Innovative Product Design Project Report

Project Title: MedPal – AI Powered Health Assistant App

Problem Statement: To design an AI-powered mobile health assistant app 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 (about the project its objective and significance)
4. Literature survey (includes background survey)
5. Problem statement
6. Methodology
7. Results
8. Future scope
9. References
10. Plagiarism report
11. (Atleast 20-25 pg)

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

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.

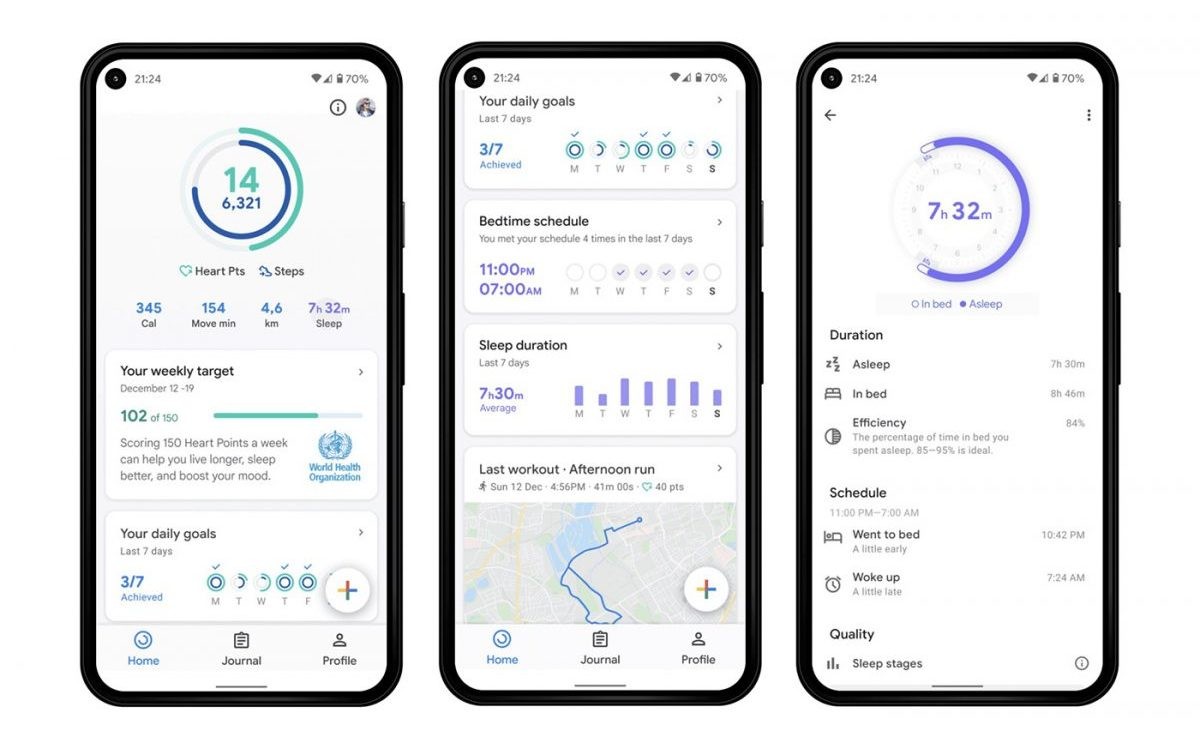
MedPal also behaves as a **caring, nurse-like assistant**, offering emotional support, which helps build a stronger relationship between the user and their well-being. The project displays how AI can be used not just for automating tasks, but also for creating **empathetic and meaningful experiences**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



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.



**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](https://www.who.int/europe/news/item/06-02-2023-artificial-intelligence-in-mental-health-research--new-who-study-on-applications-and-challenges)

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.



# Curso básico de python | Gallileu GenesisTechnology Stack

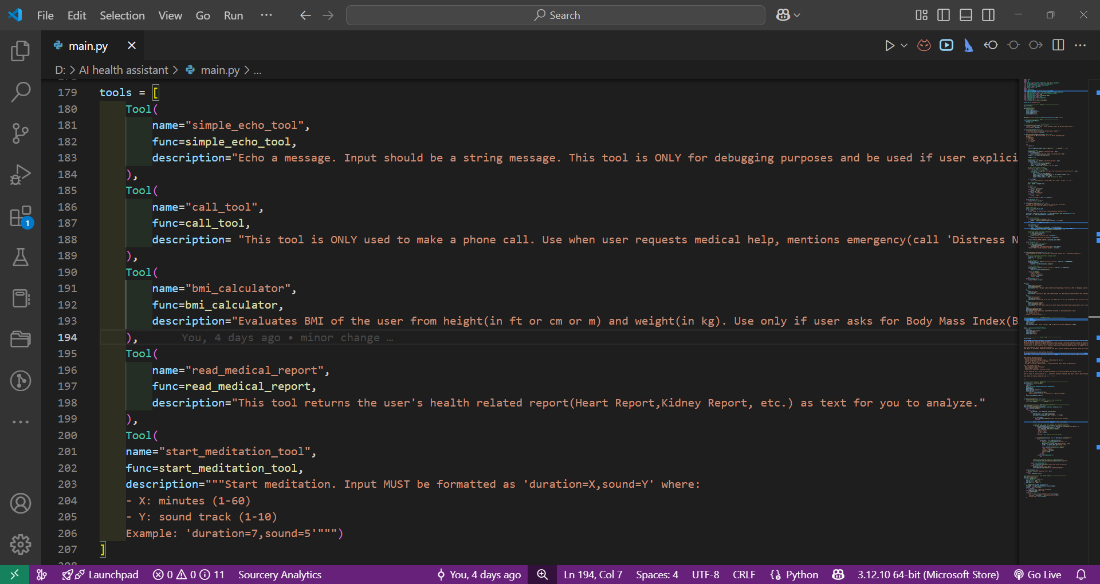
For the frontend, the app was developed using **Android Studio**, with **Kotlin** as the primary language and **Jetpack Compose** for building the user interface. This combination allowed for a modern, responsive, and visually appealing UI that feels smooth and intuitive to use. On the backend, we used **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.

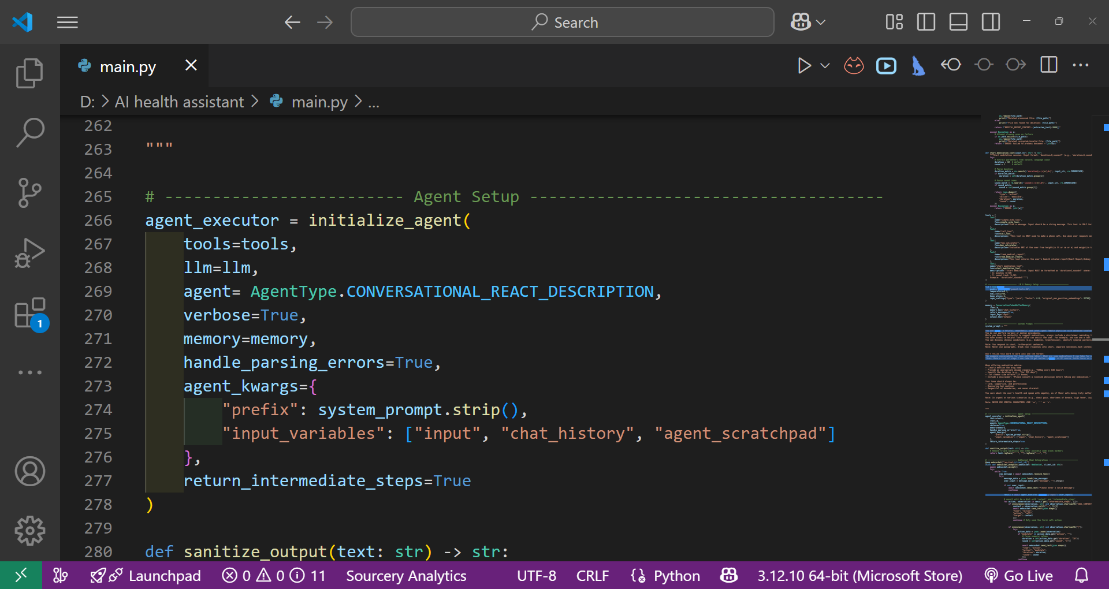


# Building the Backend

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.

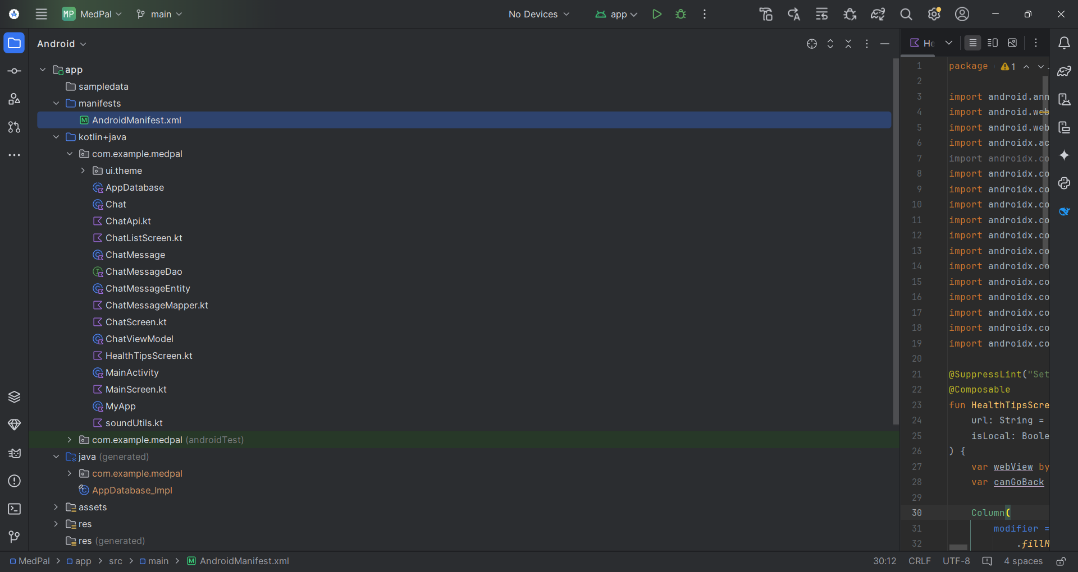
The LLM uses LangChain agent to dynamically respond to inputs, access tools, and keep track of context. This allows the app to handle complex tasks like interpreting medical PDFs or checking symptoms via dialogue.

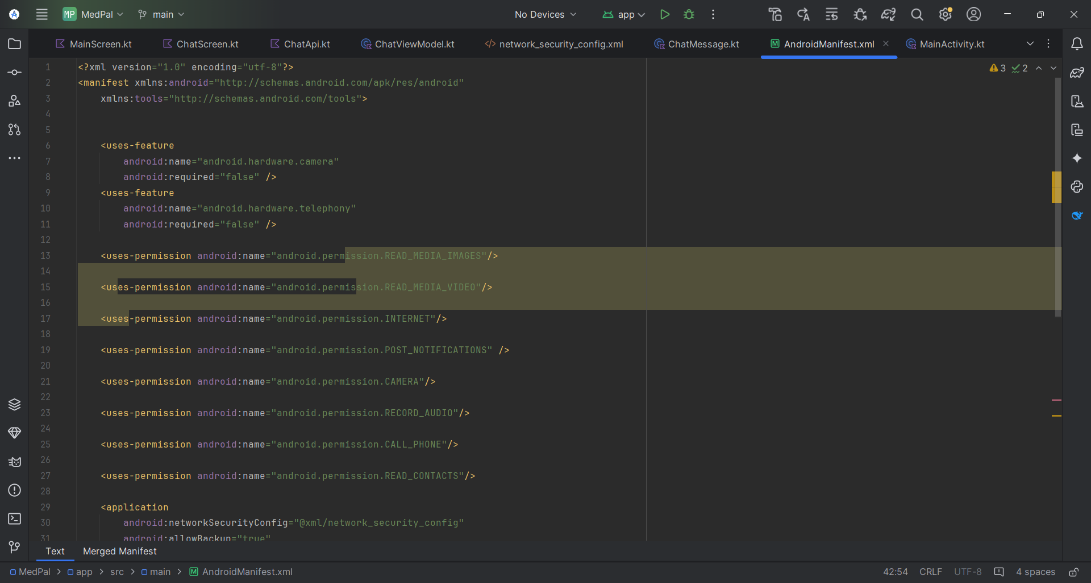




# Frontend Implementation

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.





# Tool Integration

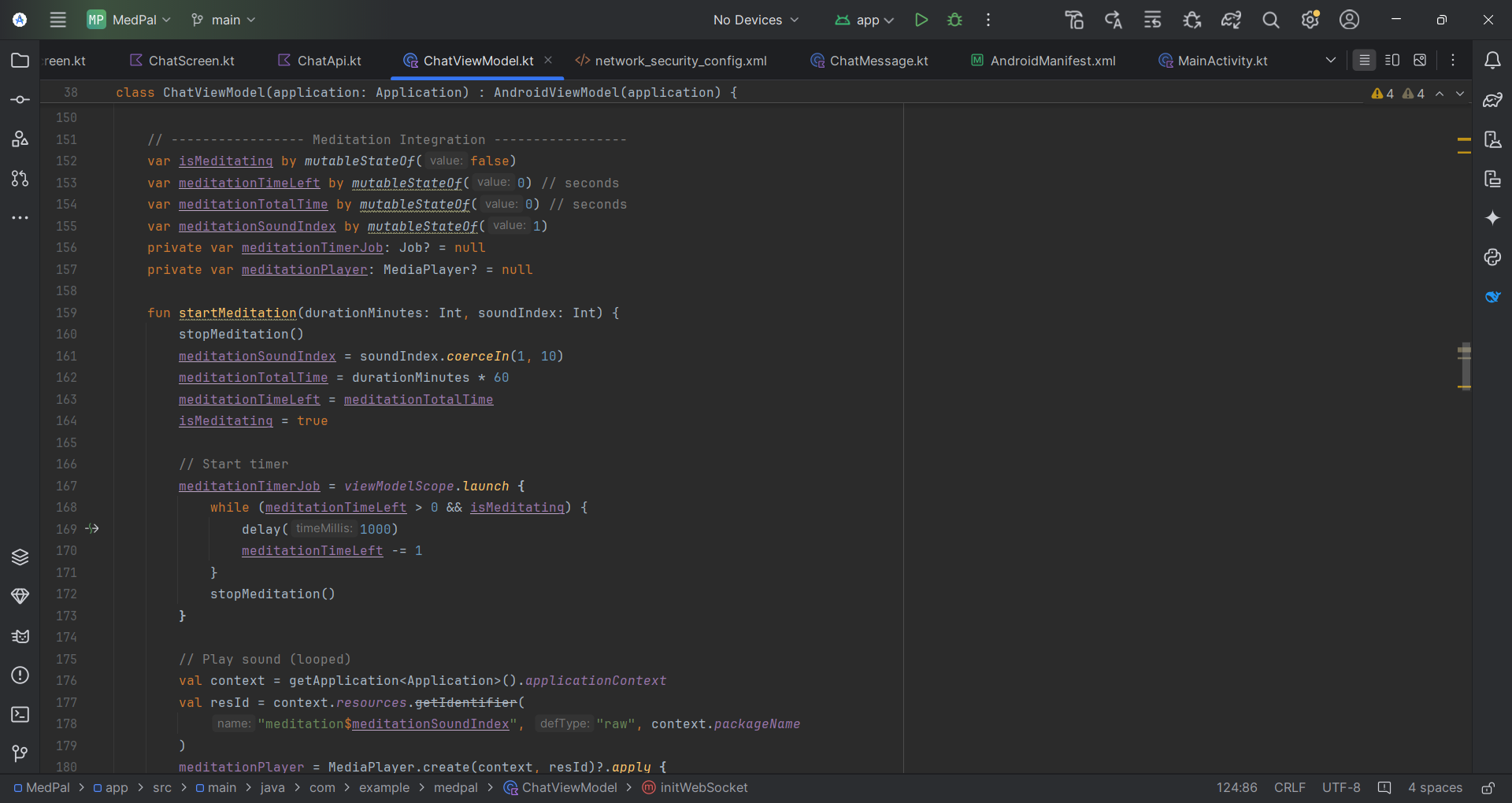
**Symptom Checker:**  
Users can describe how they feel, and the model infers possible non-serious illnesses based on trained prompt patterns.

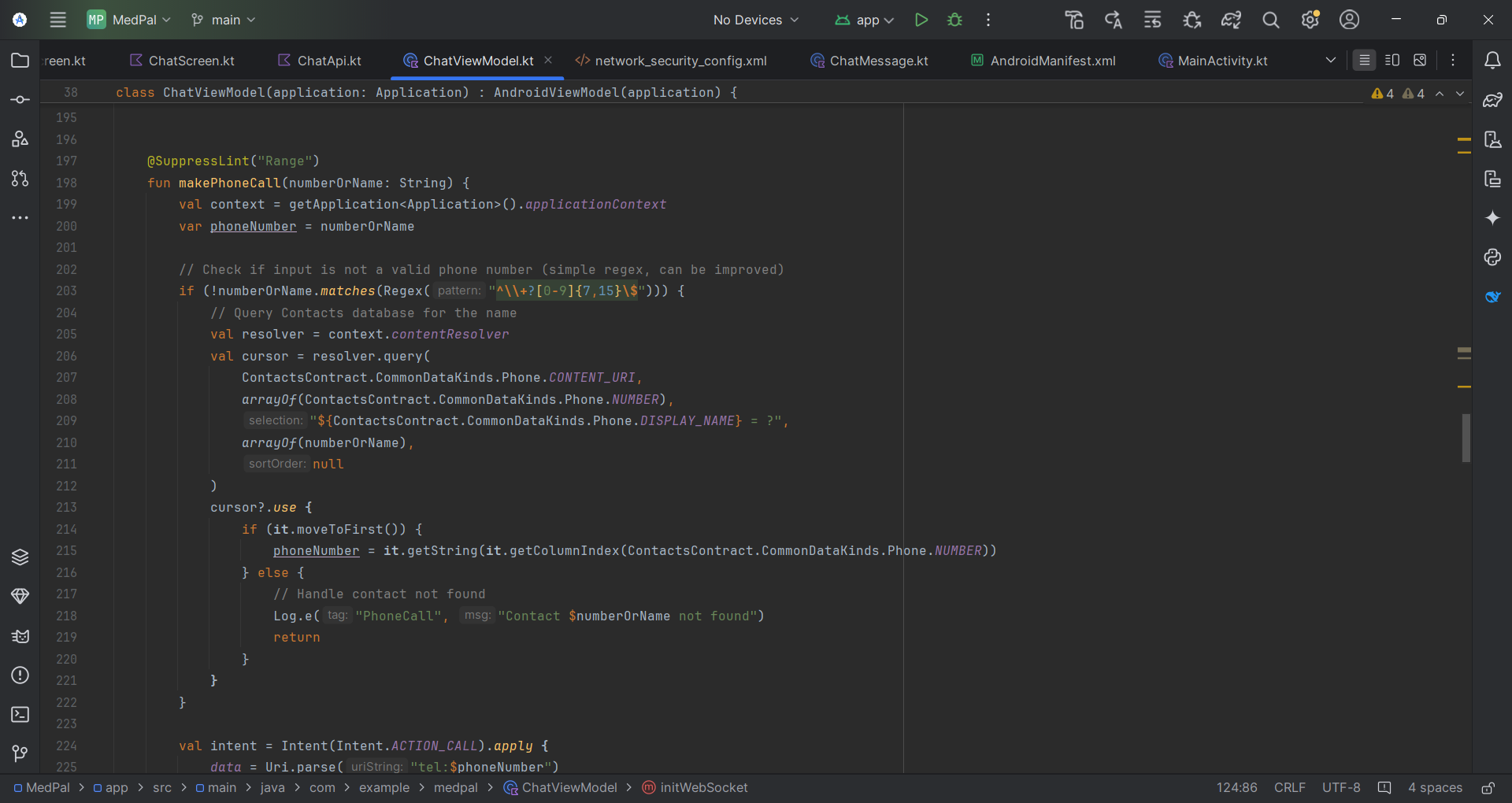
**Report Reader:**  
Uploaded PDFs or images with medical content are processed using OCR and summarized by the AI assistant.

**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”.





# 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.

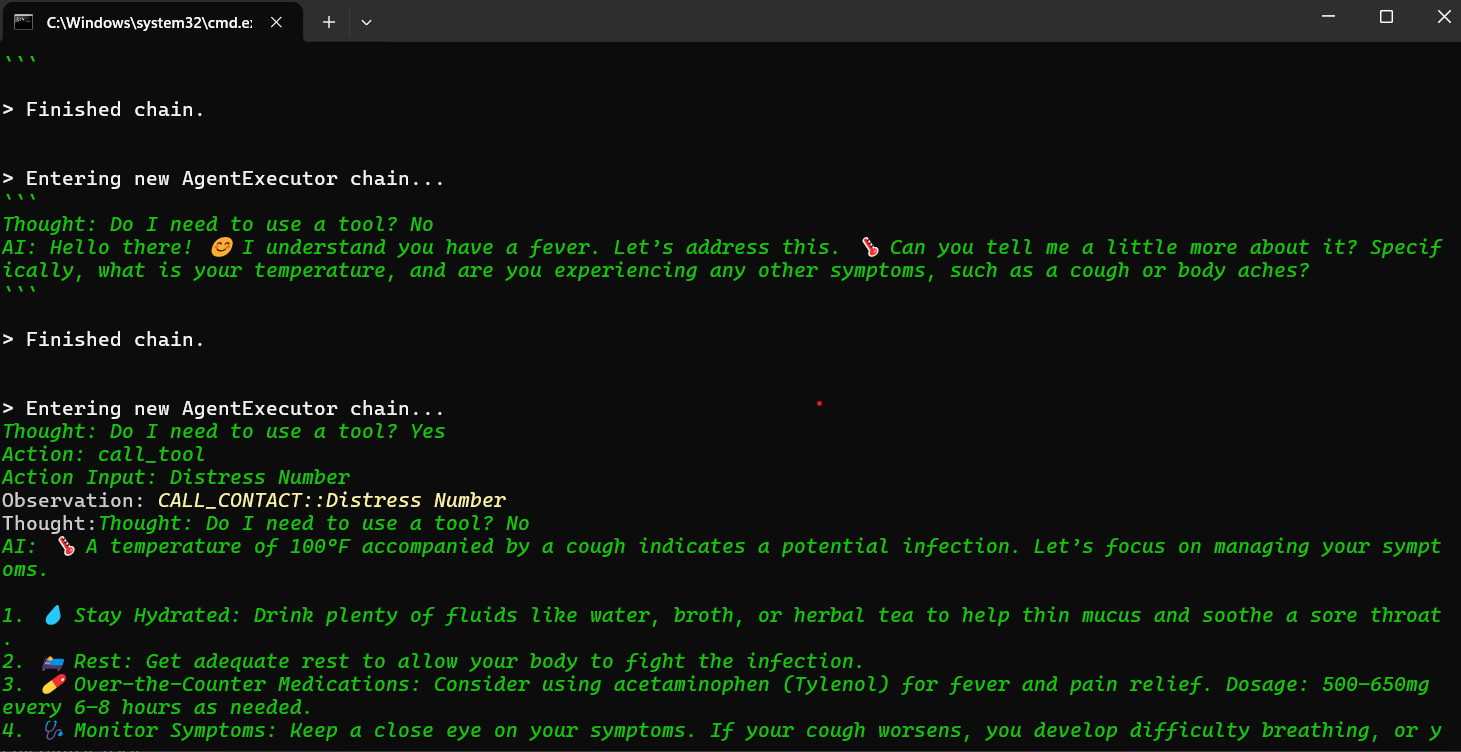
We tried different AI models of varying parameter count (1B, 2B) to get faster responses but the smaller the model, the more quality loss was encountered. Thus, we stuck with the 4B model, which seemed to generate a response within a minute or less.  
We faced slowdowns due to offline inference, especially 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.



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.



