MEDICAL CHATBOT FOR EFFICIENT PATIENT DATA MANAGEMENT

TEAM MEMBERS:

GARAPATI MANJU BHASHITA – 21BRS1517

CHINTAREDDY VARSHITHA – 21BRS1545

SADINENI VARUN KUMAR – 21BRS1540

GUIDE NAME:

DR. BENIL T

PROJECT STATEMENT:

Project Objective: Develop a Python-based medical chatbot with a Tkinter GUI to provide accurate and timely health-related information, first aid assistance, and symptom-based guidance, ensuring a seamless and intuitive user experience.

Key Features: Incorporates a comprehensive medical database, supports user-friendly interactions through text inputs, simplifies medical policy drafting and understanding, and provides personalized responses tailored to individual queries for better engagement.

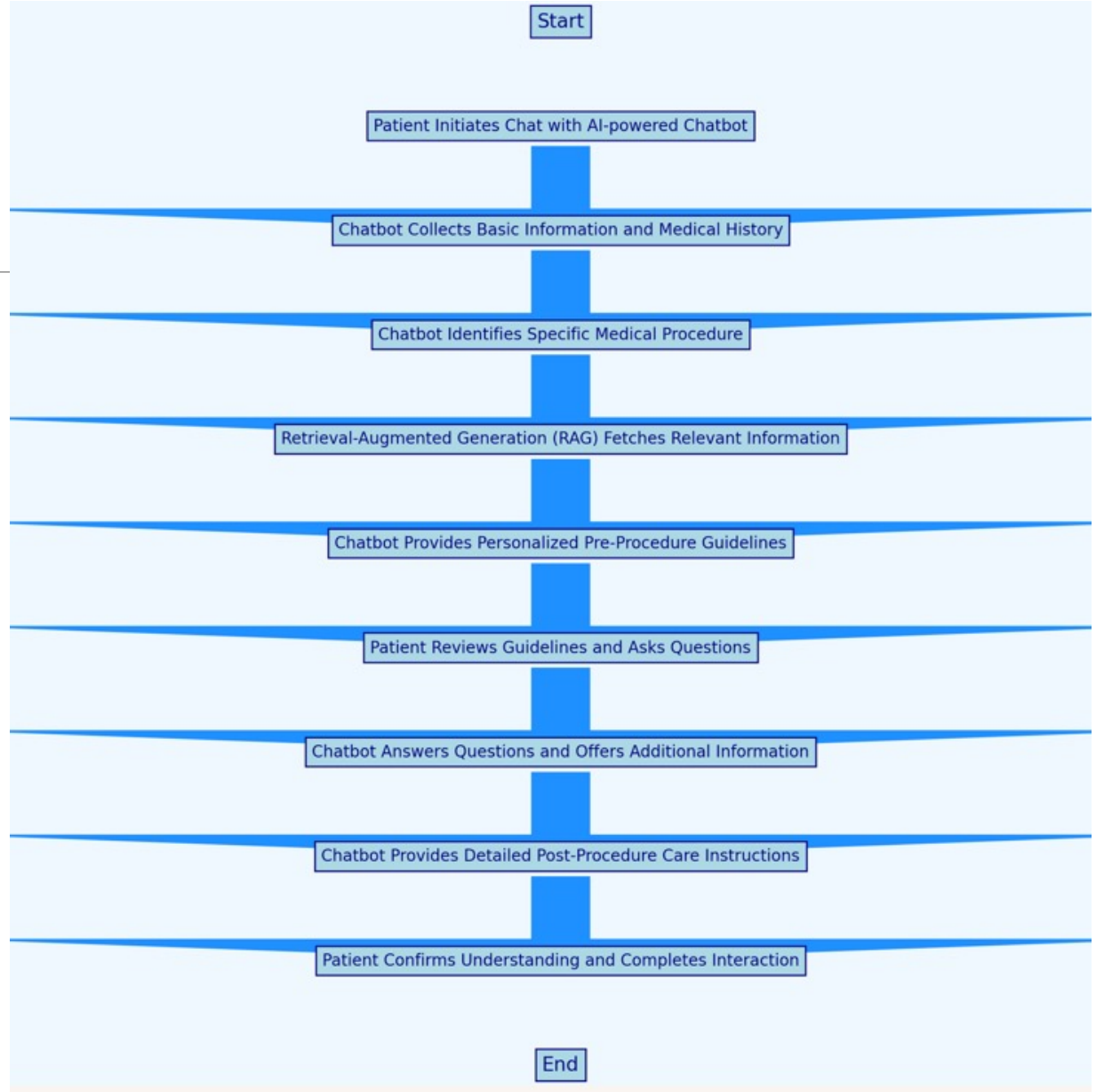
Purpose: Bridge gaps in healthcare accessibility, enhance health literacy, empower users with reliable, interactive health information, and contribute to the digital transformation of healthcare by complementing traditional medical services.

ABSTRACT:

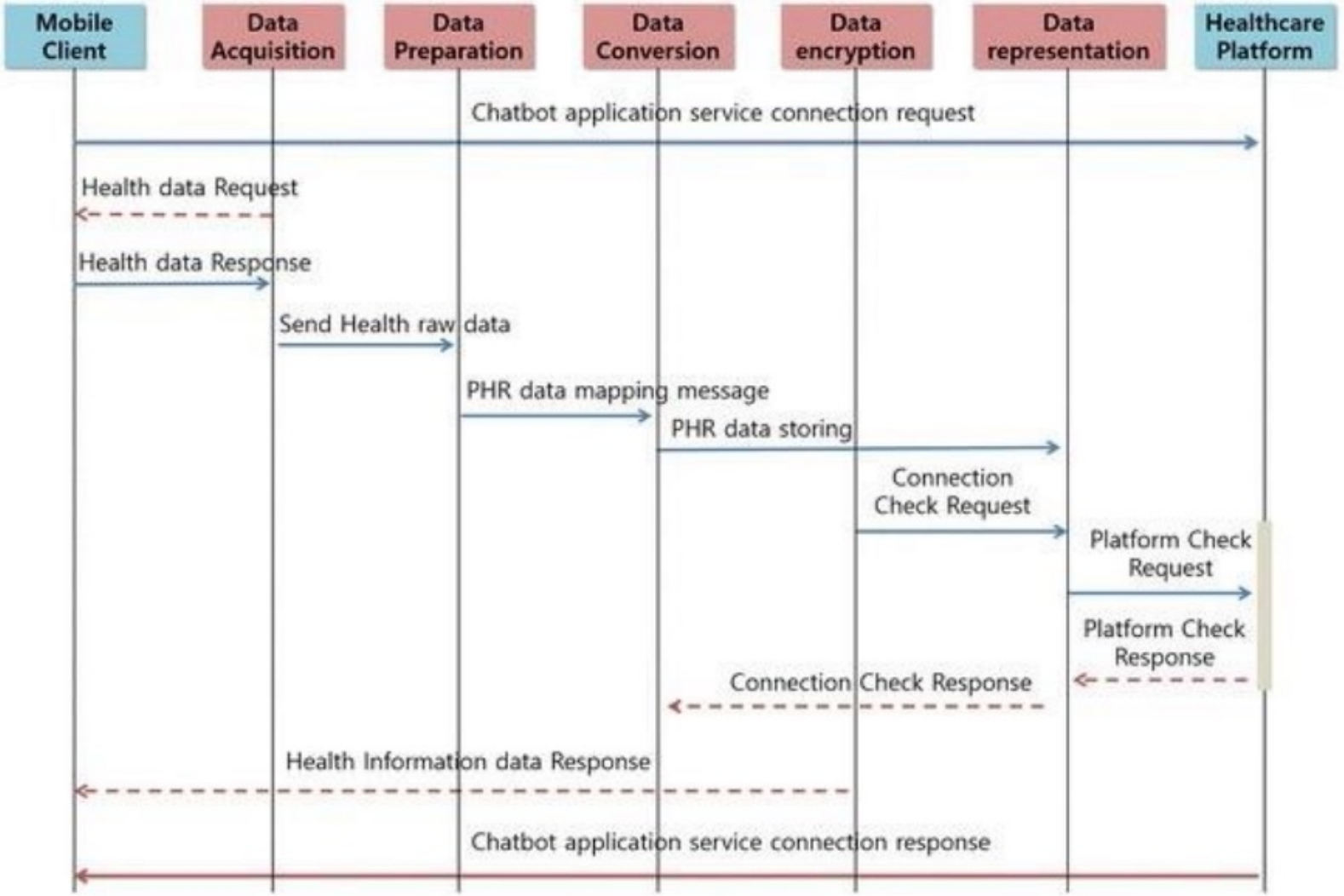
- The medical chatbot provides seamless and user-friendly access to medical information and first aid assistance, leveraging Python and a Tkinter GUI for enhanced functionality.
- Enables users to inquire about symptoms, gain insights into related diseases, and access detailed medication information through a comprehensive medical database.
- Supports both text and voice inputs, offering a dynamic interface that caters to diverse user preferences and technological comfort levels.
- Assists users in drafting medical policies by providing step-by-step instructions and integrating external resources, adding practicality to its scope.
- Serves as a supplementary tool for health-related queries while emphasizing the need for professional medical advice for critical health decisions.
- Demonstrates the potential of chatbots to enhance health literacy, improve access to health information, and contribute to the ongoing digital transformation of the healthcare industry.

PROPOSED WORK:

- Create a Python-based chatbot with a Tkinter GUI to provide users with reliable medical information, symptom analysis, and first aid guidance in a user-friendly format.
- Integrate a robust database to offer detailed insights into symptoms, potential diseases, and medication recommendations, ensuring accurate and prompt responses.
- Implement a feature to guide users through drafting and understanding medical policies, simplifying complex processes with step-by-step assistance.

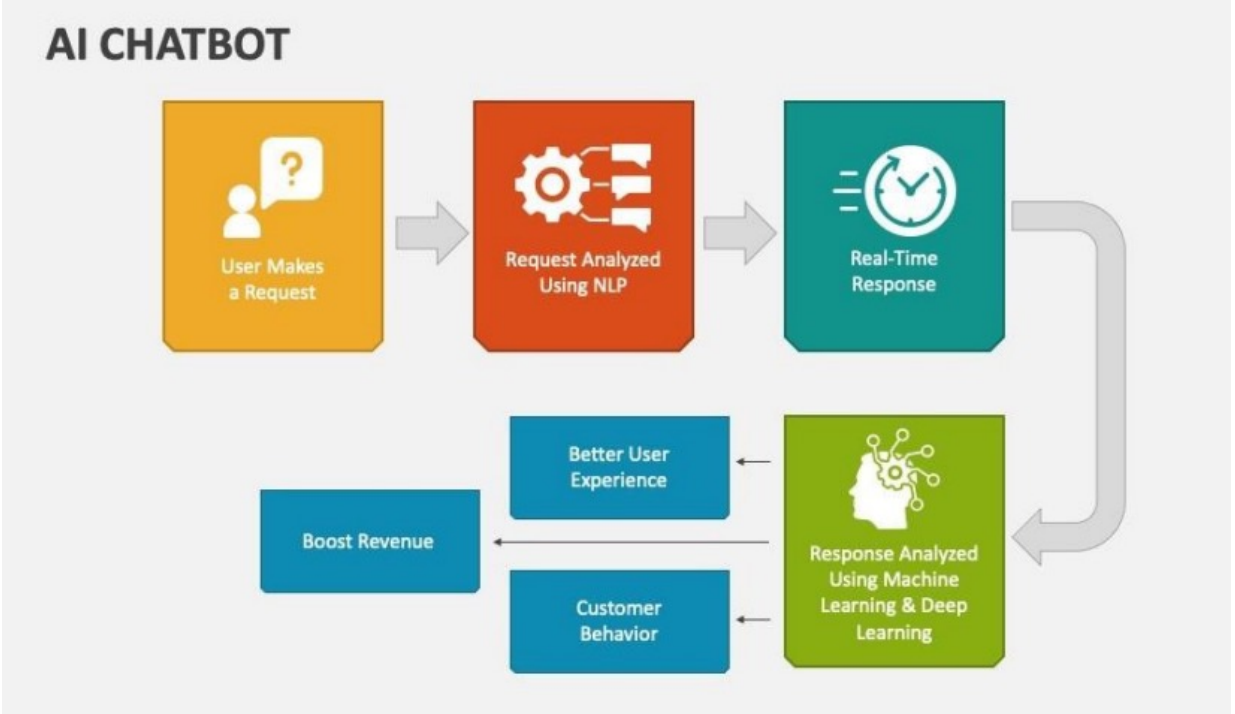


ARCHITECTURE:



CONCEPTS USED:

- Tkinter Library for GUI Development
- Dictionaries for Data Storage
- Exception Handling
- Libraries - -Tkinter, Web browser, Random
- String Manipulation
- Conditional statements and loops

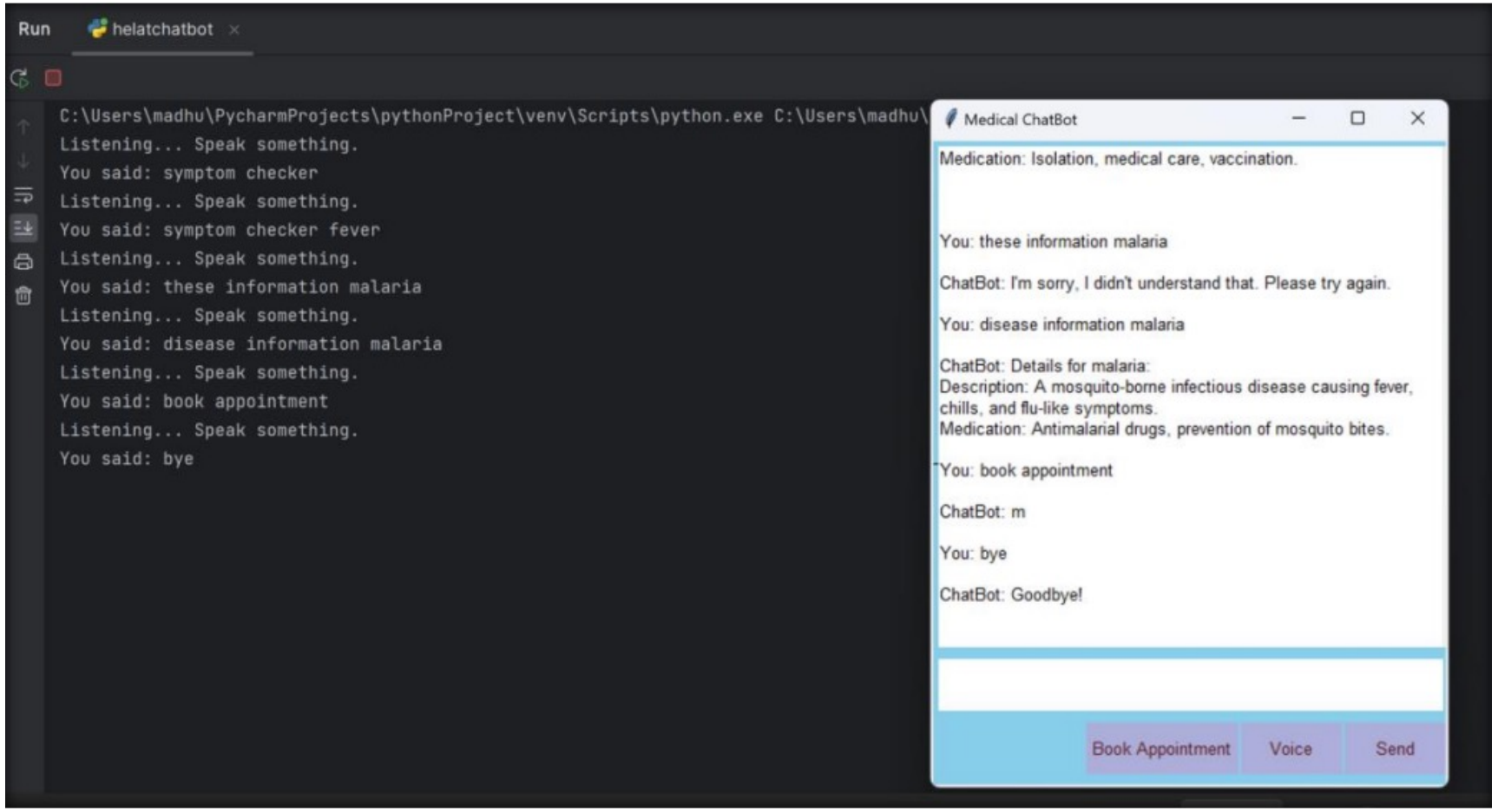


CODE EXPLANATION:

- Functionality: This is a medical chatbot application built using Tkinter for GUI, which allows text and voice-based interactions to provide medical advice based on symptoms or disease queries.
- Predefined Responses: A responses dictionary contains default replies for specific keywords like "hi," "bye," and "how are you." It also supports queries like "symptom checker," "book appointment," and "disease information.“
- Symptom and Disease Databases: The code maintains symptom_db and disease_db dictionaries, which store information about symptoms, related diseases, and recommended medications. These are used to provide meaningful responses to user queries.
- Symptom Checker: The function check_symptoms analyzes a user's input symptoms, matches them with the database, and returns related diseases, descriptions, and medications.
- Voice Input Support: Using the speech_recognition library, the function send_voice_message captures and processes spoken input, converting it to text for interaction with the chatbot.

- Appointment Booking: Clicking the "Book Appointment" button opens a predefined URL (open_appointment_website function) in the browser, demonstrating integration with external resources.
- GUI Design with Tkinter: The chatbot's interface is created using Tkinter components such as Canvas for the gradient background, Text for chat logs, and `Entry` for user input and Buttons are included for sending messages, voice input, and booking appointments.
- Real-Time Chat Display: The send_text_message function handles user input, processes it against predefined responses or database queries, and dynamically updates the chat_log widget with messages from both the user and the chatbot.
- Responsive Components: The app binds the Return key to the send_message function, enabling users to press Enter to send messages. It also uses a scroll bar for the chat log to manage large conversations.
- Gradient Background & Styling: A visual enhancement is added with a gradient background on the Tkinter canvas and custom button colors for a more appealing user interface.

OUTPUT SCREENSHOT:




Coverage And Claims
Checking Symptoms
Providing Medical Assistance
Collecting Patient Information
Instant Response to Queries
Easily Scalable Working Hours
Automated Presciption Refills
Easy Scheduling of Appointments
Recommendations for Wellness Programs
Providing Necessary Medical Information

CONCLUSION:

In conclusion, the medical chatbot represents a significant innovation in the healthcare sector, combining technology and accessibility to deliver timely, reliable, and user-friendly health information.

By offering guidance on symptoms, diseases, medication, and medical policy drafting, the chatbot addresses critical gaps in healthcare accessibility and literacy. While it serves as a valuable supplementary tool for users, it emphasizes the importance of professional medical consultation for comprehensive care.

This project highlights the transformative potential of digital solutions in enhancing health services and empowering individuals, marking a pivotal step in the ongoing digital transformation of healthcare.

THANK YOU