller batch size
def summarize_conversations(conversations, model, tokenizer):
    user_prompts = ["Summarize as the user:\n" + "\n".join([turn["value"] for turn in convo if turn["from"] == "human"]) for convo in conversations]
    assistant_prompts = ["Summarize as the assistant:\n" + "\n".join([turn["value"] for turn in convo if turn["from"] == "assistant"]) for convo in conversations]

    user_summaries = generate_summaries_with_hf(user_prompts, model, tokenizer, batch_size=1)
    assistant_summaries = generate_summaries_with_hf(assistant_prompts, model, tokenizer, batch_size=1)

    return list(zip(user_summaries, assistant_summaries))

# Evaluate Conv-ROUGE
def evaluate_conv_rouge(references, generated):
    scorer = rouge_scorer.RougeScorer(['rouge1', 'rouge2', 'rougeL'], use_stemmer=True)
    return [{
        "Conv-ROUGE-1": scores["rouge1"].fmeasure,
        "Conv-ROUGE-2": scores["rouge2"].fmeasure,
        "Conv-ROUGE-L": scores["rougeL"].fmeasure
    } for scores in map(lambda x: scorer.score(x[0], x[1]), zip(references, generated))]
```

```
# Evaluate BERTScore with batching
"""def evaluate_bertscore(references, generated):
    with torch.no_grad():
        _, _, F1 = score(generated, references, lang="en", batch_size=min(2, len(generated)), rescale_with_baseline=True)
    torch.cuda.empty_cache()  # Free memory after BERTScore computation
    return F1.tolist()"""


def evaluate_bertscore(references, generated):
    with torch.no_grad():
        _, _, F1 = score(tuple(generated), tuple(references), lang="en", batch_size=min(2, len(generated)), rescale_with_baseline=True)
    torch.cuda.empty_cache()  # Free memory after BERTScore computation
    return F1.tolist()



# Load dataset and process conversations
def load_dataset_from_hf(dataset_name, model, tokenizer):
    dataset = load_dataset(dataset_name, split="train")
    conversations = [convo for convo in dataset["conversations"] if isinstance(convo, list) and len(convo) >= 2]
    #conversations = conversations[:5] # Limit to Load First 5 conversations for summarization
    conversations = random.sample(conversations, min(25, len(conversations)))
    summarized_conversations = summarize_conversations(conversations, model, tokenizer)
    print(f"Loaded {len(summarized_conversations)} summarized user-assistant pairs.")
    return summarized_conversations

# Run Evaluation
def evaluate_model(dataset_name, model_name, output_file):
    model, tokenizer = load_hf_model(model_name)
    conversation_pairs = load_dataset_from_hf(dataset_name, model, tokenizer)
    results = []
    # Only evaluate the 5 summarized conversations
    user_summaries, expected_responses = zip(*conversation_pairs)

    # Generate responses one-by-one to reduce memory usage
    generated_responses = generate_summaries_with_hf(["Reply Concisely:\n" + user for user
in user_summaries], model, tokenizer, batch_size=1)

    # Compute Conv-ROUGE in batch
    conv_rouge_scores = evaluate_conv_rouge(expected_responses, generated_responses)

    # Compute BERTScore in batch
    bert_scores = evaluate_bertscore(expected_responses, generated_responses)

    # Store results
    for i in range(len(user_summaries)):
        results.append({
            "User Summary": user_summaries[i],
            "Expected Response": expected_responses[i],
            "Generated Response": generated_responses[i],
            "Conv-ROUGE-1":  conv_rouge_scores[i]["Conv-ROUGE-1"],
            "Conv-ROUGE-2":  conv_rouge_scores[i]["Conv-ROUGE-2"],
            "Conv-ROUGE-L": conv_rouge_scores[i]["Conv-ROUGE-L"],
            "BERTScore": bert_scores[i]
        })

    df = pd.DataFrame(results)
    df.to_csv(output_file, index=False)
    print(f"Evaluation results saved to {output_file}")

# Main Execution
if _name_ == "_main_":
    dataset_name = "savinirsekas/MediTalk-300-Conversational"  # Replace with your dataset
    model_name  =  "savinirsekas/MediTalk-300-Llama-3.1-8B-Instruct-bnb-4bit"   #  Replace
```

```
with your model
    output_file = "conv_rouge_bertscore_results_hf_llama3-8b-it-bnb-4bit_ft_25.csv"

    evaluate_model(dataset_name, model_name, output_file)
```

APPENDIX-B

SCREENSHOTS

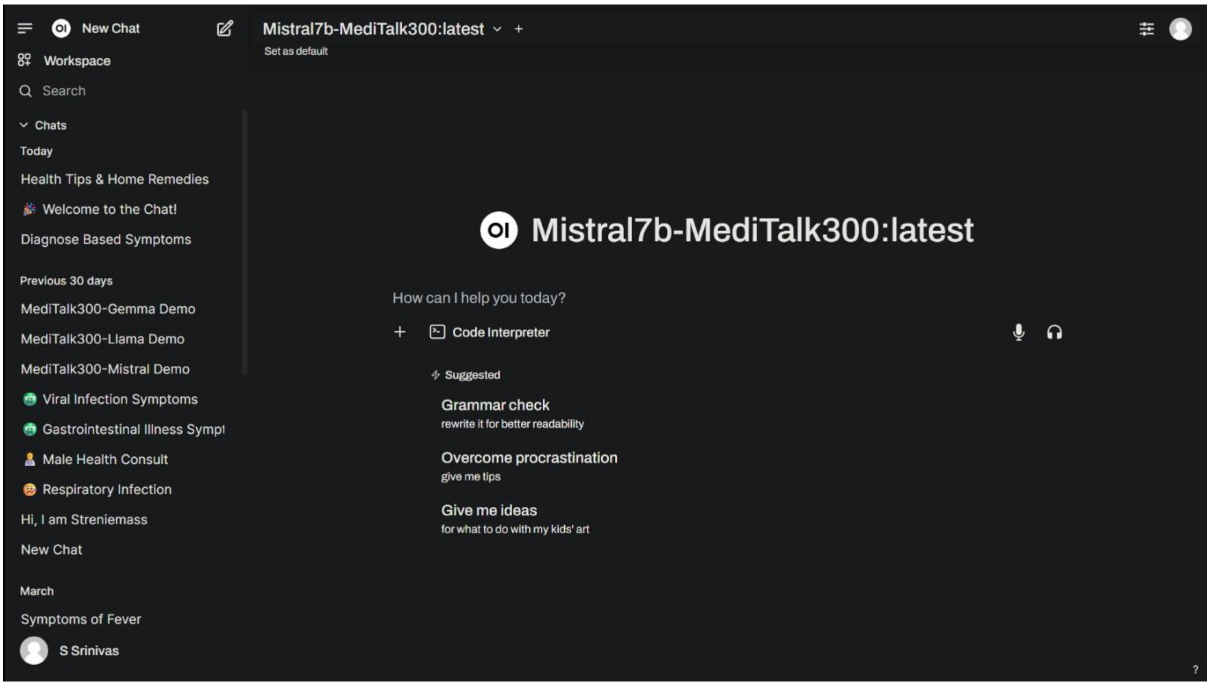


Fig.3.1 Fine-Tuned Mistral on Open Web UI

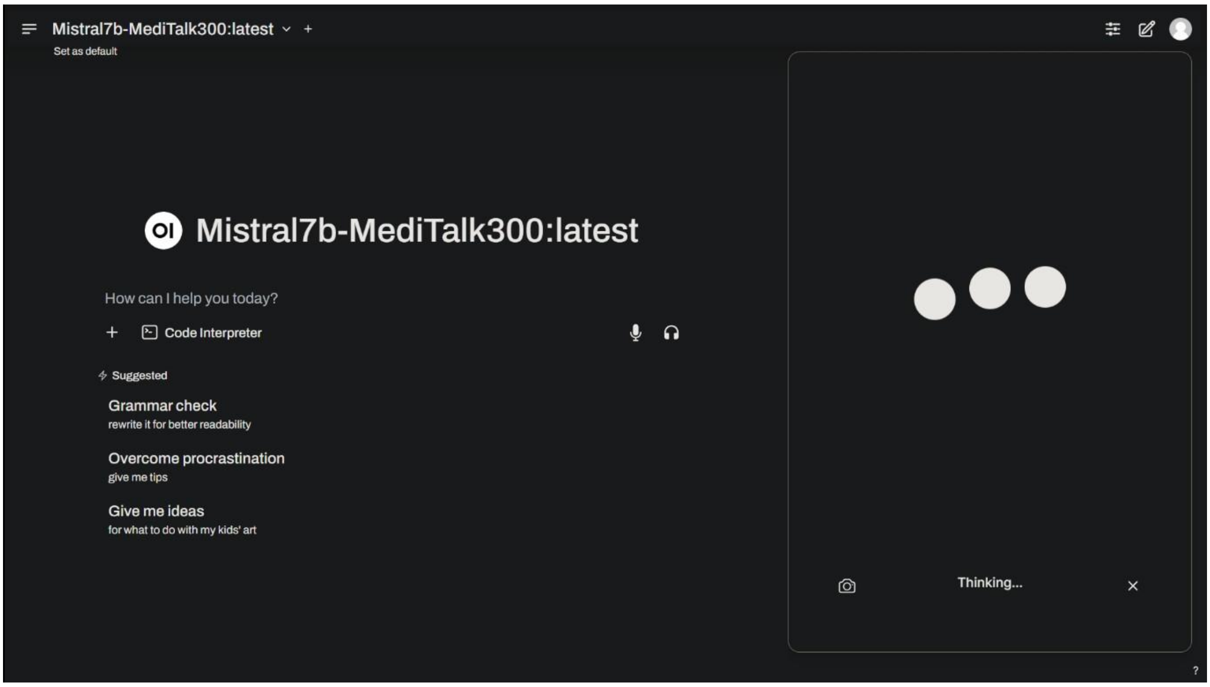
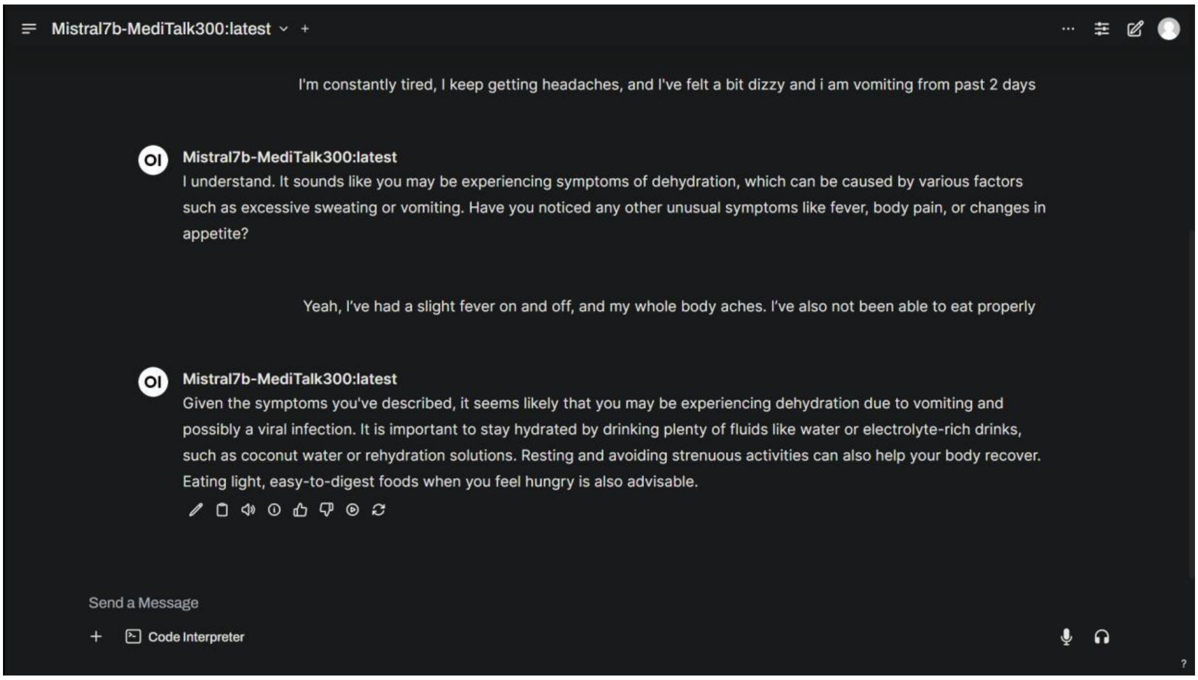

Mistral7b-MediTalk300:latest
Set as default
Mistral7b-MediTalk300:latest
How can I help you today?
Code Interpreter
Suggested
Grammar check
rewrite it for better readability
Overcome procrastination
give me tips
Give me ideas
for what to do with my kids' art
Thinking...

Fig.3.2 Mistral Chatbot



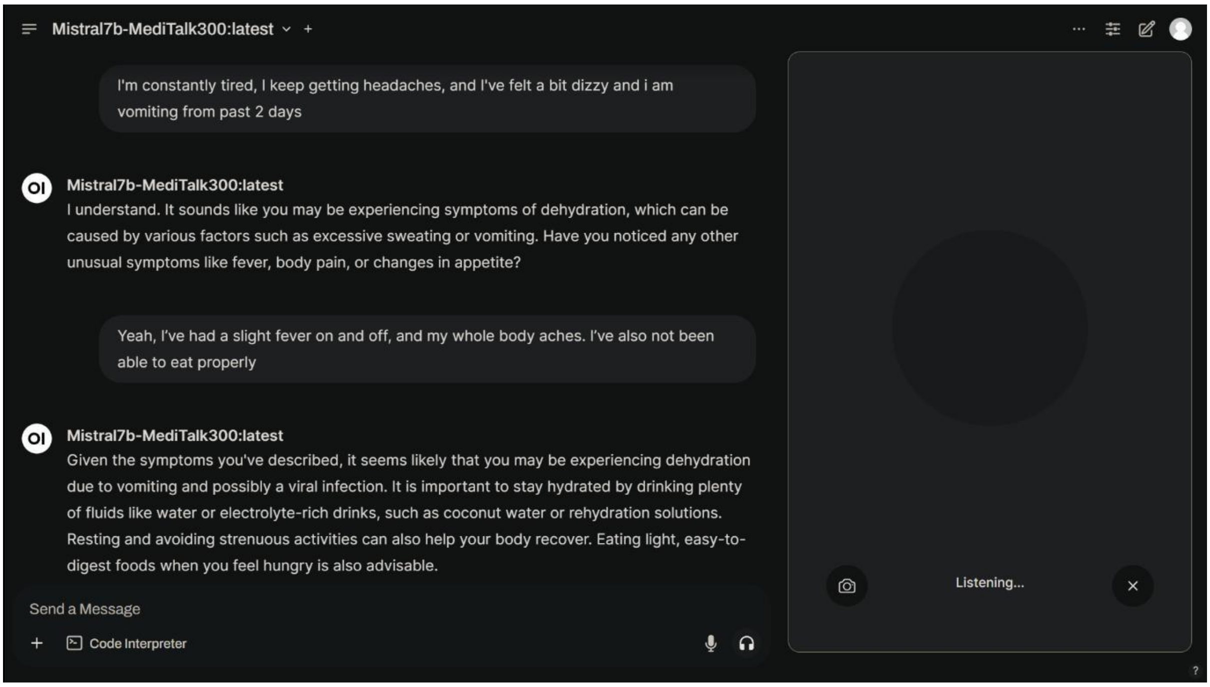
I'm constantly tired, I keep getting headaches, and I've felt a bit dizzy and i am vomiting from past 2 days

Mistral7b-MediTalk300:latest
I understand. It sounds like you may be experiencing symptoms of dehydration, which can be caused by various factors such as excessive sweating or vomiting. Have you noticed any other unusual symptoms like fever, body pain, or changes in appetite?

Yeah, I've had a slight fever on and off, and my whole body aches. I've also not been able to eat properly

Mistral7b-MediTalk300:latest
Given the symptoms you've described, it seems likely that you may be experiencing dehydration due to vomiting and possibly a viral infection. It is important to stay hydrated by drinking plenty of fluids like water or electrolyte-rich drinks, such as coconut water or rehydration solutions. Resting and avoiding strenuous activities can also help your body recover. Eating light, easy-to-digest foods when you feel hungry is also advisable.

Fig.3.3 Mistral Response



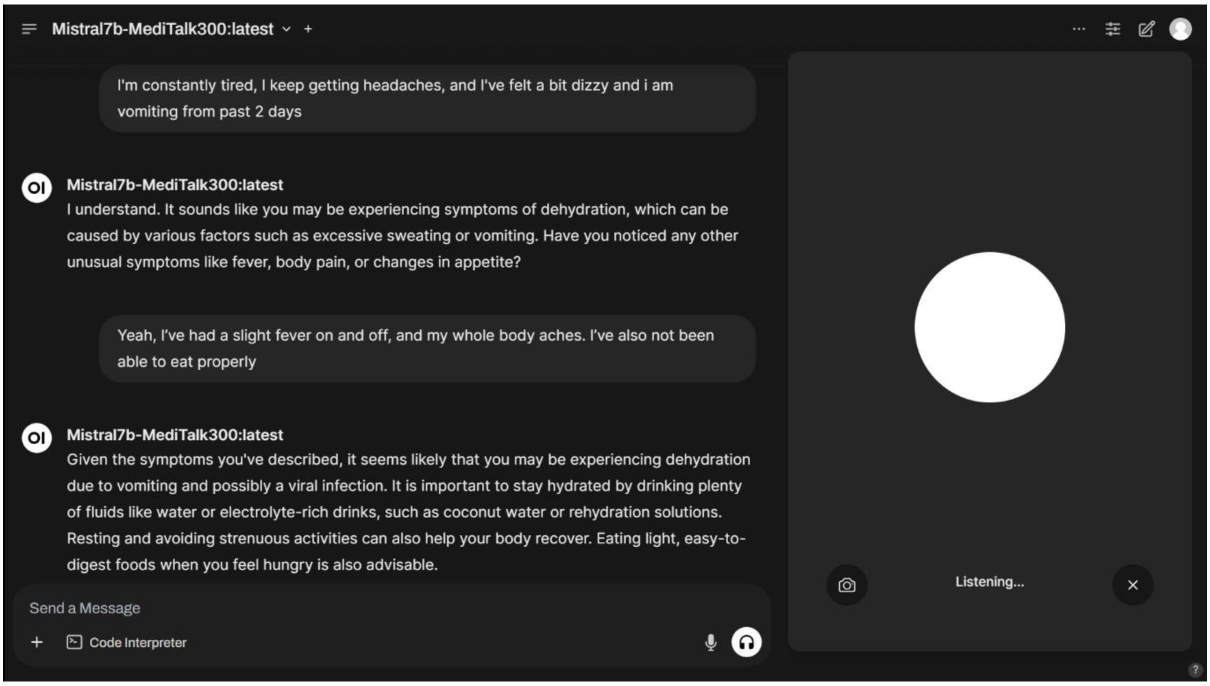
I'm constantly tired, I keep getting headaches, and I've felt a bit dizzy and i am vomiting from past 2 days

Mistral7b-MediTalk300:latest
I understand. It sounds like you may be experiencing symptoms of dehydration, which can be caused by various factors such as excessive sweating or vomiting. Have you noticed any other unusual symptoms like fever, body pain, or changes in appetite?

Yeah, I've had a slight fever on and off, and my whole body aches. I've also not been able to eat properly

Mistral7b-MediTalk300:latest
Given the symptoms you've described, it seems likely that you may be experiencing dehydration due to vomiting and possibly a viral infection. It is important to stay hydrated by drinking plenty of fluids like water or electrolyte-rich drinks, such as coconut water or rehydration solutions. Resting and avoiding strenuous activities can also help your body recover. Eating light, easy-to-digest foods when you feel hungry is also advisable.

Fig.3.4 Mistral Voice Bot- Start Recording



I'm constantly tired, I keep getting headaches, and I've felt a bit dizzy and i am vomiting from past 2 days

Mistral7b-MediTalk300:latest
I understand. It sounds like you may be experiencing symptoms of dehydration, which can be caused by various factors such as excessive sweating or vomiting. Have you noticed any other unusual symptoms like fever, body pain, or changes in appetite?

Yeah, I've had a slight fever on and off, and my whole body aches. I've also not been able to eat properly

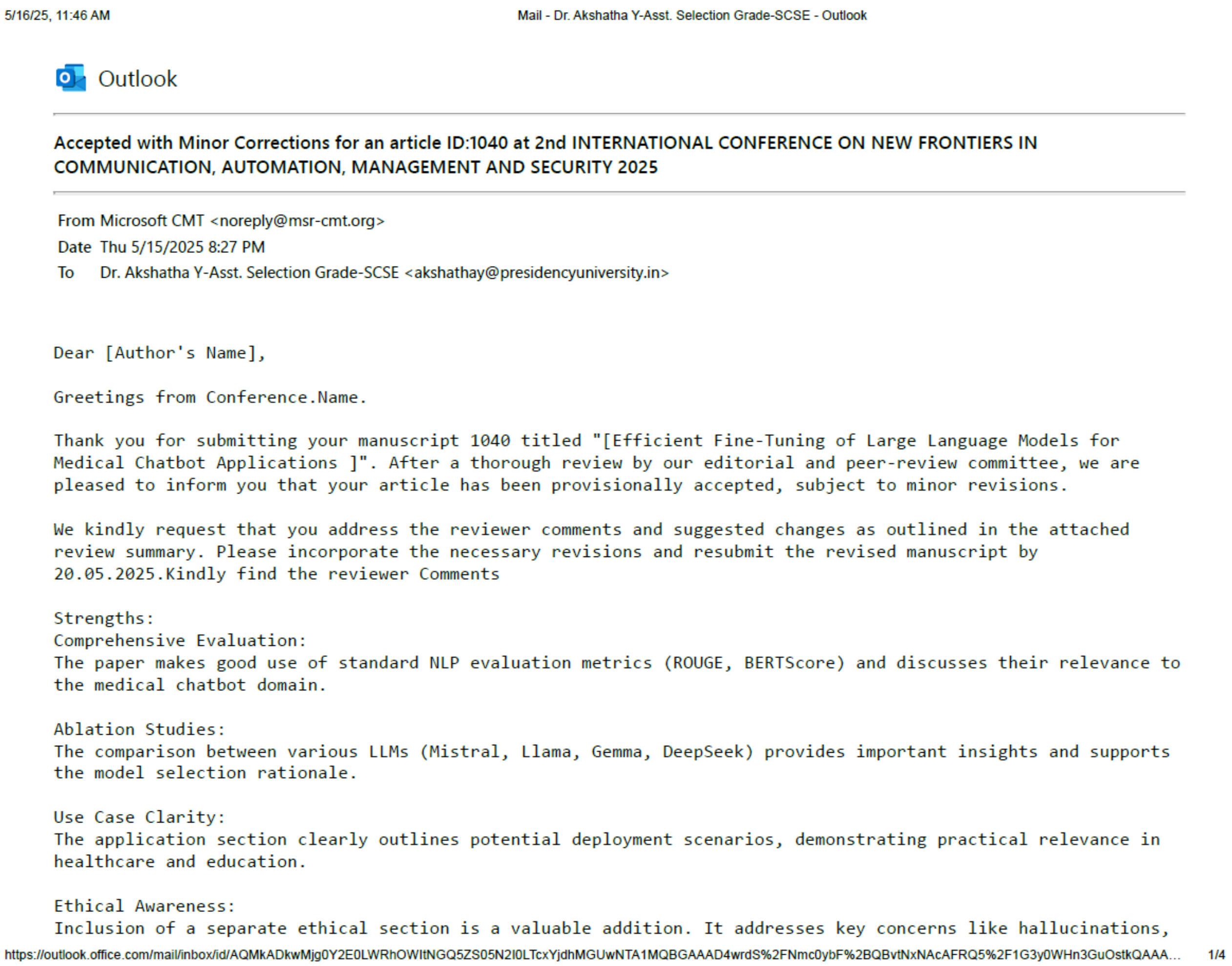
Mistral7b-MediTalk300:latest
Given the symptoms you've described, it seems likely that you may be experiencing dehydration due to vomiting and possibly a viral infection. It is important to stay hydrated by drinking plenty of fluids like water or electrolyte-rich drinks, such as coconut water or rehydration solutions. Resting and avoiding strenuous activities can also help your body recover. Eating light, easy-to-digest foods when you feel hungry is also advisable.

Fig.3.5 Mistral Voice bot- Stop Recording

APPENDIX-C

ENCLOSURES

1. Journal publication/Conference Paper Presented Certificates



5/16/25, 11:46 AM   Mail - Dr. Akshatha Y-Asst. Selection Grade-SCSE - Outlook

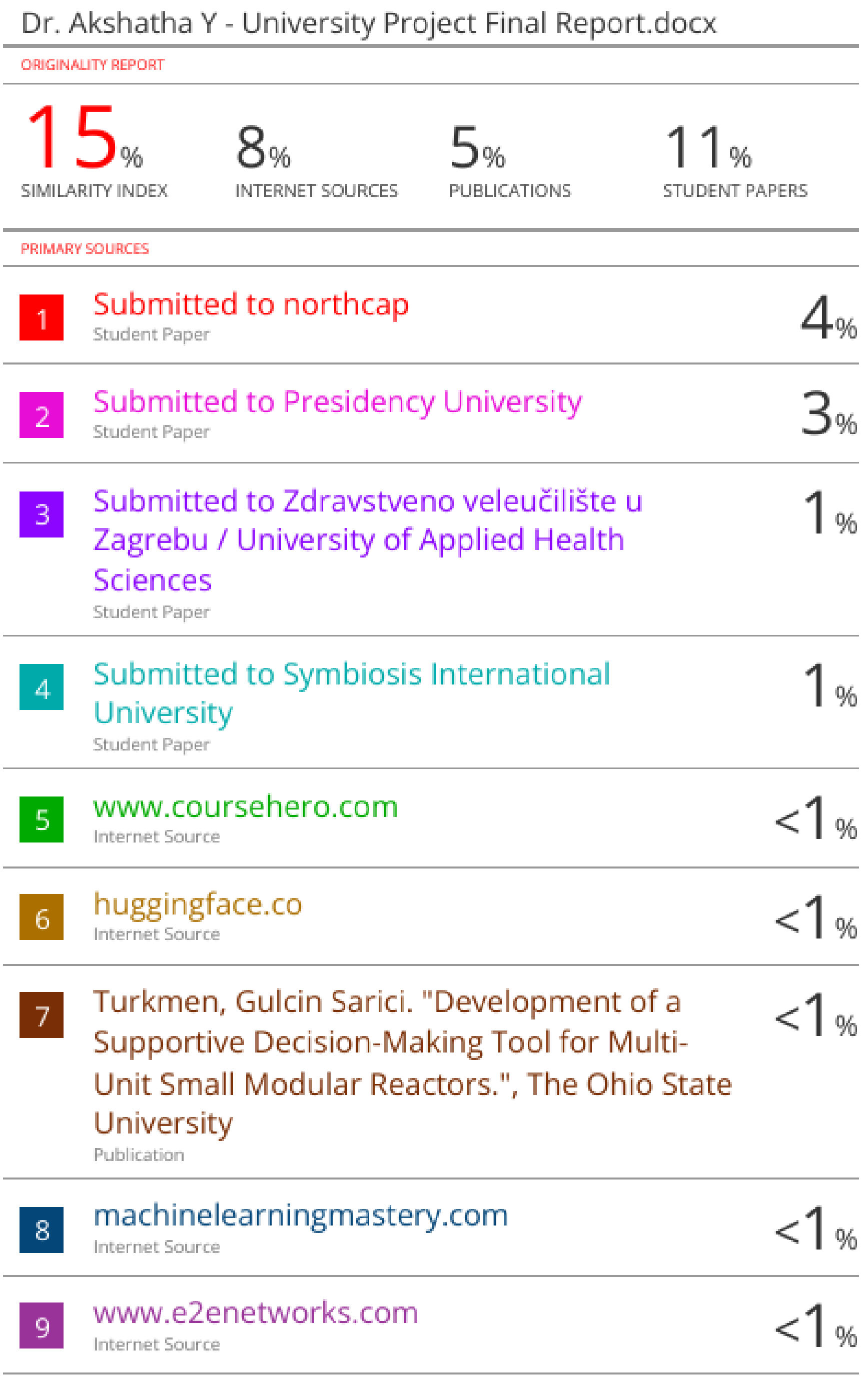
Accepted with Minor Corrections for an article ID:1040 at 2nd INTERNATIONAL CONFERENCE ON NEW FRONTIERS IN COMMUNICATION, AUTOMATION, MANAGEMENT AND SECURITY 2025

From Microsoft CMT <noreply@msr-cmt.org>
Date Thu 5/15/2025 8:27 PM
To Dr. Akshatha Y-Asst. Selection Grade-SCSE <akshathay@presidencyuniversity.in>

```
Dear [Author's Name],

Greetings from Conference.Name.

Thank you for submitting your manuscript 1040 titled "[Efficient Fine-Tuning of Large Language Models for
Medical Chatbot Applications ]". After a thorough review by our editorial and peer-review committee, we are
pleased to inform you that your article has been provisionally accepted, subject to minor revisions.

We kindly request that you address the reviewer comments and suggested changes as outlined in the attached
review summary. Please incorporate the necessary revisions and resubmit the revised manuscript by
20.05.2025.Kindly find the reviewer Comments

Strengths:
Comprehensive Evaluation:
The paper makes good use of standard NLP evaluation metrics (ROUGE, BERTScore) and discusses their relevance to
the medical chatbot domain.

Ablation Studies:
The comparison between various LLMs (Mistral, Llama, Gemma, DeepSeek) provides important insights and supports
the model selection rationale.

Use Case Clarity:
The application section clearly outlines potential deployment scenarios, demonstrating practical relevance in
healthcare and education.

Ethical Awareness:
Inclusion of a separate ethical section is a valuable addition. It addresses key concerns like hallucinations,
```

https://outlook.office.com/mail/inbox/id/AQMkADkwMjg0Y2E0LWRhOWItNGQ5ZS05N2I0LTcxYjdhMGUwNTA1MQBGAAAD4wrdS%2FNmc0ybF%2BQBvtNxNAcAFRQ5%2F1G3y0WHn3GuOstkQAAA...   1/4

2. Similarity Index / Plagiarism Check report



Dr. Akshatha Y - University Project Final Report.docx

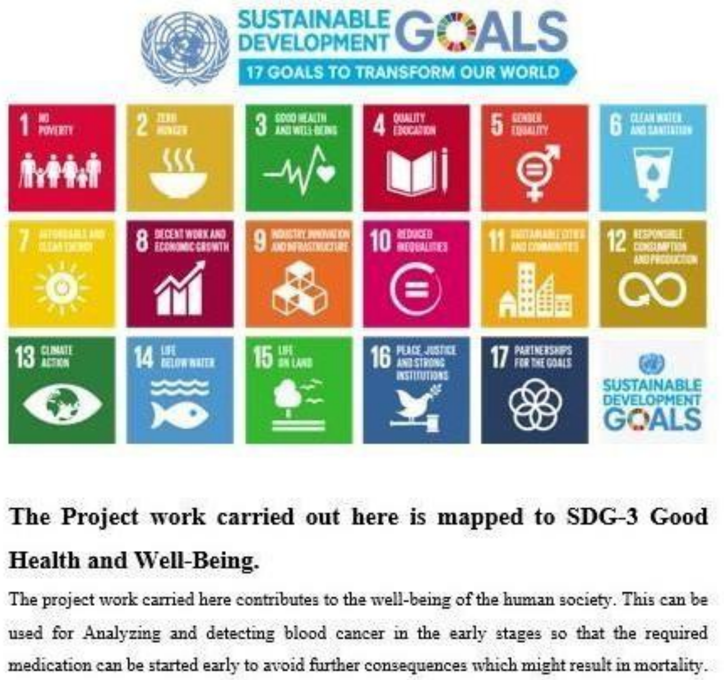
ORIGINALITY REPORT

15%	8%	5%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to northcap Student Paper	4%
2	Submitted to Presidency University Student Paper	3%
3	Submitted to Zdravstveno veleučilište u Zagrebu / University of Applied Health Sciences Student Paper	1%
4	Submitted to Symbiosis International University Student Paper	1%
5	www.coursehero.com Internet Source	<1%
6	huggingface.co Internet Source	<1%
7	Turkmen, Gulcin Sarici. "Development of a Supportive Decision-Making Tool for Multi-Unit Small Modular Reactors.", The Ohio State University Publication	<1%
8	machinelearningmastery.com Internet Source	<1%
9	www.e2enetworks.com Internet Source	<1%

SUSTAINABLE DEVELOPMENT GOALS


SUSTAINABLE DEVELOPMENT GOALS
17 GOALS TO TRANSFORM OUR WORLD
1 NO POVERTY
2 ZERO HUNGER
3 GOOD HEALTH AND WELL-BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
6 CLEAN WATER AND SANITATION
7 AFFORDABLE AND CLEAN ENERGY
8 DECENT WORK AND ECONOMIC GROWTH
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
10 REDUCED INEQUALITIES
11 SUSTAINABLE CITIES AND COMMUNITIES
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND
16 PEACE, JUSTICE AND STRONG INSTITUTIONS
17 PARTNERSHIPS FOR THE GOALS
SUSTAINABLE DEVELOPMENT GOALS

The Project work carried out here is mapped to SDG-3 Good Health and Well-Being.

The project work carried here contributes to the well-being of the human society. This can be used for Analyzing and detecting blood cancer in the early stages so that the required medication can be started early to avoid further consequences which might result in mortality.

Fig.3.6 Sustainable Development Goals (SDGs)

The project aligns with several Sustainable Development Goals (SDGs) as follows:

1. SDG 3: Good Health and Well-being – Improves healthcare delivery by automating medical transcription, symptom extraction, and report generation, leading to better diagnoses and patient care.
2. SDG 4: Quality Education – Provides an educational tool for healthcare professionals, promoting learning in AI-driven healthcare technologies. %
3. SDG 9: Industry, Innovation, and Infrastructure – Encourages technological innovation in healthcare and supports the development of smart healthcare infrastructure.
4. SDG 10: Reduced Inequalities – Enhances access to healthcare by improving efficiency, especially in underserved regions, and supports language inclusivity.
5. SDG 12: Responsible Consumption and Production – Reduces paper usage and increases efficiency, contributing to sustainable healthcare practices