

QuadraSystems

Weekly Report

Athish.CK

B-Tech(AI&DS)

Intern-QuadraSystems

Week-2

We will explore the core principles behind modern AI systems, particularly those relevant to natural language processing (NLP). Key topics will include model training, fine-tuning, prompt engineering, and the use of LLMs for conversational interfaces.

Additionally, we will begin experimenting with existing LLM frameworks such as OpenAI's GPT or Google's Gemini, understanding how to interact with them via APIs, and evaluating their performance in domain-specific tasks.

Goals

Goals for the Week:

1. Develop the initial version of the chatbot.
2. Learn the fundamentals of Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG), and begin implementing them in the project.
3. Build a full-stack chatbot and integrate it with the necessary APIs.
4. Train the model with domain-specific data and apply constraints to limit its scope appropriately.

LLMS

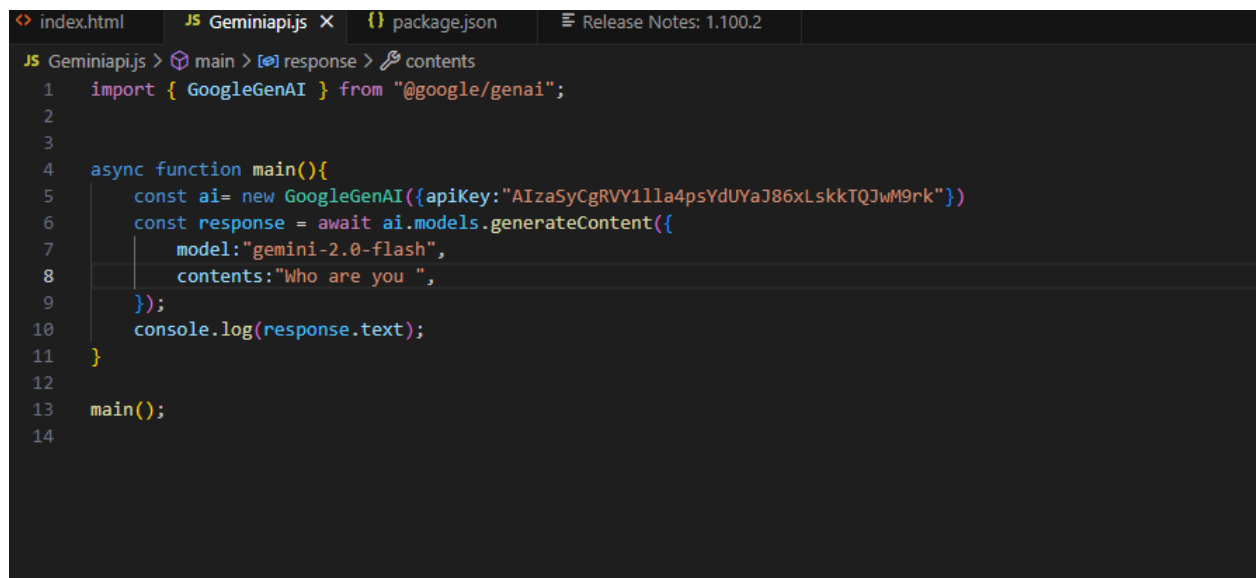
To begin with, we obtained an API key from Google's Gemini platform. The API has been successfully integrated and linked with the local machine for further development and testing purposes.

```
PS C:\Users\athis\OneDrive\Desktop\Quadra\Codes> node Geminiapi.js
Quadra is an AI-powered platform that helps teams document, organize, and share their knowledge more effectively.

PS C:\Users\athis\OneDrive\Desktop\Quadra\Codes> node Geminiapi.js
I am a large language model, trained by Google.
```

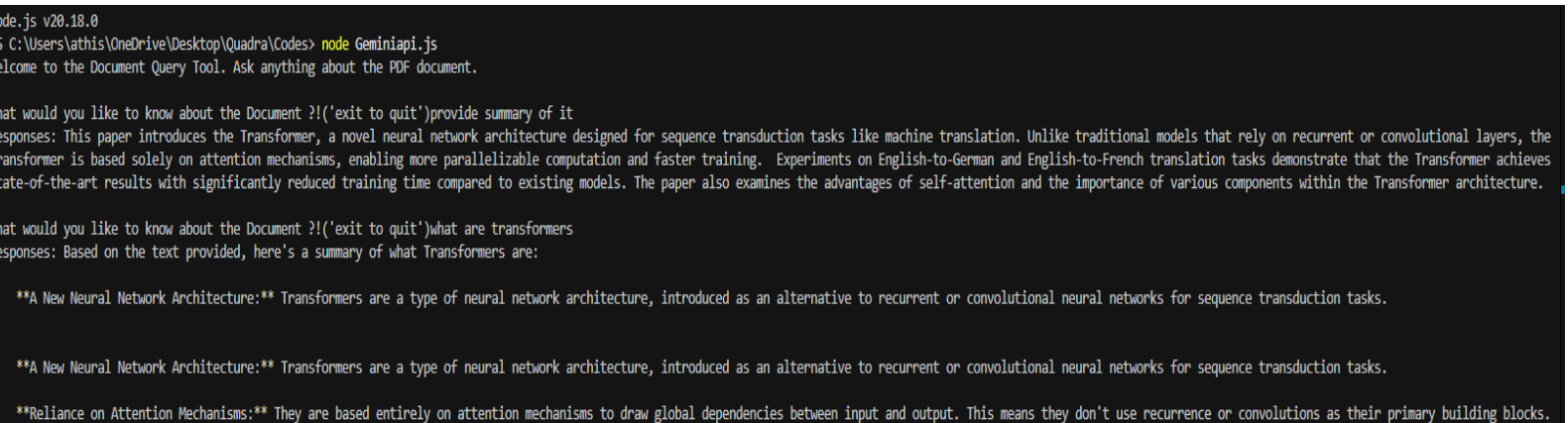
Making the API Dynamic

After successfully connecting the API to the local file, the next objective is to make the API dynamic, enabling real-time interaction with the user. For demonstration purposes, we have utilized the research paper *"Attention Is All You Need"*, allowing the user to pose



```
index.html JS Geminiapi.js X {} package.json Release Notes: 1.100.2
JS Geminiapi.js > main > response > contents
1 import { GoogleGenAI } from "@google/genai";
2
3
4 async function main(){
5   const ai= new GoogleGenAI({apiKey:"AIzaSyCgRVY1lla4psYdUYaJ86xLskkTQJwM9rk"})
6   const response = await ai.models.generateContent({
7     model:"gemini-2.0-flash",
8     contents:"Who are you ",
9   });
10  console.log(response.text);
11 }
12
13 main();
14
```

questions based on its content.



```
ode.js v20.18.0
C:\Users\athis\OneDrive\Desktop\Quadra\Codes> node Geminiapi.js
Welcome to the Document Query Tool. Ask anything about the PDF document.

What would you like to know about the Document ?!(('exit to quit')provide summary of it
Responses: This paper introduces the Transformer, a novel neural network architecture designed for sequence transduction tasks like machine translation. Unlike traditional models that rely on recurrent or convolutional layers, the transformer is based solely on attention mechanisms, enabling more parallelizable computation and faster training. Experiments on English-to-German and English-to-French translation tasks demonstrate that the Transformer achieves state-of-the-art results with significantly reduced training time compared to existing models. The paper also examines the advantages of self-attention and the importance of various components within the Transformer architecture.

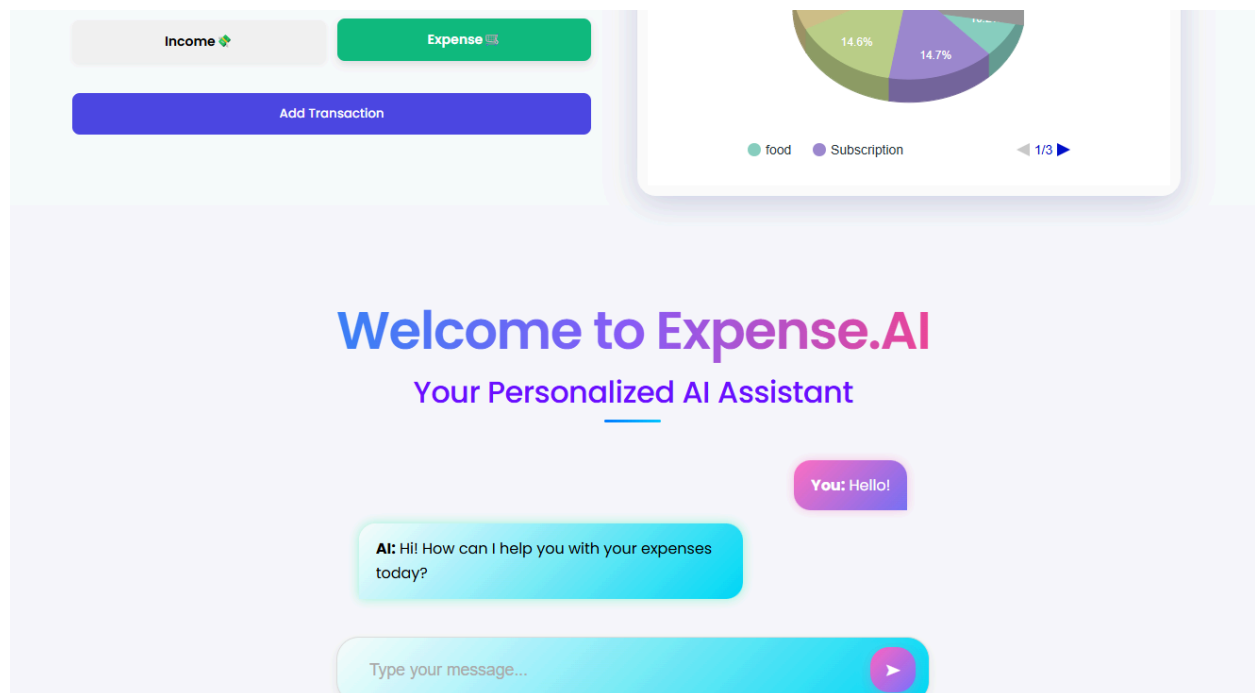
What would you like to know about the Document ?!(('exit to quit')what are transformers
Responses: Based on the text provided, here's a summary of what Transformers are:

**A New Neural Network Architecture:** Transformers are a type of neural network architecture, introduced as an alternative to recurrent or convolutional neural networks for sequence transduction tasks.

**A New Neural Network Architecture:** Transformers are a type of neural network architecture, introduced as an alternative to recurrent or convolutional neural networks for sequence transduction tasks.

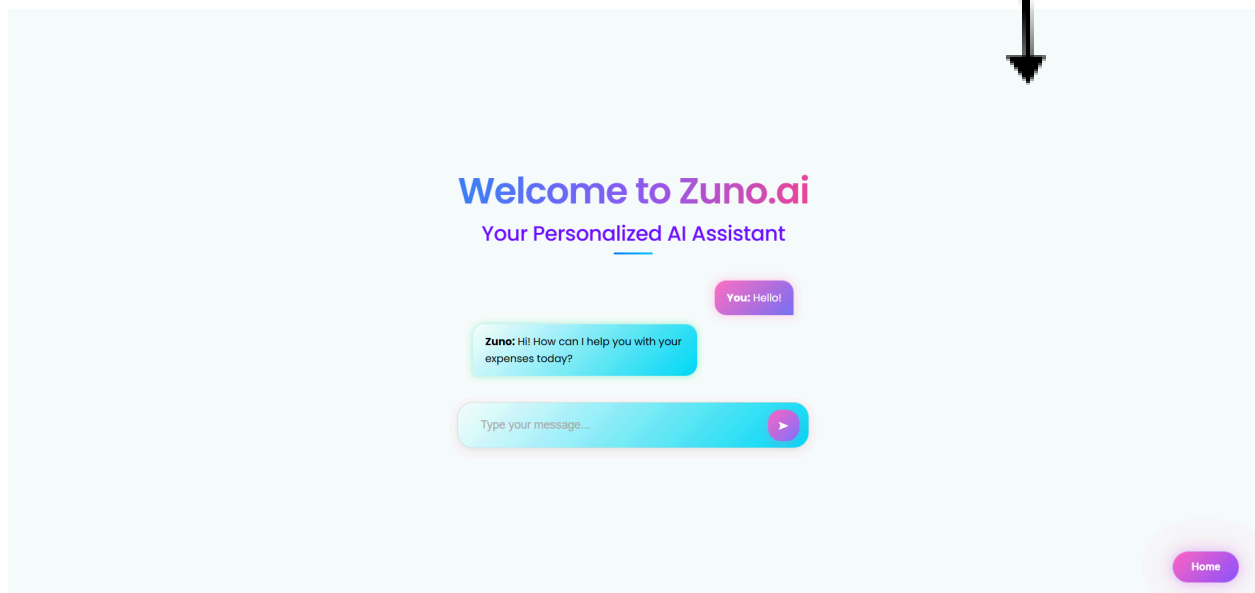
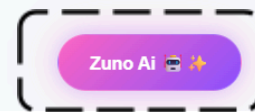
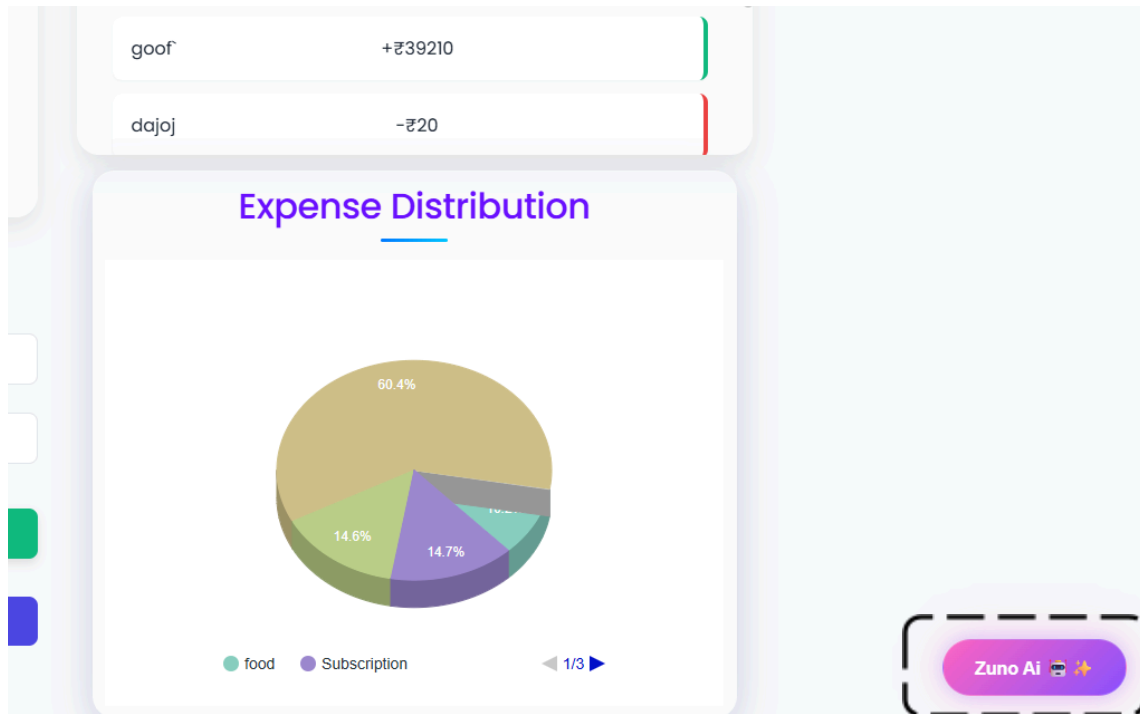
**Reliance on Attention Mechanisms:** They are based entirely on attention mechanisms to draw global dependencies between input and output. This means they don't use recurrence or convolutions as their primary building blocks.
```

Integrating with Expense tracker



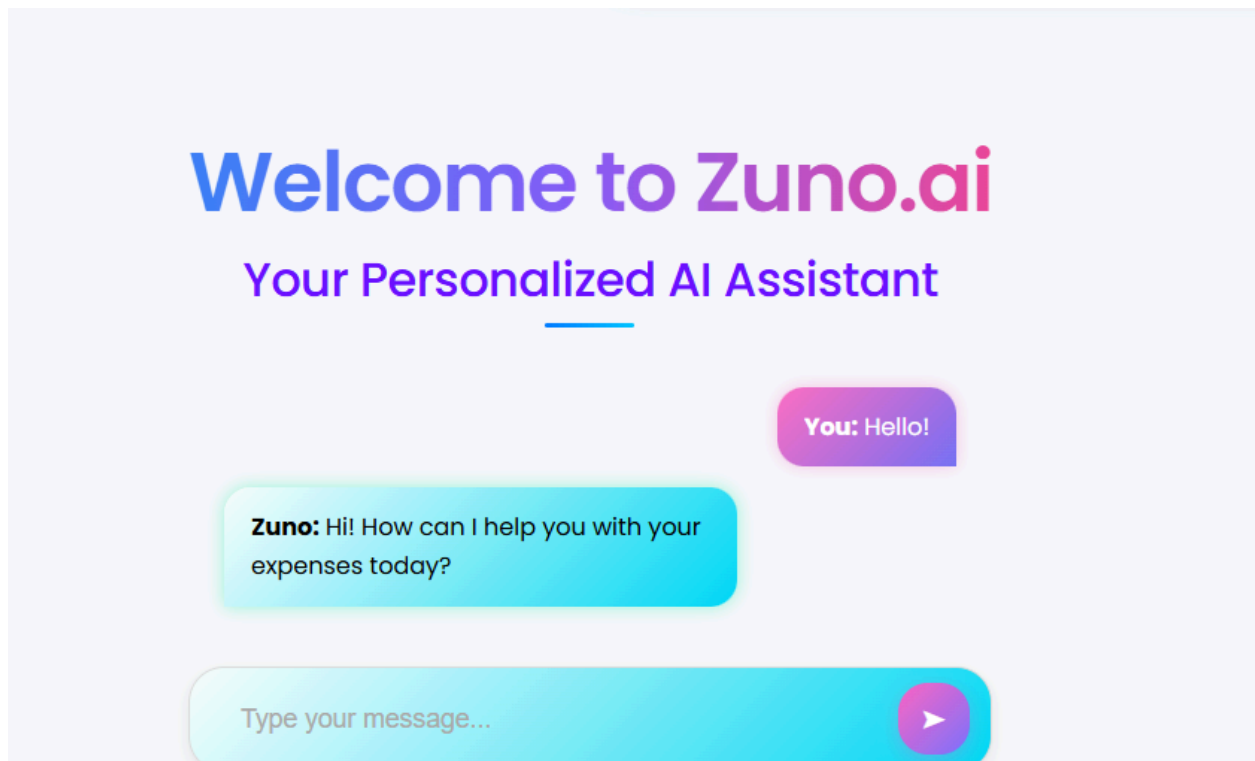
We are now integrating the API into our Expense Manager application as part of the project. This phase includes the development of a basic front-end interface that demonstrates the core concept. Moving forward, we plan to integrate the model with a local DBMS containing user expense data. This integration will enable users to analyze their financial transactions based on the data they provide, offering personalized insights and recommendations.

Now the next step is to give a submit Button where it leads to another separate page for the AI interaction .



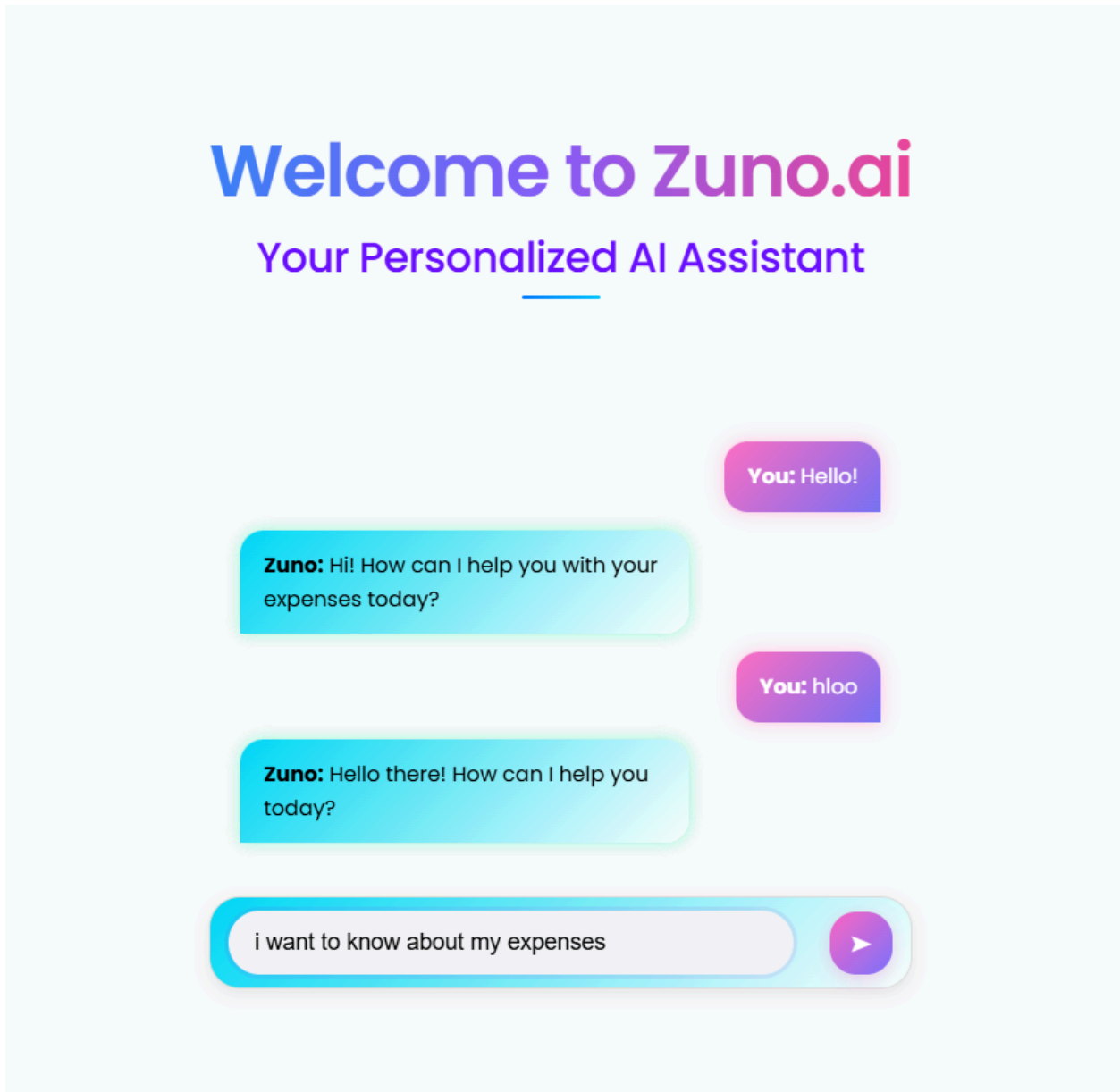
```
console.log("Welcome to Zuno.ai")
```

Btw I named this Zuno.ai . So from now on we call him Zono 😊. The original idea was to keep Quadro but it's off the table .



Connection with API

Zuno is now successfully integrated with the Gemini API, enabling seamless and connected conversations.



The next steps involve connecting Zuno with the user's data and the chatbot. It needs to be trained with domain-specific information focused on expenditure, while access to other content will remain restricted. A Retrieval-Augmented Generation (RAG) system will be implemented, along with integration of the user's DBMS server.

Conclusion

In this phase of the project, we successfully laid the groundwork for building an intelligent, domain-specific chatbot. We began by integrating the Gemini API with Zuno, enabling basic conversational capabilities. Key advancements included understanding the fundamentals of Large Language Models (LLMs), Retrieval-Augmented Generation (RAG), and the architecture required for full-stack chatbot development. The model was trained and constrained to operate within a specific domain—expenditure management—to ensure relevance and accuracy.

Future Scope

Enhanced Personalization:

Further training the model with personalized user data to offer more contextual and user-specific responses.

Advanced RAG Implementation:

Implement a robust RAG pipeline that dynamically fetches information from large datasets and document stores, improving factual accuracy and domain relevance.

Integration with DBMS and Analytics Tools:

Seamlessly connect the chatbot to the user's DBMS to allow for real-time data retrieval, analysis, and report generation.

Multi-modal Capabilities:

Expand the chatbot's functionality to include voice and visual inputs, enabling a more interactive and user-friendly experience.

Scalability and Deployment:

Optimize the full-stack application for large-scale deployment, ensuring performance, security, and reliability in real-world usage.