Final Presentation: Atorus Research RAG Application

By Michael Collins, Patrick Donlon, Spencer Colón, Ryan Calderone, Efaz Ertesum, and Connor Gabe

Atorus Research’s Product: OpenVal

- OpenVal is an Atorus Research product that validates open-source R packages for clinical research.
- Packages are thoroughly reviewed by Atorus' expert team to meet industry and regulatory standards.
- Ensures healthcare and pharma organizations have reliable tools for:
  - Accurate data analysis
  - Regulatory compliance

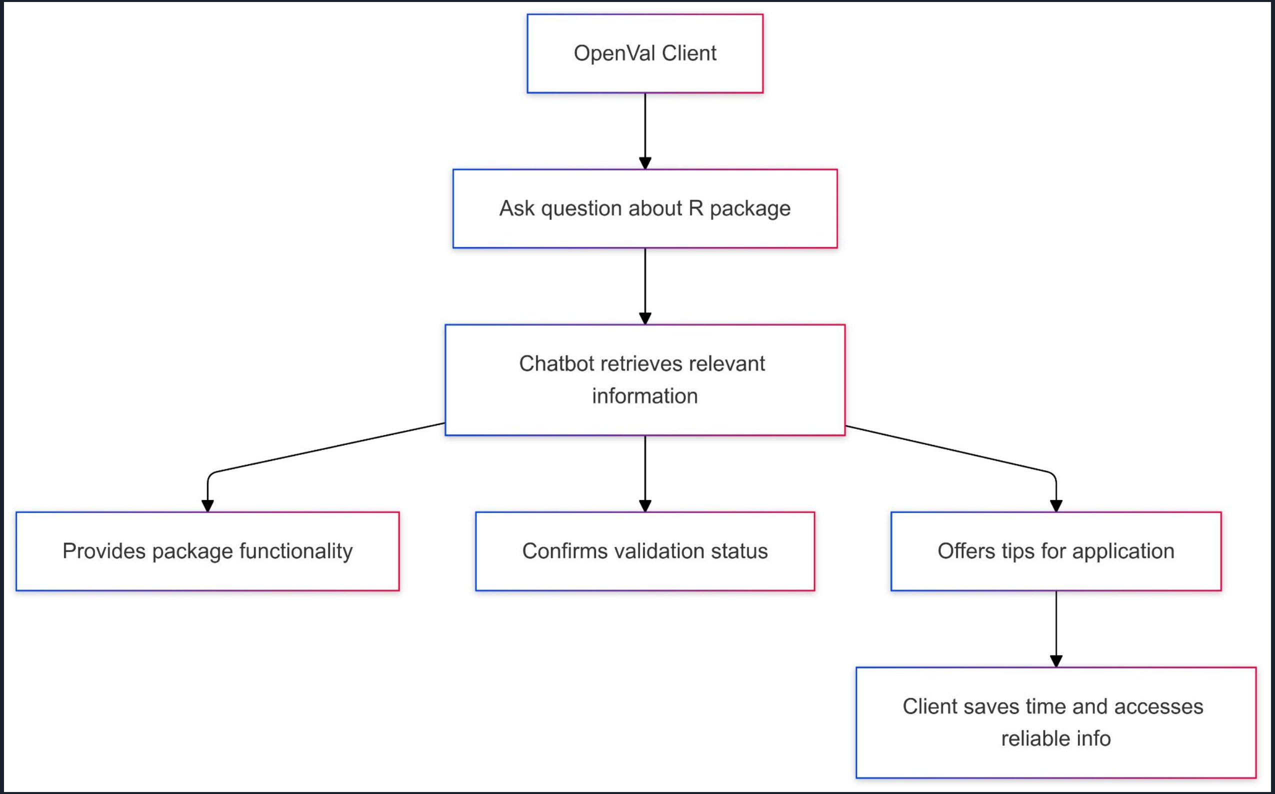
Spencer

Chatbot Goals

- Create a prototype chatbot to improve OpenVal user experience.
- Provides quick access to:
  - R package details and functionalities
  - Insights on Atorus' validation process
  - Practical guidance for effective package use
- Saves time, reduces complexity, and helps clients maximize OpenVal's value.

Spencer

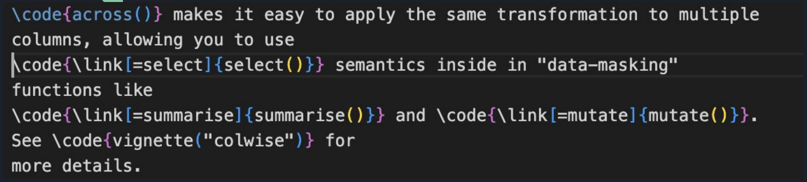
User Story


OpenVal Client
Ask question about R package
Chatbot retrieves relevant information
Provides package functionality
Confirms validation status
Offers tips for application
Client saves time and accesses reliable info

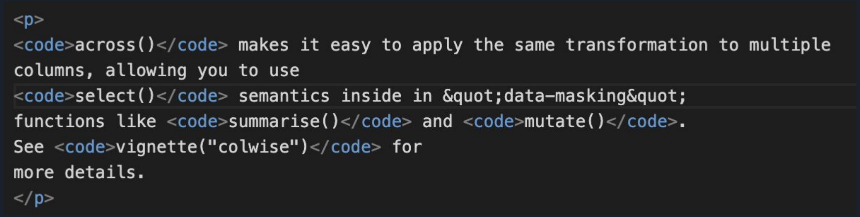
Spencer

Live Demo

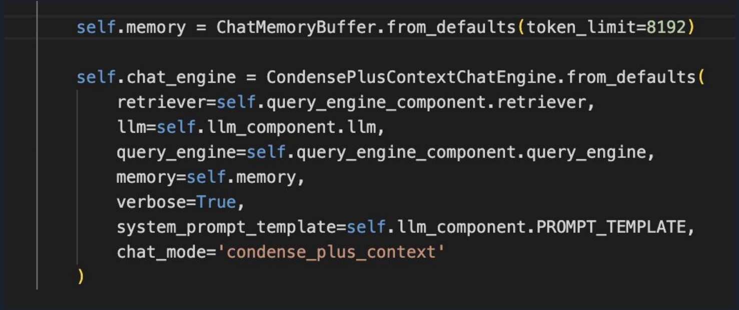
Backend Achievement - Functional Services



```
\code{across()} makes it easy to apply the same transformation to multiple
columns, allowing you to use
\code{\link[=select]{select()}} semantics inside in "data-masking"
functions like
\code{\link[=summarise]{summarise()}} and \code{\link[=mutate]{mutate()}}.
See \code{vignette("colwise")} for
more details.
```



```
<p>
<code>across()</code> makes it easy to apply the same transformation to multiple
columns, allowing you to use
<code>select()</code> semantics inside in &quot;data-masking&quot;
functions like <code>summarise()</code> and <code>mutate()</code>.
See <code>vignette("colwise")</code> for
more details.
</p>
```



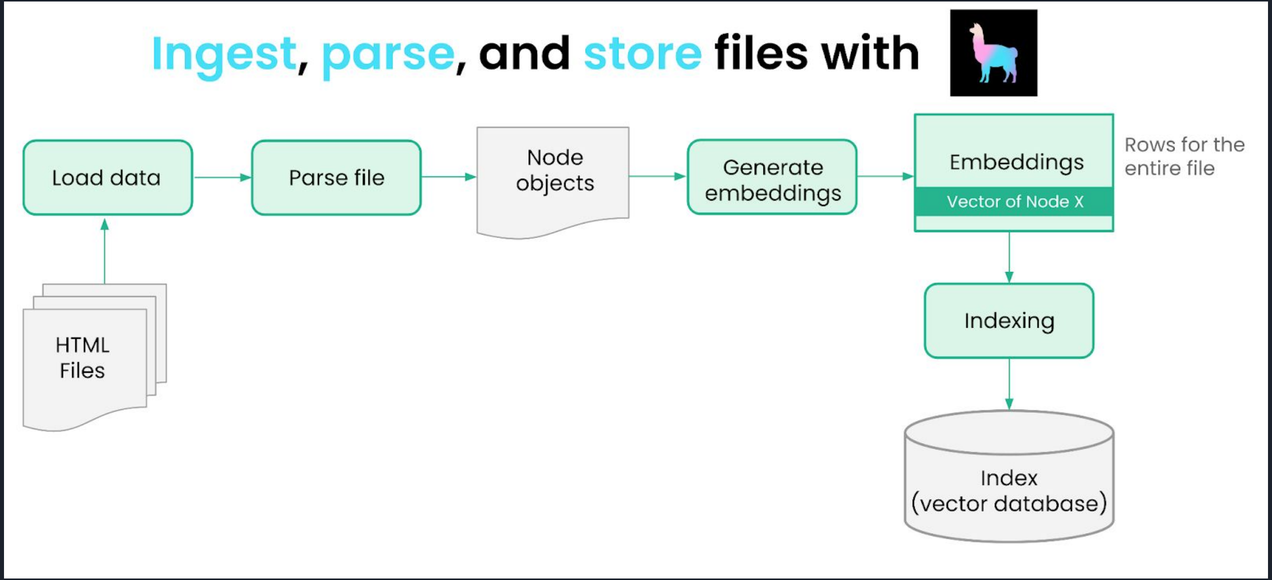
```
    self.memory = ChatMemoryBuffer.from_defaults(token_limit=8192)

    self.chat_engine = CondensePlusContextChatEngine.from_defaults(
        retriever=self.query_engine_component.retriever,
        llm=self.llm_component.llm,
        query_engine=self.query_engine_component.query_engine,
        memory=self.memory,
        verbose=True,
        system_prompt_template=self.llm_component.PROMPT_TEMPLATE,
        chat_mode='condense_plus_context'
    )
```

- Data Ingestion:
  - Ingesting documentation for 60 R packages
  - Handling .Rmd files as Markdown and converting .Rd files to .html.
  - Pipeline is ready for Atorus devs to scale

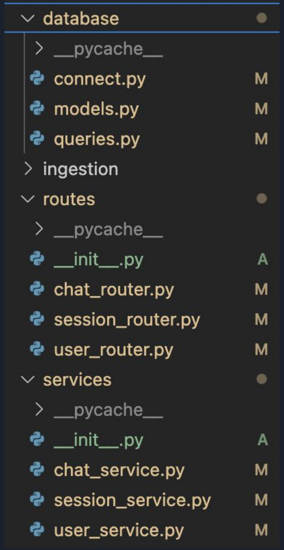
- Chat Engine Integration:
  - Query Engine -> Chat Engine
    - Multi-turn conversations using memory buffers and conversation history

Michael

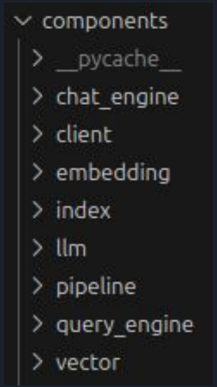

Ingest, parse, and store files with
Load data
Parse file
Node objects
Generate embeddings
Embeddings
Vector of Node X
Rows for the entire file
HTML Files
Indexing
Index (vector database)

Michael

Backend Achievement - Modular Structure



```
∨ database
  > __pycache__
  connect.py            M
  models.py             M
  queries.py            M
> ingestion
∨ routes
  > __pycache__
  __init__.py           A
  chat_router.py        M
  session_router.py     M
  user_router.py        M
∨ services
  > __pycache__
  __init__.py           A
  chat_service.py       M
  session_service.py    M
  user_service.py       M
```

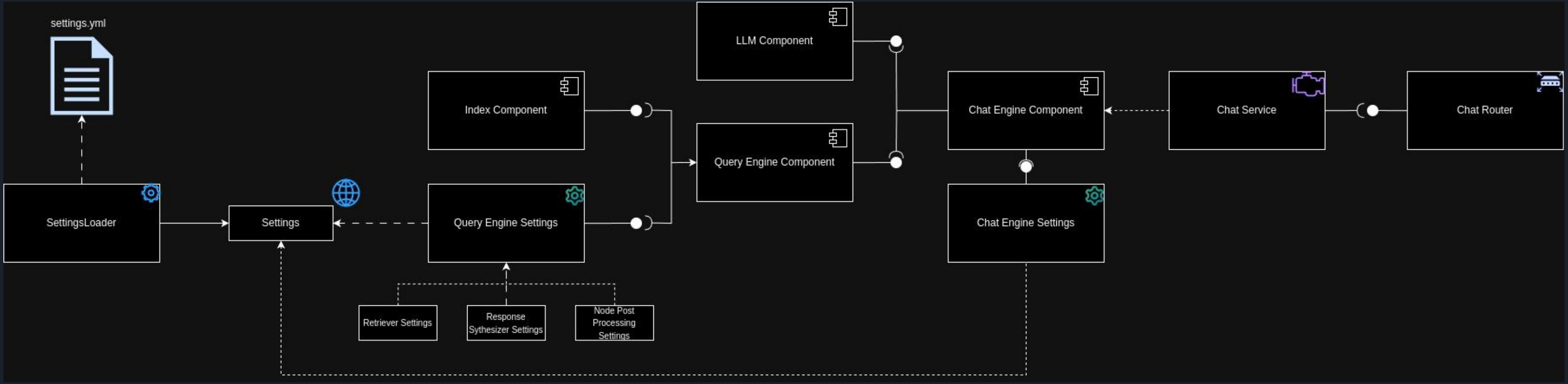


```
∨ components
  > __pycache__
  > chat_engine
  > client
  > embedding
  > index
  > llm
  > pipeline
  > query_engine
  > vector
```

- Design Patterns (Gamma, Helm, Johnson, Vissides)
  - **Singleton** - Global Settings
  - **Dependency Injection** - Services
- Modularity allows for faster adoption of new features
  - **Chat Engine -** Builds upon query engine
  - **Ingestion Pipeline -** Builds upon embedding, vector, client
- Reusability allows for easier debugging
  - New features can rely on **tested components lower in hierarchy**
  - Breaking code into s**mall digestible chunks**, allows for faster development and bug fixing

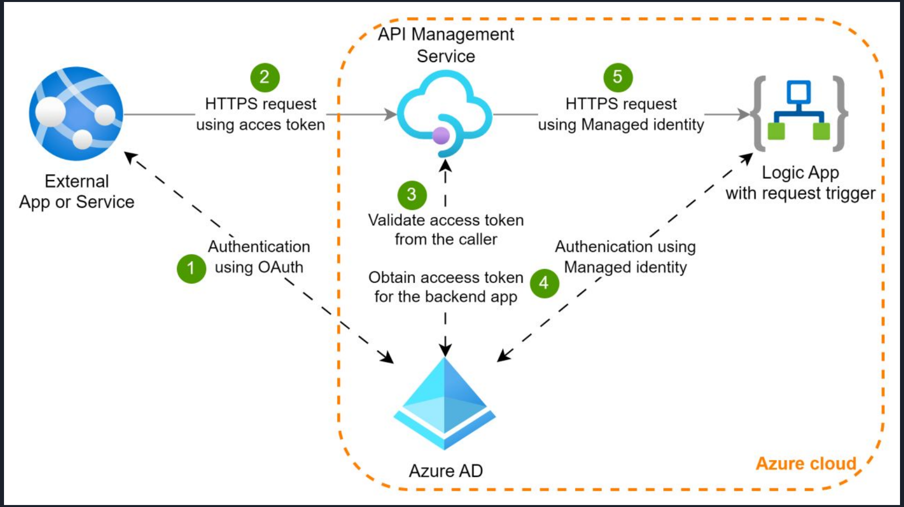
Patrick

Backend Achievement - Modular Structure Cont.



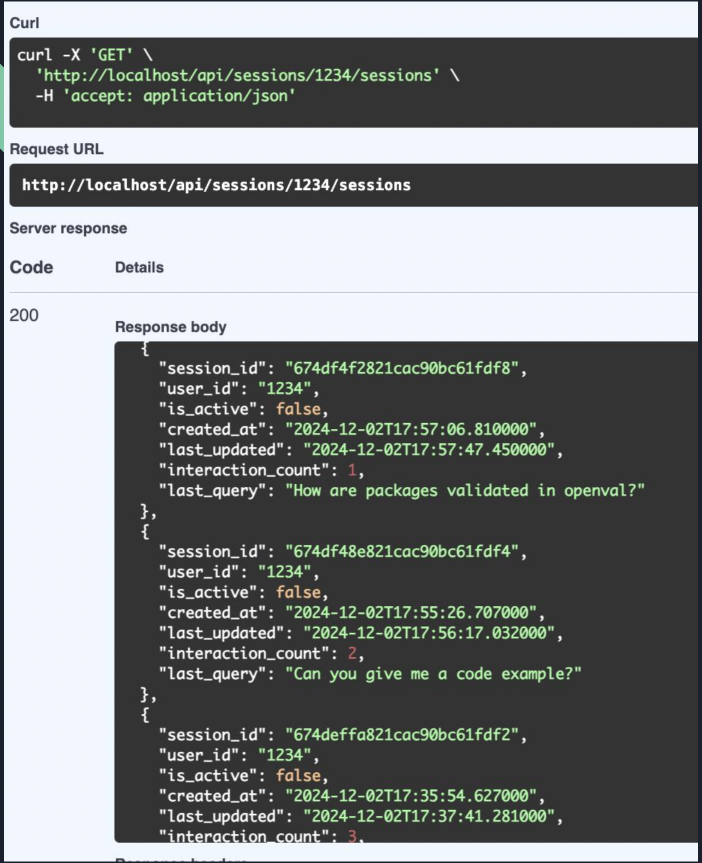
Patrick

Backend Weakness - Authentication/User Storage


API Management Service
2 HTTPS request using acces token
5 HTTPS request using Managed identity
External App or Service
Logic App with request trigger
3 Validate access token from the caller
1 Authentication using OAuth
Authenication using Managed identity
Obtain acceess token for the backend app 4
Azure AD
Azure cloud

- No Authentication with Atorus' Azure Active Directory
  - Out of scope but still hoped to get it done
  - Found it would be best for devs at Atorus to set this up.
  - **Fix -** In the scope of the project, the demo does not need to have authentication set up.
- User Storage partially working
  - We are still working on setting up user storage and session storage.
  - **Fix -** Continuing to work on integrating this with frontend.
- General Security Concerns
  - Prompt Injection
  - API access
  - **Fix -** Atorus told us security concerns are not in the scope of the project

Michael


Curl

```
curl -X 'GET' \
  'http://localhost/api/sessions/1234/sessions' \
  -H 'accept: application/json'
```

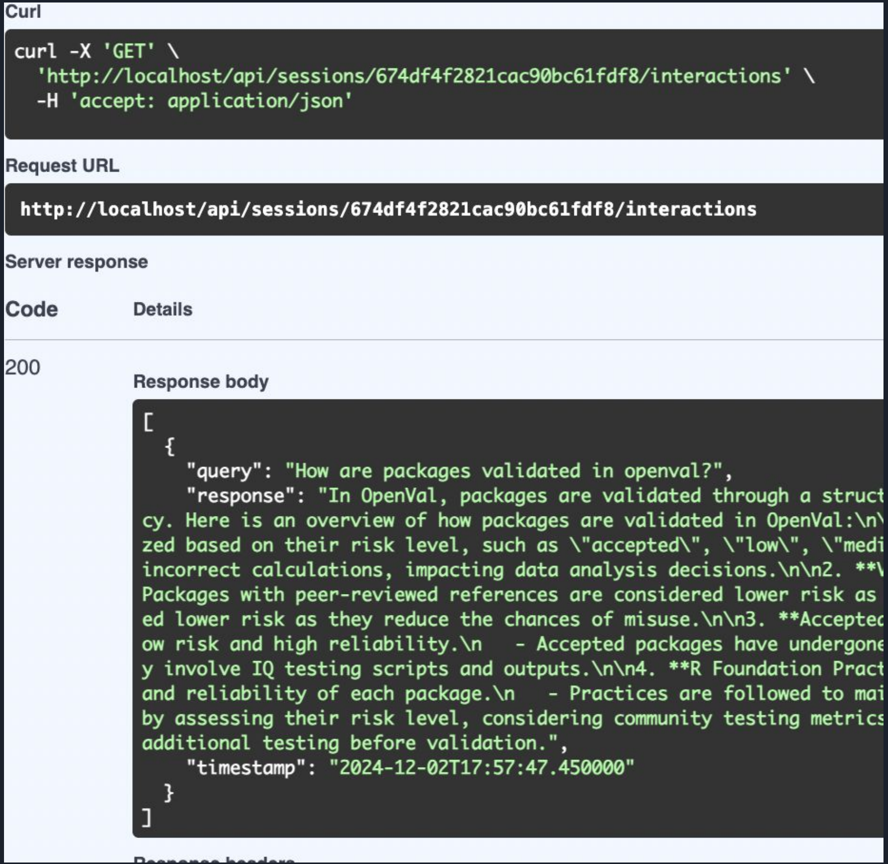
Request URL

```
http://localhost/api/sessions/1234/sessions
```

Server response

Code	Details
200	Response body

```
  {
    "session_id": "674df4f2821cac90bc61fdf8",
    "user_id": "1234",
    "is_active": false,
    "created_at": "2024-12-02T17:57:06.810000",
    "last_updated": "2024-12-02T17:57:47.450000",
    "interaction_count": 1,
    "last_query": "How are packages validated in openval?"
  },
  {
    "session_id": "674df48e821cac90bc61fdf4",
    "user_id": "1234",
    "is_active": false,
    "created_at": "2024-12-02T17:55:26.707000",
    "last_updated": "2024-12-02T17:56:17.032000",
    "interaction_count": 2,
    "last_query": "Can you give me a code example?"
  },
  {
    "session_id": "674deffa821cac90bc61fdf2",
    "user_id": "1234",
    "is_active": false,
    "created_at": "2024-12-02T17:35:54.627000",
    "last_updated": "2024-12-02T17:37:41.281000",
    "interaction_count": 3,
```


Curl

```
curl -X 'GET' \
  'http://localhost/api/sessions/674df4f2821cac90bc61fdf8/interactions' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost/api/sessions/674df4f2821cac90bc61fdf8/interactions
```

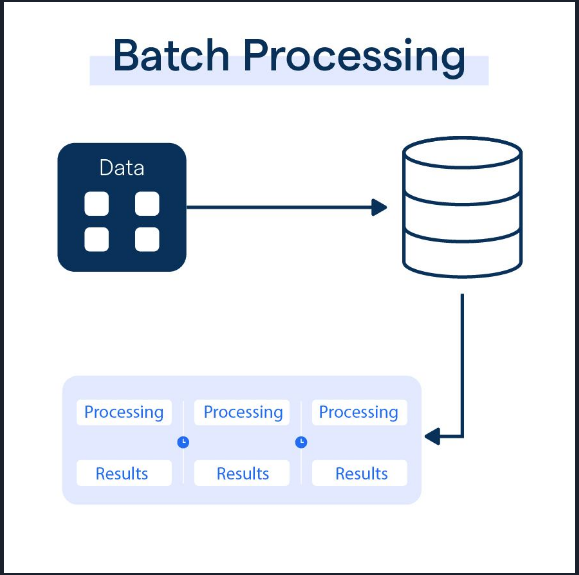
Server response

Code	Details
200	Response body

```
[
  {
    "query": "How are packages validated in openval?",
    "response": "In OpenVal, packages are validated through a struct
cy. Here is an overview of how packages are validated in OpenVal:\n\
zed based on their risk level, such as \"accepted\", \"low\", \"medi
incorrect calculations, impacting data analysis decisions.\n\n2. **V
Packages with peer-reviewed references are considered lower risk as
ed lower risk as they reduce the chances of misuse.\n\n3. **Accepted
ow risk and high reliability.\n   - Accepted packages have undergone
y involve IQ testing scripts and outputs.\n\n4. **R Foundation Pract
and reliability of each package.\n   - Practices are followed to mai
by assessing their risk level, considering community testing metrics
additional testing before validation.",
    "timestamp": "2024-12-02T17:57:47.450000"
  }
]
```

Michael

Backend Weakness - Speed (Ingestion / Chat)

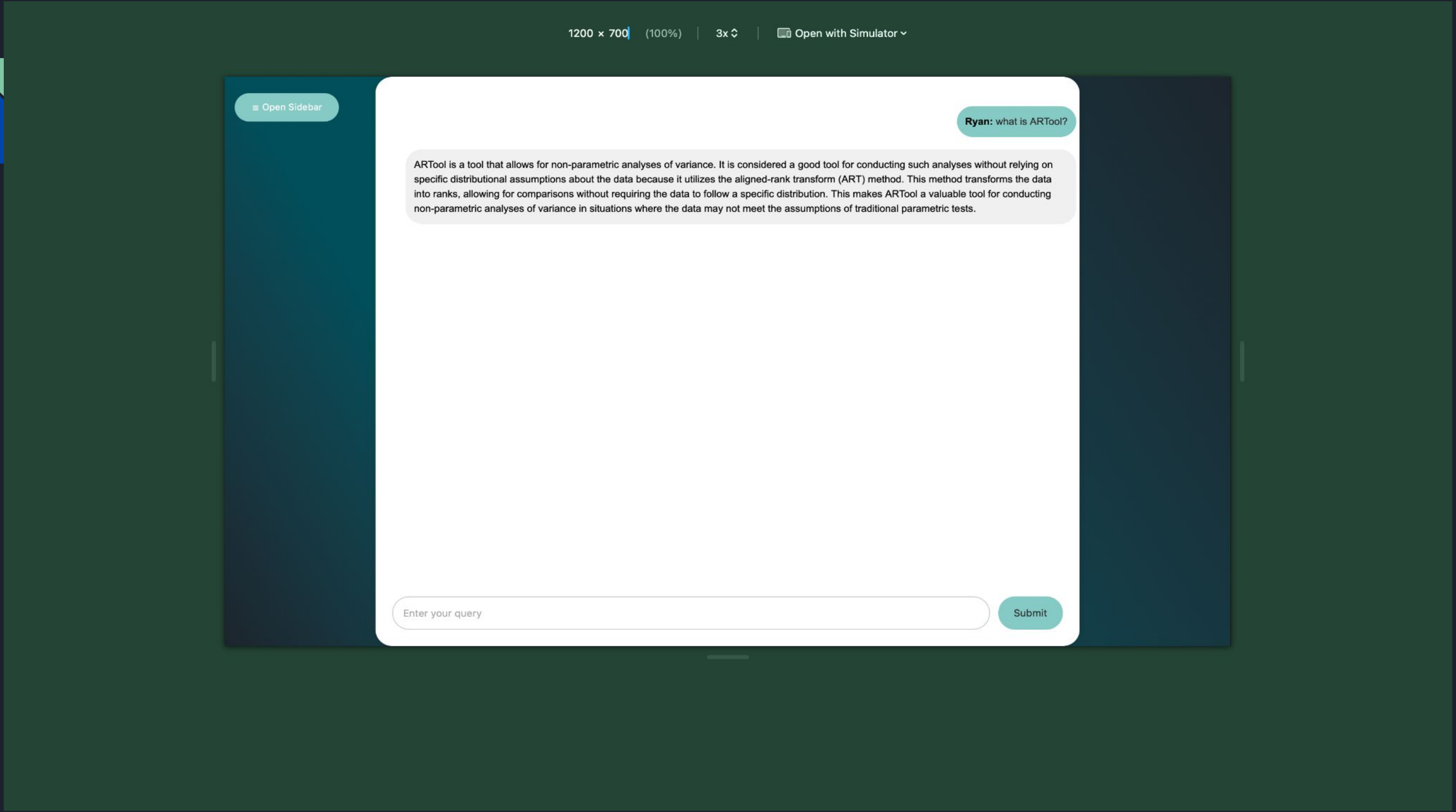


- Rate Limiting with Ingestion
  - Ingestion large amount of text will result in rate limit from OpenAI
  - Fix - OpenAI Batch API allows for async batch processing at reduced cost
- Lack of Parallel Processing
  - Reliance on CPU power leads to long wait time for ingestion and buffering for chat service
  - **Fix** - LLamaindex CUDA functionality through LlamaCPP, would need established deployment architecture to implement
- Chat Latency
  - Fix - Implement “Streaming” to reduce user perceived latency i.e ChatGPT

Patrick

Frontend - Achievement #1

- Aspect Ratio adjustments for smaller devices.



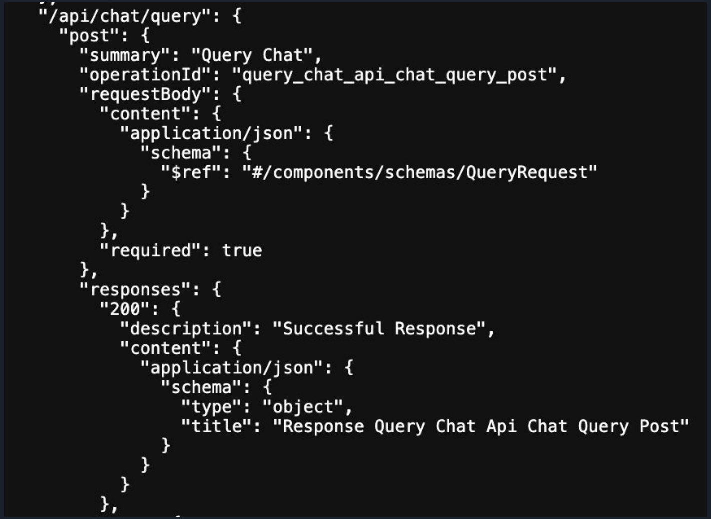
Ryan: what is ARTool?

ARTool is a tool that allows for non-parametric analyses of variance. It is considered a good tool for conducting such analyses without relying on specific distributional assumptions about the data because it utilizes the aligned-rank transform (ART) method. This method transforms the data into ranks, allowing for comparisons without requiring the data to follow a specific distribution. This makes ARTool a valuable tool for conducting non-parametric analyses of variance in situations where the data may not meet the assumptions of traditional parametric tests.

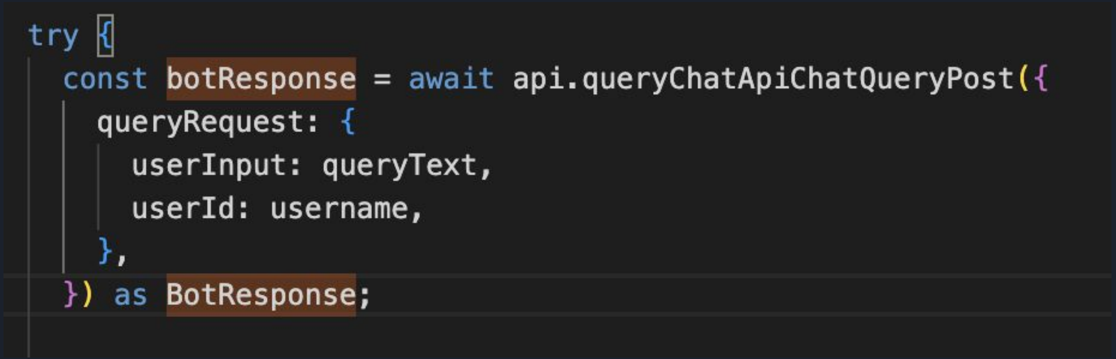
Efaz

Frontend - Achievement #2 - OpenAPI

- **Officially connected the frontend to the backend**
  - **Users make request to /api/chat/query endpoint through frontend UI**
- **This was accomplished by providing an OpenAPI.json schema to a Client API generator**
  - **We import the generated functions to make calls to our endpoints.**

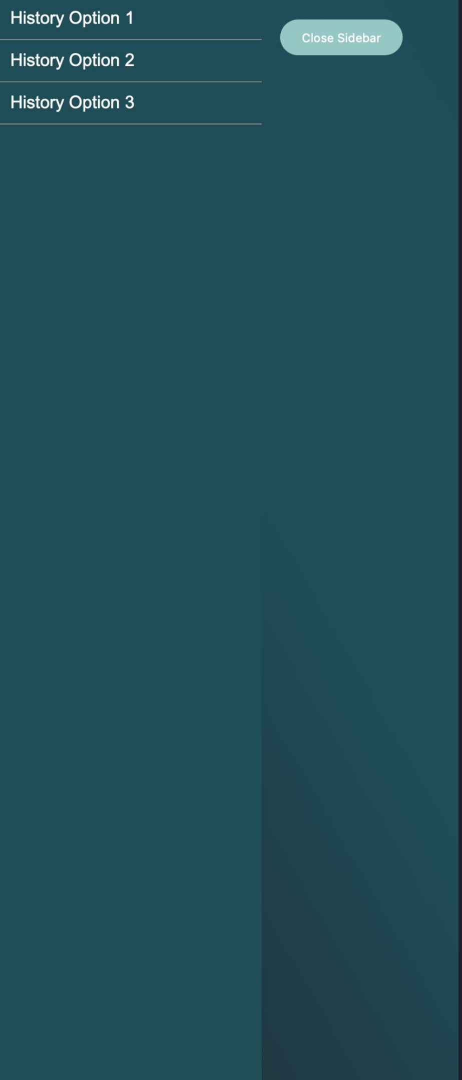


```
"/api/chat/query": {
  "post": {
    "summary": "Query Chat",
    "operationId": "query_chat_api_chat_query_post",
    "requestBody": {
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/QueryRequest"
          }
        }
      },
      "required": true
    },
    "responses": {
      "200": {
        "description": "Successful Response",
        "content": {
          "application/json": {
            "schema": {
              "type": "object",
              "title": "Response Query Chat Api Chat Query Post"
            }
          }
        }
      },
```



```
try {
  const botResponse = await api.queryChatApiChatQueryPost({
    queryRequest: {
      userInput: queryText,
      userId: username,
    },
  }) as BotResponse;
```

Ryan

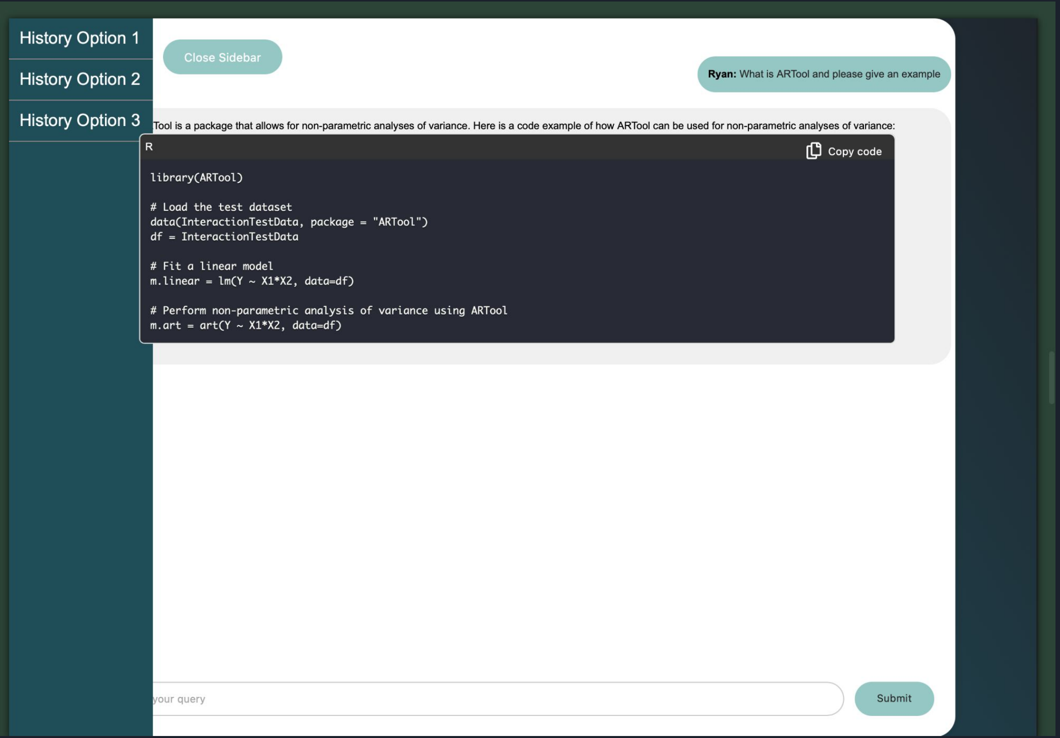


Frontend - Weakness #1

- The sidebar is not integrated with the backend
- We have a placeholder for the sidebar
- We want the sidebar to have previous chat history and when one is clicked, leads to a previous chat that the user2

Ryan

Frontend - Weakness #2



Ryan: What is ARTool and please give an example

Tool is a package that allows for non-parametric analyses of variance. Here is a code example of how ARTool can be used for non-parametric analyses of variance:

```
library(ARTool)

# Load the test dataset
data(InteractionTestData, package = "ARTool")
df = InteractionTestData

# Fit a linear model
m.linear = lm(Y ~ X1*X2, data=df)

# Perform non-parametric analysis of variance using ARTool
m.art = art(Y ~ X1*X2, data=df)
```

- The newly added sidebar does not work with the aspect ratio changes
- The button and sidebar gets in the way of the chat interface
- When the button is clicked, the sidebar sticks out, making the sidebar out of place

Ryan

Summary

Ultimately, we have a product that has the necessary functionality to demo to clients and gauge interest.

- Ready for production?
  - No! In the scope of our project, we do not need to have a production ready product.
- Should they invest more to deploy/maintain it?
  - Absolutely. We have created a great foundation to build off of.
    - It is well organized and maintainable. We are working on writing up final documentation.
  - Data Scientists will need to pick up on the ingestion side of the project to improve the data going into OpenSearch
- What has been accomplished?
  - We have effectively built a demo for Atorus' OpenVal RAG application.

- What has been learned?
  - Backend
    - We have learned the RAG framework in depth, backend development fundamentals, API development, Database integration, Docker
  - Frontend
    - We have learned how to develop user friendly UI and integrate it with the backend.
    - We learned how to use react JS
  - Team
    - We have learned how to effectively communicate with a client and build a relationship. We learned how to organize a large group to get tasks done.
- Is the client satisfied?
  - Yes! They were very happy in last week's demo.

Connor

Q&A Session