Generative AI with IBM Cloud

Documentation format

1. Introduction

- **Project Title:** HEALTHAI_APPLICATION
- **Team Members:** list as follows

- **Team Leader :** Gogada Santhosh

- **Team member :** Gaddipati Pushpasri

- **Team member :** Mohammad Nazeemunnisa

- **Team member :** Sk Shabbeer

2. Project Overview

- **Purpose:** the objective of the project is such as creating a health-focused AI assistant using generative AI models hosted on IBM Cloud.
- **Features:** Summarize major functionalities like symptom identification, home remedy suggestions, and AI-generated treatment plans.

3. Architecture

- **Frontend:** the interface is built—Streamlit, Gradio, or another tool.
- **Backend:** using IBM's Granite-3B-Instruct or similar models through the IBM Cloud platform
- **Database:** Detail integration with MongoDB Atlas for secure patient data storage.

4. Setup Instructions

- **Prerequisites:** List tools like Python, IBM Cloud CLI, MongoDB, and required Python packages.
- **Installation:** Step-by-step guide for cloning the repo, installing dependencies, and setting up .env configurations.

5. Folder Structure

- **Client:** Structure of the user interface components
- **Server:** Organization of AI inference, database, and API logic.

6. Running the Application

- Provide commands to start the frontend and backend servers locally.
  - **Frontend:** streamlit run app.py
    Backend: MongoDB connection setup, IBM
    API setup

7. API Documentation

- Document all endpoints exposed by the backend.
- Include request methods, parameters, and example responses.

8. Authentication

Secure handling of API keys, user sessions, and database access with .env and IAM principles

9. User Interface

- C:\Users\gsant\OneDrive\Pictures\Screenshot 2025-06-27 074010.png.

10. Testing

- testing methods (unit testing, integration tests) and tools used (e.g., pytest, Postman).

11. Screenshots or Demo

- https://drive.google.com/file/d/16auOM5V7y-NQxOdPLFaa9H1afzDCYoYU/view?usp=sharing.

12. Known Issues

- Document existing limitations—API latency, incomplete model responses, etc.

13. Future Enhancements

- Document existing limitations—API latency, incomplete model responses, etc.