Developing an AI-Powered Customer Support Website with an Offline Chatbot

## Group 2:

=>Tanzeem Haque, 2131122642

=> SK Asafur Rahman, 2112430642

=> Sajid Hasan, 2222684042

=> Masud Ahmed Niloy, 1912477042

=> S. K. Sajid Anam, 2211624042

Abstract—

This proposal outlines the development of a customer support website integrated with an AI-powered chatbot. The chatbot will be a locally hosted, offline ChatGPT-equivalent system powered by a custom multimodal model. An open-source web interface supporting multi-user chat and authentication will serve as the front end. The model is serialized in GGUF format (quantized from “.Safetensors”) to enable CPU-based computation while offloading intensive layers to a GPU for efficiency. Our key contributions include:

\* Developing a lightweight yet powerful transformer-based language model;

\* Optimizing offline inference on consumer-grade hardware; and

\* Implementing an API-based approach for seamless chatbot integration.

Additionally, the proposal introduces “Instant Context (Instant Train),” a feature designed to enable rapid fine-tuning in seconds rather than prolonged retraining.

Keywords— AI, customer support, offline chatbot, transformer model, model quantization, API integration, multimodal.

I. INTRODUCTION

Large Language Models (LLMs) such as GPT have demonstrated exceptional performance but often require cloud-based deployment. This dependency raises concerns regarding data privacy, accessibility, and model control. Our goal is to create a fully offline AI system capable of real-time inference on local hardware by balancing GPU offloading with CPU fallback for efficient resource use. In addition, we will develop a customer support website that provides users with a seamless support experience through an intuitive interface, knowledge base integration, and live chat functionality.

II. LITERATURE REVIEW

A. Transformer Models

Self-attention-based transformer architectures have revolutionized natural language processing (NLP) by significantly enhancing contextual understanding and computational efficiency. Advances by organizations such as OpenAI demonstrate the power of large-scale pre-trained networks.

B. Model Quantization & Offloading

Quantization techniques reduce model size and computational demand, making offline operation feasible. Frameworks like Hugging Face Transformers allow for selective GPU offloading, ensuring efficient workload distribution while maintaining performance on consumer-grade hardware.

III. PROBLEM STATEMENT

Developing a customer support website with an integrated AI chatbot poses several challenges, including:

\* Website Performance: Ensuring smooth operation for all users.

\* User-Friendly Design: Creating an intuitive and easy-to-navigate interface.

\* Security: Safeguarding user data.

\* Chatbot Integration: Seamlessly integrating the chatbot with the website.

\* Offline Functionality: Enabling the chatbot to operate without an internet connection.

Our solution directly addresses these challenges by ensuring full local deployment for data privacy, efficient resource management using a GTX 1660 GPU with CPU fallback, and a familiar ChatGPT-like interface that guarantees a user-friendly experience.

IV. PROPOSED SOLUTION

A. Model Design

We propose a multimodal LLM focused initially on text-based interactions. The model will be stored in the .safetensors format to ensure compatibility with GPU acceleration.

B. Website Integration

A basic consumer website will be developed to serve as the customer support portal. The integrated AI chatbot will function as an information center, providing real-time assistance to users.

C. API-Based Chatbot Deployment

An API-based approach will enable seamless integration between the website frontend and the locally hosted AI model. The API will:

\* Handle user queries from the website;

\* Interface directly with the AI model;

\* Maintain session context for continuous conversations; and

\* Provide endpoints for analytics and logging.

D. Local Deployment

The entire system is designed for offline operation, using transformer architectures that allow local hosting without external network dependencies.

V. IMPLEMENTATION

A. System Architecture

The system architecture is divided into three main components:

\* Client: A web-based interface accessed via an icon for the AI assistant.

\* Server: A Python-based backend that manages model handling using Transformers.

\* API Layer: An API conforming to standard OpenAI API practices to facilitate communication between the frontend and the AI model.

B. Model Training & Storage

The transformer model will be trained on a diverse dataset using high-end hardware (e.g., RTX 4090) and stored in a format optimized for GPU inference (.safetensors).

C. Security & Privacy

\* No external network access will be permitted.

\* User data will be stored locally in an encrypted format.

\* The system will not rely on any third-party APIs, ensuring full control over data and operations.

VI. ANTICIPATED RESULTS

A. Website & Chatbot Performance

The website is expected to deliver a smooth, lag-free user experience with rapid load times, while the chatbot should provide near-instant AI-driven responses facilitated by GPU acceleration.

B. Performance Metrics

Performance tests will be conducted on systems equipped with an NVIDIA GTX 1660 GPU and 16GB RAM, with expected outcomes including:

\* Rapid responses for short queries (~30 tokens).

\* Feasible processing times for longer responses (~200 tokens) with GPU support.

C. Accuracy

Benchmarks against standard GPT models will be performed, with minor trade-offs in long-context coherence addressed through targeted fine-tuning.

D. User Feedback

Pilot testing will be conducted to gather insights regarding speed, reliability, and usability, which will inform subsequent optimizations.

VII. FUTURE OUTLOOK

After the initial deployment, the following enhancements are planned:

\* Expansion of multimodal support (e.g., image and audio inputs).

\* Optimization of GPU offloading policies to improve performance further.

\* Enhancement of fine-tuning mechanisms for domain-specific improvements.

This system is envisioned as a privacy-focused, offline alternative to commercial chatbot solutions, offering complete model control and secure data handling on consumer-grade hardware.

VIII. AI-POWERED CUSTOMER SUPPORT WEBSITE

A. Purpose

The project aims to develop a customer support website that integrates an AI-powered chatbot to ensure efficient user assistance and real-time query resolution.

B. Features

\* Automated Responses: The AI chatbot will deliver instant replies to common inquiries.

\* Knowledge Base Integration: The system will pull relevant articles and information to address user queries.

C. Implementation

\* Frontend: Developed using HTML, CSS, and JavaScript to create an engaging UI.

\* Backend: A Python-based API will manage chatbot interactions.

\* AI Model: A locally hosted ChatGPT-equivalent model will power the chatbot responses.

\* Deployment: Both the website and chatbot will be hosted locally, ensuring offline functionality.

\* API-Based Chatbot Integration: The chatbot will be accessible via an API modeled on OpenAI standards, enabling seamless interaction from the website.

D. Expected Outcomes

\* An enhanced customer experience through quick and relevant support.

\* A reduction in the workload for human support agents.

\* Improved user retention and satisfaction through a reliable and secure offline system.