



Our **Code**
is **Open**



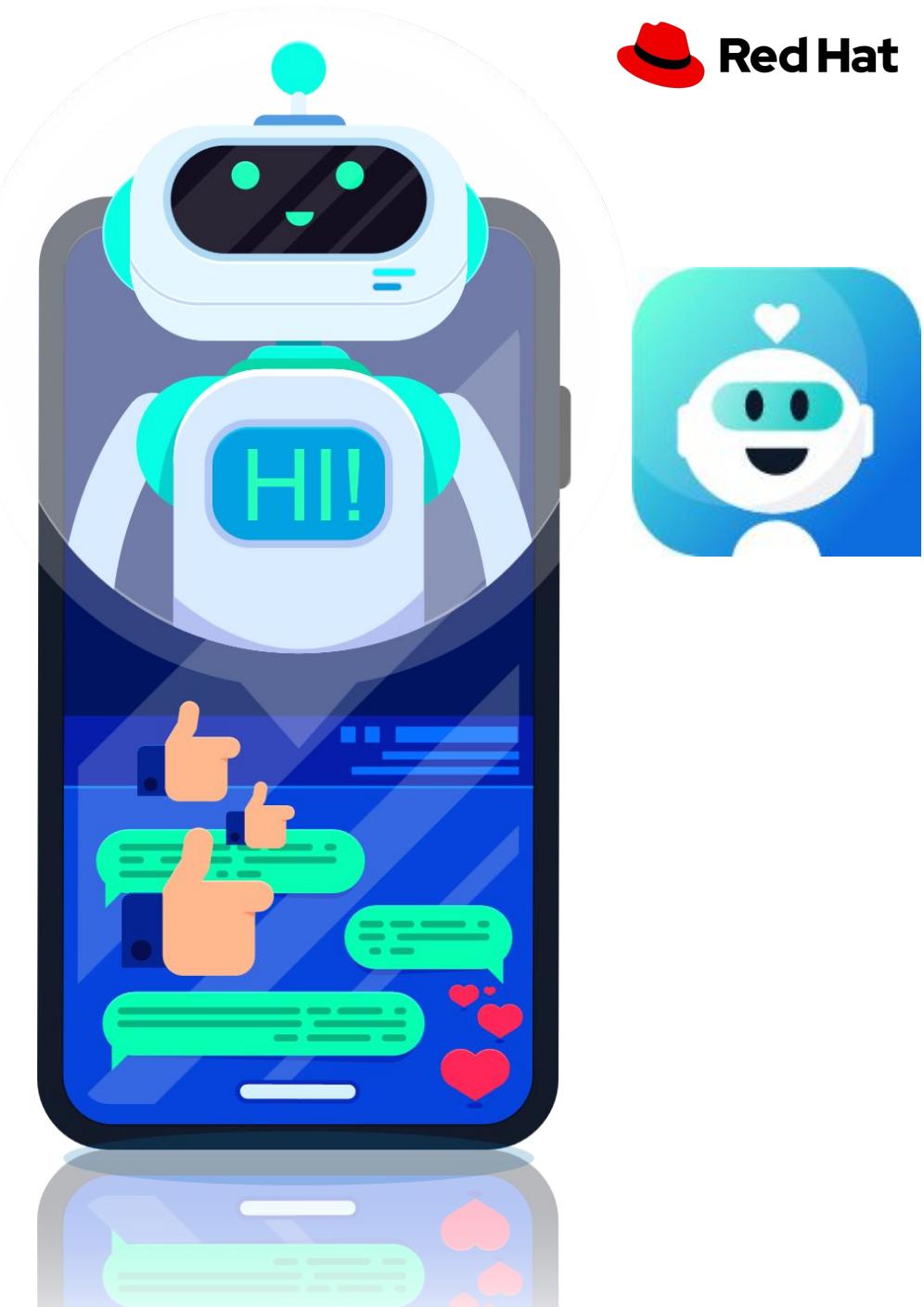
REDHAT CODESHIFT

HACKATHON

Developing cloud-native apps with Red Hat's
OpenShift application platform

FINAPP

- › Tim Lam
- › Red Hat Hong Kong
- › tinlam@redhat.com
- › Deadline: Nov 13, 2023 @ 5:00pm EST



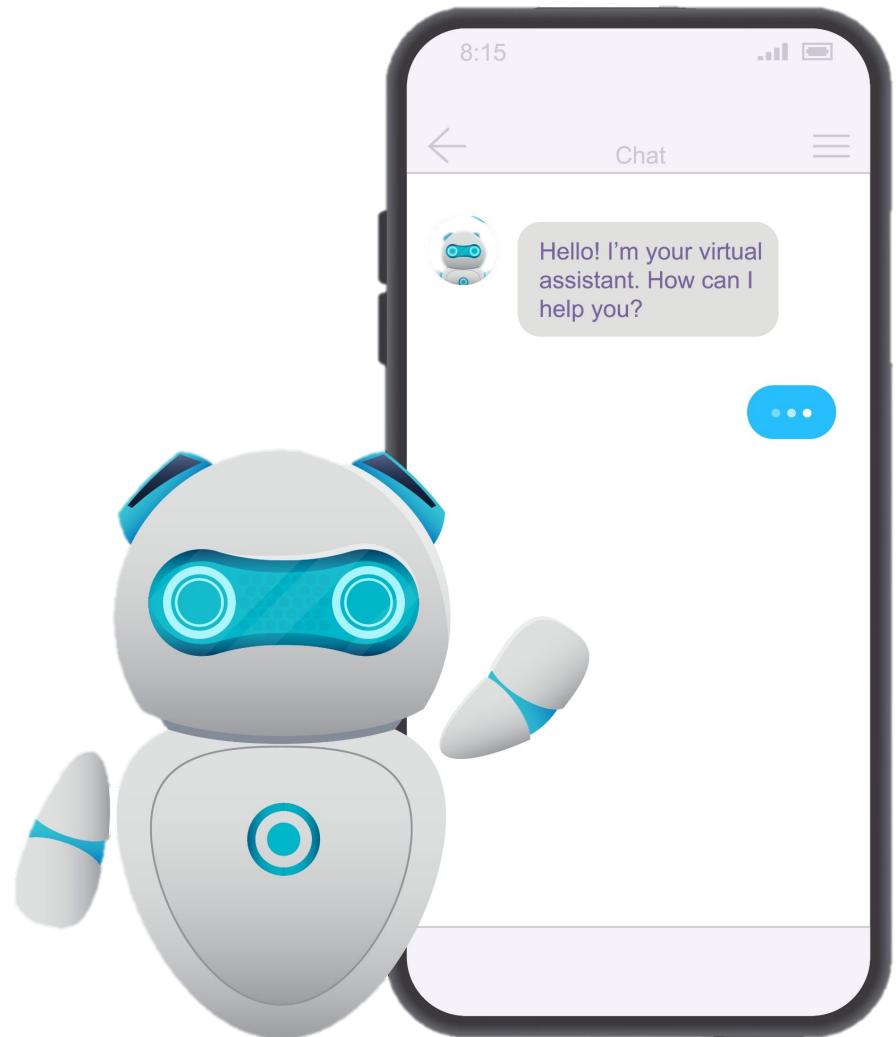
REDHAT CODESHIFT HACKATHON

Developing cloud-native apps with Red Hat's OpenShift application platform



What to submit

- › Open-source project built with (or updated) Red Hat and Intel tools that is deployed on the OpenShift application platform (or Kubernetes community equivalent)
- › URL to a publicly visible code repository (Source code submission must be released under an OSI-approved license) - don't forget to add a README with
 - Architectural diagram
 - Any prerequisites
 - Demo guide
 - Deployment instructions for running the project on red hat openshift or kubernetes.
- › Text description that provides a project overview and guide on how to deploy and run your project
- › Description of the project architecture
- › Demonstration video of the project (approximately <5 mins)



FINAPP CHATBOT



My Project Name

- FinApp



GitHub Repository

https://github.com/timlam007/finance_chatbot



Demo Video

<https://drive.google.com/file/d/1S-F-ErfxQgBrhvzONaBgCEkG9bhgqL1/view?usp=sharing>



Docker Hub

<https://hub.docker.com/repository/docker/timlam007/finchat>



Elevator Pitch

- FinApp Chatbot Document and Data Retrieval System, a transformative digital tool that streamlines access to private company data powered by OpenAI GPT LLM.



Value of the Project

- The project streamlines the process of navigating through extensive company documents by using a context-aware chatbot. Powered by OpenAI's GPT, the chatbot can answer, summarize, and rephrase queries accurately, ensuring context-driven information retrieval.

FINAPP CHATBOT FRONT END UI

Options

OpenAI API Key

***** 

[Get an OpenAI API Key](#)

Choose LLM:

- gpt-3.5-turbo-0613
- gpt-3.5-turbo-16K-0613
- gpt-4
- text-davinci-003

Temperature: 0.00 1.00

What would you like to query?

Database

Clear Conversation

Costs

Total Cost: \$0.03732

- \$0.03732

Input your Open API Key

Select your LLM Model

Drag and Drop the Files here

Temperature (Control the Randomness)

Showing the Cost

Input your question

 Chat Document

Finance Chatbot

Upload Document or Connect to a Database

Drag and drop files here
Limit 200MB per file . CSV, XLSV, PDF, DOCX, TXT

Personal FinanceGPT

The table in the Customer Database are customers, Product01, smartphones.

Show me the tables in Customer Database

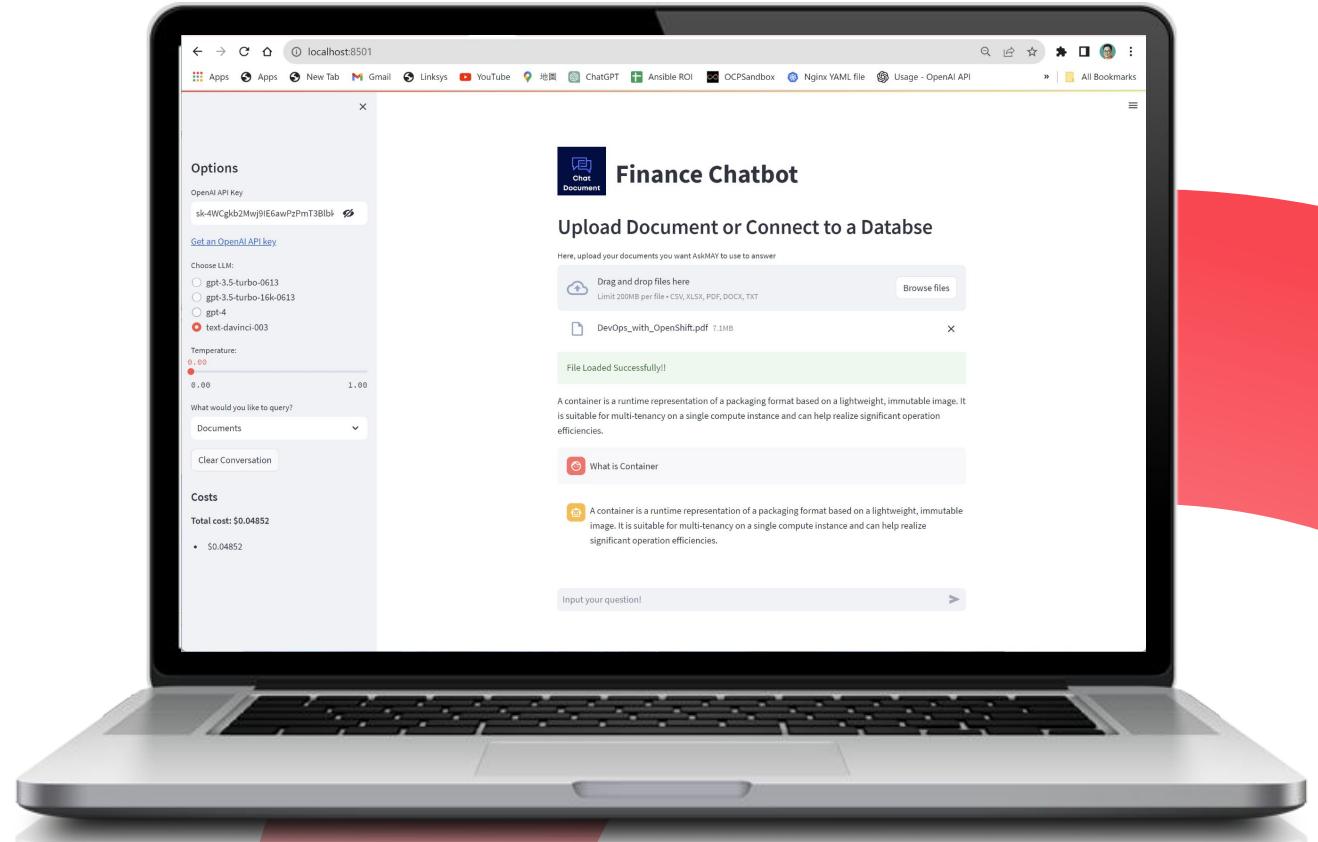
The table in the Customer Database are Customers, Product01, Smartphones

Input your question

FINAPP CHATBOT FUNCTIONS

- › Provides a chat interface for user interaction.
- › Accepts various document formats, including PDF, TXT, and DOCX.
- › Handles structured data in Excel and CSV files.
- › Connects seamlessly to databases.
- › Extracts relevant information from documents and data sources to respond to user queries.
- › Ensures responses are contextually accurate by not answering questions when the answer isn't in the provided context.

For commercial purposes, utilizing open-source models for document processing is a cost-effective option. However, it's important to note that while these models come without associated costs, their performance may not match that of premium alternatives.



MY PROJECT: FINAPP CHATBOT

Developing cloud-native apps with Red Hat's OpenShift application platform



Well Architected

Can your application scale well at a linear cost? Can it accommodate additional features without rewriting most of the codebase?

Answer

FinAPP's architecture is robust and scalable. The use of vector databases like Chroma, Pinecone, and Faiss ensures efficient querying, even as the knowledge base grows. The modular design, facilitated by tools like LangChain, allows for easy integration of additional features and datasets without the need for extensive codebase modifications.



What's next for FinAPP

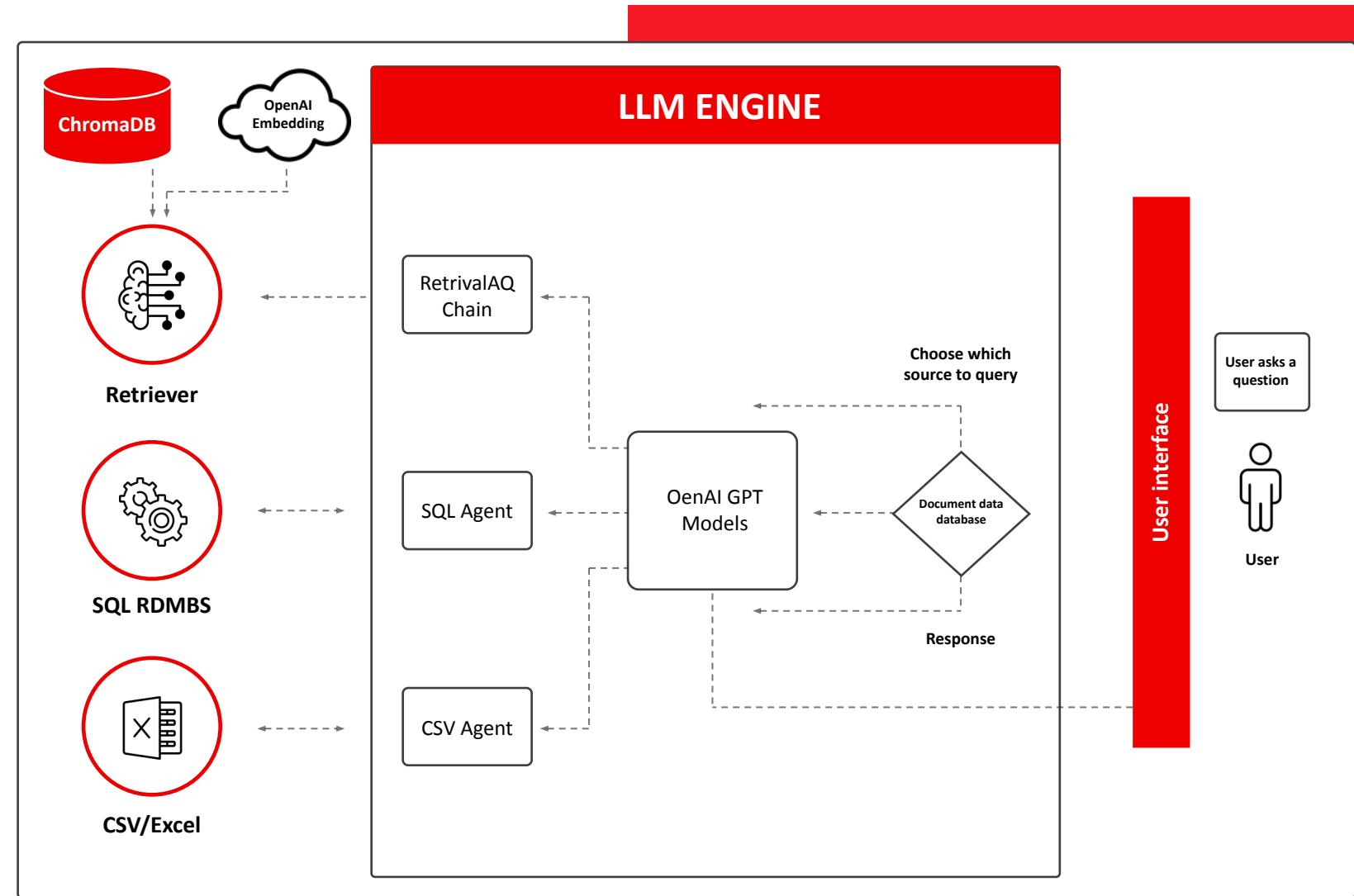
As technology advances, we aim to integrate more databases, support additional document formats, and refine the contextual understanding further. We're also looking into real-time collaborative features and expanding the system's capabilities to predict user needs.



HIGH LEVEL ARCHITECTURE

Technology Behind the Project

- › GPT (OpenAI Large Language Model): This state-of-the-art language model is central to the system's ability to provide accurate and contextually relevant responses.
- › LangChain Python Library: LangChain simplifies the chaining of large language models, making it easier to handle complex interactions.
- › Vector Stores (Chroma): These serve as a database of vectorized words, enhancing the system's capability to understand and process text.
- › Streamlit: This user-friendly web application framework streamlines the creation of the chatbot interface, making it highly accessible for users without extensive technical expertise.



HOW DO YOU BUILD A LLM BASED CHATBOT TO QUERY YOUR PRIVATE KNOWLEDGE BASE

Low Level Architecture



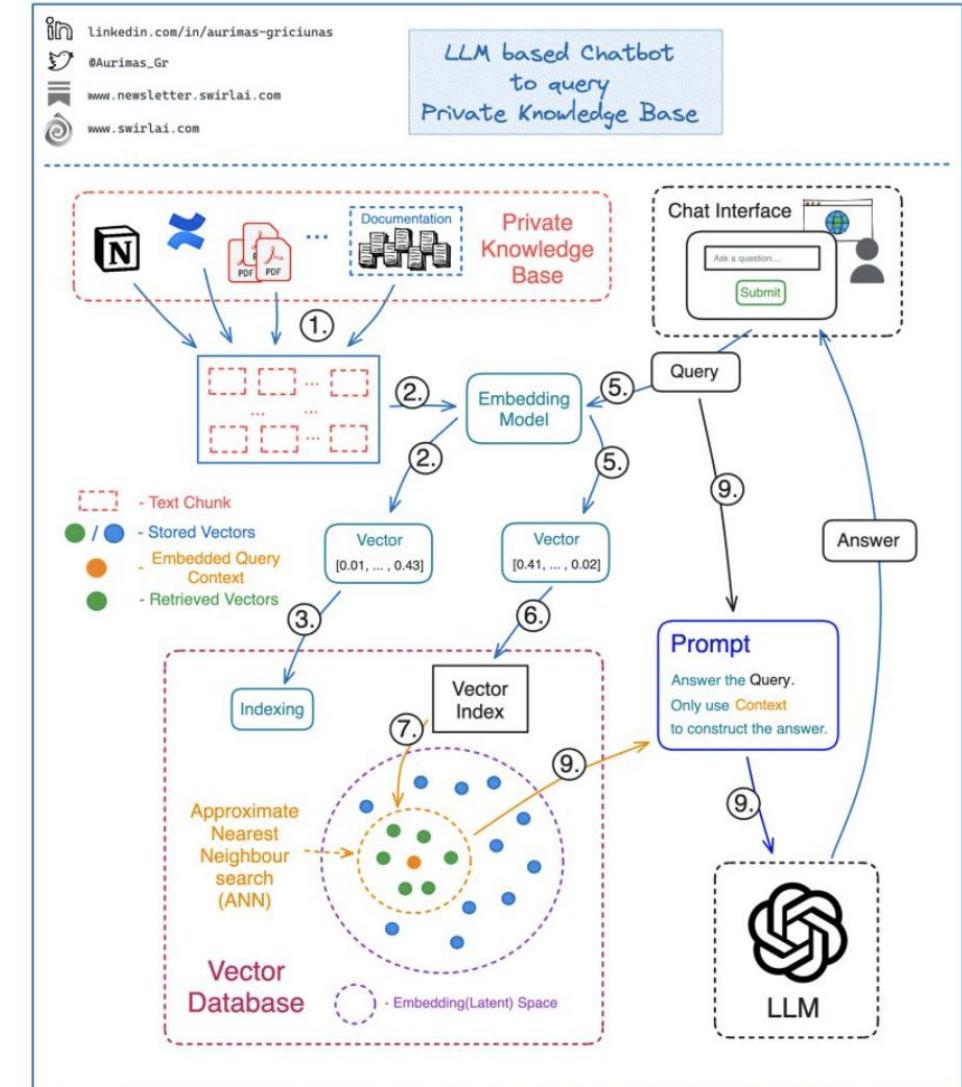
Preparing the Knowledge Base

- › Split Text: Divide the entire knowledge base (e.g., Confluence documentation, PDF reports) into manageable chunks.
- › Embed Text: Use an embedding model to convert each text chunk into a vector representation.
- › Store Vectors: Save these vectors in a Vector Database like Chroma, Pinecone, or Faiss.
- › Save Text with Pointers: Keep the original text associated with its vector for later retrieval.



Answering a Query

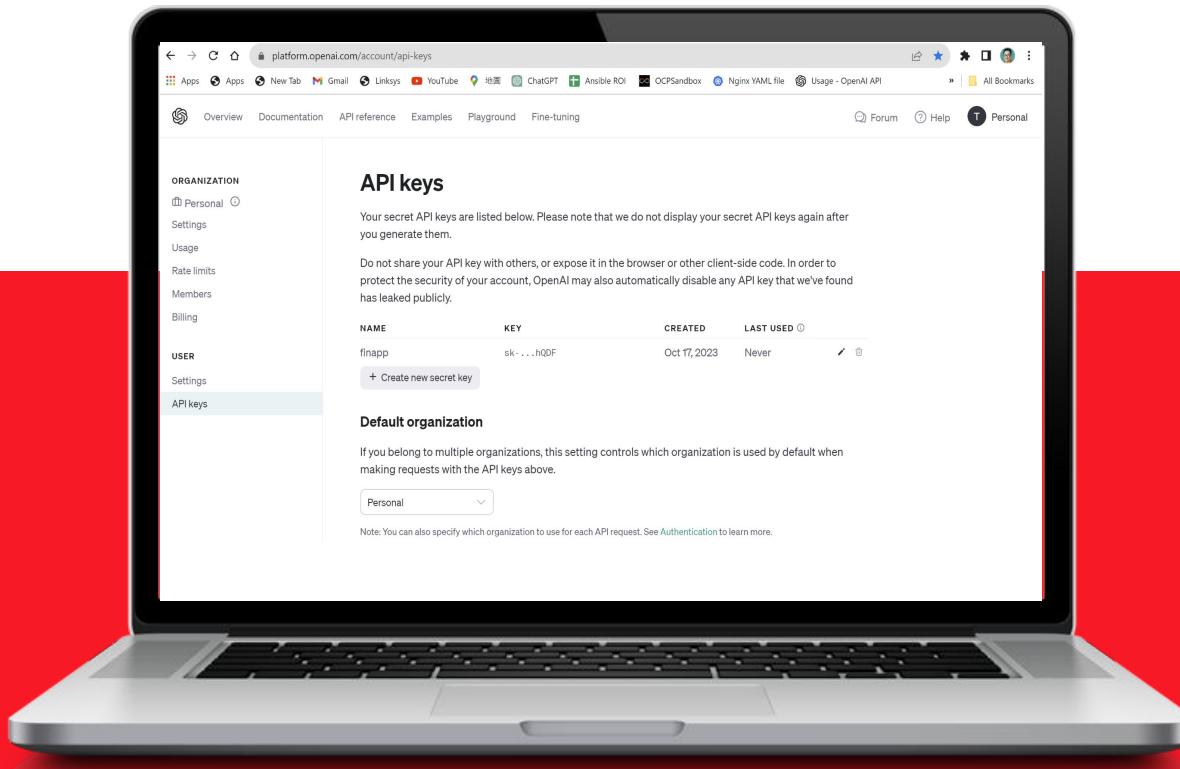
- › Embed the Query: Convert the user's question into a vector using the same embedding model.
- › Search for Relevant Vectors: Query the Vector Database to find the most similar vectors to the question vector.
- › Retrieve Context: Use the Approximate Nearest Neighbour (ANN) search to get the most relevant context vectors.
- › Map Vectors to Text: Link the returned vectors to their corresponding text chunks.
- › Generate Answer with LLM: Feed the question and the retrieved text chunks to the LLM. Ensure the LLM uses only the provided context to generate the answer, avoiding fabrications.



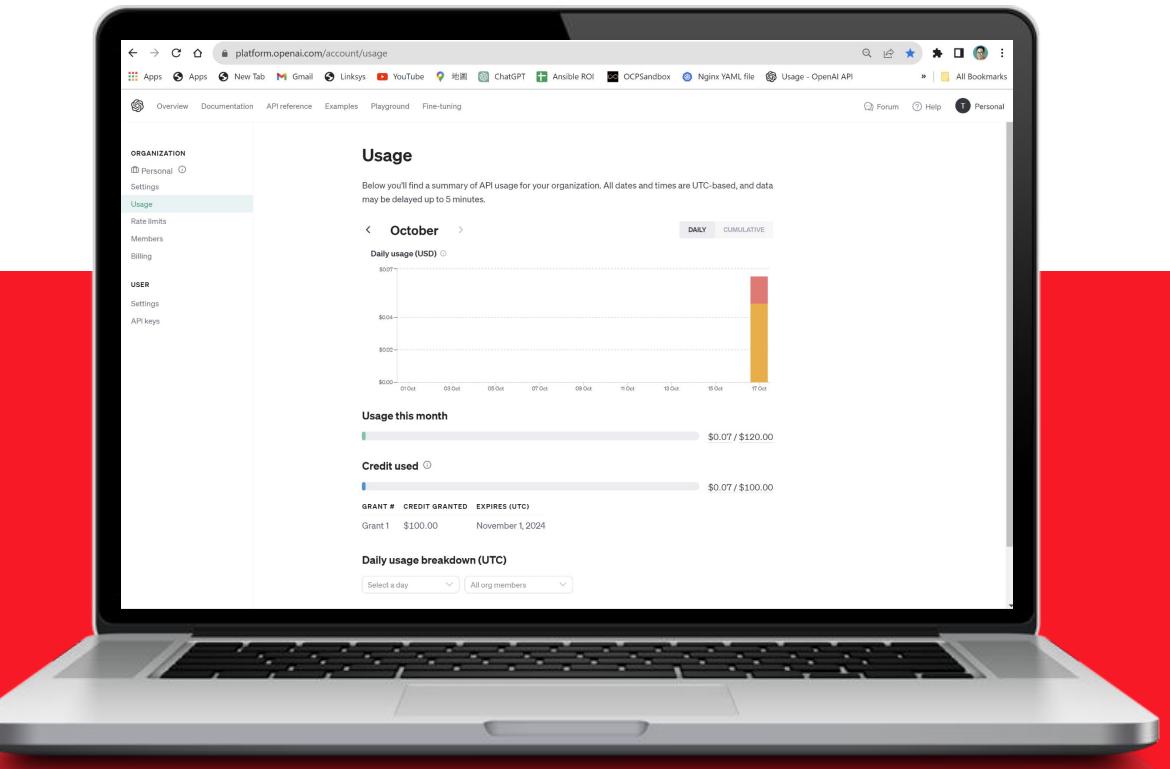
MY PROJECT: FINAPP CHATBOT

Enable OpenAI API Keys and Track Usage

The costs are favorable, **US\$0.04852** for one API call



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



<https://platform.openai.com/account/usage>

GUIDELINES FOR DEPLOYING THE CHATBOT APP

USING STREAMLIT AND DOCKER



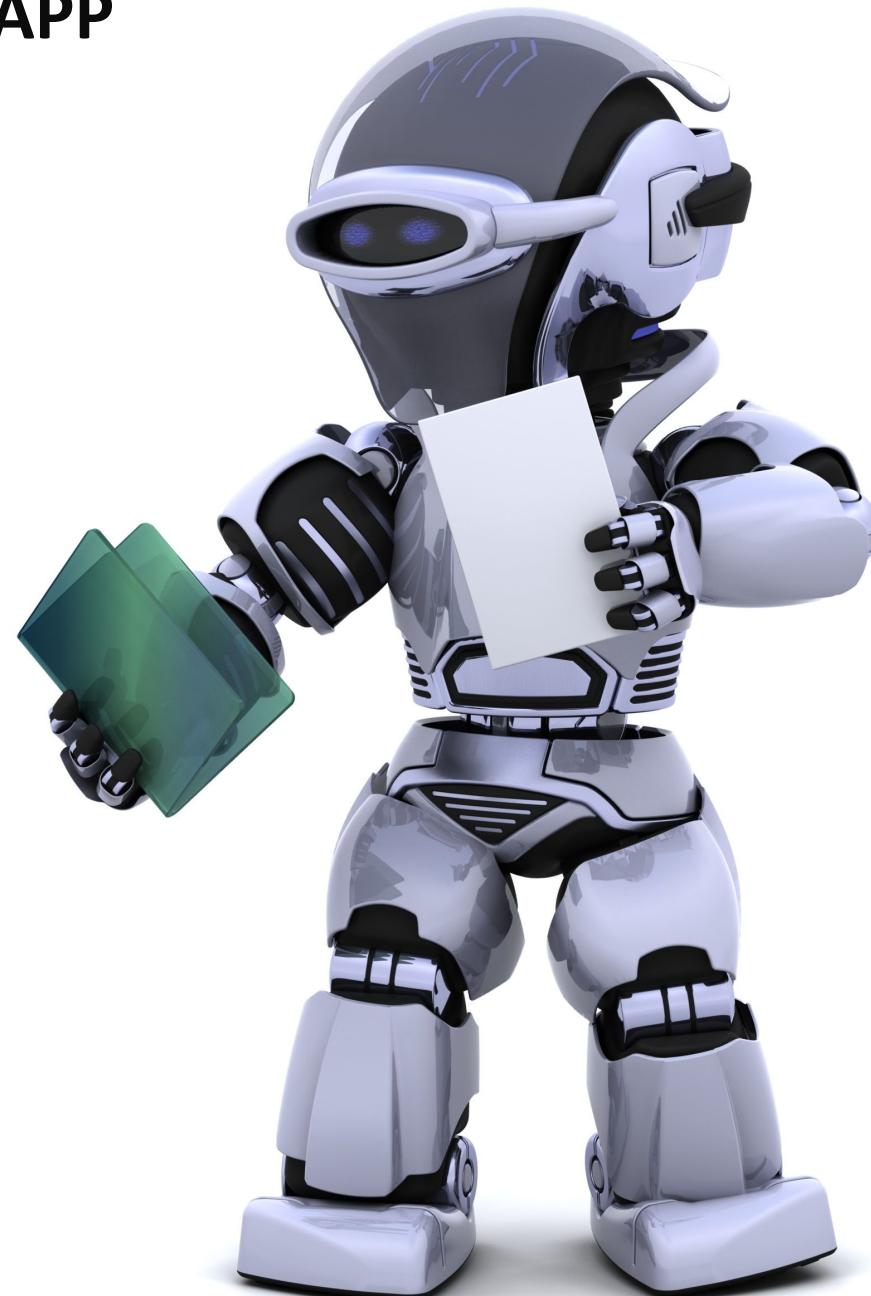
Steps to run the app via streamlit

- › Run in cmd
- › pip -r install requirements.txt.
- › streamlit run app.py



Steps to run the app via docker

- › 1. Pull docker image from the hub docker
`docker pull timlam007/finchat:latest`
- › 2. `docker run -d -p 8501:8501 timlam007/finchat:latest`
- › 3. You can navigate to the localhost endpoint and interact
with the application. <http://localhost:8501>



RUNNING CHATBOT IN OPENSIFT 4 SANDBOX

The repository located at https://github.com/timlam007/finance_chatbot, which is deployed as an application on the OpenShift Application Platform

Within this repository:

- The deployment.yaml and services.yaml files outline the procedures to deploy and install the finapp chatbot on OpenShift.

```
D:\github\finance_chatbot>oc apply -f deployment.yaml  
deployment.apps/finchat created
```

```
D:\github\finance_chatbot>oc apply -f services.yaml  
service/finchat created  
route.route.openshift.io/finchat created
```

```
D:\github\finance_chatbot>oc get route  
NAME      HOST/PORT                                PATH  SERVICES  PORT  TERMINATION  WILDCARD  
finchat   finchat-tinlam-dev.apps.sandbox-m2.ll9k.p1.openshiftapps.com          finchat    8501        None
```

```
D:\github\finance_chatbot>oc get pods  
NAME           READY  STATUS  RESTARTS  AGE  
finchat-75dbb655f6-wkff9  1/1    Running  0          5m48s
```

OCP Deployment Steps

RUNNING CHATBOT IN OPENSIFT 4 SANDBOX

The repository located at https://github.com/timlam007/finance_chatbot, which is deployed as an application on the OpenShift Application Platform

Within this repository:

- The tekton directory houses several YAML files that illustrate the automation process for deploying the application using OpenShift Pipelines.

```
D:\github\finance_chatbot\tekton>oc apply -f deploy-finchat-task.yaml  
task.tekton.dev/deploy-finchat-task created
```

```
D:\github\finance_chatbot\tekton>oc apply -f deploy-finchat-service-task.yaml  
task.tekton.dev/deploy-finchat-service-task created
```

```
D:\github\finance_chatbot\tekton>oc apply -f deploy-finchat-route-task.yaml  
task.tekton.dev/deploy-finchat-route-task created
```

```
D:\github\finance_chatbot\tekton>tkn task list  
NAME          DESCRIPTION     AGE  
deploy-finchat-route-task      14 seconds ago  
deploy-finchat-service-task    29 seconds ago  
deploy-finchat-task           53 seconds ago
```

```
D:\github\finance_chatbot\tekton>oc apply -f deploy-finchat-pipeline.yaml  
pipeline.tekton.dev/deploy-finchat-pipeline created
```

```
D:\github\finance_chatbot\tekton>oc apply -f deploy-finchat-pipeline-run.yaml  
pipelinerun.tekton.dev/deploy-finchat-pipeline-run created
```

```
D:\github\finance_chatbot\tekton>tkn pipelinerun describe pipelinerun.tekton.dev/deploy-finchat-pipeline-run  
Error: failed to find pipelinerun "pipelinerun.tekton.dev/deploy-finchat-pipeline-run"
```

```
D:\github\finance_chatbot\tekton>tkn pipelinerun list  
NAME        STARTED        DURATION   STATUS  
deploy-finchat-pipeline-run  3 minutes ago  37s       Succeeded
```

Tekton
Deployment
Steps

RUNNING CHATBOT IN OPENSHIFT 4 SANDBOX

Project: tinlam-dev ▾

PipelineRuns > PipelineRun details

PLR **deploy-finchat-pipeline-run** ✓ Succeeded

[Details](#) [YAML](#) [TaskRuns](#) [Parameters](#) [Logs](#) [Events](#)

PipelineRun details

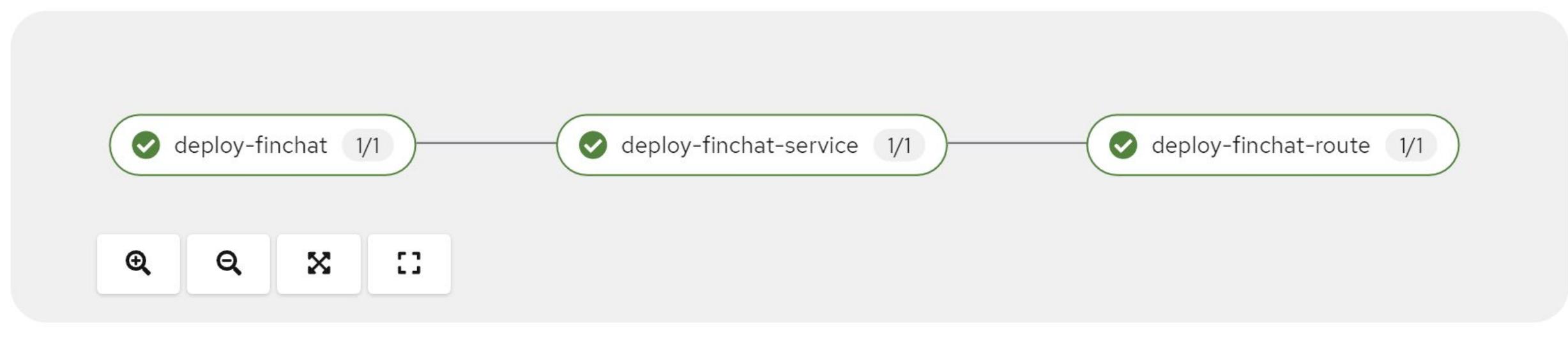
```
graph LR; A[✓ deploy-finchat 1/1] --- B[✓ deploy-finchat-service 1/1]; B --- C[✓ deploy-finchat-route 1/1]
```

🔍 🔍 ✖ ⋮

Name	Status
deploy-finchat-pipeline-run	✓ Succeeded
Namespace	Pipeline
NS tinlam-dev	PL deploy-finchat-pipeline
Labels	Edit ✎
<code>tekton.dev/pipeline=deploy-finchat-pipeline</code>	
Annotations	
2 annotations ✎	
Created at	
⌚ Nov 2, 2023, 11:40 PM	

RUNNING CHATBOT IN OPENShift 4 SANDBOX

PipelineRun details

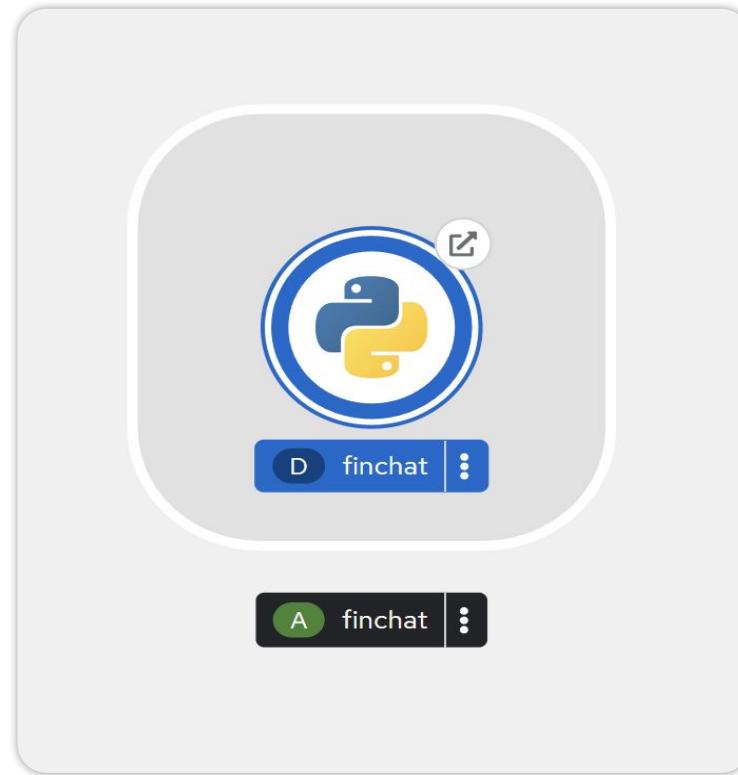


Tasks

Name	Namespace	Created
T deploy-finchat-route-task	NS tinlam-dev	Nov 2, 2023, 11:38 PM
T deploy-finchat-service-task	NS tinlam-dev	Nov 2, 2023, 11:38 PM
T deploy-finchat-task	NS tinlam-dev	Nov 2, 2023, 11:38 PM

RUNNING CHATBOT IN OPENSHIFT 4 SANDBOX

<https://console-openshift-console.apps.sandbox-m2.ll9k.p1.openshiftapps.com/topology/ns/tinlam-dev?view=graph&selectId=dd1e736f-dd10-43c9-86aa-35360e1ea407>



OCP Application FinChat

A screenshot of a laptop displaying the Red Hat OpenShift Dedicated interface. The left pane shows a sidebar with options like Developer, Topology, Observe, Search, Builds, Pipelines, Helm, Project, ConfigMaps, Secrets, and Desktop Tools. The main pane displays the "finchat" application under the "tinlam-dev" project. It shows a summary card with the Python logo and the text "finchat". On the right, there are tabs for Details, Resources, and Observe. Under the Details tab, it shows a Pod named "finchat-cdb8c548-72mtf" in the "Running" state, a Service named "finchat" (Service port: TCP/8501 → Pod port: 8501), and a Route named "finchat" (Location: http://finchat-tinlam-dev.apps.sandbox-m2.ll9k.p1.openshiftapps.com). A terminal window at the bottom shows the command "oc apply -f deployment.yaml" being run, followed by "D:\projects\finance_chatbot>oc apply -f services.yaml". The Red Hat logo is visible in the bottom right corner of the laptop screen.

RUNNING CHATBOT IN OPENSIFT 4 SANDBOX (Application GUI)

Finapp Chatbot also supports Chinese characters for searching and display and is multilingual, powered by the OpenAI LLM model

The screenshot shows a web browser window with a red border, displaying the Finapp Chatbot application. The URL in the address bar is `finchat-tinlam-dev.apps.sandbox-m2.ll9k.p1.openshiftapps.com`. The page has a header with various links: YouTube, 地圖, Sales, Google Translate, AI, Q12023, Personal, Product, Accounts, Win Case, BU, OCP Sandbox, Workday, TBR, GCG HR.

The main interface is divided into two main sections:

- Options (Left Panel):** Includes fields for "OpenAI API Key" (redacted), "Get an OpenAI API key", "Choose LLM:" (radio buttons for gpt-3.5-turbo-0613, gpt-3.5-turbo-16k-0613, gpt-4, text-davinci-003, with gpt-4 selected), "Temperature" (0.00 to 1.00 slider at 0.00), "What would you like to query?" dropdown (set to "Documents"), and "Clear Conversation" button.
- Upload Document or Connect to a Database (Right Panel):** A section for uploading files with a "Drag and drop files here" area (with a limit of 200MB per file for CSV, XLSX, PDF, DOCX) and a "Browse files" button. A file named "Victor-doc.docx" (12.1KB) is listed with a delete "X" icon. A green message bar below says "File Loaded Successfully!!".
- Personal FinanceGPT (Bottom Right):** A conversational interface showing the following exchange:
 - User: Who is Victor?
 - Bot: Victor is the person in charge of business in the Greater China region at Red Hat and has been in this position for five years.
 - User: 誰是Victor
 - Bot: Victor is the person in charge of business in the Greater China region at Red Hat and has been in this position for five years.
- Input Field:** A text input field at the bottom right with placeholder text "Input your question!" and a right-pointing arrow button.

FINAPP CHATBOT UI + FUNCTIONS

Developing cloud-native apps with Red Hat's OpenShift application platform

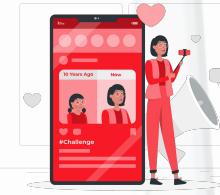


CREATIVITY

Is this a new and original idea, or has this been done before?

Answer

While chatbots for document retrieval are not new, FinAPP Chatbot stands out with its context-aware integration across various knowledge sources, including documents and databases. By focusing on context, FinAPP reduces potential inaccuracies, positioning it as an innovative solution in the chatbot landscape.



BUSINESS AVAILABILITY

How well does this solve a real business problem?

Answer

FinAPP Chatbot simplifies the process of navigating through vast company documents by allowing users to interact with a chatbot. This enables companies to easily access and extract crucial information from their knowledge bases, boosting productivity.

MY PROJECT: FINAPP CHATBOT

Developing cloud-native apps with Red Hat's OpenShift application platform



RELEVANCE

How have you combined relevant and interesting dataset and tools?



Answer

FinAPP seamlessly integrates diverse data sources, including PDFs, TXT, DOCX files, Excel, CSV, and databases like SQL Server and PostgreSQL. It leverages state-of-the-art technologies like GPT-3.5, LangChain Python Library, and Vector Stores (Chroma) to ensure accurate and contextually relevant responses. The use of Streamlit as a front-end tool further enhances user experience and accessibility.



THOROUGHNESS

Is your application or project easy for the end user to understand? Does it provide relevant and insightful information?

Answer

Absolutely! FinAPP is designed with the end user in mind. Its user-friendly chat interface, powered by Streamlit, ensures a smooth interaction. The system not only answers queries but also summarizes, rephrases, and presents information in a clear manner. By focusing on context, it ensures that the information provided is both relevant and accurate.

DEMO 3 USE CASES

These use cases illustrate the wide range of data sources and formats that the FinApp Chatbot can tackle, making it a valuable asset for businesses seeking intelligent data analysis solutions.

Use Case 1

Extracting data from an Excel file

[video 1](#)

Use Case 2

Extracting Data from a PDF File

[video2](#)

Use Case 3

Extracting Data from Postgres Database

[video3](#)



Excel Data Query

In the first use case, I demonstrated how the chatbot seamlessly loads an Excel file, enabling users to query and interact with the data within it effortlessly.



PDF Data Extraction

The second use case highlighted the chatbot ability to extract data from PDF files, making it a powerful tool for analyzing information stored in this common document format.



MySQL Database Integration

In the third use case, I showcased how the chatbot can connect to a Postgre database and extract data, expanding its capabilities to handle data stored in databases efficiently.

DEMO

Use Case 1: Loading an Excel File to query the data

Here, upload your documents you want AskMAY to use to answer

**Drag and drop files here**
Limit 200MB per File , CSV, XLSX, PDF, DOCX, TXT

Browse File

Loading the file via the GUI



DELL_OEM_Q4CY23.xlsx 48.8kb

X

	SKU	SKU Description	Product Line	CURRENCY	1-Year MSRP	1-Year Transfer Rate (25%)
	MCT0696	Red Hat Directory Server		USD	15000	
	MCT0697	Red Hat Certificate System		USD	6	
	MCT0826	Red Hat Directory Server (Replica)		USD	3000	
	MCT0840	Red Hat Certificate System (Starter Pack)		USD	10000	
	MCT0954	Red Hat Directory Server Small Business Bundle		USD	5000	
	MCT0955	Red Hat Directory Server Small Business Bundle w/ Red Hat Enterprise Linux Premium		USD	9000	
	MCT2735	Red Hat OpenShift Container Platform Premium (2 Cores or 4 vCPUs)		USD	4000	
	MCT2736	Red Hat OpenShift Container Platform Standard (2 Cores or 4 vCPUs)		USD	2700	
to	MCT2739	Red Hat Jobs Enterprise Application Platform for OpenShift Container Platform Premium 2-Core		USD	1500	
1	MCT2748	Red Hat JBoss Enterprise Application Platform for OpenShift Container Platform Standard 2-Core		USD	1000	
2	MCT2884	Red Hat OpenStack Platform (without guest os). Premium (2-sockets)		USD	3098	
3	MCT2885	Red Hat OpenStack Platform (without guest os). Standard (2-sockets)		USD	2383	
4	MCT2886	Red Hat OpenStack Platform Premium (2-sockets)		USD	6298	
5	MCT2887	Red Hat OpenStack Platform Standard (2-sockets)		USD	4845	
6	MCT2979	Red Hat OpenStack Platform with Satellite Premium (2-sockets)		USD	7518	
7	MCT2980	Red Hat OpenStack Platform with Satellite, Standard (2-sockets)		USD	5783	
8	MCT2981	Red Hat OpenStack Platform (without guest os) with Satellite Premium (2-sockets)		USD	3448	
9	MCT2982	Red Hat OpenStack Platform (without guest os) with Satellite Standard (2-sockets)		USD	2652	
to	MCT3326	Red Hat OpenShift Dedicated Single AZ Cluster Fee (General Purpose (4x 4vCPU,16GB RAM). Yearly)		USD	36000	
1	MCT3327	Red Hat OpenShift Dedicated Additional Node (General Purpose (4vCPU,16GB RAM). Yearly)		USD	3000	

File: DELL_OEM_Q4CY23

USE CASE 1: EXTRACTING DATA FROM AN EXCEL FILE (EXAMPLES 1A, 1B)

DELL_OEM_Q4CY23.xls



What is SKU MCT2735 Product Description?

Question 1a



The product description for SKU MCT2735 is “Red Hat OpenShift Container Platform Premium



How many items are in the SKU column?

Question 1b



There are 69 items in the SKU column in df1 and 586 items in the SKU column in df2.

USE CASE 1: EXTRACTING DATA FROM AN EXCEL FILE (EXAMPLES 1C, 1D)

DELL_OEM_Q4CY23.xls



What is the price of SKU MCT2735 and MCT2736 for 1 year subscription?

Question 1c



The price of SKU MCT2735 for 1 year subscription is 4000.0 and the price of SKU MCT2736 for 1 year subscription is 2700.0



What is the price of SKU MCT2735 and take a 15% Discount Please?

Question 1d



The price of SKU MCT2735 after a 15% discount is \$3400

USE CASE 2 : EXTRACTING DATA FROM A PDF FILE (EXAMPLE 1)

DevOps_with_OpenShift.pdf



What is container?



A.container is the runtime representation of a packaging format based on a lightweight, immutable image. Runtime dependencies are resolved within the image which facilitates portability. This makes possible the agreement on a standardized software work product. Container-based workloads are suitable for multi-tenancy on a single compute instance and when implemented securely can realize significant operation efficiencies. An important corollary is that launching a new workload does not incur the cost of provisioning new compute infrastructure. This enables a true on-demand, self-service experience for users.

USE CASE 2 : EXTRACTING DATA FROM A PDF FILE (EXAMPLE 2)



What are the keys difference between DevOps and containers? Please provide a table summarizing these differences



DevOps and Containers are two different concepts in the field of software development and deployment. DevOps is a methodology that aligns the constituents in the software delivery process, fostering professional respect and shared sensibility across the delivery team. On the other hand, Containers are the runtime representation of a packaging format based on a lightweight, immutable image. They resolve runtime dependencies within the image, facilitating -portability and enabling a standardized software work product. Containers are suitable for multi- tenancy on a single compute instance and can realize significant operation efficiencies.

DevOps	Containers
A methodology aligning the constitutes in the software delivery process	The runtime representation of a packaging format based on a lightweight, immutable image.
Fosters professional respect and ahared sensibility across the delivery team.	Resolves runtime dependencies within the image, facilitating profitability
-	Enables a standardized software work product
-	Suitable for multi-tenancy on a single compute instance
-	Can realize significant operation efficiencies

USE CASE 3 : EXTRACTING DATA FROM MYSQL DATABASE (EXAMPLE 1)



How many tables in the timedb database and what they are?



There are 3 tables in the timeDB database: product01, and smartphones.