

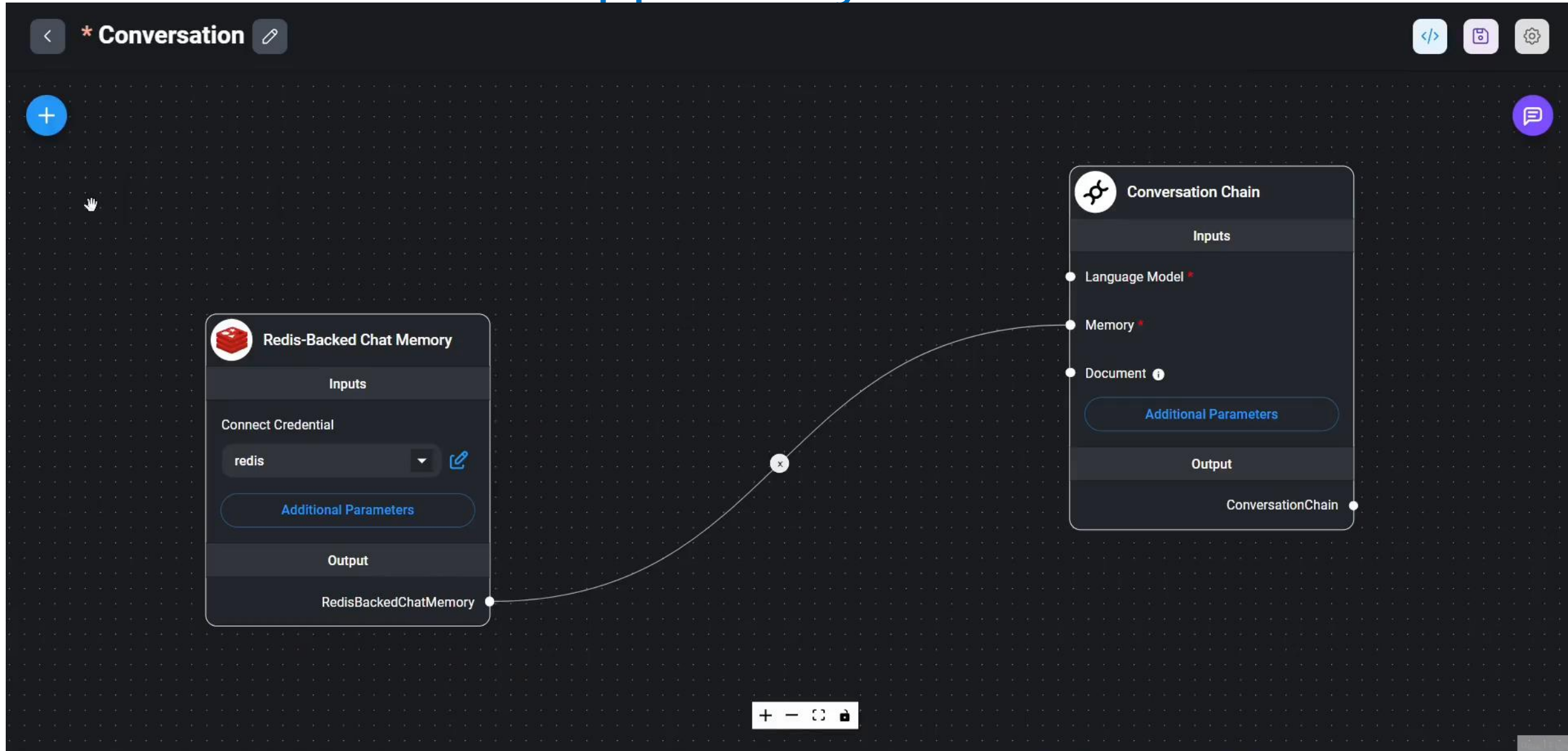
# 8. 노코딩 시애플 빌드

How to  
**Build an App**  
Without Coding

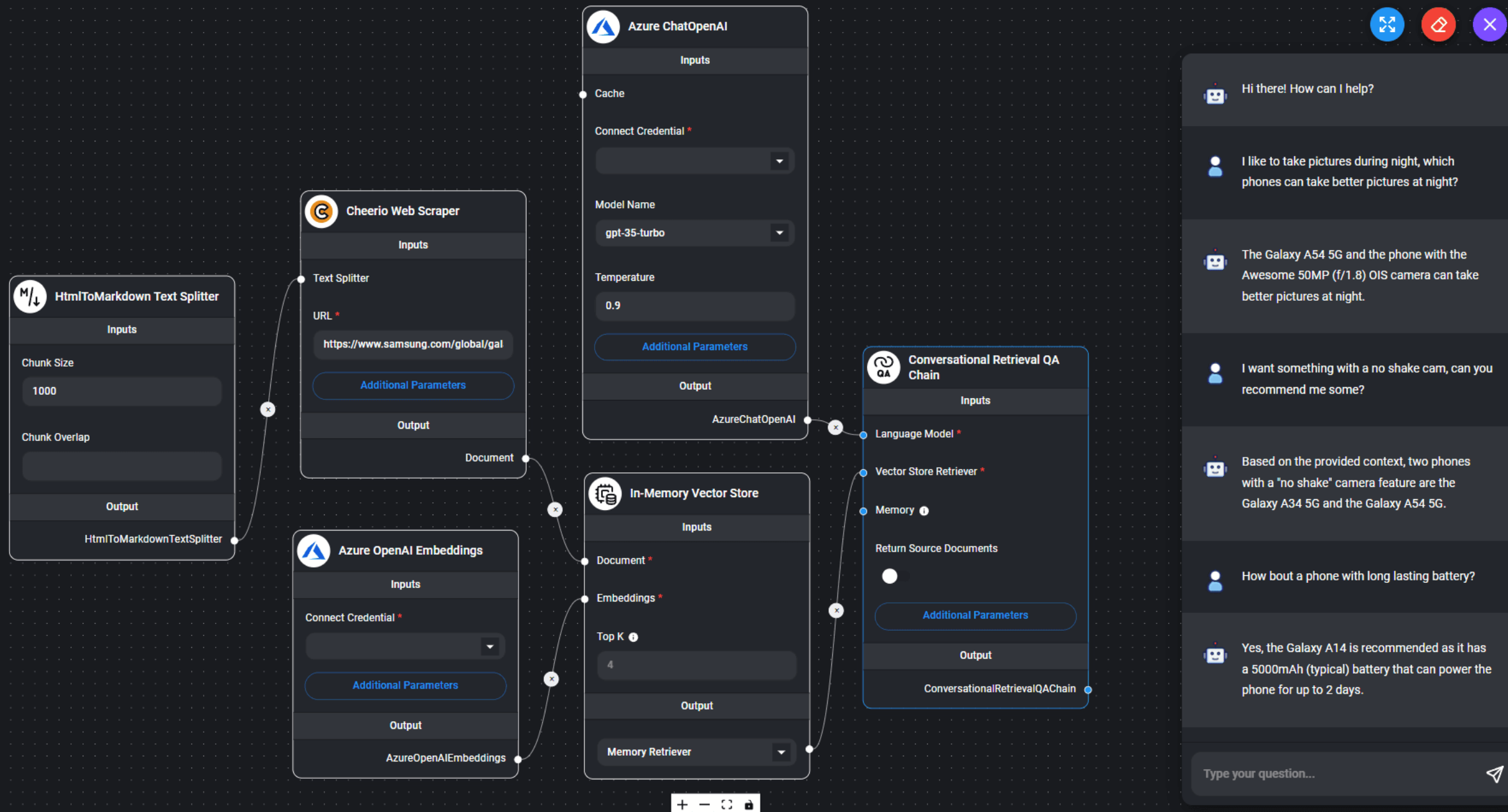
---




# Flowise – Build LLM Apps Easily



# 유스케이스- 제품과 관련된 모든 질문에 답변하는 제품 카탈로그 챗봇



# 유스케이스- 주어진 제품 사양에서 제품 설명 생성

 ChatOpenAI

Inputs

Cache

Connect Credential \*

Model Name

gpt-3.5-turbo


Temperature

0.9

Additional Parameters

Output

ChatOpenAI

 Prompt Template

Inputs

Langchain Hub

Template \*


You are a product description writer. Using the following specification, write a product description in 50 words.

Format Prompt Values

Format Prompt Values

Output

PromptTemplate

 LLM Chain

Inputs

Language Model \*

Prompt \*

Output Parser

Input Moderation ⓘ

Chain Name

Name Your Chain

Output

LLM Chain

Hi there! How can I help?

6.9" edge Quad HD+ Dynamic AMOLED 2X (3088x1440)  
496 ppi  
Infinity-O Display

Introducing our cutting-edge 6.9" edge Quad HD+ Dynamic AMOLED 2X display with an impressive 3088x1440 resolution. With a pixel density of 496 ppi, this Infinity-O display delivers stunning visuals and an immersive viewing experience. Experience the future of display technology with our top-of-the-line product.

- 12.0 MP + 12.0 MP Rear Camera Resolution (Multiple)
- F1.8 , F2.2 Rear Camera F Number (Multiple)
- Rear Camera Auto Focus
- Rear Camera OIS
- Digital Zoom up to 10x
- 10.0 MP Front Camera Resolution
- UHD 4K (3840 x 2160) @60fps Video Recording Resolution

Type your question...

# 유스케이스- 자연어를 사용하여 SQL 데이터베이스 쿼리하기



Hi there! How can I help?

how many rows are there

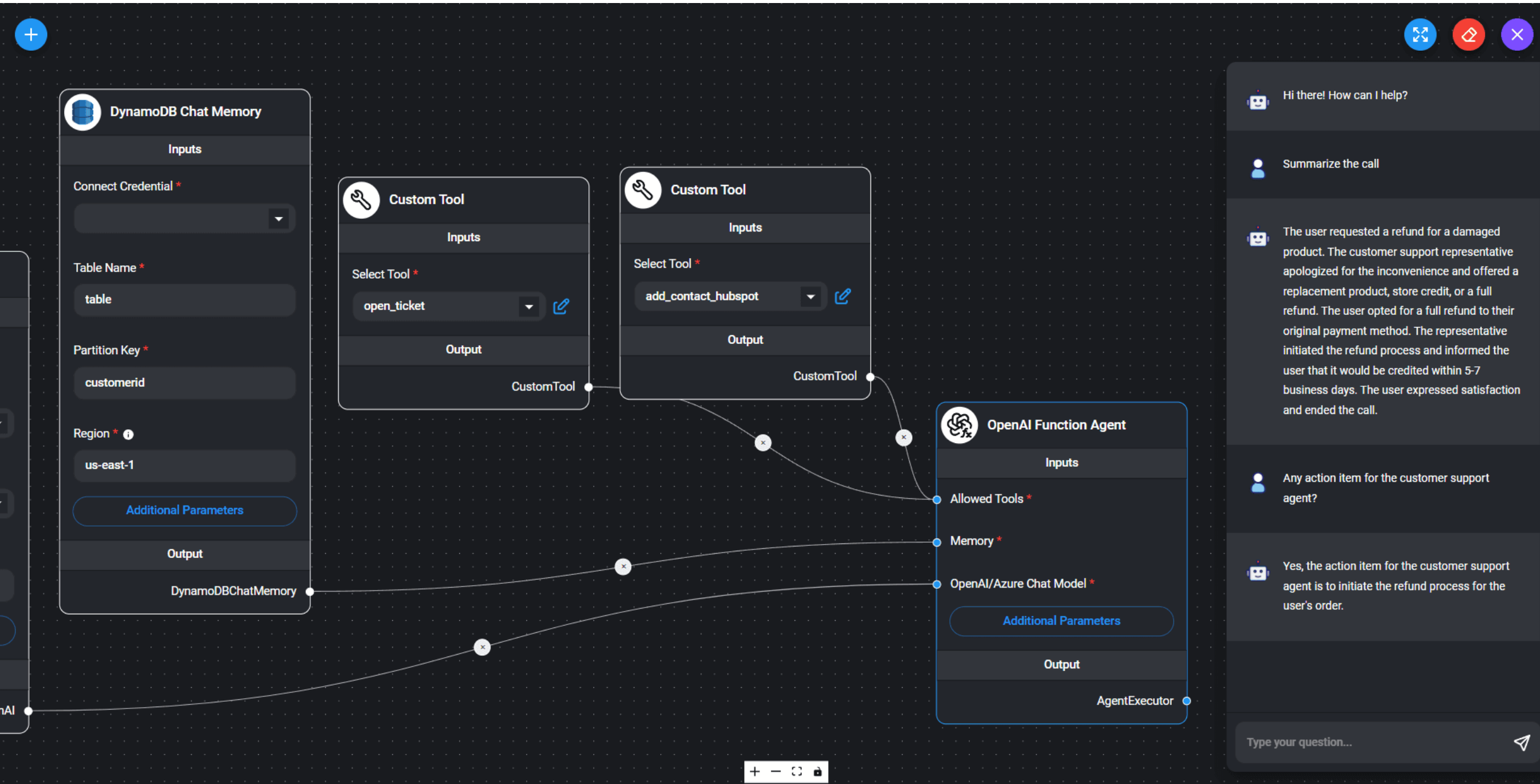
There are a total of 5 rows.

what is the address of steven

Steven's address is 2, 120 Jefferson St.

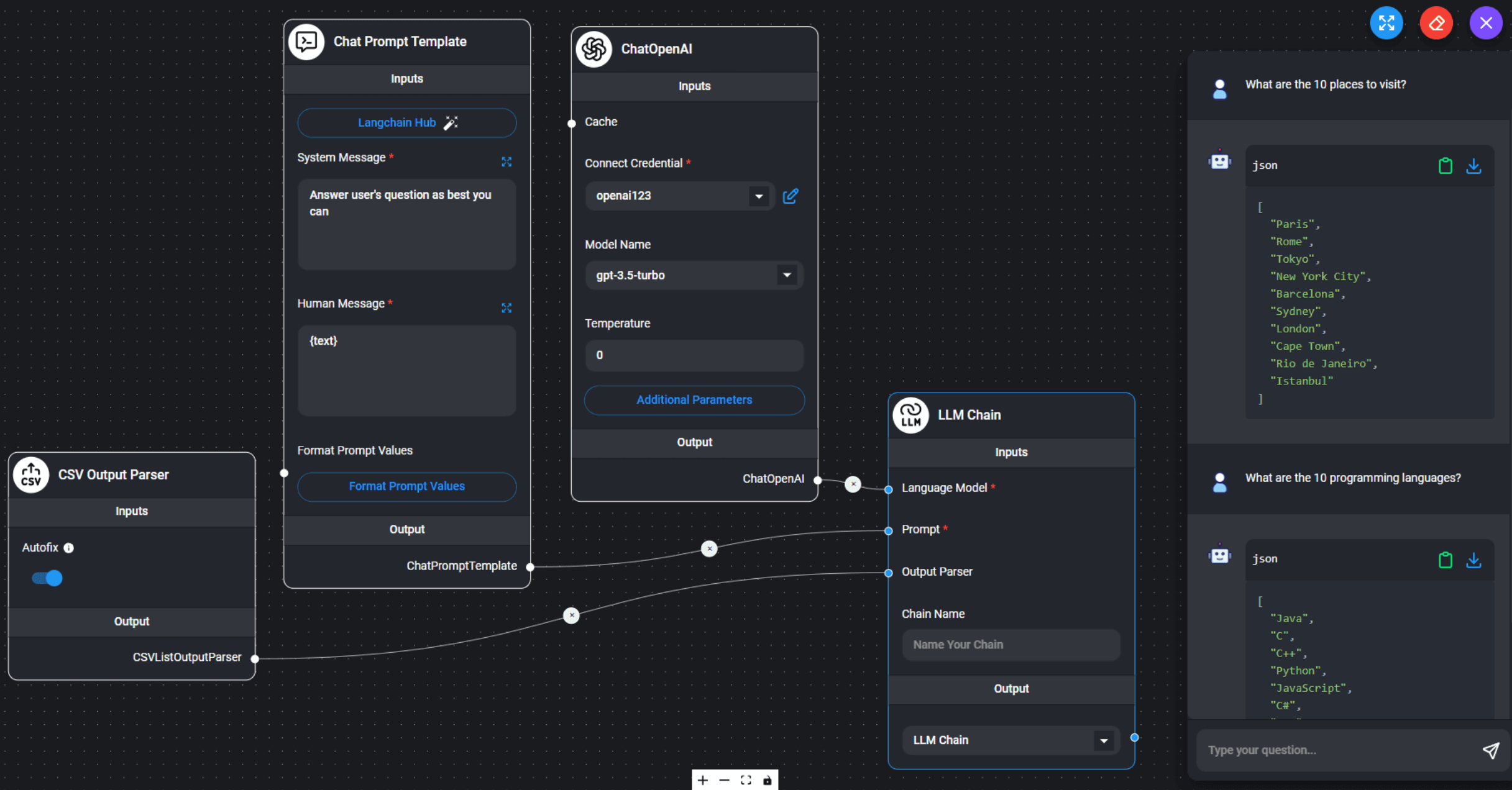
Type your question...

# 유스케이스- 고객 지원 티켓에서 후속 작업 요약 및 만들기



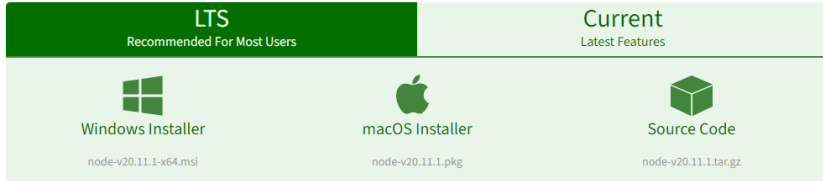


# 유스케이스- 다운스트림 작업에 사용할 구조화된 데이터 생성 또는 추출



# Flowise 설치

## NodeJS 설치



## ⚡ Quick Start

### 1. Flowise 설치

```
npm install -g flowise
```

### 2. Flowise 시작

```
npx flowise start
```

### 3. <http://localhost:3000> 접속

## Developers (선택사항)

### 1. Yarn 설치

```
npm i -g yarn
```

### 2. Repository 복제

```
git clone
```

```
https://github.com/FlowiseAI/Flowise.git
```

### 3. 모듈 설치

```
cd Flowise
```

```
yarn install
```

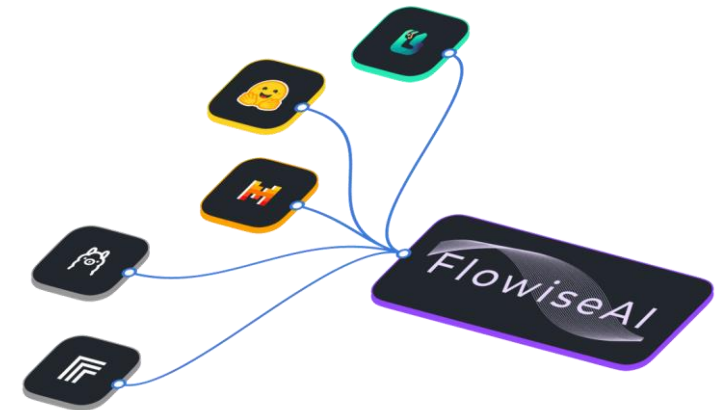
### 4. 빌드

```
yarn build
```

### 5. Flowise 시작

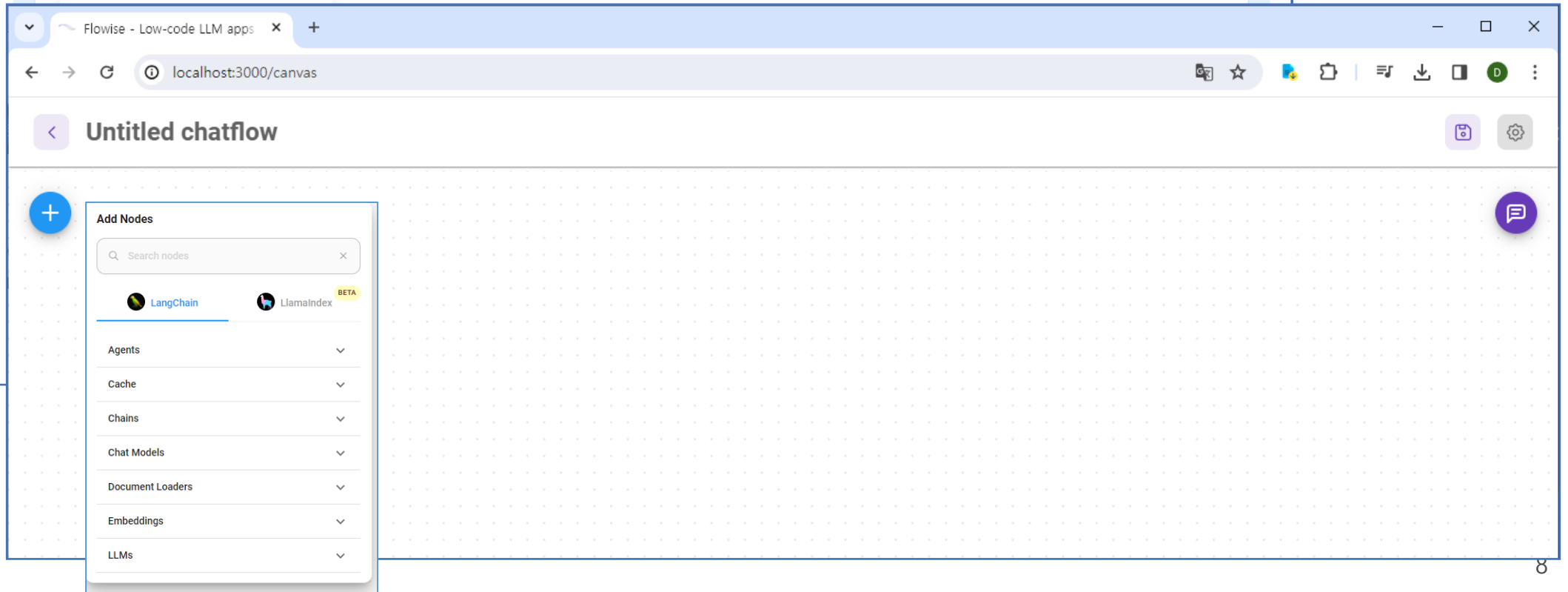
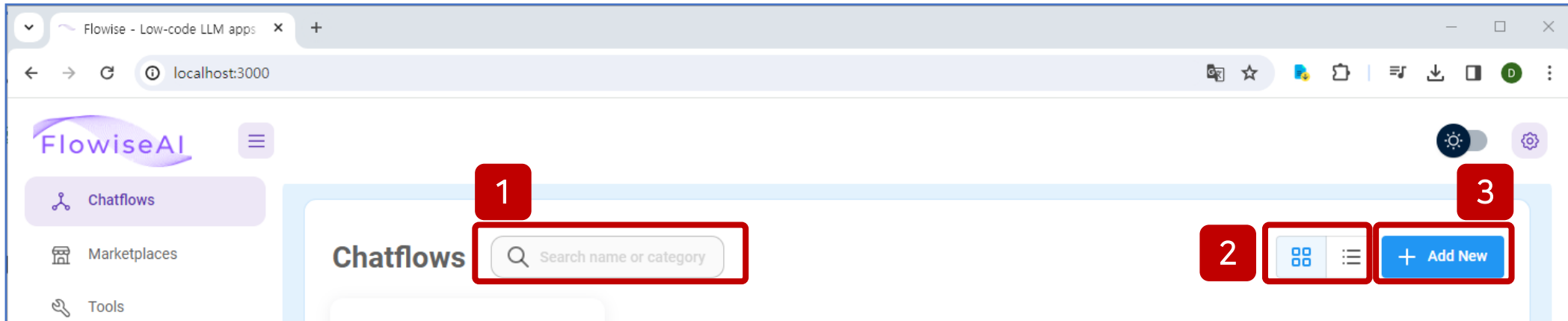
```
yarn start
```

### 6. <http://localhost:3000> 접속





# Chatflows



# Marketplace

The image displays two screenshots of the FlowiseAI interface, showing the Marketplace and Chatflows sections.

**Top Screenshot: Marketplace**

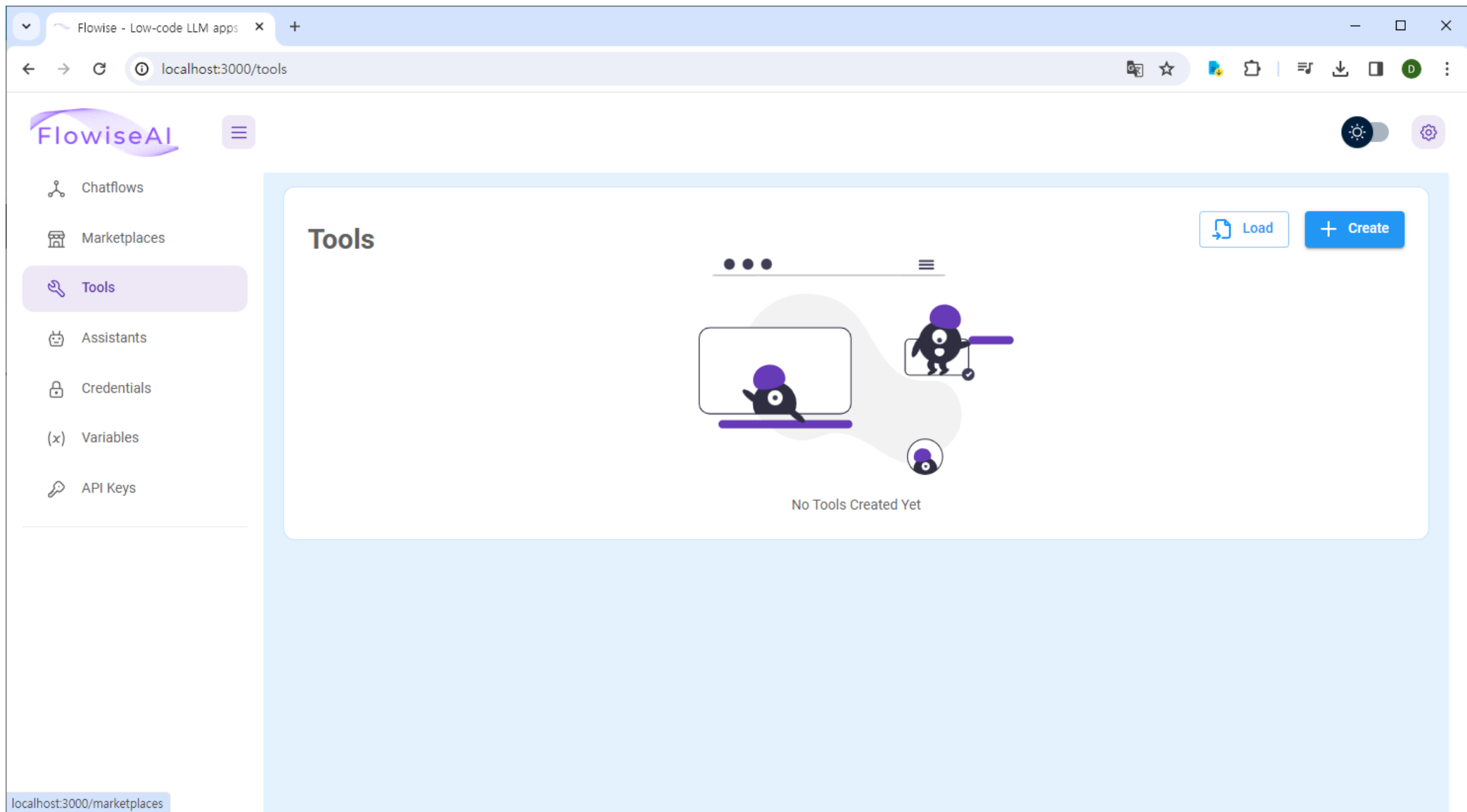
- URL: localhost:3000/marketplaces
- Search bar: Search name or description or node name
- Buttons: Show Filters, Use Template
- Templates displayed:
  - Add Hubspot Contact**: Add new contact to Hubspot
  - Advanced Structured Output Parser** (NEW): Return response as a JSON structure as specified by a Zod
  - Antonym**: Output antonym of given user input using few-shot prompt template built with examples
  - API Agent**: Given API docs, agent automatically decides call, generating url, request from conversation

**Bottom Screenshot: Chatflows**

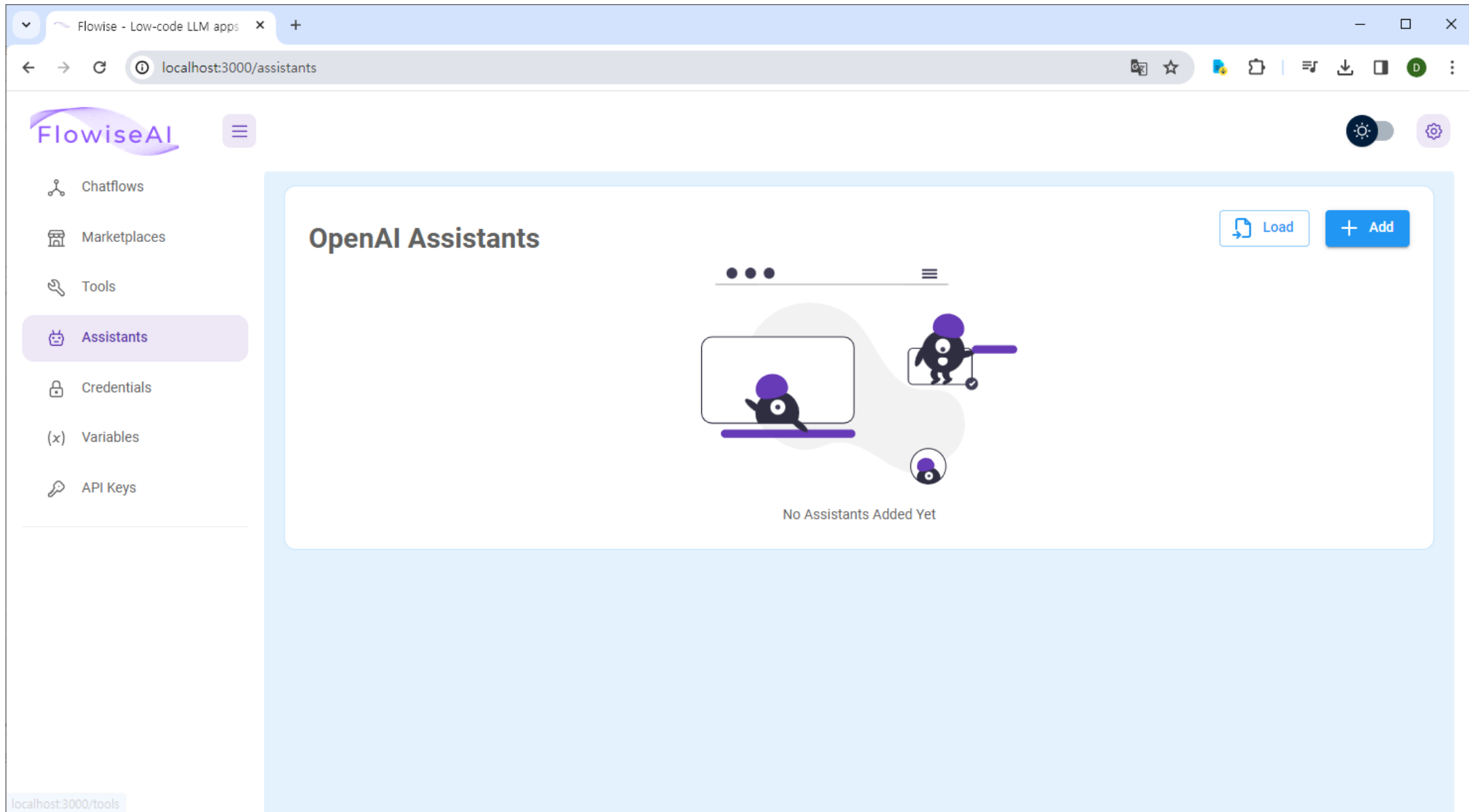
- URL: localhost:3000
- Search bar: Search name or category
- Buttons: Add New
- Table of Chatflows:

Name	Category	Nodes	Last Modified Date	Actions
Demo		M/L, GPT, QA, LLM, + 1 More	March 10th, 2024	Options

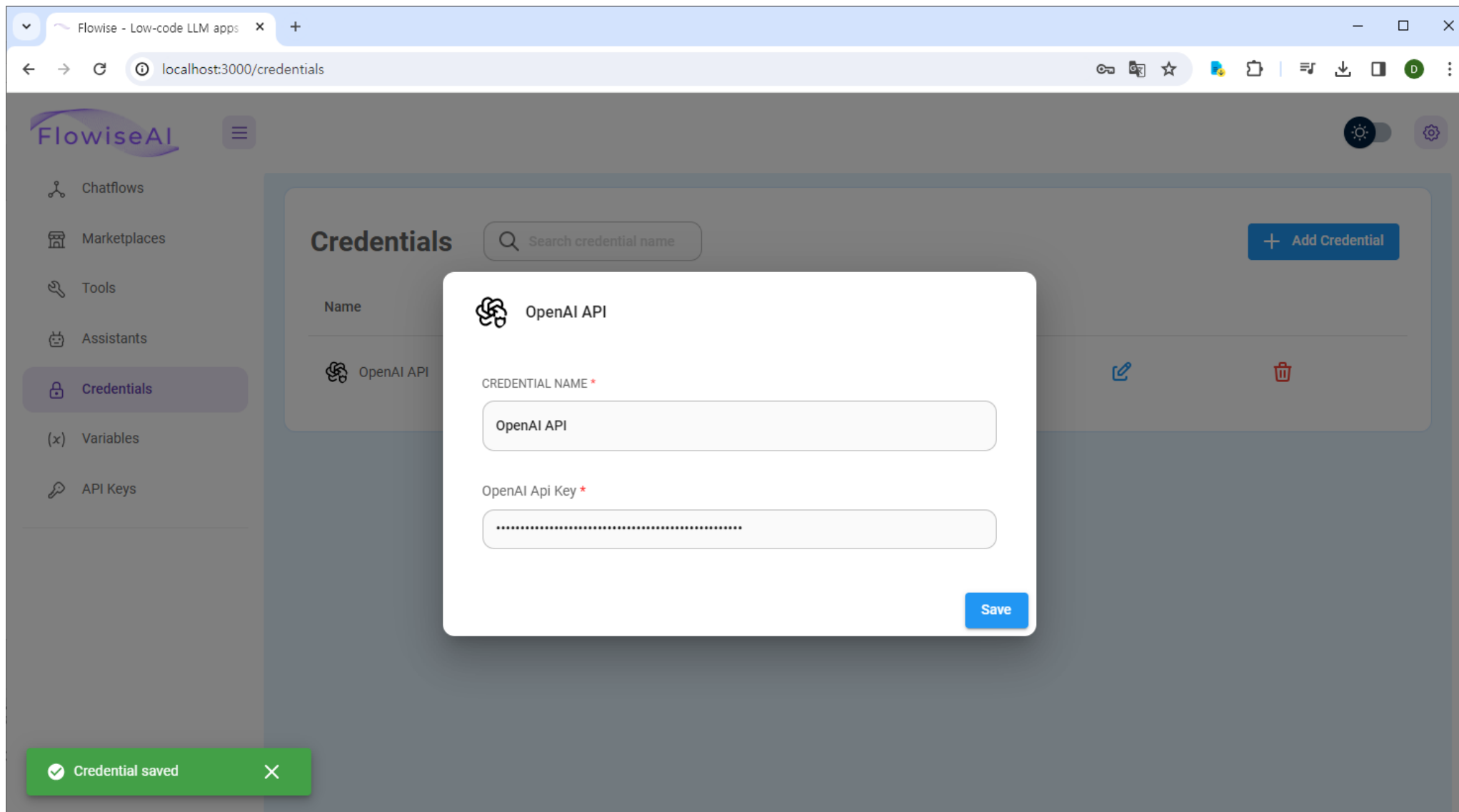
# Tools



# Assistants



# Credentials



# 실습 - Food Recipe App

<https://platform.openai.com/api-keys>

Flowise - Low-code LLM apps

localhost:3000/canvas/5a24d695-a2e1-4a4b-aba7-f149aa1bc0d9

## FoodRecipe

**3** Prompt Template  
Schema to represent a basic prompt for an LLM

**2** OpenAI  
Wrapper around OpenAI large language models

**1** LLM Chain  
Chain to series against

**OpenAI API Configuration:**

- CREDENTIAL NAME: OepnAI
- OpenAI Api Key: [Redacted]

**Prompt Template:**

Inputs: Langchain Hub

Template: 당신은 숙련된 요리사입니다. 주 재료 {ingredient} 를 사용하여 맛있는 레시피를 만들어 주세요.

Format Prompt Values: ingredient: {{question}}

Select Variable: question (User's question from chatbox), chat\_history (Past conversation history between user and AI)

**OpenAI Configuration:**

- Cache: [X]
- Connect Credential: OpenAI
- Model Name: gpt-3.5-turbo-instruct
- Temperature: 0.7
- Additional Parameters: [Link]

**LLM Chain Configuration:**

- Language Model: [Link]
- Prompt: [Link]
- Output Parser: [Link]
- Input Moderation: [Link]
- Chain Name: chef
- LLM Chain: [Link]

**Chat Interface:**

감자

감자는 다양한 방식으로 요리할 수 있기 때문에 여러 가지 레시피를 소개해 드리겠습니다.

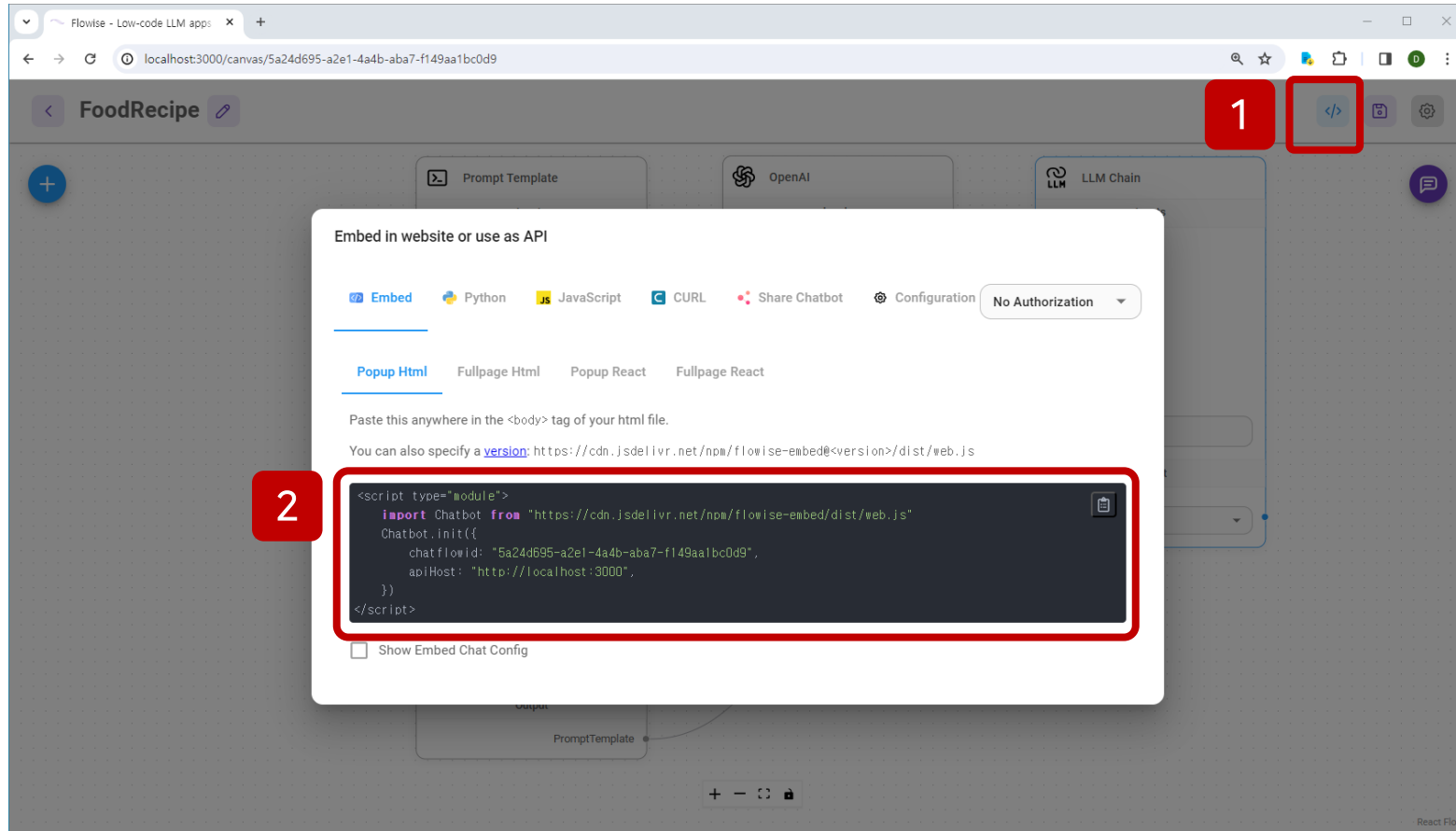
1. 감자조림

- 감자를 깨끗이 씻어 껍질을 벗긴 후 2cm 두께로 썬다.
- 양파 1/2개와 당근 1개는 비슷한 크기로 썰어 준다.
- 냄비에 참기름을 두르고 양파와 당근을 넣어 볶은 후 감자를 넣고 함께 볶는다.
- 물 1컵, 간장 2큰술, 설탕 1큰술을 넣어 감자가 잠길 정도로 물을 부어 뚜껑을 닫고 약한 불에서 10분 정도 끓인다

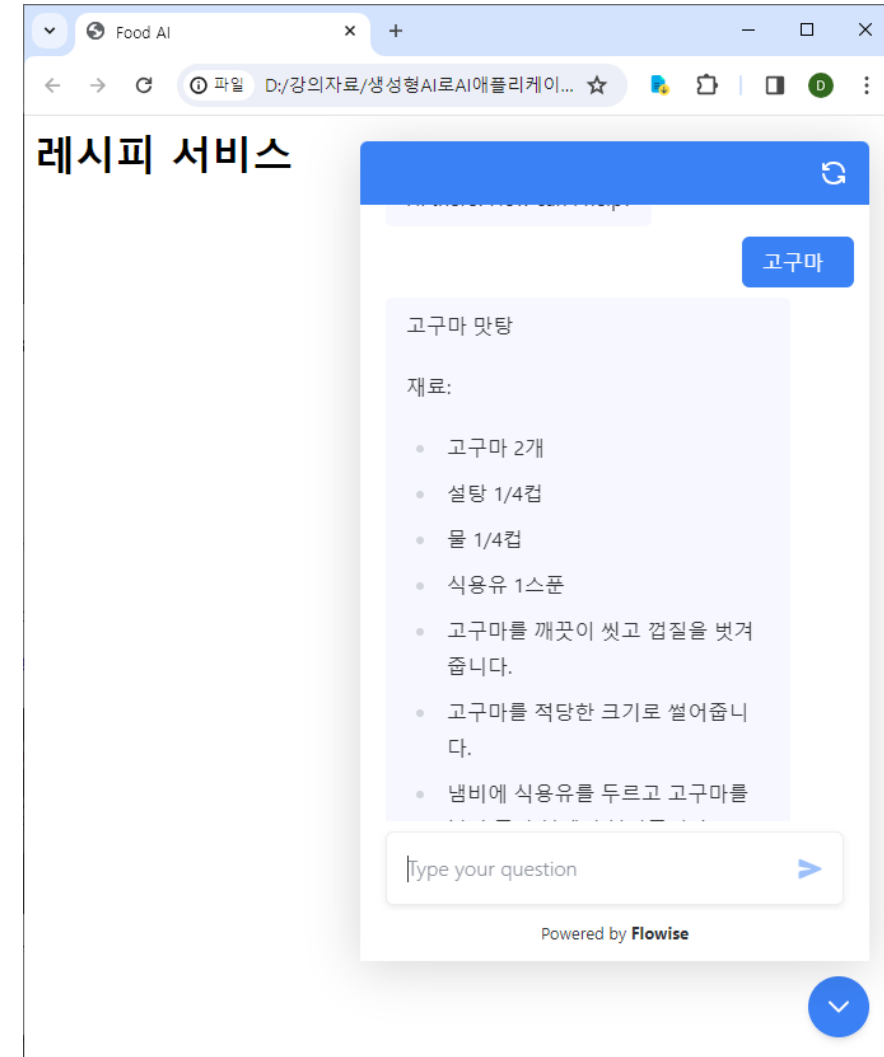
Type your question...



# 실습 - Food Recipe App



index.html



# 실습 - Food Recipe Critic App

The screenshot displays the Flowise Low-code LLM app interface for a project named "FoodRecipeCriticApp". The workflow is composed of several interconnected components:

- Prompt Template (Left):** Contains a template for generating a recipe based on ingredients: "당신은 숙련된 요리사입니다. 주 재료 (ingredient) 를 사용하여 맛있는 레시피를 만들어 주세요."
- OpenAI:** Configured with "Cache", "Connect Credential" (OpenAI), "Model Name" (gpt-3.5-turbo-instruct), and "Temperature".
- LLM Chain (Left):** Configured with "Language Model", "Prompt", "Output Parser", "Input Moderation", and "Chain Name" (chef). The output is labeled "Output Prediction" and marked with a red "1".
- Prompt Template (Middle):** Contains a template for generating a review: "You are a harsh and rude food critic. Write a review about the following recipe in korean: {recipe}" (highlighted in yellow). The output is labeled "Output Prediction" and marked with a red "2".
- ChatOpenAI:** Configured with "Cache", "Connect Credential" (OpenAI), "Model Name" (gpt-3.5-turbo-1106), "Temperature" (0.9), and "Allow Image Uploads".
- LLM Chain (Right):** Configured with "Language Model" and "Prompt". The output is labeled "Output Prediction" and marked with a red "3".

Two "Format Prompt Values" popups are visible:

- Format Prompt Values (Left):** Shows a JSON structure for "recipe": 

```
{ 1 item recipe: {{llmChain_0.data.instance}} }
```

.
- Format Prompt Values (Middle):** Shows a JSON structure for "question": 

```
{ 1 item question: User's question from chatbox }
```

.

A "Select Variable" popup is also visible, showing variables: "question" (User's question from chatbox), "chat\_history" (Past conversation history between user and AI), and "chef" (Output Prediction from LLM Chain).

On the right, a chat interface shows a conversation with a user (고구마) and an AI assistant (고구마). The AI assistant provides a recipe and a review. The chat interface includes a "Type your question..." input field and a "Send" button.


The diagram illustrates a complex LangChain workflow for sentiment analysis and response generation. The workflow is composed of several interconnected nodes:

- Prompt Template (Initial):** Contains a JSON structure for a list of items, each with a `sentence` field. The prompt is: `{ { item sentence : "{{question}}" } }`.
- OpenAI:** Uses the `gpt-3.5-turbo-instruct` model to analyze the sentiment of the input sentences. The output is a `setiment` (sic).
- Structured Output Parser (JSON):** Parses the output from the OpenAI node into a structured format.
- LLM Chain:** Uses the `gpt-3.5-turbo` model to generate a response based on the sentiment analysis. It includes an **IfElse Function** that checks if the sentiment is positive.
  - If the sentiment is positive (`if ($sentiment.positive == true){ return true; }`), it triggers a **ChatOpenAI** node to generate a positive response.
  - If the sentiment is negative, it triggers another **ChatOpenAI** node to generate a negative response.
- ChatOpenAI (Positive):** Generates a positive response based on the prompt: `당신은 고객 지원 도우미입니다. 먼저 고객에게 리뷰에 대해 감사 인사를 전한 다음 웹사이트에 리뷰를 게시할 것인지 고객에게 문의합니다. 또한, 리뷰를 요약합니다: {review}`.
- ChatOpenAI (Negative):** Generates a negative response based on the prompt: `당신은 부정적인 리뷰를 처리하는 고객 지원 도우미입니다. 다음 리뷰에 대한 사과문을 작성하고 사용자에게 고객 서비스 담당자에게 연락을 받고 싶은지를 문의합니다. 리뷰: {review}`.
- JSON Structure:** A table defining the output structure:
 


Pro...	Type	Description
sentiment	boolean	the sentence is positive

The workflow is designed to handle a list of customer reviews, analyze their sentiment, and generate appropriate responses based on whether the sentiment is positive or negative.


# 실습 - Review AI App




제품의 가성비가 좋은 거 같아요.



고객님, 안녕하세요! 리뷰를 남겨 주셔서 감사드립니다. 고객님의 소중한 의견을 전달해주셔서 감사합니다. 제품의 가성비가 좋다고 말씀해주셔서 저희도 기쁘게 생각합니다. 웹사이트에 리뷰를 게시해도 괜찮을까요? 게시를 원하시면 언제든지 말씀해주세요. 감사합니다!



배송 제품에 스크래치가 있어요.



고객님,  
저희 제품을 구매해 주셔서 감사합니다. 그러나 제품이 스크래치가 있어서 심려를 끼쳐드려 죄송합니다. 제품이 완벽하지 않아서 정말 죄송합니다.  
  
고객 서비스 담당자가 귀하의 문제를 즉시 해결하도록 도와드릴 수 있습니다. 귀하의 문제를 빠르게 해결해 드릴 수 있도록 도와드릴 수 있도록, 고객 서비스 담당자에게 연락을 받고 싶으신

```
C:\Windows\system32\cmd.exe
2024-03-10 19:58:08 [INFO]: POST /api/v1/internal-prediction/7f981a57-89d2-4e33-99bf-e382023913b5

****OUTPUT PREDICTION****

{ json: { positive: true } }
2024-03-10 19:58:09 [INFO]: [server]: Chatflow 7f981a57-89d2-4e33-99bf-e382023913b5 added into ChatflowPool

****FINAL RESULT****

고객님, 안녕하세요! 리뷰를 남겨 주셔서 감사드립니다. 고객님의 소중한 의견을 전달해주셔서 감사합니다. 제품의 가성비가 좋다고 말씀해주셔서 저희도 기쁘게 생각합니다. 웹사이트에 리뷰를 게시해도 괜찮을까요? 게시를 원하시면 언제든지 말씀해주세요. 감사합니다!
2024-03-10 19:58:38 [INFO]: POST /api/v1/internal-prediction/7f981a57-89d2-4e33-99bf-e382023913b5

****OUTPUT PREDICTION****

{ json: { positive: false } }
2024-03-10 19:58:38 [INFO]: [server]: Chatflow 7f981a57-89d2-4e33-99bf-e382023913b5 added into ChatflowPool

****FINAL RESULT****

고객님,
저희 제품을 구매해 주셔서 감사합니다. 그러나 제품이 스크래치가 있어서 심려를 끼쳐드려 죄송합니다. 제품이 완벽하지 않아서 정말 죄송합니다.
고객 서비스 담당자가 귀하의 문제를 즉시 해결하도록 도와드릴 수 있습니다. 귀하의 문제를 빠르게 해결해 드릴 수 있도록 도와드릴 수 있도록, 고객 서비스 담당자에게 연락을 받고 싶으신가요?
다시 한번 사과의 말씀을 전하며, 고객님의 양해를 부탁드립니다.
```

# 실습 - Conversational Chatbot

<https://upstash.com/>

The screenshot shows the Upstash dashboard with the following elements and numbered annotations:

- 1**: Points to the 'Redis' tab in the top navigation bar.
- 2**: Points to the 'Create database' button in the 'Create a Redis Database' section.
- 3**: Points to the 'Name' field, which contains 'ChatHist'.
- 4**: Points to the 'Region' dropdown, which is set to 'N. Virginia, USA (us-east-1)'.
- 5**: Points to the 'Create' button at the bottom of the 'Create Database' modal.
- 6**: Points to the 'REST API' section in the modal, which explains that the REST API enables access to the database.
- 7**: Points to the 'REST API' toggle switch, which is currently turned on.

Below the 'REST API' section, there are fields for 'UPSTASH\_REDIS\_REST\_URL' and 'UPSTASH\_REDIS\_REST\_TOKEN', and a code block showing a cURL command:

```
curl "https://us1-equal-tuna-40994.upstash.io/set/foo/bar" \
-H "Authorization: Bearer *****"
```

# 실습 - Conversational Chatbot

Flowise - Low-code LLM apps

localhost:3000/canvas/73a737cf-72d0-4f6d-9ec7-db2ac1ff679

## \* Conversational Chatbot

**Simple Prompt Moderation**

Inputs

- Chat Model
- Deny List

나쁜 강아지

Error Message

Cannot Process! Input violates content moderation policies.

**Upstash Redis-Backed Chat Memory**

Inputs

- Connect Credential \*  
Upstash Redis
- Upstash Redis REST URL \*  
https://us1-equal-tuna-40994.upstash
- Additional Parameters

Output

UpstashRedisBackedChatMemory

**ChatOpenAI**

Inputs

- Cache
- Connect Credential \*  
OpenAI

Session Id \*  
chat001

Session Timeouts \*

Additional Parameters

Output

ChatOpenAI

**Upstash Redis Memory API**

Refer to [official guide](#) on how to create redis instance and get redis REST Token

CREDENTIAL NAME \*  
Upstash Redis

Upstash Redis REST Token \*

Add

**Conversation Chain**

Inputs

- Chat Model \*
- Memory \*
- Chat Prompt Template
- Input Moderation

Additional Parameters

Output

ConversationChain

**Buffer Memory**

Inputs

- Memory Key \*  
chat\_history
- Input Key \*  
input

Allow Image Uploads

Additional Parameters

Output

ChatOpenAI

내 이름은 대니

반가워요, 대니님! 좋은 하루 보내세요. 어떤 정보가 필요하신가요?

니 이름은?

저는 인공지능 어시스턴트입니다. 제 이름은 따로 없지만, 대니님과 함께 대화를 나누고 도와드리기 위해 여기 있어요. 어떤 것이든 물어보세요!

나쁜 강아지

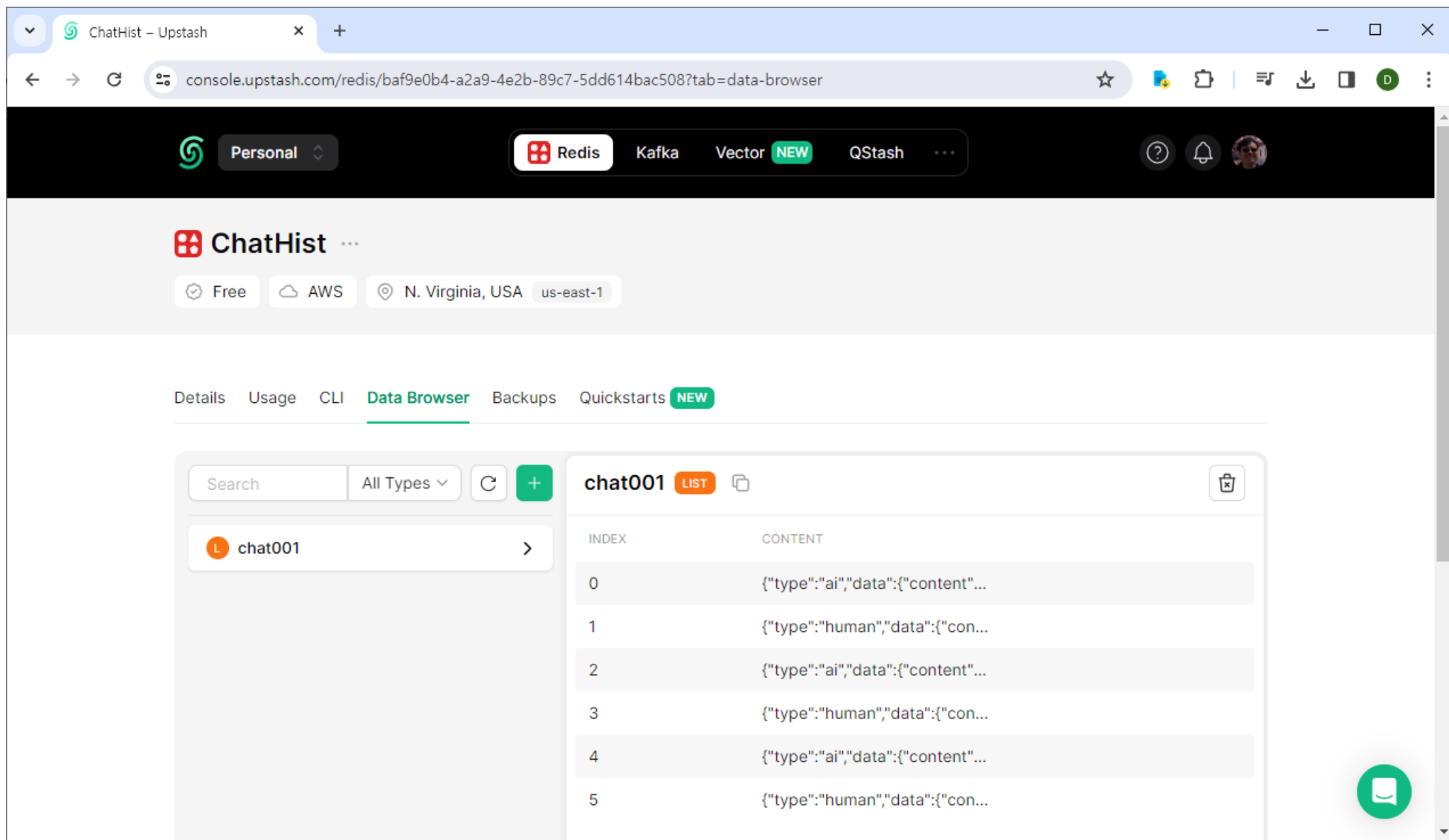
Cannot Process! Input violates content

내 이름은 oo

React Flow



# 실습 - Conversational Chatbot



The screenshot shows the ChatHist Upstash console interface. The browser address bar displays the URL: `console.upstash.com/redis/baf9e0b4-a2a9-4e2b-89c7-5dd614bac508?tab=data-browser`. The top navigation bar includes the ChatHist logo, a 'Personal' dropdown, and tabs for 'Redis', 'Kafka', 'Vector' (marked as 'NEW'), and 'QStash'. Below the navigation bar, the 'ChatHist' header is followed by filters: 'Free', 'AWS', 'N. Virginia, USA', and 'us-east-1'. The main content area features a tabbed interface with 'Details', 'Usage', 'CLI', 'Data Browser' (selected), 'Backups', and 'Quickstarts' (marked as 'NEW'). The 'Data Browser' tab shows a search bar, a filter dropdown set to 'All Types', and a list of chat sessions. The selected session is 'chat001', which is displayed in a table with two columns: 'INDEX' and 'CONTENT'. The table contains six rows of chat data, alternating between AI and human messages. A green chat icon is visible in the bottom right corner.

ChatHist ...

Free AWS N. Virginia, USA us-east-1

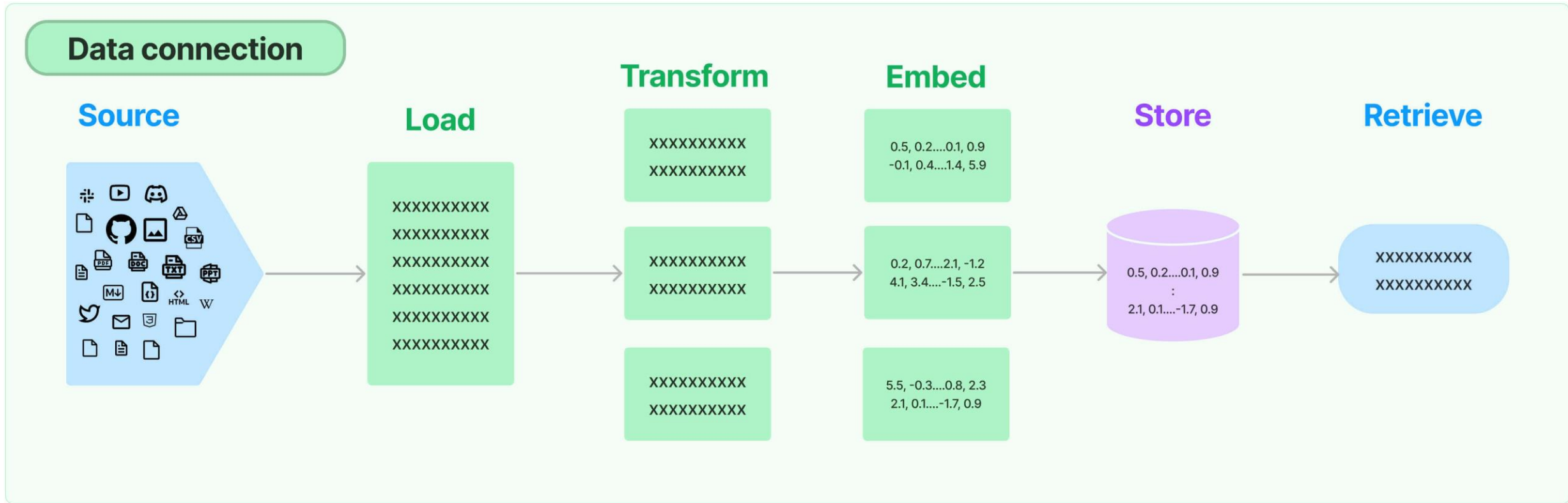
Details Usage CLI **Data Browser** Backups Quickstarts **NEW**

Search All Types **+**

chat001

INDEX	CONTENT
0	{"type":"ai","data":{"content"...
1	{"type":"human","data":{"con...
2	{"type":"ai","data":{"content"...
3	{"type":"human","data":{"con...
4	{"type":"ai","data":{"content"...
5	{"type":"human","data":{"con...

# 실습 - RAG Chatbot



# 실습 - RAG Chatbot

Flowise - Low-code LLM apps

localhost:3000/canvas/f7724231-4aae-4e2a-94b4-db93fef0f5e8

## RAGChatbot

**Recursive Character Text Splitter**

Inputs

Chunk Size: 1000

Chunk Overlap: 50

Additional Parameters

Output: RecursiveCharacterTextSplitter

**Pdf File**

Inputs

Text Splitter

손익계산서.pdf

Upload File

Usage: One document per page

Additional Parameters

Output: Document

**In-Memory Vector Store**

Inputs

Document

Embeddings \*

Top K: 4

Output: Memory Retriever

**OpenAI Embeddings**

Inputs

Connect Credential: OpenAI

Model Name: text-embedding-ada-002

Additional Parameters

**ChatOpenAI**

Inputs

Cache

Connect Credential: OpenAI

Model Name: gpt-3.5-turbo

Temperature: 0.4

Allow Image Uploads

Additional Parameters

Output: ChatOpenAI

**Conversational Retrieval QA Chain**

Inputs

Chat Model \*

Vector Store Retriever \*

Memory

Return Source Documents

Additional Parameters

Output: ConversationalRetrievalQAChain

Upset Vector Store

- Recursive Character Text Splitter (recursiveCharacterTextSplitter\_0)
- Pdf File (pdfFile\_0)
- OpenAI Embeddings (openAIEmbeddings\_0)
- In-Memory Vector Store (memoryVectorStore\_0)

Show API

Upset

Hi there! How can I help?

2020년 영업수익이 얼마야?

2020년 영업수익은 실액원으로 23,917입니다.

2023년 당기순이익은?

2023년 당기순이익은 실액원으로 989입니다.

2020년 영업수익이 얼마야?

2023년 당기순이익은?

# 실습 - RAG Chatbot

Flowise - Low-code LLM apps

localhost:3000/canvas/6384ca1b-c034-481e-a588-3e3fbd640e0

## RAGWebChatbot

**Recursive Character Text Splitter**

Inputs

Chunk Size: 1000

Chunk Overlap: 50

Additional Parameters

Output: RecursiveCharacterTextSplitter

**Cheerio Web Scraper**

Inputs

Text Splitter

URL: <https://docs.flowiseai.com/configuration/deployment/render>

Manage Links

Additional Parameters

Output: Document

**In-Memory Vector Store**

Inputs

Document

Embeddings

Output: 4

Memory Retriever

**ChatOpenAI**

Inputs

Cache

Connect Credential: OpenAI

Model Name: gpt-3.5-turbo

Temperature: 0.4

Allow Image Uploads

Additional Parameters

Output: ChatOpenAI

**Conversational Retrieval QA Chain**

Inputs

Chat Model

Vector Store Retriever

Memory

Return Source Documents

Additional Parameters

Output: ConversationalRetrievalQA

**OpenAI Embeddings**

Inputs

Connect Credential: OpenAI

Model Name: text-embedding-ada-002

Additional Parameters

**Scraped Links**

<https://docs.flowiseai.com/sitemap.xml>

Fetch Links

Scraped Links

- <https://docs.flowiseai.com/>
- <https://docs.flowiseai.com/getting-started>
- <https://docs.flowiseai.com/contributing>
- <https://docs.flowiseai.com/using-flowise>
- <https://docs.flowiseai.com/using-flowise/api>
- <https://docs.flowiseai.com/using-flowise/streaming>
- <https://docs.flowiseai.com/using-flowise/embed>

Cancel Save

**Web Crawl**

Scrape XML Sitemap

Scrape relative links from XML sitemap URL

Selector (CSS)

Metadata

By following these steps, you should be able to

6. Connect your GitHub account.
7. Choose your forked Flowise repo and click "Connect".
8. Fill in your preferred Name and Region.
9. Select Docker as your Runtime.
10. Optionally, add app level authorization by clicking "Advanced" and adding Environment Variables like FLOWISE\_USERNAME, FLOWISE\_PASSWORD, and NODE\_VERSION with the value 18.18.1.
11. Click on "Create Web Service".
12. Navigate to the deployed URL to access your deployed Flowise application.

How can I deploy Flowise to Render?

How can I deploy Flowise to AWS?

How can I set up Flowise as a developer?

React Flow

# 실습 - RAG Chatbot

<https://www.pinecone.io/>

**Pinecone Console**

[Back to Indexes](#)

## Create a new Index

Name:

**Configure your Index**

The dimensions and metric depend on the model you select.

Dimensions:  Metric: cosine

**Capacity Mode**

SERVERLESS PODS STARTER

**flowise** Starter

METRIC	DIMENSIONS	POD TYPE	HOST
cosine	1536	starter	<a href="https://flowise-d2c940b.svc.gcp-starter.pinecone.io">https://flowise-d2c940b.svc.gcp-starter.pinecone.io</a>

CLOUD	REGION	ENVIRONMENT
gcp-starter	Iowa (us-central1)	gcp-starter

**Starter**

You are currently on the **Free Plan**

Cancel Create Index

# 실습 - RAG Chatbot

Flowise - Low-code LLM apps

localhost:3000/canvas/6384ca1b-c034-481e-a588-3e3fdbd640e0

## RAGWebChatbot

**Recursive Character Text Splitter**

Inputs

Chunk Size: 1000

Chunk Overlap: 50

Additional Parameters

Output: RecursiveCharacterTextSplitter

**Cheerio Web Scraper**

Inputs

Text Splitter

URL: [https://python.langchain.com/docs/expression\\_language/](https://python.langchain.com/docs/expression_language/)

Manage Links

Additional Parameters

Output: Document

**Pinecone**

Inputs

Document

Embeddings: Pinecone API

Pinecone Index: flowise

Additional Parameters

Output: Pinecone Retriever

**ChatOpenAI**

Inputs

Cache

Connect Credential: OpenAI

Model Name: gpt-3.5-turbo

Pinecone API

CREDENTIAL NAME: Pinecone API

Pinecone Api Key

Add

**Conversational Retrieval QA Chain**

Inputs

Chat Model

Vector Store Retriever

Memory

Return Source Documents: ☒

What is a LCEL?

LangChain Expression Language, or LCEL, is a declarative way to easily compose chains together. It was designed to support putting prototypes into production without code changes, from simple chains to complex ones with hundreds of steps. Some reasons to use LCEL include streaming support, optimized parallel execution, async support, retries and fallbacks configuration, access to intermediate results, and seamless integration with LangServe and LangSmith.

/docs/expresio...

Type your question: What is a LCEL?



# 실습 - Chatflow 분석

The screenshot displays the Flowise web application interface. The main workspace shows a chatflow titled "RAGWebChatbot" with a visual flow diagram. The flow starts with a "Recursive Character Text Splitter" node, which feeds into a "Cheerio Web Scraper" node. The "Recursive Character Text Splitter" node has inputs for "Chunk Size" (set to 1000) and "Chunk Overlap" (set to 50), and an output labeled "RecursiveCharacterTextSplitter". The "Cheerio Web Scraper" node has an input for "URL" (set to "https://python.langchain.com/docs/ex") and an output labeled "Document". A modal window titled "Analyse Chatflow" is open in the foreground. This modal contains a list of credentials: "LangSmith" (https://smith.langchain.com) with a green "ON" toggle, "LangFuse" (https://langfuse.com), and "Lunary" (https://lunary.ai). The "LangSmith" credential is selected, and its "Connect Credential" dropdown is set to "Langsmith". The "Project Name" is set to "RAG Web Chatbot". There is an "On/Off" toggle for the selected credential. A blue arrow points from the "Analyse Chatflow" button in the right-hand menu to the modal. The right-hand menu also includes options like "Starter Prompts", "View Messages", "Speech to Text", "Duplicate Chatflow", "Load Chatflow", "Export Chatflow", and "Delete Chatflow".

Flowise - Low-code LLM apps

localhost:3000/canvas/6384ca1b-c034-481e-a588-3e3fbd640e0

RAGWebChatbot

Recursive Character Text Splitter

Inputs

Chunk Size

1000

Chunk Overlap

50

Additional Parameters

Output

RecursiveCharacterTextSplitter

Cheerio Web Scraper

Inputs

Text Splitter

URL \*

https://python.langchain.com/docs/ex

Manage Links

Additional Parameters

Output

Document

Analyse Chatflow

LangSmith  
https://smith.langchain.com

ON

Connect Credential \*

Langsmith

Project Name

RAG Web Chatbot

On/Off

LangFuse  
https://langfuse.com

Lunary  
https://lunary.ai

Save

Conversational Retrieval QA Chain

Inputs

Model \*

Store Retriever \*

Source Documents

Additional Parameters

Output

ConversationalRetrievalQAChain

Starter Prompts

View Messages

Speech to Text

Duplicate Chatflow

Load Chatflow

Export Chatflow

Analyse Chatflow

Delete Chatflow

# 실습 - Chatflow 분석

smith.langchain.com/o/99048fc5-5e46-5edd-8a00-98665c3d24e7/projects/p/534faf7c-a1df-4f31-9d47-5793da9a8aa2?columnVisibilityModel=%7B"outputs"%3Afalse%2C"feedback\_stats"%3Afalse%2C"reference\_...

Personal > Projects > RAG Web Chatbot

## RAG Web Chatbot

Traces LLM Calls All Runs Monitor Setup

Filters Last 7 days

		✓	Name	Input	Start Time	Latency	Dataset	Annotation Queue	Tokens	Cost
<input type="checkbox"/>	>	✓	RunnableSequence	Explain about Async s...	2024. 3. 11. 오후 10:4...	3.03s	<input type="checkbox"/>		1,837	\$0.0009985
<input type="checkbox"/>	>	✓	RunnableSequence	Show an example of u...	2024. 3. 11. 오후 10:4...	2.73s	<input type="checkbox"/>		1,735	\$0.0008845
<input type="checkbox"/>	>	✓	RunnableSequence	LCEL 주요 기능을 요약...	2024. 3. 11. 오후 10:4...	4.01s	<input type="checkbox"/>		1,693	\$0.0008635
<input type="checkbox"/>	>	✓	RunnableSequence	What is a LCEL?	2024. 3. 11. 오후 10:4...	4.53s	<input type="checkbox"/>		1,570	\$0.000883

Trace Collapse Stats Most relevant

RunnableSequence SUCCESS 4.53s 1,570

map:key:context 2.03s

FindDocs 2.02s

RetrievalChain... 2.02s

CondenseQ... 0.93s

ChatOpenAI 0.93s

VectorStoreRetr... 1.09s

ChatOpenAI 2.49s

Some runs have been hidden. [Show 14 hidden runs](#)

### Runnable...

Run Feedback Metadata

Input

```
1 question: What is a LCEL?
2 chat_history:
3 -
4 message: What is a LCEL?
5 type: userMessage
6 message: LangChain Expression
```

YAML

Output





수고하셨습니다. 감사합니다.