PROJECT NAME - Conversational Requirement Elicitation Tool for Reporting Domain

TEAM MEMBERS-

* ANKITA POREL (MTECH CSE, 2024201043)
* ACHANTI PAVAN KUMAR (MTECH CSE, 2024201053)
* ASHWANI RAJ (MTECH PDM, 2024204002)
* SRIRANGAM VENKAT MOHIT KUMAR (MTECH CSE, 2024201010)

SCOPE-

1. Inputs:

Meta-Model Input:

Priyamwada: Creates and uploads a meta-model that outlines the essential attributes for report generation (in JSON/XML format).

User Input:

Sushma: Interacts with the chat-based tool, which probes her for reporting requirements. The tool uses the meta-model to guide the conversation and capture key details like report type, visualization preferences, required filters, sub-reports, and graph types.

2. Step-by-Step Control Flow/Data Flow:

Meta-Model Creation:

Priyamwada creates a meta-model in JSON/XML format containing necessary fields for report creation.

Meta-model is uploaded to the tool, forming the basis for subsequent interactions.

Chat-Based Requirement Elicitation:

Sushma accesses the chat tool.

The tool prompts Sushma with questions based on the meta-model to gather specific report requirements.

Questions may include: Type of report, Visualization type, Filters, Sub-reports, Graphs, etc.

The tool dynamically adjusts the conversation flow based on Sushma's responses.

Requirement Summarization:

Once all necessary data is collected, the tool summarizes the requirements in plain English.

The summary is prepared in a format ready for Priyamwada to forward to Data Engineers.

Output:

A plain English summary of the report requirements that Priyamwada can review and send to the Data Engineers for report creation.

SYSTEM DESGIN

Client-Side:

UI/UX: HTML, CSS, JS for the chat interface.

Chatbot: Frontend script interacting with the user.

Server-Side:

API Layer: Middleware that handles requests from the chat interface.

Meta-Model Processor: Component that parses and processes the uploaded meta-model.

Conversation Engine: Logic to drive the chat-based conversation, probing the user based on the meta-model.

Summarization Engine: Component that generates the plain English summary of the captured requirements.

Database:

Meta-Model Storage: Stores uploaded meta-models.

Chat History/Logs: Stores interaction history for auditing and improvement purposes.

STAKEHOLDERS

\* Priyamwada: Should be able to create a meta-model and upload it to the Tool

\* Sushma (Any Department Head): Should be able to converse with the tool and get her report requirement submitted to the respective department.

USE CASES:

UC1: Priyamwada creates a new meta-model and uploads it to the system.

UC2: Sushma logs into the tool and starts a conversation about a new report requirement.

UC3: The tool probes Sushma for the type of report needed.

UC4: The tool queries Sushma about the visualization style preferred for the report.

UC5: The tool asks Sushma about specific filters or data fields needed in the report.

UC6: The tool inquires if there are any sub-reports required.

UC7: The tool summarizes the collected requirements in plain English.

UC8: Priyamwada reviews the summary and sends it to Data Engineers.

UC9: The system stores a history of Sushma’s interactions for future reference.

UC10: Priyamwada updates the meta-model as requirements evolve.

UI/UX SAMPLE SCREEN:

Meta-Model Upload Screen:

Interface for Priyamwada to upload and manage meta-models.

Chat Interface for Sushma:

Simple chat window with conversational prompts.

Summary Review Screen:

Interface for Priyamwada to review and finalize the summary before sending it to Data

Engineers.