Hackathon

Macroeconomic Researcher Food Security Time Series and Large Language Chat GPT Dashboard

Team members:
1. Harshith Akkapelli
2. Neeharika Singh
3. Sanjay Bhargav Kudupudi

Introduction

This is a system for processing and responding to natural language prompts, specifically tailored for economic research and government use cases, such as the analysis of budget speeches. This system is engineered to accommodate both extensive data interaction by researchers and straightforward query-response interactions for government officials.

User Personas:

Economic Researcher: Engages deeply with the system, possibly requiring tools for data annotation and analysis to refine the AI model or to delve into specific research areas.

Government Representative: Uses the client side of the system for obtaining information or generating reports, necessitating a straightforward and efficient user interface.

Architecture:
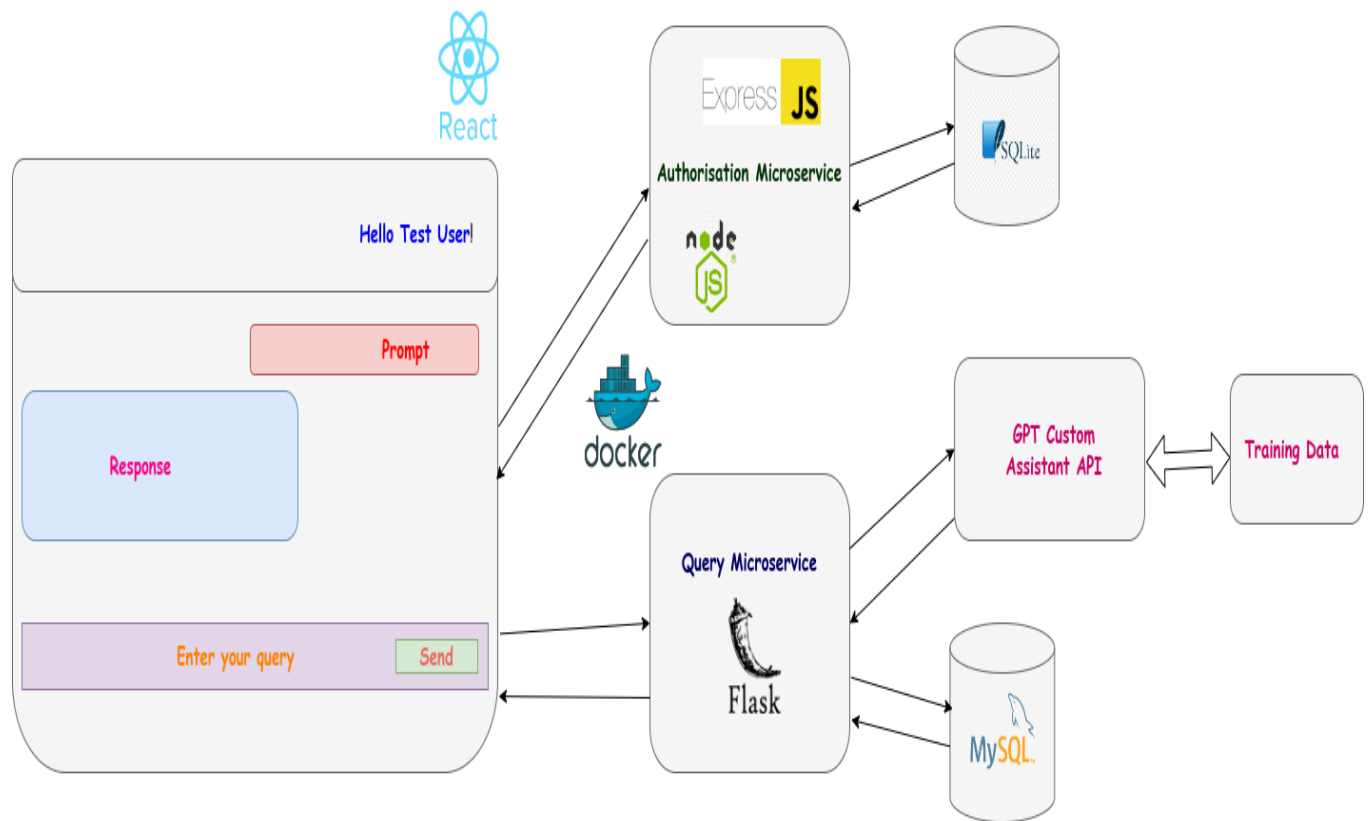
Authorization Microservice:

Purpose: Serves as the entry point for users, handling secure access to the system.

Workflow: Users identify themselves and gain access, ensuring that only authorized personnel can submit queries and access the data.

Query Microservice:

Role: Acts as the system's brain, processing complex natural language queries submitted by users.

Interaction with AI: Employs an advanced language model, trained on specialized data (such as budget speeches), to interpret the queries and generate meaningful responses.

Data Management: Manages the flow of prompts and responses, ensuring that user queries are correctly interpreted and that the responses are accurate and relevant.

Client Application:

User Interface: Provides a user-friendly platform for submitting prompts and viewing responses, with clear areas for interaction and information display.

User Experience: Designed to be intuitive, allowing users such as government representatives to easily navigate and utilize the system without needing in-depth technical knowledge.

Data Storage:

Databases: Utilizes two separate data storage systems, suggesting a division of data management tasks, such as storing user information separately from query data.

Data Integrity: Ensures that the data is stored securely and efficiently, allowing for quick retrieval and processing when needed.

Deployment:

Containerization: The use of containers indicates that the system is designed for easy deployment and maintenance, with the ability to scale up or down as required.

The architecture is built with the end-user in mind, ensuring that complex data interactions are managed seamlessly in the background, providing a smooth and accessible front-end experience for users seeking to interact with economic and budget-related data. It's structured to support high-volume data processing while maintaining user accessibility and system integrity.