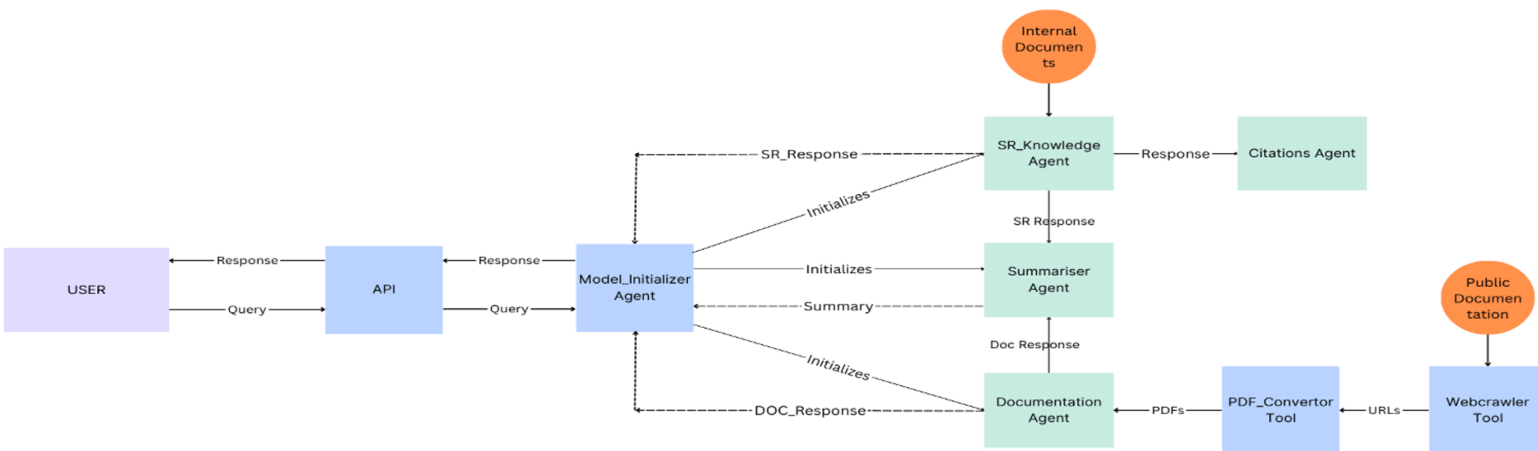
OTM ASSISTANT

WORKFLOW DIAGRAM



DETAILED EXPLANATION OF THE WORKFLOW

- The user queries the API first.
- The API sends a request with the query in the request's body, to the Model_Initializer agent.
- The Model_Initializer agent then initializes all the three models and runs them.
- The SR_Knowledge AGENT parses the Support Request documents available to check if the query has been answered previously, and generates an answer based on the documents available and sends it to the Summariser Agent.
- The Documentation Agent then fetches the pdfs made by the Webcrawler tool, parses them and returns the answer to the query.
- The SR response is sent to the Citations Agent which extracts the citations and sends these back. Citations are important so that the support people can directly go that document ID or bug no. referenced.
- The Webcrawler tool crawls the URLs in the public documentation, fetches the text from those URLs.
- The PDF_Convertor tool converts these into pdfs.

- These both responses are then sent to the summarizer agent which then summarizes both the answers and returns the summarized answer to the user.

DATASET

- 2 Datasets:
  - SR Knowledge Modules: 88 PDFs, given by POC as we cannot access those.
  - Public documentation of OTM: Due to computing restraints, we have crawled and used data of URLs which have the keyword “Planning” in it. Ex: docs.oracle.com/en/cloud/saas/transportation/planning. This is the starting URL.

MODELS/ LIBRARIES USED

- Parsing: LlamaParse
  - For efficient retrieval and context augmentation.
  - We generate a parsed file parsed_data.pkl
- Embedding: FastEmbed Embedding
  - It is light and fast
  - Accuracy/Recall better than OpenAI Ada-002

- LLM Model: Groq mixtral-8x7b-3276
  - Proven nice performance
  - Relatively free tier

- Summarizing : t5-small from Hugging Face

- HTML to Text: BeautifulSoup

- Text to pdf: fpdf

- Indexing documents: VectorStoreIndex from Qdrant
  - Qdrant is a high-performance vector database that is used for storing and retrieving large datasets of vectors. It is particularly well-suited for applications that require fast and efficient similarity search
  - We need fast similarity searches, hence Qdrant.

BACKEND

Created an API in python using FastAPI

- On startup the API loads up the model and prompts it when the request arrives.
- The server is hosted using ngrok which sets up an external server and redirects the requests to our locally exposed port.
- We used ngrok because the SR model contains sensitive SR documents which cannot be uploaded to any site/cloud provider.

CRITERIA

CRITERIA	
Creativity and Innovation	Oracle Transport Management has a bunch of internal SR documents. These are answers to customer queries given by support people at Oracle. There is also a huge load of documentation that they need to go through for finding the solution. Our tool enables employees providing support to find answers that are already there in SR documentation in a simple summarized form, and if not there it can crawl public documentation and provide answers. It can also be exposed directly to customers where our summarizer will provide clear and concise answers to the queries.
Business Impact and Relevance	The agents can efficiently resolve redundant service requests and proactively address new ones. This streamlines our support team's workflow, eliminating the need to manually search for and review documents before assisting customers. As a result, service delivery to customers becomes faster and more seamless. It is also very scalable, if we get more computing units.
Implementation and Execution	The agents developed are accurate and have been implemented without errors. Their solutions for new service requests are consistent with previous SRs and documentation, providing reliable and precise resolutions. Our project uses three agents. Two agents are basically using a Groq LLM model (“mixtral-8x7b-32768”). One agent works on SR documentation, and the other works on the public documentation available. The third agent summarizes both the answers and provides the user with a simple answer to the query (uses Hugging Face ‘t5-small’ summariser). We also use a tool, which is a web crawler script that crawls all the URLs after giving a starting URL. We use BeautifulSoup pulling data out of HTML available. We use fpdf

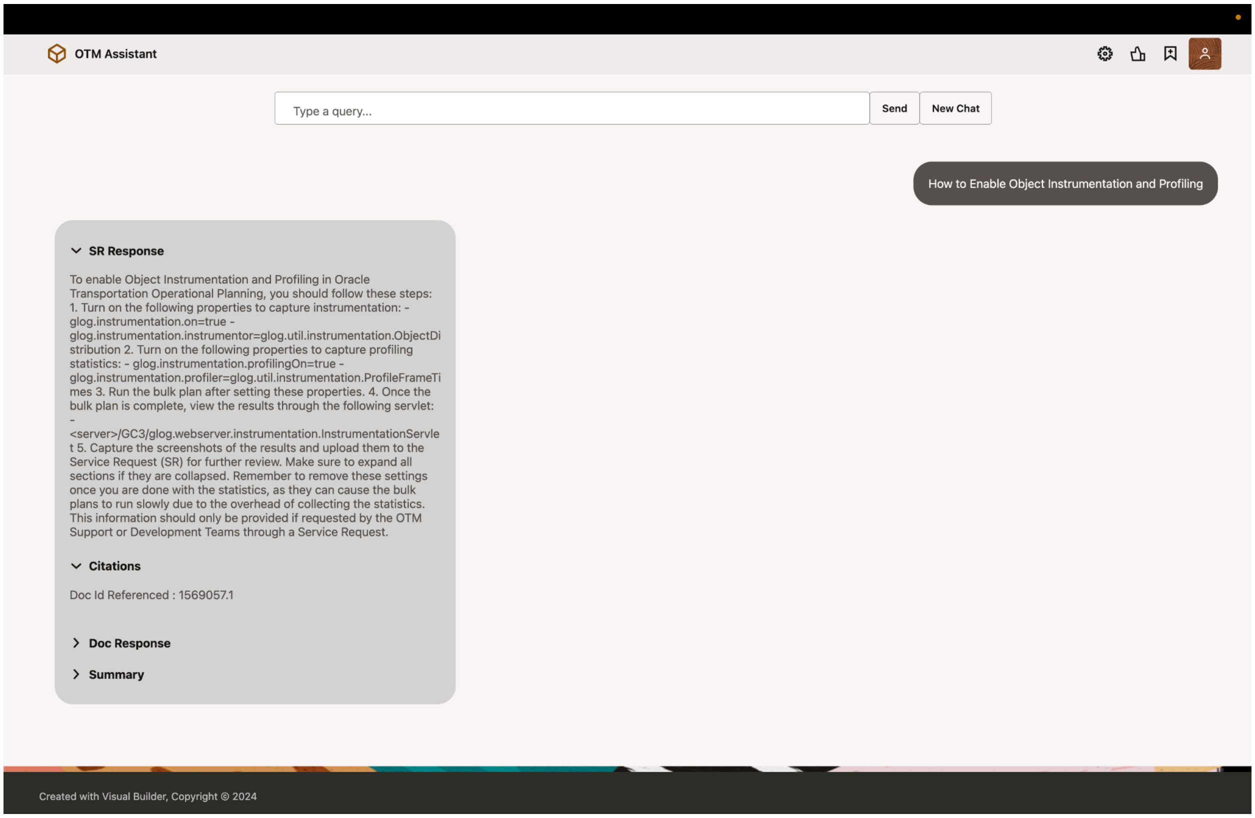
	to convert these html texts to pdf files for passing to the LLM models. We have built our own custom agents.
Technical Proficiency	We have used state of the art models for every step.
Data Handling and Preprocessing	We have used BeautifulSoup for extracting text out of html files and fpdf for converting these files to pdfs. We are using LlamaParse for parsing the pdfs and storing it into a .pkl file. As the links are static and the SR knowledge docs are also static, we have generated the parsed files and kept it so that the model does not have to spend time parsing it again and again. For embeddings, we have used FastEmbed embeddings. We also store these embeddings and pass it to the model once the querying starts.
Evaluation and Metrics	We made some sample queries and tested our model on those queries. We fetched the required answers and then manually calculated the similarity score (Jaccard Similarity) of the texts to see the quality of the answers generated. The accuracy of the answers that we checked manually was around 80% (subject to interpretation)

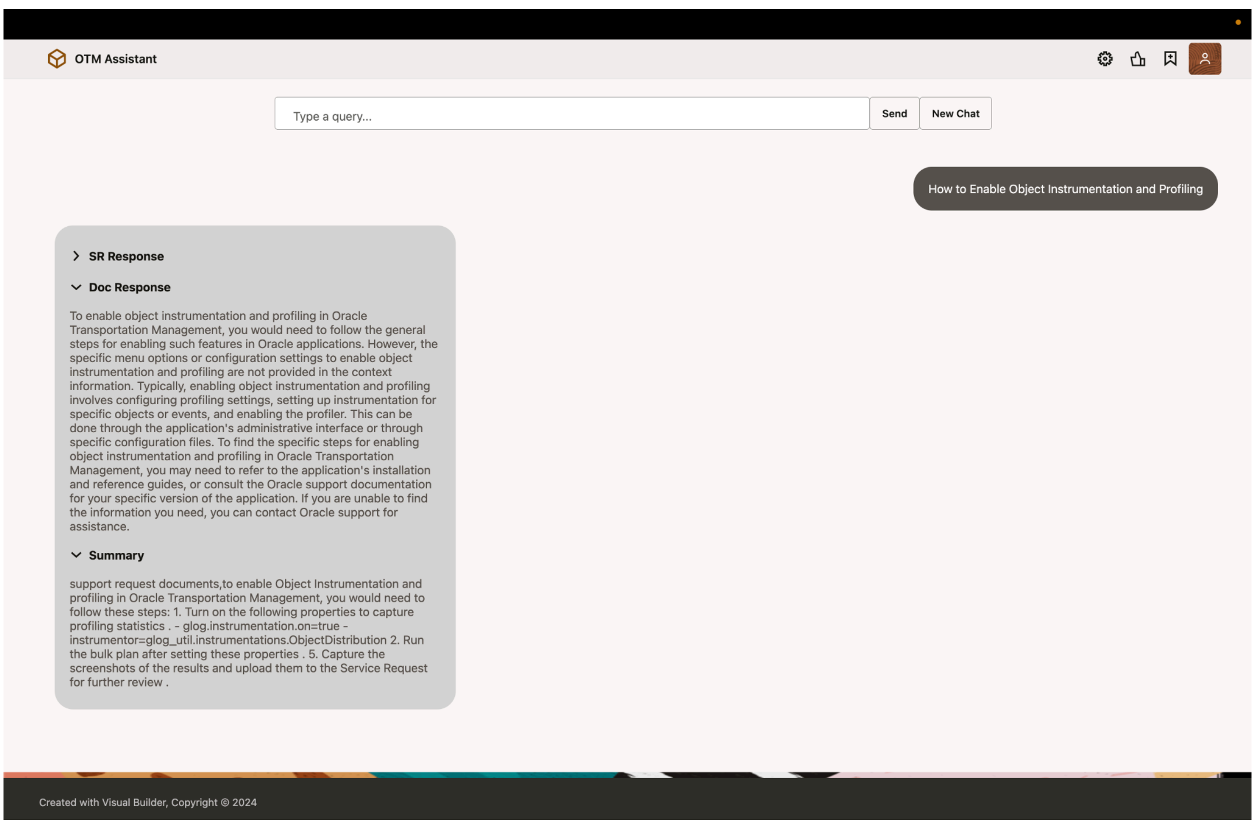
FRONTEND UI DEVELOPMENT

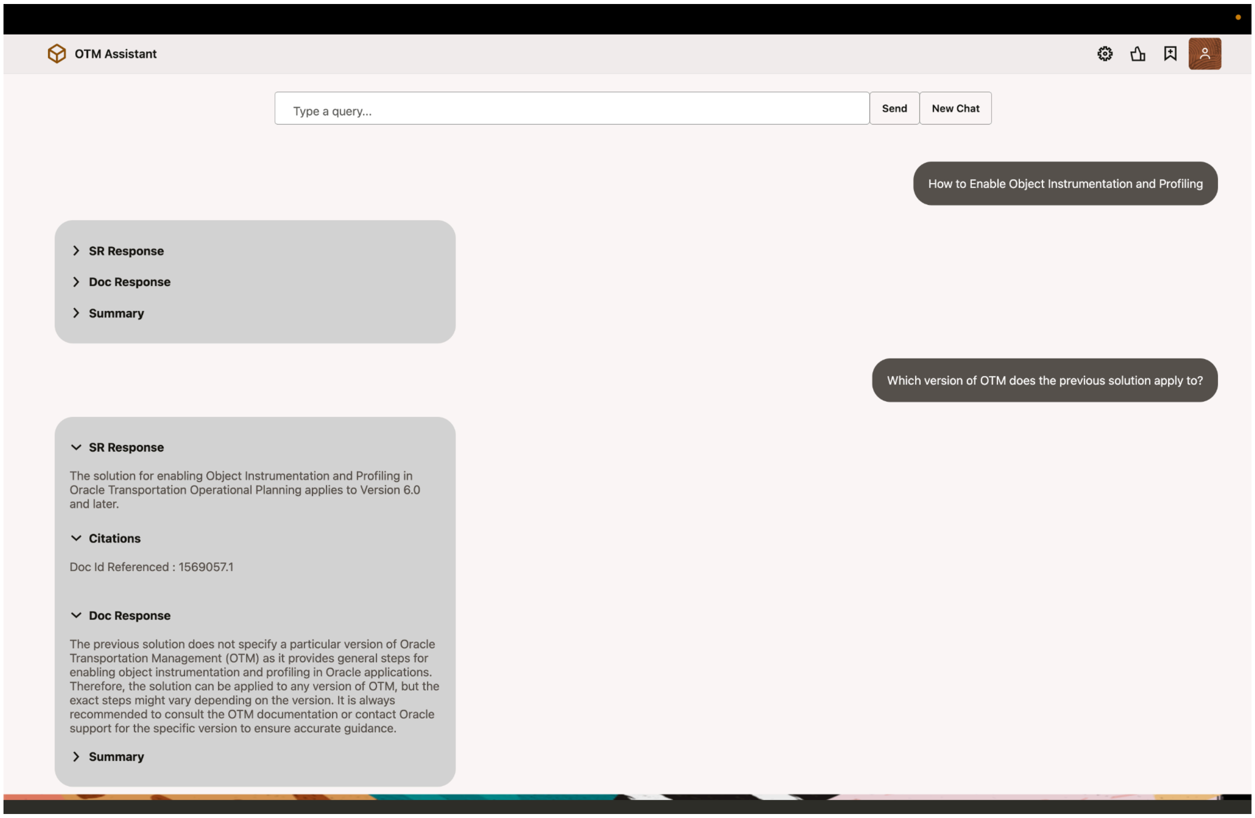
We developed a seamless user interface for our project using **Visual Builder Studio**.

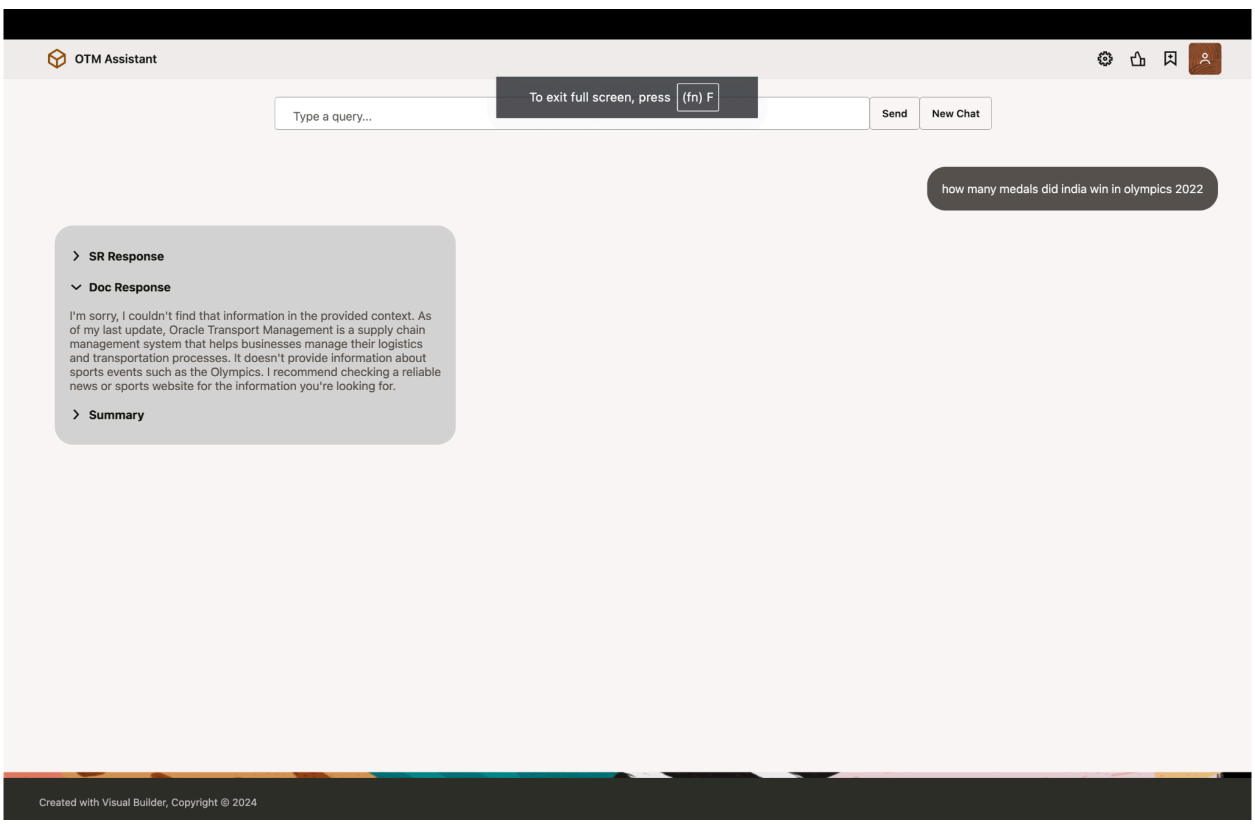
Implementation Details:

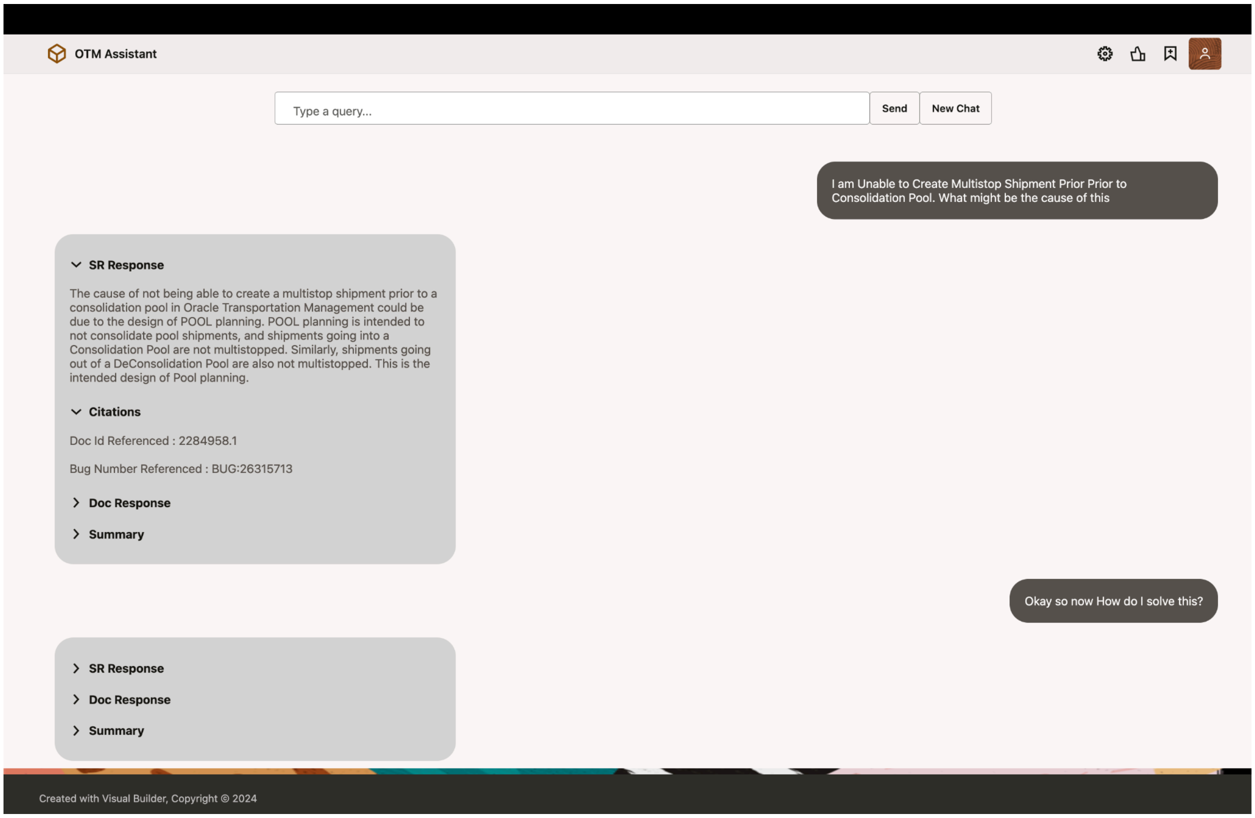
- **Service Connection:** A service connection was established by consuming the endpoint URL provided by Ngrok.
- **User Input:** An input box was added to allow users to submit their queries. The input is stored in a bind variable `[[ $variables.query ]]`, which is then sent to the backend.
- **Data Handling:** Both the user's input and the bot's responses are stored in an array within the `chats` variable, an object that contains two key properties: `current.is_user` and `current.data`.
- **Chat Display:** The chat container dynamically displays user inputs and bot responses, which are conditionally segregated based on the API's replies.
- **New Chat Functionality:** A "New Chat" button triggers an ActionChain that resets the `query` and `chats` variables, preparing the interface for a new interaction.

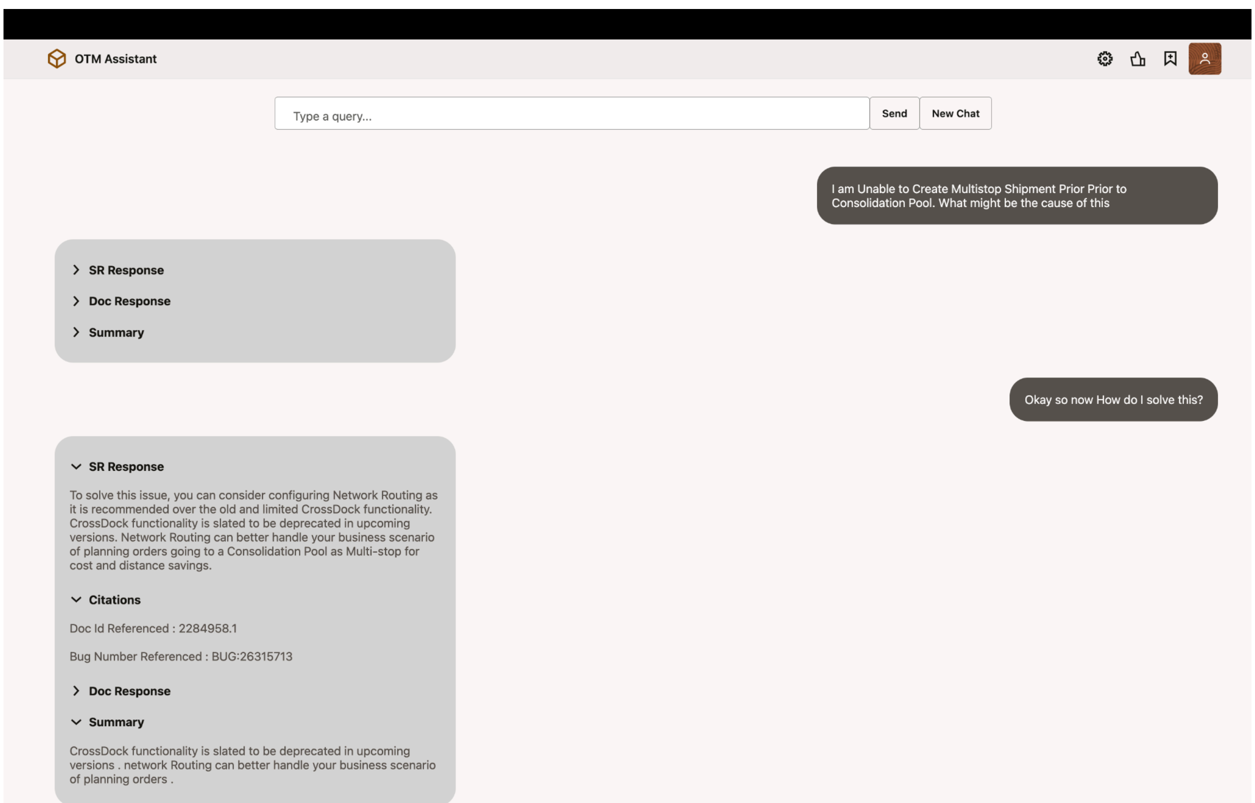

OTM Assistant
Type a query...
Send
New Chat
How to Enable Object Instrumentation and Profiling
SR Response
To enable Object Instrumentation and Profiling in Oracle Transportation Operational Planning, you should follow these steps: 1. Turn on the following properties to capture instrumentation: - glog.instrumentation.on=true - glog.instrumentation.instrumentor=glog.util.instrumentation.ObjectDistribution 2. Turn on the following properties to capture profiling statistics: - glog.instrumentation.profilingOn=true - glog.instrumentation.profiler=glog.util.instrumentation.ProfileFrameTimes 3. Run the bulk plan after setting these properties. 4. Once the bulk plan is complete, view the results through the following servlet: - <server>/GC3/glog.webserver.instrumentation.InstrumentationServlet 5. Capture the screenshots of the results and upload them to the Service Request (SR) for further review. Make sure to expand all sections if they are collapsed. Remember to remove these settings once you are done with the statistics, as they can cause the bulk plans to run slowly due to the overhead of collecting the statistics. This information should only be provided if requested by the OTM Support or Development Teams through a Service Request.
Citations
Doc Id Referenced : 1569057.1
Doc Response
Summary
Created with Visual Builder, Copyright © 2024


OTM Assistant
Type a query...
Send
New Chat
How to Enable Object Instrumentation and Profiling
SR Response
Doc Response
To enable object instrumentation and profiling in Oracle Transportation Management, you would need to follow the general steps for enabling such features in Oracle applications. However, the specific menu options or configuration settings to enable object instrumentation and profiling are not provided in the context information. Typically, enabling object instrumentation and profiling involves configuring profiling settings, setting up instrumentation for specific objects or events, and enabling the profiler. This can be done through the application's administrative interface or through specific configuration files. To find the specific steps for enabling object instrumentation and profiling in Oracle Transportation Management, you may need to refer to the application's installation and reference guides, or consult the Oracle support documentation for your specific version of the application. If you are unable to find the information you need, you can contact Oracle support for assistance.
Summary
support request documents,to enable Object Instrumentation and profiling in Oracle Transportation Management, you would need to follow these steps: 1. Turn on the following properties to capture profiling statistics . - glog.instrumentation.on=true - instrumentor=glog_util.instrumentations.ObjectDistribution 2. Run the bulk plan after setting these properties . 5. Capture the screenshots of the results and upload them to the Service Request for further review .
Created with Visual Builder, Copyright © 2024


OTM Assistant
Type a query...
Send
New Chat
How to Enable Object Instrumentation and Profiling
SR Response
Doc Response
Summary
Which version of OTM does the previous solution apply to?
SR Response
The solution for enabling Object Instrumentation and Profiling in Oracle Transportation Operational Planning applies to Version 6.0 and later.
Citations
Doc Id Referenced : 1569057.1
Doc Response
The previous solution does not specify a particular version of Oracle Transportation Management (OTM) as it provides general steps for enabling object instrumentation and profiling in Oracle applications. Therefore, the solution can be applied to any version of OTM, but the exact steps might vary depending on the version. It is always recommended to consult the OTM documentation or contact Oracle support for the specific version to ensure accurate guidance.
Summary


OTM Assistant
To exit full screen, press (fn) F
Type a query...
Send
New Chat
how many medals did india win in olympics 2022
SR Response
Doc Response
I'm sorry, I couldn't find that information in the provided context. As of my last update, Oracle Transport Management is a supply chain management system that helps businesses manage their logistics and transportation processes. It doesn't provide information about sports events such as the Olympics. I recommend checking a reliable news or sports website for the information you're looking for.
Summary
Created with Visual Builder, Copyright © 2024


OTM Assistant
Type a query...
Send
New Chat
I am Unable to Create Multistop Shipment Prior Prior to Consolidation Pool. What might be the cause of this
SR Response
The cause of not being able to create a multistop shipment prior to a consolidation pool in Oracle Transportation Management could be due to the design of POOL planning. POOL planning is intended to not consolidate pool shipments, and shipments going into a Consolidation Pool are not multistopped. Similarly, shipments going out of a DeConsolidation Pool are also not multistopped. This is the intended design of Pool planning.
Citations
Doc Id Referenced : 2284958.1
Bug Number Referenced : BUG:26315713
Doc Response
Summary
Okay so now How do I solve this?
SR Response
Doc Response
Summary
Created with Visual Builder, Copyright © 2024


OTM Assistant
Type a query...
Send
New Chat
I am Unable to Create Multistop Shipment Prior Prior to Consolidation Pool. What might be the cause of this
SR Response
Doc Response
Summary
Okay so now How do I solve this?
SR Response
To solve this issue, you can consider configuring Network Routing as it is recommended over the old and limited CrossDock functionality. CrossDock functionality is slated to be deprecated in upcoming versions. Network Routing can better handle your business scenario of planning orders going to a Consolidation Pool as Multi-stop for cost and distance savings.
Citations
Doc Id Referenced : 2284958.1
Bug Number Referenced : BUG:26315713
Doc Response
Summary
CrossDock functionality is slated to be deprecated in upcoming versions . network Routing can better handle your business scenario of planning orders .

SOME IMPORTANT POINTS:

- We are using the PDF_Convertor tool so that the same model can be instantiated twice which takes in PDFs. The pdfs are made as such they behave like HTML files, by which we mean, every header element is converted to a heading and the paragraphs are converted to paragraphs under the heading. We also capture the URL and store it in the pdf itself. Only text can be passed once we get more computing units.

- The Webcrawler tool prevents same links to come in since we are using a set data structure . Also recursion depth is kept tp 3 so that infinite loops can be prevented.

- We have improved the prompting process so that the OTM Assistant maintains context of the given documentation.

HOW TO RUN THE PROJECT

- Unzip the entire project
- BACKEND
    - Create virtualenv
      python3 -m venv .venv && source .venv/bin/activate

    - Install packages
      pip install -r requirements.txt

    - Environment variables
      env variables are in .env (All API keys- Groq, Qdrant etc.)

    - Run the server
      python3 index.py

    - The body of the request looks like this. (No authentication required)

      ```
      {
         "query": "What does the Bulk manager do in OTM"
      }
      ```

FUTURE ASPECTS:

- We can scale it by providing all the SRs and the whole public documentation.

- We can implement role-based access control so that SR response is only visible to the internal employees and public documentation response can be exposed to everyone.

- We can keep auto updating the SR responses. The chat history can be put as another SR that has been solved.

- We can store past queries for reference

- We can integrate a feedback loop for continuous improvement of the model.