**PROJECT DESIGN**

**4.1 Problem–Solution Fit**

Understanding the “problem-solution fit” is essential before any implementation begins. This section explores how the HEALTH-AI project is designed to address the specific pain points of the user.

**🎯 The Core Problem:**

Users often struggle with:

* Access to affordable medical consultation.
* The complexity and unreliability of online medical content.
* Lack of simple, trusted explanations for symptoms.
* Difficulty interpreting clinical terms or treatment options.

**💡 Our Solution:**

HEALTH-AI addresses these challenges by:

* Offering a **chat-based interface** that simulates a health assistant.
* Providing responses using **IBM Granite**, a large language model fine-tuned for instruction-following.
* Responding to **symptom queries** with:
  + Probable **disease names**
  + **Natural remedies**
  + Suggested **treatment plans**
  + A **bar chart** of disease likelihood
* Running on **Python code**, deployed via Google colab using Hugging Face APIs.

Thus, the solution is **conversational**, **intelligent**, **visual**, and **non-invasive**, ideal for users seeking guidance without clinical consultation.

**4.2 Proposed Solution**

**🧠 Functional Overview:**

HEALTH-AI is a Python-based system that accepts symptom-related inputs from users and produces intelligent, contextual answers using an AI model. The following are the key components of the proposed solution:

**🔍 Key Features:**

| **Feature** | **Description** |
| --- | --- |
| **Chat Input Handler** | Accepts natural language input |
| **LLM Query Module** | Sends input to IBM Granite via Hugging Face API |
| **Response Parser** | Extracts relevant sections (disease, treatment, remedies) |
| **Visualization Generator** | Converts prediction confidence into a **bar chart** using matplotlib |
| **Context Management** | Maintains recent conversation context |
| **Disclaimer System** | Appends advisory that this is not a professional diagnosis |

**📊 Sample Output Example:**

User Input: I have a sore throat and headache.

AI Response:

Possible diseases:

- Common Cold (40%)

- Sinus Infection (35%)

- Influenza (25%)

Natural Remedies:

- Warm salt water gargles

- Honey + lemon tea

- Stay hydrated

**4.3 Solution Architecture**

Below is the **logical architecture** of the HEALTH-AI solution.

**🏗️ Solution Layers:**

1. **User Interface Layer**
   * Medium: Google Colab (or web front-end)
   * Function: Accepts user inputs in chat format
2. **Application Logic Layer**
   * Written in Python
   * Handles query formatting, model invocation, and response parsing
3. **Model Inference Layer**
   * IBM Granite model accessed via Hugging Face Inference API
   * Returns health-related answers
4. **Visualization Layer**
   * Uses matplotlib to generate disease probability charts
5. **Output Presentation Layer**
   * Returns combined chart + response to user in a conversational format

🖼️ **Suggested Architecture Diagram**

**[user input]**

↓

[Symptom Prompt]

↓

[IBM Granite Model via API/HF]

↓

[AI-generated Response]

↓

[Chat Output Displayed to User]

**✅ Summary**

* The **problem-solution fit** confirms that HEALTH-AI directly addresses the user need for accessible, conversational health support.
* The **proposed solution** outlines all components and their integration.
* The **architecture design** shows how the input flows through various layers, using generative AI and Python to deliver intelligent results.