HEALTH AI: INTELLINGENT HEALTHCARE ASSISTANT

**Project Documentation**

1. **Introduction:**

* Project Title: HealthAI-Intellingent Healthcare Assistant.
* Team Member: Karthika A
* Team Member: Harini A
* Team Member: Harini R
* Team Member: Jayasree G

1. **Project Overview:**

**Purpose:**

The Medical AI Assistant is designed to provide users with preliminary health insights based on their symptoms and medical history. By leveraging IBM Granite LLM models and a Gradio-based interactive interface, the assistant helps generate possible disease predictions and personalized treatment plans. While it is not a substitute for professional medical advice, it empowers individuals to seek timely consultations and gain awareness of their health conditions.

**Features:**

**Disease Prediction:**

* In Diesease Prediction we can know the Symptom-based condition suggestions
* Users enter symptoms and receive possible conditions with general recommendations.

**Treatment Plan Generation:**

* In Treatment Plan Generation we get some Personalized guidance.
* Provides a treatment plan including home remedies and general medication guidelines based on age, gender, and medical history.

1. **Architecture:**

**Frontend (Gradio UI):**

* Built using Gradio Blocks.
* Provides a tabbed interface with input fields for symptoms, patient details, and medical history.
* Displays outputs in structured text boxes for readability.

**Backend (PyTorch + Transformers):**

* Uses IBM Granite LLM (ibm-granite/granite-3.2-2b-instruct) for text generation.
* Tokenization handled by Hugging Face AutoTokenizer.
* Model optimized for GPU with torch.float16 and device\_map="auto" when CUDA is available.

1. **Setup Instructions:**

**Prerequisites:**

* Python 3.9 or later
* pip and virtual environment tools
* Installed libraries:(pip install gradio torch transformers)
* Internet access to download the IBM Granite model

**Installation Process**:

* Clone or copy the project files into your local machine.
* Install dependencies using the above pip command.
* Run the Python script:(python medical\_ai\_assistant.py)
* The Gradio interface will launch locally. Copy the provided URL into a web browser

1. **Folder Structure:**

* Since the project is run entirely in google colab,there is only one main file used that is ,smartSDLC.ipynb.
* Structure:

project/

└─- smartSDLC.ipynb

1. **Running the Application:**

* Launch the script using Python.
* Access the Gradio interface through the generated local/temporary share URL.
* Navigate between Disease Prediction and Treatment Plan tabs.
* Enter input data (symptoms, condition, medical history, etc.).
* View AI-generated suggestions instantly.

1. **API Documentation:**

* generate\_response(prompt, max\_length=1024)

Inputs: Prompt (string)

Outputs: Model-generated response (string)

* disease\_prediction(symptoms)

Inputs: User-entered symptoms.

Outputs: List of possible conditions & general recommendations.

* treatment\_plan(condition, age, gender, medical\_history)

Inputs: Patient details (condition, age, gender, history)

Outputs: Personalized treatment plan with home remedies & medication guidelines.

1. **Authentication:**

* Current version is open for demonstration purposes.
* Secure deployment options:
* Token-based authentication (JWT/API keys)
* Role-based access (e.g., doctor, patient, researcher)
* User session logging for medical data privacy (planned feature)

1. **User Interface:**

**Tabs:**

* Disease Prediction
* Treatment Plans

**Input Fields:**

* Symptoms, condition, medical history.
* Age
* Gender

**Outputs:**

* In Disease Prediction: AI will generate possible medical conditions along with general suggestion.
* In Treatment Plans: The system will generate a personalized treatment suggestion including home remedies and over the counter medications.

1. **Testing:**

Testing was done in multiple phases.

**Unit Testing:** Tested model loading, tokenizer initialization, and function responses.

**Functional Testing:**Checked disease prediction with different symptom inputs.

Validated treatment plans with multiple patient profiles.

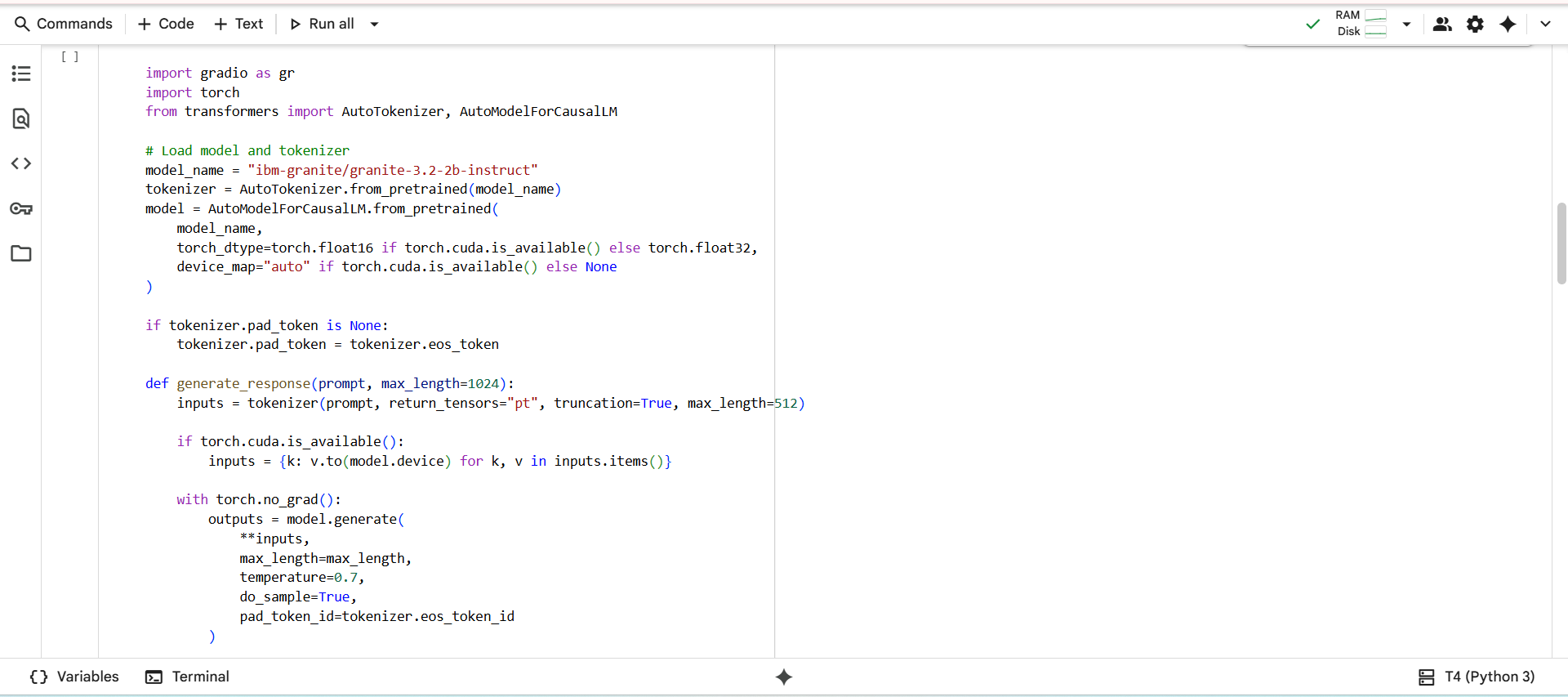
**Edge Case Handling**:Empty input fields

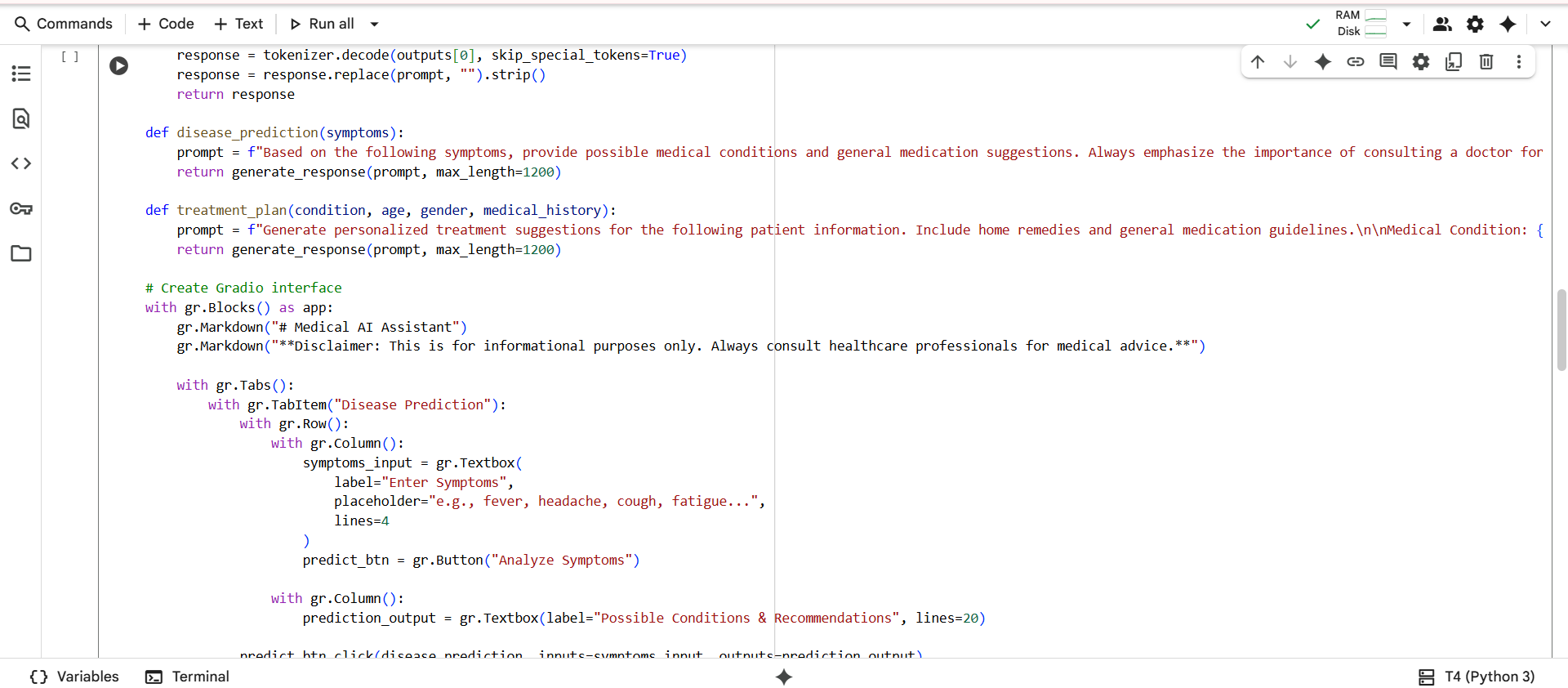
Extremely long symptom descriptions

Unsupported characters

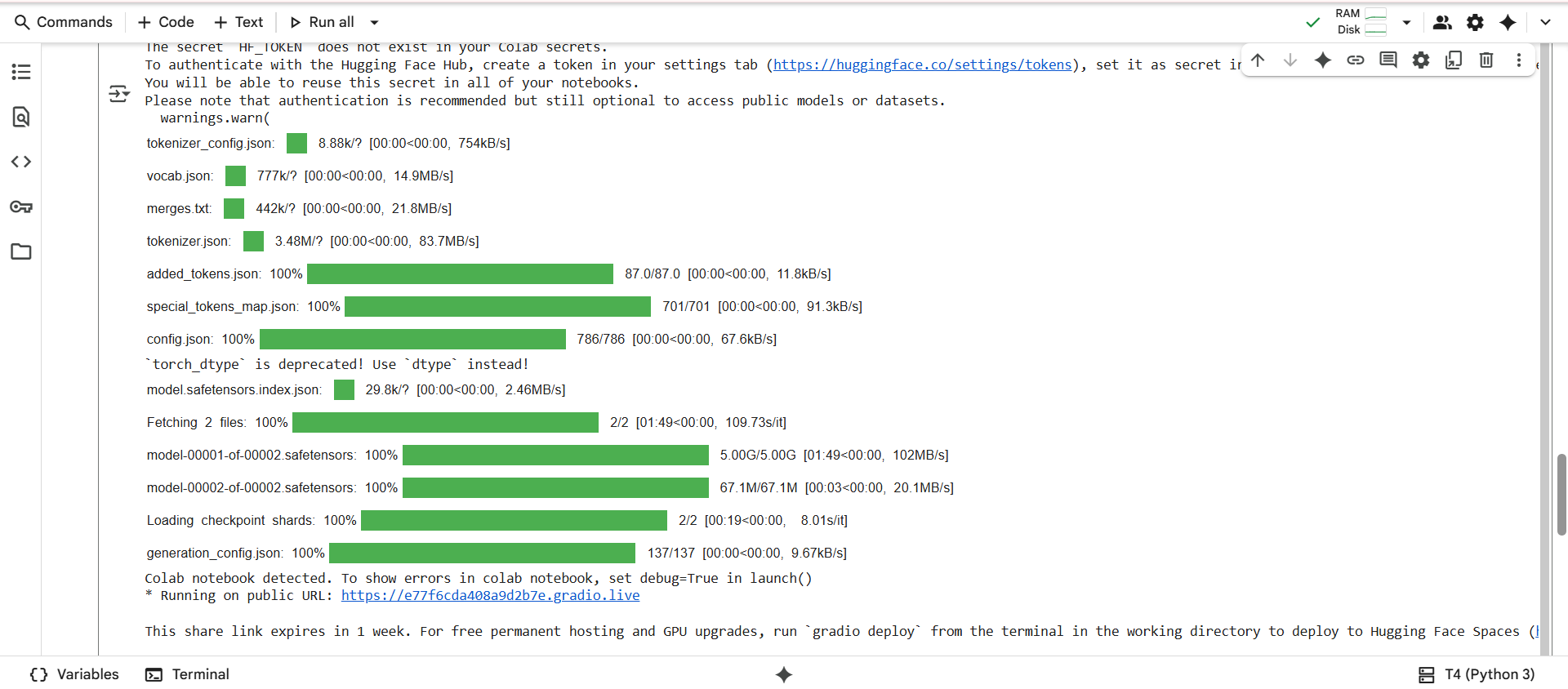
1. **Screenshots:**

**Program:**

****







**Output:**

