

Innovative Product Design Project Report

Project Title: MedPal – AI Powered Health Assistant Android App

Problem Statement: To design an AI-powered mobile health assistant app for Android capable of performing agentic tasks and provide general health tips to the user.

Mentor: Dr. Gulshan Goyal

College Name: Chandigarh College of Engineering and Technology, Sector - 26

Submitted By: Himender Sharma (CO24326), Ginisha Miglani (CO24323)

Academic Year: 2024 – 2025

Contents

1. Acknowledgment
2. Abstract
3. Introduction
4. Literature survey
5. Problem statement
6. Methodology
7. Results
8. Future scope
9. References
10. Plagiarism report

Acknowledgement

We would like to express our sincere gratitude to everyone who contributed to the successful completion of this project, “MedPal – AI-powered health assistant app”.

First and foremost, we would like to thank our mentor **Dr. Gulshan Goyal** for his valuable guidance and encouragement, and support for the development of this project. His insights and suggestions greatly helped in shaping the direction and quality of this project.

We would also like to thank the **faculty members and staff** of the Computer Science & Engineering department for providing the resources and environment necessary for making this innovative journey possible.

Finally, we would like to acknowledge all those who, directly or indirectly, contributed to the successful completion of this project.

Abstract

This project presents *MedPal*, an AI-powered health assistant app designed for Android devices. The aim was to create a more emotionally intelligent and capable alternative to existing health applications. Most health apps today either focus on physical health tracking or basic chatbot replies. *MedPal* takes a different route by integrating a locally hosted large language model (LLM) through Ollama and FastAPI, enabling the app to engage in natural, human-like conversations, understand symptoms, and even interpret medical reports.

With features like a WhatsApp-inspired chat interface, offline functionality, and tool-based reasoning using LangChain, the app is designed to feel more like a helpful friend than a machine. It also supports basic Over-The-Counter (OTC) suggestions, stress-relieving tools like meditation prompts, and contextual awareness through memory.

The project aims to bridge the gap between health tech and emotional support by bringing together AI, local computing, and user-friendly design into a single, cohesive assistant.

Introduction (about the project its objective and significance)

About the Project

MedPal is an AI powered health assistant app. At its core, is an LLM (Large Language Model) which is the brain of the AI assistant. It has conversational abilities and has access to tools which can help accomplish health-related tasks. The app is designed to help users manage their basic health needs by providing features such as **medication advice**, **symptom-based assessments**, **health report analysis**, and **guided meditation sessions**, among others all while engaging with the user in a warm, friendly tone. All these features are accessible to the user in a polished, **WhatsApp-like interface**, making it user-friendly. Lastly, the current version of the app connects to a **FastAPI server** hosted on a personal laptop, which runs the LLM model locally. Due to the offline nature and hardware limitations, this can result in **response delays of up to a minute** in extreme cases.

Objective

The main objective of this project is to design and develop an AI-powered health assistant app that helps users access basic health services that do not require a medical professional.

The specific goals include:

- Providing **symptom-based assessments** to help users understand common health issues
- Offering **medication suggestions** based on user input, with clear disclaimers to consult a medical professional before taking any action
- **Analyzing health reports** such as PDFs or images to extract meaningful information
- Delivering **meditation sessions** to support mental wellness
- Serving as an **emotionally supportive AI assistant**, designed to offer empathetic responses and practical guidance
- Maintain a **conversational, chat-based interface** that is easy to use and intuitive

In addition to these, there were some goals that were envisioned for the app:

- Doctor Appointment Booking
- Integration with wearable health devices to monitor heart rate, blood pressure and other significant health measures to provide extended support.
- Emergency health alerts and reminders

Significance

The significance of this project is in its ability to provide accessible and intelligent health support for everyday users. In a world where minor health concerns are often ignored due to lack of time, hesitation, or unavailability of medical professionals, MedPal acts as a **first option for guidance**.

By integrating conversational AI with practical tools, the app helps users to **actively take steps** in managing their health. It has features like **symptom assessments**, **medication suggestions**, **health report analysis**, and **mental wellness support**, all through a user-friendly chat interface.

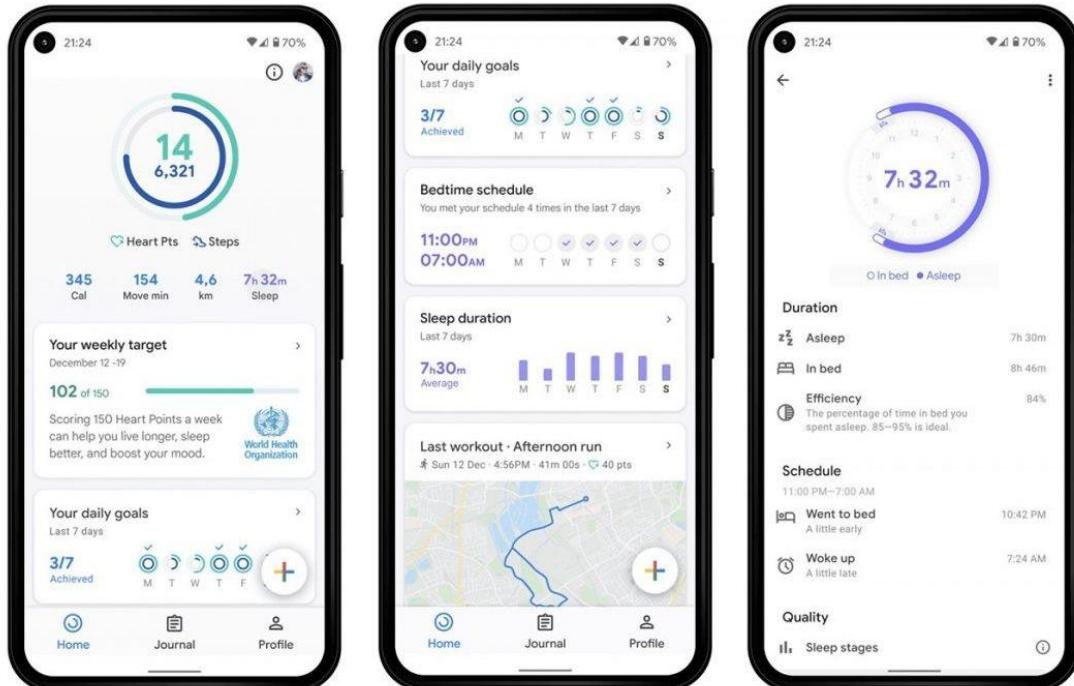
MedPal also behaves as a **caring, nurse-like assistant**, offering emotional support, which helps build a stronger relationship between the user and their well-being. The project displays how AI can be used not just for automating tasks, but also for creating **empathetic and meaningful experiences** in health technologies.

Literature and Background Survey

Introduction

There are many health-related apps like **Google Fit**, **Strava**, **Samsung Health** and countless others which provide basic health services to the user. However, **most of them** lack the ability to provide emotionally intelligent feedback to the user. They also lack the ability to automate basic health related tasks for the user. AI assistants like **ChatGPT**, **Grok**, **Meta Llama**, **Microsoft Copilot** and many more are revolutionizing the health sector. When fine tuned and trained for medical support, they provide meaningful responses helping medical students and professionals alike. This calls for a need to take a step further in extending the capabilities of the health-related apps, making them more automated, emotionally intelligent and human centered.

A glimpse of Google Fit's UI showcasing no support for any AI assistant:



A major direction to our endeavors was provided by the **Zepp** app which takes things to the next level with **Zepp Aura**, which is a personalized wellness assistant focused on mental health and sleep improvement. It uses AI to offer calming sounds and mood tracking, aiming to enhance emotional well-being. In addition to this, many more apps are integrating AI to provide a better user experience.



WHO has recently acknowledged AI's potential for mental health care especially in supporting clinicians and early identification of issues but also warned of serious limitations such as bias, lack of diverse disorder coverage, and inadequate validation.

See - [Artificial intelligence in mental health research: new WHO study on applications and challenges](#)

For example, the **HealthifyMe** app gives access to an AI-powered nutrition assistant called **Ria**, which provides users with personalized diet suggestions and reminders based on their habits and goals. Ria can answer queries about food choices, calorie counts, and general dietary advice through a chat-like interface.

Redesigning HealthifyMe's AI-powered meal tracking
UX case study

Limitations of the aforementioned apps with AI assistant:

- Only give basic replies, not real conversations
- Don't understand feelings or give emotional support
- Can't read medical reports or explain test results
- Symptom checkers are usually multiple choice
- Need internet all the time, not offline
- Don't remember past chats or build connection
- Rarely help with mental health or stress
- Don't use multiple tools to solve a problem together

Need for an Improved Approach

In light of recent events and growing use of AI in health apps. We decided to create an app that tries to overcome the shortcomings of other similar AI health apps, by incorporating an LLM that can perform more than just basic calculations and can engage in human-like conversations to support the user's mental and physical wellbeing. Our assistant aims to ask users for symptoms and diagnose an illness. It keeps conversational memory to remember key information about the user, offering a more personalized experience over time. It can also read medical reports which let's be honest, most of us are unable to, and interpret it for the user. It can also suggest users, Over-The-Counter (OTCs) medicines to treat non-serious illnesses like headache, common cold, etc. with due **caution, without attempting to replace a medical professional.**

Summary

While existing apps have made great progress in health tracking or AI support individually, there remains a noticeable gap in **emotional intelligence, automation, and holistic health conversation.** MedPal aims to take a step forward by experimenting with a conversational AI model that can support users' mental and physical wellness in a more engaging and empathetic manner.

Problem Statement

In the current digital health space, most mobile applications either provide generic fitness tracking or offer limited medical assistance through predefined options. While AI-powered apps do exist, many lack emotional intelligence, conversational depth, or the ability to automate multiple health-related tasks. There is a noticeable gap in providing users with a virtual health assistant that not only offers medical support but also interacts in a human-like manner, interprets complex reports, and supports user's mental well-being.

To design and implement an AI powered health app that can do more than just general problem solving, an app that can engage with the users at an emotional level through human-like conversation using a Large Language Model (LLM) and perform agentic tasks like booking an appointment for a doctor, read medical reports from PDFs and other complex tasks.

Methodology

Planning and Design

The initial phase focused on identifying the limitations of the current AI health apps and understanding user expectations. Based on this, we decided to design an app that focuses on more than just basic health tracking. Something that was emotionally supportive and had access to tools for a variety of complex tasks that those other apps didn't. We also focused on creating a sleek WhatsApp-like UI to make the app seem more familiar and easier to use. It was also meant to make the user experience the app more like a friendly chat with a companion, rather than just using a boring tool.

Technology Stack



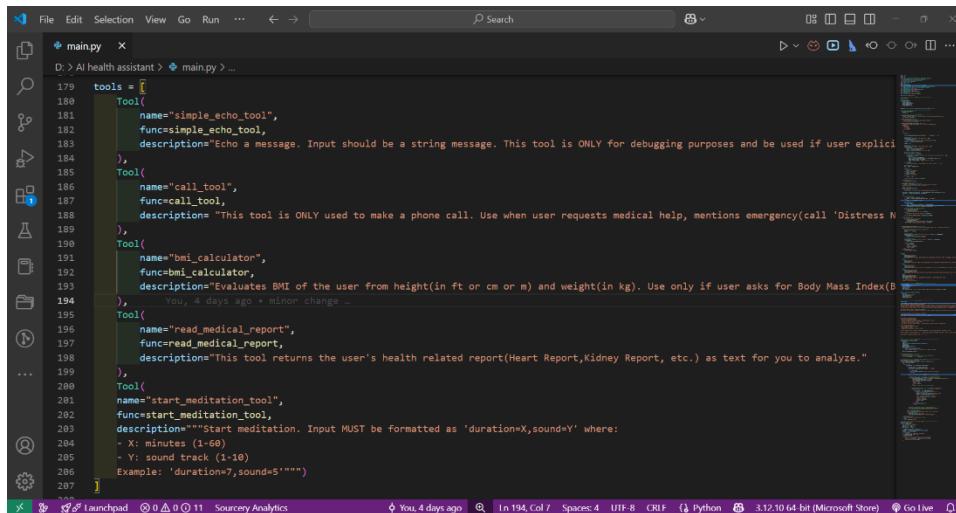
For the frontend, the app was developed using **Android Studio**, with **Kotlin** as the primary language and **Jetpack Compose** for building the user interface. This combination allowed for a modern, responsive, and visually appealing UI that feels smooth and intuitive to use. On the backend, we used **FastAPI**, a fast and lightweight Python web framework, to serve as the core API layer that handles user requests and AI interactions. The backend was coded in **Visual Studio Code**. For the AI integration, the app connects to **Ollama**, a local framework that allows running large language models directly on personal machines without requiring cloud access. The specific model used was **gemma3-tools:4b**, a custom variant of Google's Gemma model, enhanced with tool support to handle tasks like text extraction from images, health calculations, and more. **LangChain**, a framework for advanced AI application development, was used to create the intelligent agent. This agent is powered by an LLM with tool usage capabilities and reasoning logic, enabling more complex and context-aware interactions. Additionally, **pytesseract** (a Python library) was used for text extraction from images.



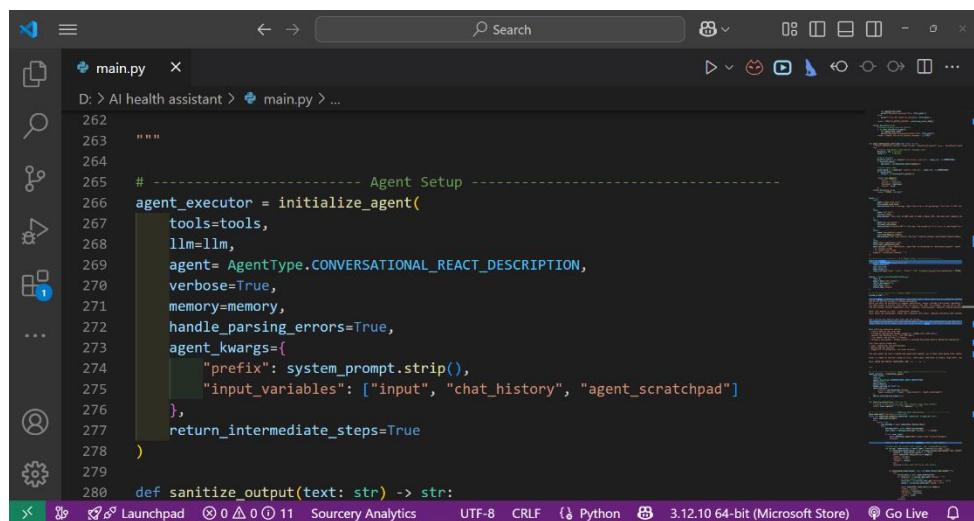
Building the Backend

A FastAPI server runs locally on the laptop, hosting the LLM. The LLM itself runs using Ollama. Requests are sent from the Android app to the server via a WebSocket, avoiding timeout. Documents and images are sent through an HTTP POST request. The Python backend defines the agent's personality, tool access, and WebSocket-based architecture.

The LLM uses LangChain agent to dynamically respond to inputs, access tools, and keep track of context. This allows the app to handle complex tasks like interpreting medical PDFs or checking symptoms via dialogue.



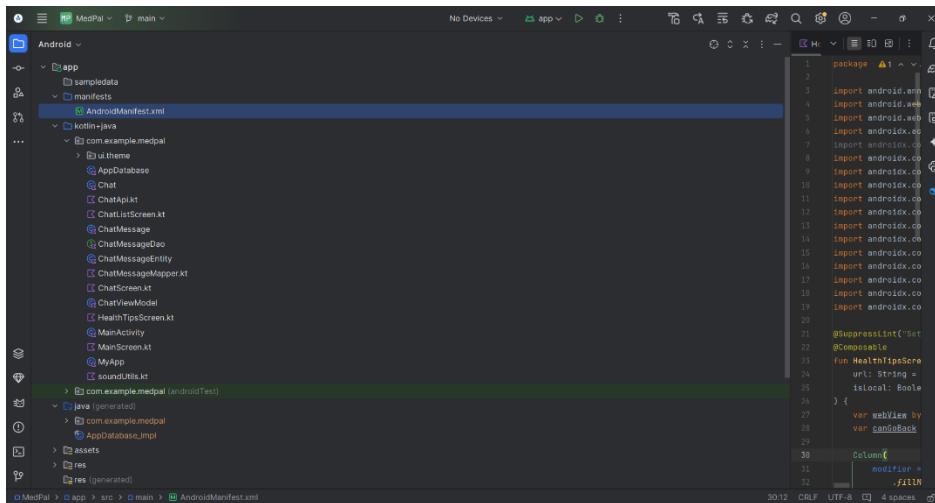
```
D: > AI health assistant > main.py > ...
179 tools = [
180     Tool(
181         name="simple_echo_tool",
182         func=simple_echo_tool,
183         description="Echo a message. Input should be a string message. This tool is ONLY for debugging purposes and be used if user explicitly asks for it."
184     ),
185     Tool(
186         name="call_tool",
187         func=call_tool,
188         description="This tool is ONLY used to make a phone call. Use when user requests medical help, mentions emergency(call 'Distress Number') etc."
189     ),
190     Tool(
191         name="bmi_calculator",
192         func=bmi_calculator,
193         description="Evaluates BMI of the user from height(in ft or cm or m) and weight(in kg). Use only if user asks for Body Mass Index(BMI)."
194     ),
195     Tool(
196         name="read_medical_report",
197         func=read_medical_report,
198         description="This tool returns the user's health related report(Heart Report,Kidney Report, etc.) as text for you to analyze."
199     ),
200     Tool(
201         name="start_meditation_tool",
202         func=start_meditation_tool,
203         description="Start meditation. Input MUST be formatted as 'duration=X,sound=Y' where:
204             - X: minutes (1-60)
205             - Y: sound track (1-10)
206             Example: 'duration=7,sound=5'"
207     )
208 ]
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287 ]
```



```
262
263 """
264
265 # ----- Agent Setup -----
266 agent_executor = initialize_agent(
267     tools=tools,
268     llm=llm,
269     agent= AgentType.CONVERSATIONAL_REACT_DESCRIPTION,
270     verbose=True,
271     memory=memory,
272     handle_parsing_errors=True,
273     agent_kwargs={
274         "prefix": system_prompt.strip(),
275         "input_variables": ["input", "chat_history", "agent_scratchpad"]
276     },
277     return_intermediate_steps=True
278 )
279
280 def sanitize_output(text: str) -> str:
```

Frontend Implementation

The Android app was built using Jetpack Compose for a modern, WhatsApp-style UI on **Android Studio**, the recommended IDE for Android Development. Each message the user sends is sent to the backend, and responses are displayed in a clean, chat-like format. The UI supports typing indicators, error handling, and even emoji-based reactions for emotional feedback. A Room Persistence library stores the data and **Data Access Objects** (DAO) interface is used to update, delete, query and insert data in the database. This ensures that when the user closes the app, the messages are retained. A few non-copyrighted mediation songs were added to the app's resources. Important permissions required to make a phone call, internet access for the web interface, access to media files for uploading to the server, etc. were defined in the `AndroidManifest.xml` file.



This screenshot shows the same Android Studio session with the `AndroidManifest.xml` file open again. In this view, several permission declarations are highlighted in green, indicating they are selected or being edited. The XML code is identical to the one shown in the previous screenshot.

Tool Integration

Symptom Checker:

Users can describe how they feel, and the model infers possible non-serious illnesses based on trained prompt patterns.

Report Reader:

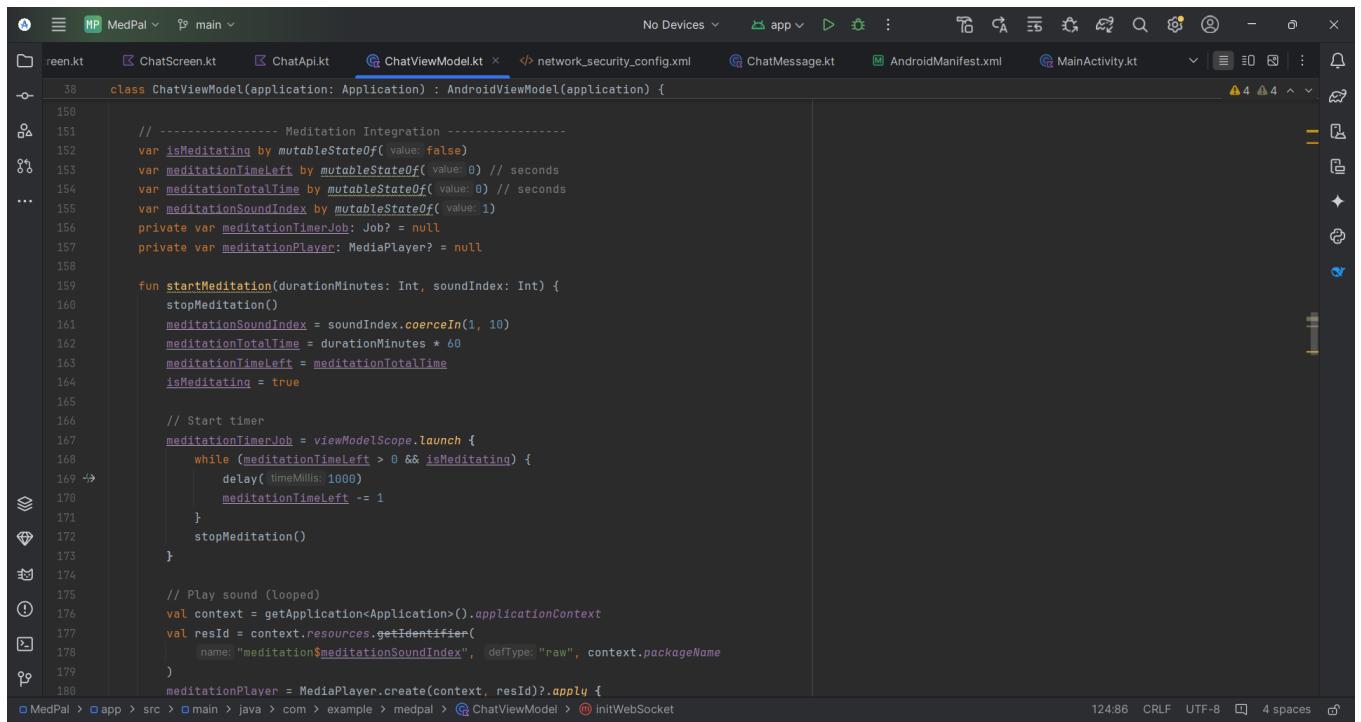
Uploaded PDFs or images with medical content are processed using OCR and summarized by the AI assistant.

Mental Wellness:

A tool initiates calming scripts or guided meditation based on mood cues detected in the conversation.

User-initiated Call:

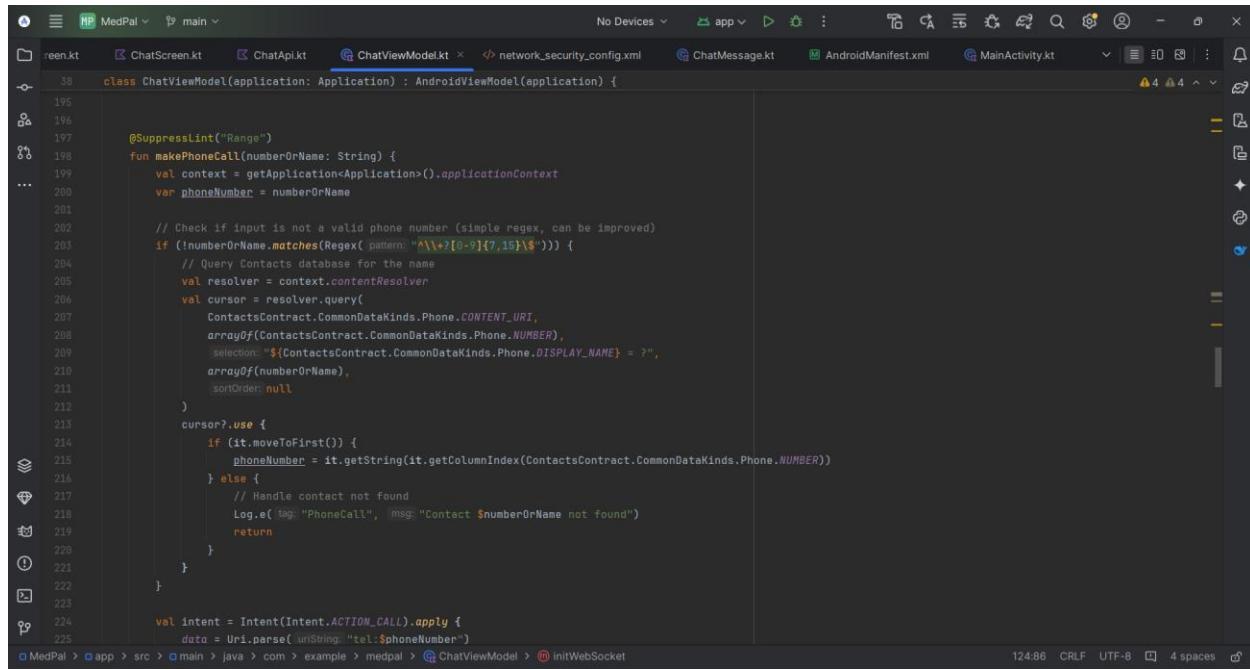
A tool that dials a number based on user input by contact name or number. It is also invoked in times of emergency, if it's detected in the user input, for example, if the user says "I am hurt and need help".



The screenshot shows the Android Studio code editor with the file `ChatViewModel.kt` open. The code is part of the `MedPal` application. The code implements a `ChatViewModel` class that extends `AndroidViewModel`. It contains logic for managing meditation states, starting a timer, playing sounds, and creating a `MediaPlayer`.

```
150
151     // ----- Meditation Integration -----
152     var isMeditating by mutableStateOf( value: false )
153     var meditationTimeLeft by mutableStateOf( value: 0 ) // seconds
154     var meditationTotalTime by mutableStateOf( value: 0 ) // seconds
155     var meditationSoundIndex by mutableStateOf( value: 1 )
156     private var meditationTimerJob: Job? = null
157     private var meditationPlayer: MediaPlayer? = null
158
159     fun startMeditation(durationMinutes: Int, soundIndex: Int) {
160         stopMeditation()
161         meditationSoundIndex = soundIndex.coerceIn(1, 10)
162         meditationTotalTime = durationMinutes * 60
163         meditationTimeLeft = meditationTotalTime
164         isMeditating = true
165
166         // Start timer
167         meditationTimerJob = viewModelScope.launch {
168             while ( meditationTimeLeft > 0 && isMeditating ) {
169                 delay( timeMillis: 1000 )
170                 meditationTimeLeft -= 1
171             }
172             stopMeditation()
173         }
174
175         // Play sound (looped)
176         val context = getApplication<Application>().applicationContext
177         val resId = context.resources.getIdentifier(
178             name: "meditation$meditationSoundIndex", defType: "raw", context.packageName
179         )
180         meditationPlayer = MediaPlayer.create(context, resId)?.apply {
```

The status bar at the bottom shows the project path as `MedPal > app > src > main > java > com > example > medpal > ChatViewModel > initWebSocket`, and the bottom right corner shows the time as 124:86, CRLF, UTF-8, 4 spaces, and a file icon.



```
class ChatViewModel(application: Application) : AndroidViewModel(application) {

    @SuppressLint("Range")
    fun makePhoneCall(numberOrName: String) {
        val context = getApplication<Application>().applicationContext
        var phoneNumber = numberOrName

        // Check if input is not a valid phone number (simple regex, can be improved)
        if (!numberOrName.matches(Regex(pattern: "^(\\+?\\d{1,3}\\)?\\d{7,15}$")))) {
            // Query Contacts database for the name
            val resolver = context.contentResolver
            val cursor = resolver.query(
                ContactsContract.CommonDataKinds.Phone.CONTENT_URI,
                arrayOf(ContactsContract.CommonDataKinds.Phone.NUMBER),
                selection: "${(ContactsContract.CommonDataKinds.Phone.DISPLAY_NAME)} = ?",
                array: arrayOf(numberOrName),
                sortOrder: null
            )
            cursor?.use {
                if (it.moveToFirst()) {
                    phoneNumber = it.getString(it.getColumnIndex(ContactsContract.CommonDataKinds.Phone.NUMBER))
                } else {
                    // Handle contact not found
                    Log.e("PhoneCall", msg: "Contact $numberOrName not found")
                    return
                }
            }
        }

        val intent = Intent(Intent.ACTION_CALL).apply {
            data = Uri.parse("tel:$phoneNumber")
        }
    }
}
```

Testing

The app was tested on different Android devices to check latency, UI responsiveness, and LLM accuracy. After **multiple rounds of debugging**, the final UI was achieved.

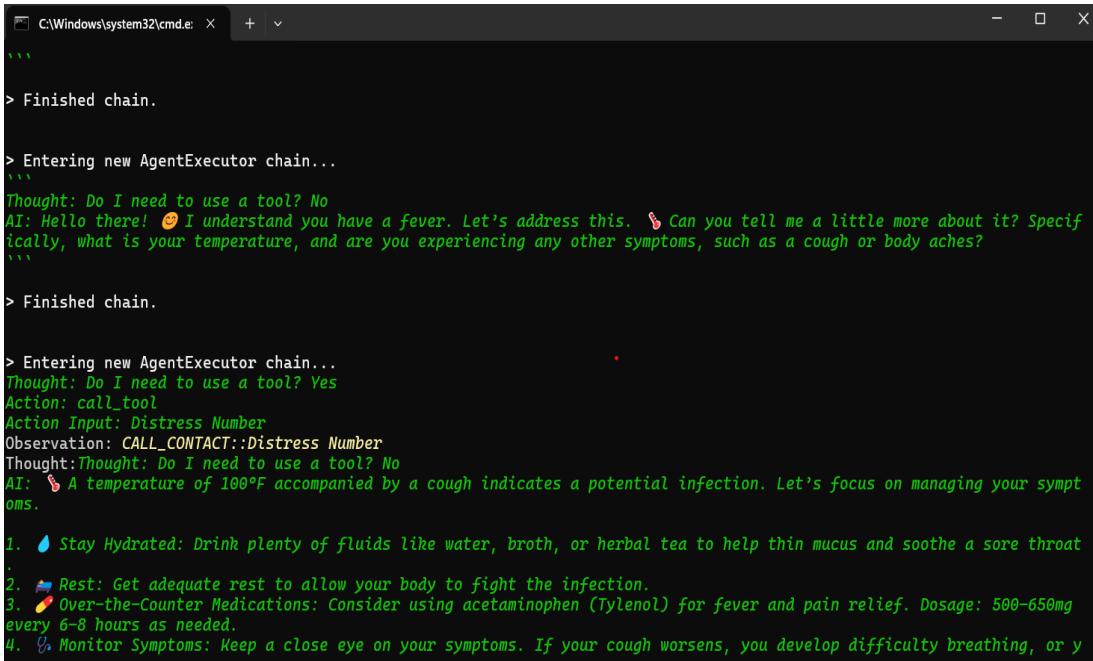
We tried different AI models of varying parameter count (1B, 2B) to get faster responses but the smaller the model, the more quality loss was encountered. Thus, we stuck with the 4B model, which seemed to generate a response within a minute or less. We faced slowdowns due to offline inference, especially on large inputs, but implemented optimization like prompt shortening and lazy loading to keep things functional.

Results

Working Features

The MedPal app was successfully developed with several core features working as intended. The symptom checker allowed users to describe their issues in natural language and receive helpful guidance, while the health report reader could extract text from both PDFs and images, then interpret the results for the user. A meditation tool was also integrated to support mental wellness through guided sessions. These features were powered by a tool-using LangChain agent, which enabled dynamic reasoning and multitool access for handling more complex tasks.

The user interface, inspired by WhatsApp, was designed to feel intuitive and friendly, making the app easy and familiar to interact with. All communication between the Android frontend and the FastAPI backend happened through Web Sockets for chat and HTTP POST requests for document uploads. While the app ran entirely offline—relying on a locally hosted LLM via Ollama—it still managed to respond accurately, though slightly slower during heavy tasks like PDF parsing.



```
C:\Windows\system32\cmd.exe: X + - □ ×

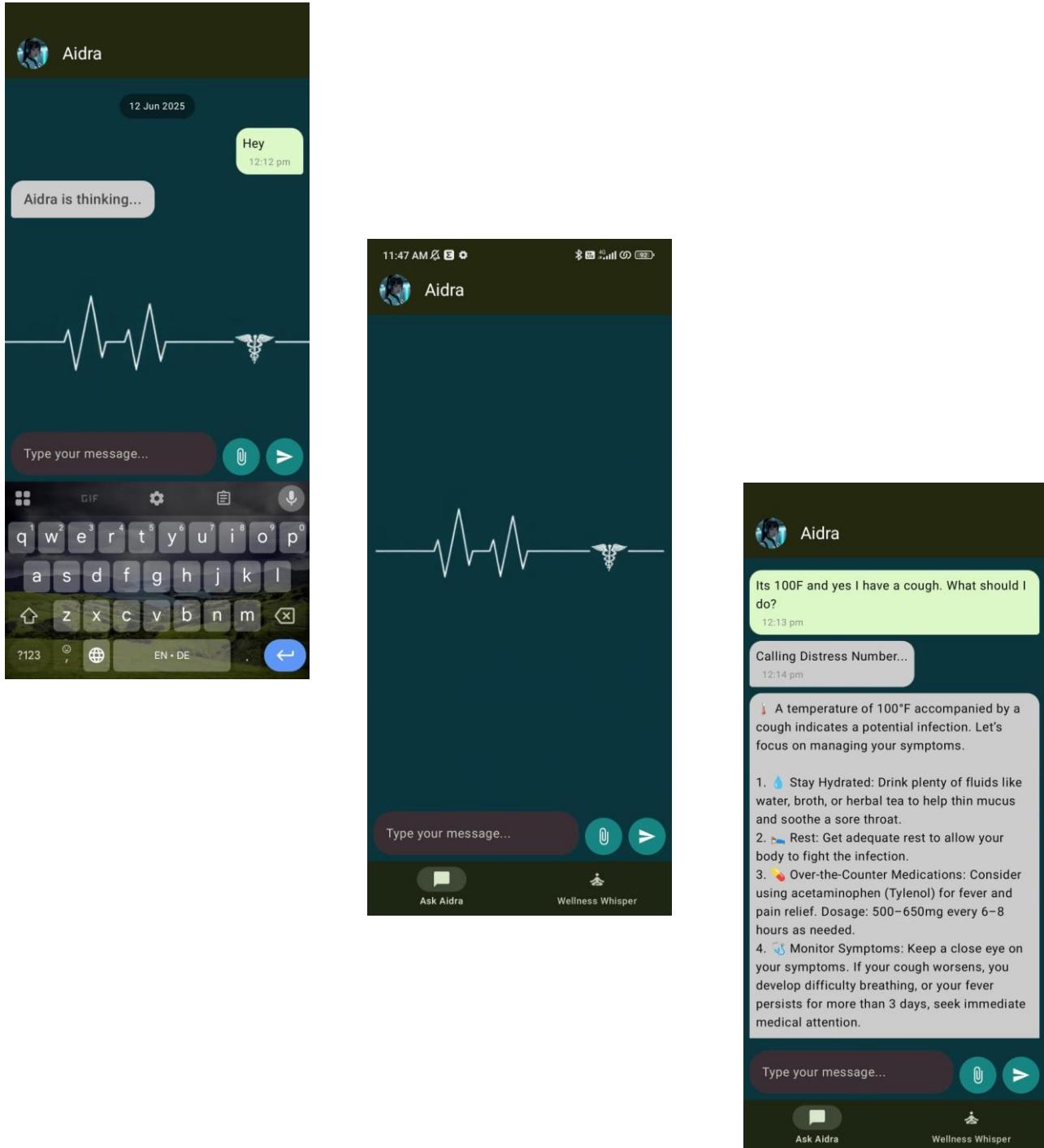
> Finished chain.

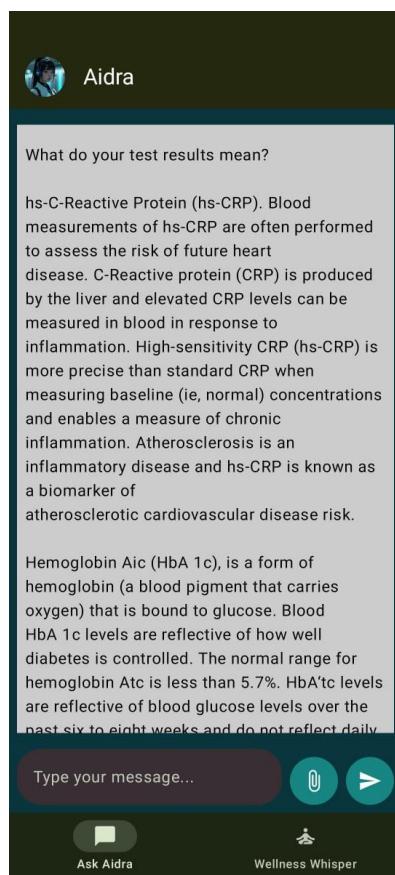
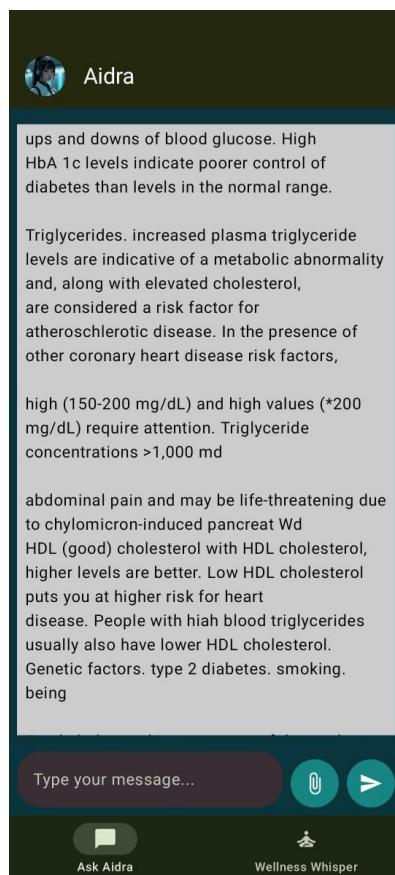
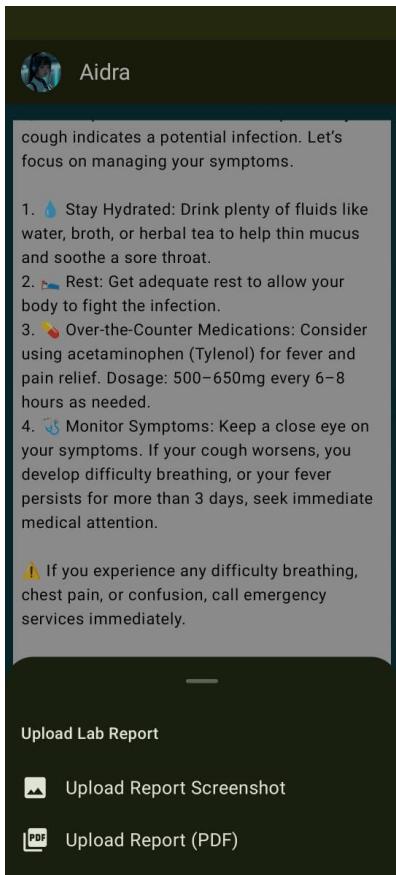
> Entering new AgentExecutor chain...
```
Thought: Do I need to use a tool? No
AI: Hello there! 😊 I understand you have a fever. Let's address this. 🌺 Can you tell me a little more about it? Specifically, what is your temperature, and are you experiencing any other symptoms, such as a cough or body aches?
```
> Finished chain.

> Entering new AgentExecutor chain...
```
Thought: Do I need to use a tool? Yes
Action: call_tool
Action Input: Distress Number
Observation: CALL_CONTACT::Distress Number
Thought: Do I need to use a tool? No
AI: 🌺 A temperature of 100°F accompanied by a cough indicates a potential infection. Let's focus on managing your symptoms.
1. 💧 Stay Hydrated: Drink plenty of fluids like water, broth, or herbal tea to help thin mucus and soothe a sore throat.
2. 🛌 Rest: Get adequate rest to allow your body to fight the infection.
3. 💊 Over-the-Counter Medications: Consider using acetaminophen (Tylenol) for fever and pain relief. Dosage: 500-650mg every 6-8 hours as needed.
4. 🌸 Monitor Symptoms: Keep a close eye on your symptoms. If your cough worsens, you develop difficulty breathing, or y
```

```

Screenshots have been included in the report to showcase the working features of the app. These include the chat interface, symptom checker, report analysis, tool definitions in code, and optional backend logs. Although features like appointment booking weren't implemented due to time limitations, the core concept and execution of the app were successfully demonstrated.





MedPal

- [Home](#)
- [Articles](#)
- [FAQ](#)

Your Guide to Health & Wellness

Explore articles and resources to help you live a healthier life.

Health Articles

- Eating for a Healthy Heart**
Learn about heart-healthy foods and simple recipes for a stronger heart.
- Benefits of Regular Exercise**
Discover how physical activity improves your health and reduces stress.
- Managing Stress Effectively**
Strategies for reducing stress and improving your mental and physical well-being.
- Understanding Type 2 Diabetes**
Get informed about causes, symptoms, and management of type 2 diabetes.
- Tips for Better Sleep**
Healthy sleep habits for a restful night and improved daily health.

Quote of the Day

"Take care of your body. It's the only place you have to live in."

Ask Aidra Wellness Whisper

12:42 PM Aidra

I feel sad today
12:30 pm

I understand you're feeling sad today. 😔 That's completely understandable, and it's good you're acknowledging it. Let's explore what might be contributing to this feeling.

Could you tell me a little more about what's making you feel sad? Perhaps sharing some details will help us find ways to feel better. ❤️
12:30 pm

I didn't score enough marks 😢
12:41 pm

Oh dear, I'm so sorry to hear that you're feeling down about your marks. It's incredibly frustrating when you don't achieve the results you hoped for. 😔 Let's address this.

It's important to remember that a single setback doesn't define your abilities or potential. Feeling disappointed is a natural reaction, but it's crucial to manage those feelings constructively.

Would you like to talk about what happened, or perhaps we can explore some strategies to help you feel more confident and motivated moving forward? ❤️ Let's focus on finding a path to success.
12:41 pm

Type your message...

Ask Aidra Wellness Whisper

12:42 PM Aidra

Could you tell me a little more about what's making you feel sad? Perhaps sharing some details will help us find ways to feel better. ❤️
12:30 pm

I didn't score enough marks 😢
12:41 pm

Oh dear, I'm so sorry to hear that you're feeling down about your marks. It's incredibly frustrating when you don't achieve the results you hoped for. 😔 Let's address this.

It's important to remember that a single setback doesn't define your abilities or potential. Feeling disappointed is a natural reaction, but it's crucial to manage those feelings constructively.

Would you like to talk about what happened, or perhaps we can explore some strategies to help you feel more confident and motivated moving forward? ❤️ Let's focus on finding a path to success.
12:41 pm

Type your message...

Ask Aidra Wellness Whisper

Future Scope

While the current version of MedPal demonstrates the core concept of a conversational health assistant, there's still a lot of potential for future improvements. One of the most anticipated upgrades is adding a **doctor appointment booking system**, which would allow users to schedule real consultations directly through the app—bridging the gap between AI guidance and professional care.

Another major enhancement would be **cloud-based model support** for faster response times, especially during heavy tasks like parsing large medical reports. Currently, the app runs completely offline, which is helpful for privacy, but also comes with limitations in speed and model complexity.

We also plan to add **user profile memory**, enabling the assistant to remember health history, medication preferences, and emotional patterns over time—making the app more personalized and emotionally intelligent.

Additionally, more **mental health features** such as journaling support, mood tracking, and daily check-ins could be included to strengthen the emotional support system of MedPal. With proper fine-tuning and medical validation, the assistant can also be improved to offer more accurate diagnostics and medication suggestions with even higher safety standards.

Finally, there's a possibility of building **wearable integrations**, such as syncing with smartwatches for real-time health data like heart rate, sleep, and steps—making MedPal not just a companion, but a full-time health buddy.

References

- WHO Regional Office for Europe. *Artificial Intelligence in Mental Health Research: Applications and Challenges*. World Health Organization, 6 Feb. 2023.
<https://www.who.int/europe/news/item/06-02-2023-artificial-intelligence-in-mental-health-research--new-who-study-on-applications-and-challenges>
- Google. *Gemma: A New Generation of Open Models*. <https://ai.google.dev/gemma>
- Zepp Health. *Zepp Aura - AI-Powered Wellness*. <https://www.zepp.com>
- HealthifyMe. *Your Personal AI Health and Fitness Coach*. <https://www.healthifyme.com>
- Android Developers. *Jetpack Compose*.
<https://developer.android.com/jetpack/compose>
- FastAPI. *FastAPI: The modern web framework for Python*. <https://fastapi.tiangolo.com>
- LangChain. *Framework for developing applications powered by language models*.
<https://www.langchain.com>
- Ollama. *Run open-source large language models locally*. <https://ollama.com>
- OpenAI. *ChatGPT (GPT-4 Turbo)*. Accessed via <https://chat.openai.com>
- Perplexity AI. *Perplexity: Ask Anything*. Accessed via <https://www.perplexity.ai>

Plagiarism Report

The final project report was analyzed using online plagiarism detection software (plagiarismsdetector.net). The analysis shows the document is **91% unique**, with only **9% matching existing content** and **0% partial matches**. The flagged content is from common formal acknowledgment statements that appear in many academic documents and does not impact the originality of the core report.

This confirms that the report is primarily original and self-authored. All references and supporting research used in the report have been properly cited in the References section.

- Overall Plagiarism Score: 9% (Acceptable)
- Unique Content: 91%
- Plagiarized Content: 9% (mostly formal acknowledgment phrases)

