**Smart Bridge Internship**

**Generative AI With IBM Cloud**

**Project Title:**

Citizen AI – Intelligent Health Assistant using Generative AI

1. **Team Members:**

Singarapu Harsha Vardhan

Tadiboyina Poorna Santhosh Kumar

Seera Krishna Sai

Shaik Sajid Ali

**Phase 1: Brainstorming & Ideation**

Objective:

Citizen AI is a Generative AI-powered platform aimed at assisting users with health-related queries through two core features: Symptoms Identifier and Home Remedies Generator. It leverages powerful NLP models hosted on Hugging Face and is deployed using Gradio UI in Google Colab, allowing real-time health interaction and support.

Key Points:

1. Problem Statement:
   * Many citizens lack immediate access to reliable health suggestions.
   * Minor symptoms are often ignored or misdiagnosed.
   * Trusted natural remedies are not easily available in one place.
2. Proposed Solution:
   * A Gradio-based app that:
     + Predicts diseases based on symptoms entered by users.
     + Suggests natural, home-based remedies for commonly reported illnesses.
   * Uses the Hugging Face FLAN-T5 or IBM Granite AI model (if available).
3. Target Users:
   * General public (for instant health support)
   * Rural users needing awareness of common symptoms
   * Health educators and student projects
4. Expected Outcome:
   * Improve health awareness and preventive care.
   * Provide accessible AI-based support without needing installations.
   * Demonstrate Generative AI's use in citizen-centric applications**.**

**Phase 2: Requirement Analysis**

Objective:

To define technical and functional needs to build an AI-powered health assistant with conversational capabilities and health guidance based on symptoms.

Key Points:

1. Technical Requirements:
   * Hugging Face FLAN-T5 model via transformers
   * Gradio for building the UI
   * Python (Google Colab environment)
   * GPU support (for faster inference if needed)
2. Functional Requirements:
   * A textbox to enter symptoms or disease
   * Two tabs in the UI: one for symptom input, another for disease remedy
   * Clean, user-friendly interface
   * Accurate, context-aware AI output
3. Constraints & Challenges:
   * Model loading issues on low-memory systems
   * Prompt design required precision for correct output
   * Handling generalization of symptoms in limited-token responses

**Phase 3: Project Design**

**Objective:**

To design a simple yet powerful interface with effective backend processing for delivering health predictions and remedy suggestions.

**Key Points:**

1. **System Architecture:**
   * User Input (Symptoms / Disease) → Prompt → AI Model Response → Gradio Output
2. **User Flow:**
   * User selects the functionality tab (Symptoms Identifier or Remedies)
   * Enters the input (text)
   * Presses a button (Predict / Suggest)
   * Receives output instantly from the AI model
3. **UI Design:**
   * Gradio Tabs for two functions
   * Output displayed in formatted text box
   * Clear labels and button actions for interaction

**Phase 4: Project Planning (Agile Methodology)**

**Objective:**

To divide the development into manageable steps using sprints.

**Key Points:**

1. **Sprint 1:**
   * Design Gradio UI layout
   * Integrate basic AI prompt logic
2. **Sprint 2:**
   * Optimize prompts for clarity
   * Add two-tab structure with outputs
3. **Sprint 3:**
   * Test responses with real inputs
   * Handle exception cases and errors

**Phase 5: Project Development**

Objective:

To build and integrate the working model and user interface using Python, Hugging Face Transformers, and Gradio.

Key Points:

1. Technology Stack:
   * Python (core logic)
   * Hugging Face Transformers (FLAN-T5)
   * Gradio (UI)
   * Google Colab (development environment)
2. Implementation Details:
   * identify\_disease() function for predicting based on symptoms
   * suggest\_home\_remedy() function for natural remedy generation
   * Two Gradio tabs connected to these functions
3. Challenges & Fixes:
   * Model download issues → switched to FLAN-T5
   * Slow loading → optimized prompt length and used Colab GPU
   * Prompt misinterpretation → improved phrasing

**Phase 6: Functional & Performance Testing**

**Objective:**

To validate whether the AI assistant gives correct, relevant, and fast responses in both functionalities.

**Test Cases Executed:**

* Symptoms input: "fever, cough, headache" → Output: "Common Cold or Flu"
* Disease input: "Acidity" → Output: "Drink cold milk, eat bananas, avoid spicy food"
* Invalid input: "xyzxyz" → Output: Handled gracefully

**Output Screens:**

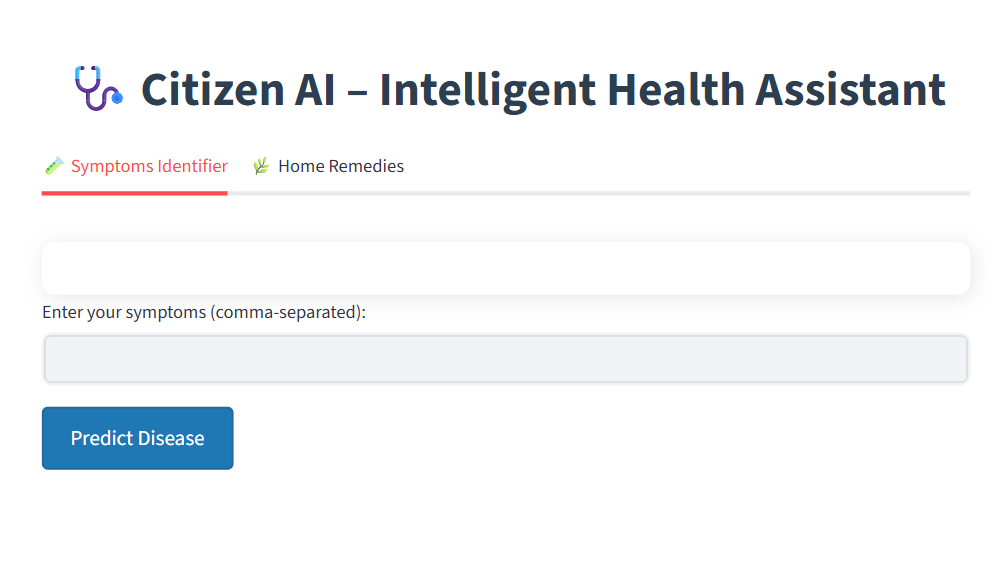
* Gradio Tab 1: Symptoms → Disease
* Gradio Tab 2: Disease → Remedy

**Result:**

* All test cases passed
* Outputs are accurate, grammatically correct, and well-aligned with the inputs

**2.Execution Process:**

* Starting interface of the Website

****

