PROBLEM STATEMENT

INTRODUCTION TO GenAI AND SIMPLE LLM INTERFACE ON CPU AND FINETUNING OF LLM MODEL TO CREATE A CUSTOM CHATBOT

UNIQUE IDEA BRIEF

The project aims to introduce Generative AI (GenAI) and Language Models (LLMs) with a focus on their application in creating a custom chatbot. The chatbot will provide personalized mental health support, it also provides fitness tips for people who doesn't know where to begin , leveraging a fine-tuned LLM to offer empathetic and context-aware responses. The entire solution will be designed to run efficiently on a CPU, making it accessible for users without high-end hardware.

Solution Components:

- Introduction to GenAI and LLMs
- Setting Up a Simple LLM Interface on CPU
- Fine-Tuning the LLM Model
- Developing the Custom Chatbot
- **Integration fine-tuned LLM model with Custom Chatbot Framework**
- **Test the Chatbot and evaluate its performance**

FEATURES OFFERED

FEATURES OFFERED BY THE LLM MODEL:

- Provides resources and articles and tips based on user request
- Generating coherent and contextually appropriate responses
- Adaptability and Scalability

FEATURES OFFERED BY THE CHATBOT:

- Real Time Conversation and User Interaction
- User-friendly interface
- Ensuring User Privacy by allowing anonymity
- Data Security and Privacy
- Multi-platform Support
- It also provides fitness tips for people who doesn't know where to begin
- Third-Party Integration
- Adheres to ethical guidelines

PROCESS FLOW

1.Setting the Environment and Importing Required Functions

•Environment Setup:

- Install Python and necessary libraries
- Set up a code repository (e.g., GitHub) for version control.

•Import Required Libraries:

- Import essential libraries for data processing, model configuration, and training .

2.Configuring the Device to CPU

- Ensure that the model and training process are set to use the CPU

3.Loading the Dataset

- **Dataset Collection :**
  - Gather domain-specific data relevant to the chatbot's intended use case.
- **Load the Dataset :**
  - Load the dataset into your environment using pandas or other data handling libraries.

```
import pandas as pd
dataset = pd.read_csv('path_to_your_dataset.csv')
```

4.Model Configuration and Preparation

- Select Pre-trained Model:
- Choose a pre-trained LLM suitable for fine-tuning.
- Load Pre-trained Model and Tokenizer

5.Data Processing

- Data Cleaning:
- Clean and preprocess the data to make it suitable for training.
- Tokenization:
- Tokenize the data using the model’s tokenizer.

6.Training Configuration

- Set Training Parameters:
- Define hyperparameters such as batch size, learning rate, and epochs.

7.Training the Model

- Create Trainer Instance:
- Create an instance of `Trainer` with the model, training arguments, and training dataset.
- Start Training.

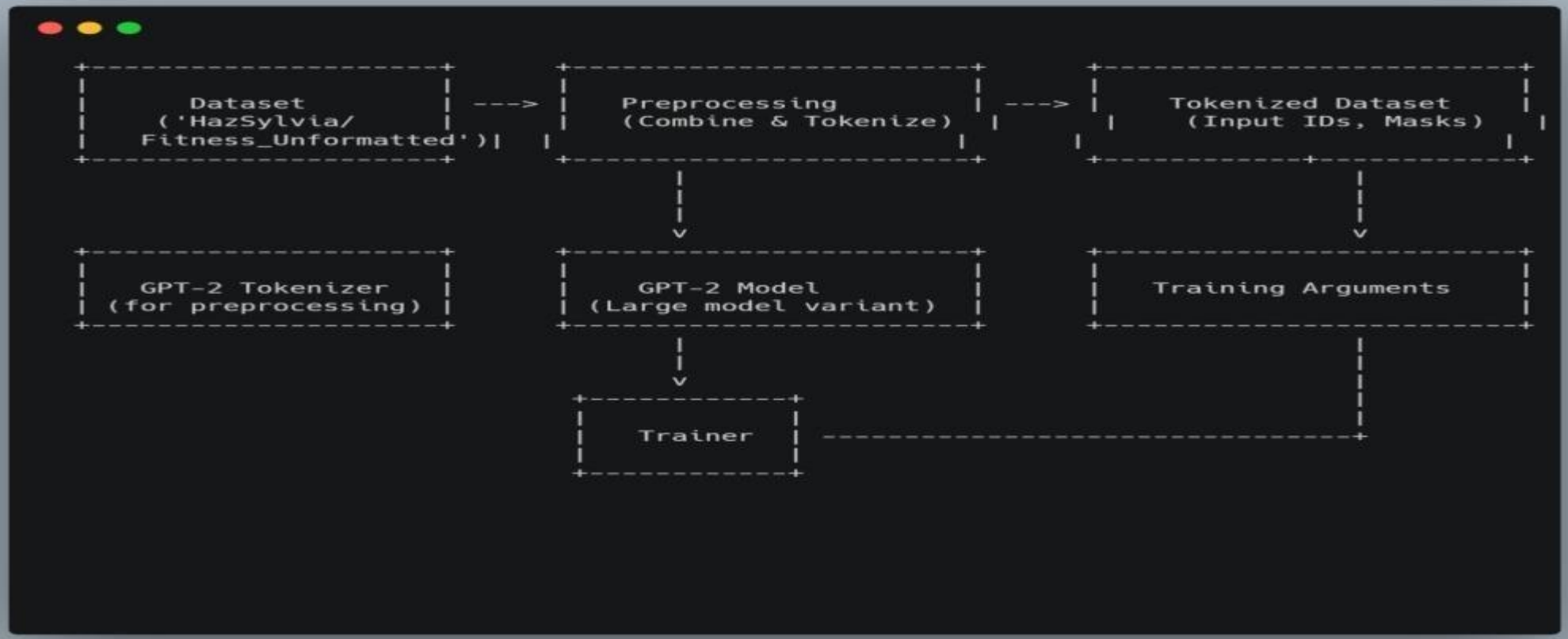
8.Inferencing the Model to Create a Chatbot

- Inference:
- Generate responses using the fine-tuned model.

9.Deployment of the Chatbot

- Frontend Development:
- Develop a user-friendly web interface using HTML, CSS, and JavaScript.
- Backend Integration:
- Set up a backend server to handle API requests and integrate the chatbot model.
- Hosting:
- Deploy the frontend on a platform like streamlit

ARCHITECTURAL DIAGRAM



TECHNOLOGIES USED

1 . User Interface (UI)

- HTML: For structuring the web pages.
- CSS: For styling the web pages to make them visually appealing and responsive

2. Backend Server

- Python: The primary programming language for backend development.

3. Fine-Tuned LLM

- Hugging Face Transformers: A library for using pre-trained LLMs and fine-tuning them for specific tasks.
- PyTorch or TensorFlow: Deep learning frameworks for training and deploying machine learning models.
- Transformers (library): For utilizing pre-trained language models and fine-tuning them.

- Datasets: Datasets can be loaded using libraries such as `datasets` from Hugging Face.
- Tokenizers: Convert text into a format that can be processed by models

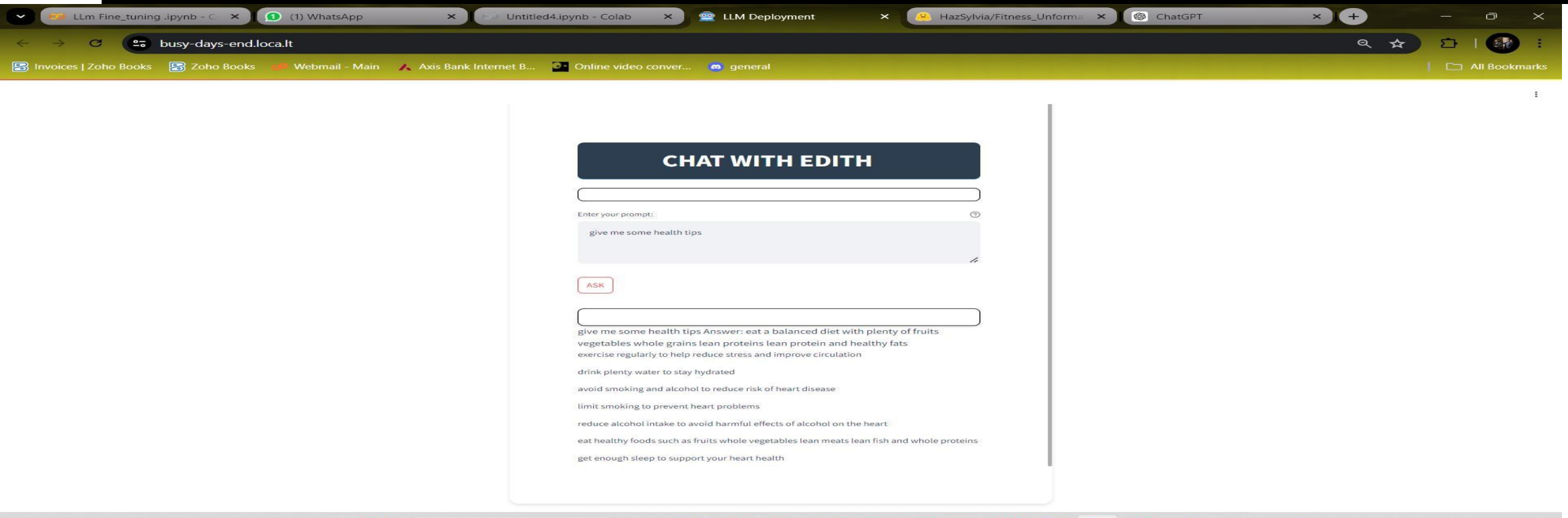
4.Deployment Infrastructure

- Streamlit: For hosting the frontend static files.

Additional Tools and Libraries:

- Google Collab
- Github

SAMPLE OUTPUT



TEAM MEMBERS AND CONTRIBUTIONS

1. ABINAV GANESH - Fine Tuning and Interference
2. PADMANATHAN- Fine Tuning
3. ARUNA DEVI- Deployment of model to stream lit
4. REASHMA SHAKTHI- Html Coding
5. DIVYA DHARSHINI- Testing and Evaluation

CONCLUSION

Creating a custom chatbot using Generative AI (GenAI) and fine-tuned Language Models (LLMs) on a CPU offers a powerful and accessible solution for various conversational applications.This project guides through the entire process, from setting up the development environment and selecting the right tools to fine-tuning an LLM and deploying the chatbot. The end result is an efficient and user-friendly chatbot capable of delivering coherent and relevant responses in real-time, providing a practical demonstration of the capabilities of modern AI in natural language understanding and generation.