

DSO 599 – Hands-on Data Analytics and Machine Learning on AWS Cloud
Saturday: 9:00 - 12:00 noon (3.0 Units)

Instructor:	<i>Sudi Bhattacharya</i>
Office:	<i>Virtual</i>
Office Hours:	<i>by Appointment</i>
Phone:	<i>(818) 264 8006</i>
Email:	sudibhat@marshall.usc.edu

Group Project Two - Creating an LLM Driven Chatbot (Total Points 250 – The Document and Code will carry 200 points and your presentation on Final Day carries 50 points)

Students: Linh Pham, Faez Aroos, Grace Li, Johnson Liu

Instructions for Submission:

- Please name the file with your group name-spring24-project 2
- Please write the names of students in the group in the section above
- Note the tasks and instructions and perform the tasks.
- Every task has a set of outputs, code files and/or a small writeup.
- Include the write-up in the main body of the document
- Attach your code files with this submission doc and refer to the attachment in the main body (don't paste code in this doc)
- Prepare and present your work on Final Examination day

Grading Rubric

	Unsatisfactory (30%)	Satisfactory (60%)	Good (80%)	Excellent (100%)
Requirements Coverage (30%): Does the code do what it is supposed to do?	I am not expecting anybody to get unsatisfactory here. The requirements are straightforward in the assignments. If you cover edge cases, optimize any specific way, that's when you get excellent.			
Coding - Logical and Maintainable (10%): Is the logic used clear, is this code modular?	I am looking for modularity (if you are using functions, or even modules) so that the code is easy to change. Remember the person maintaining the code may not have the same background as you have. There are no complex logic in the assignments.			
Coding Standards (10%): indentation, naming convention, use of variables, documentation	Following good coding practice is important. You can get a lot of details on Python practices here . Name variables meaningfully, indent properly and consistently. See this article for SQL standards that people follow. Document your code (SQL and Python code) so that somebody coming in can understand what you are writing.			
Actual Results Answering the	Be concise, thoughtful. Discuss alternatives, better ways of doing it. Discuss what you learned. No more than 250 words per answer. I will			

questions (30%): Answering the questions in assignment will count in this section.	deduct 10 points for answers that are over 250 words, 20 points for answers that are over 300 words.
Presentation (20%): Is your code achieving the intended outcome	Approach taken to solve the problem, performance improvement on the queries, reporting of the performance improvement will be considered in this section.

Task 1 – GenAI Application Development - Create a Streamlit Chatbot (15 points)

Task Instructions

- Write a Python program to create Streamlit chatbot that has the following functionality
 - Input box that accepts user inputs
 - Radio buttons to choose one of the three temperature values as an input to your LLM (0, .4, .9)
 - Select box to choose the ChatModel/LLM
 - Bot should display at least 3 last messages from the user
 - **Extra credit – A way for the users to change the System Message
-

Output from Task 1:

- Include screenshots of the Bot in various actions
- Attach your python file(s) with this doc
- Comment on your observations, challenges

Task 1 Code File: task1_openai_final.py

Task 1 Observations & Challenges:

In our initial task, we experimented with various OpenAI models and seamlessly integrated the OpenAI API with Streamlit, enabling user interaction through a chat interface. Our approach was heavily influenced by Streamlit's documentation, which provided a structured approach (<https://docs.streamlit.io/develop/tutorials/llms/build-conversational-apps>). This resource guided us in designing the interface, incorporating a sidebar for adjusting temperature settings and selecting different OpenAI models, offering users flexibility and customization options.

Utilizing Streamlit's session state, we managed chat history, and temperature settings, and selected OpenAI models. This allowed the chatbot to maintain conversational context, deliver coherent responses based on previous messages, and store and display chat history within the application, enabling users to review previous messages and better understand the chatbot's responses. By mimicking the behavior of popular messaging applications, the chat history enhances the user experience, creating a natural and intuitive conversational flow.

For extra credit question, being able to change the system message is a nice feature, as it allows users to experiment with different personas or prompts for the chatbot. However, integrating the system message presented challenges initially. Occasionally, the chatbot's responses did not align with our expectations set by specific system messages. This limitation is inherent to LLMs, as they may struggle to accurately interpret and adhere to complex or nuanced prompts. We understand that this

challenge arises due to LLMs' training on vast internet data, and lack of a deep understanding of specific personas or contexts that humans possess.

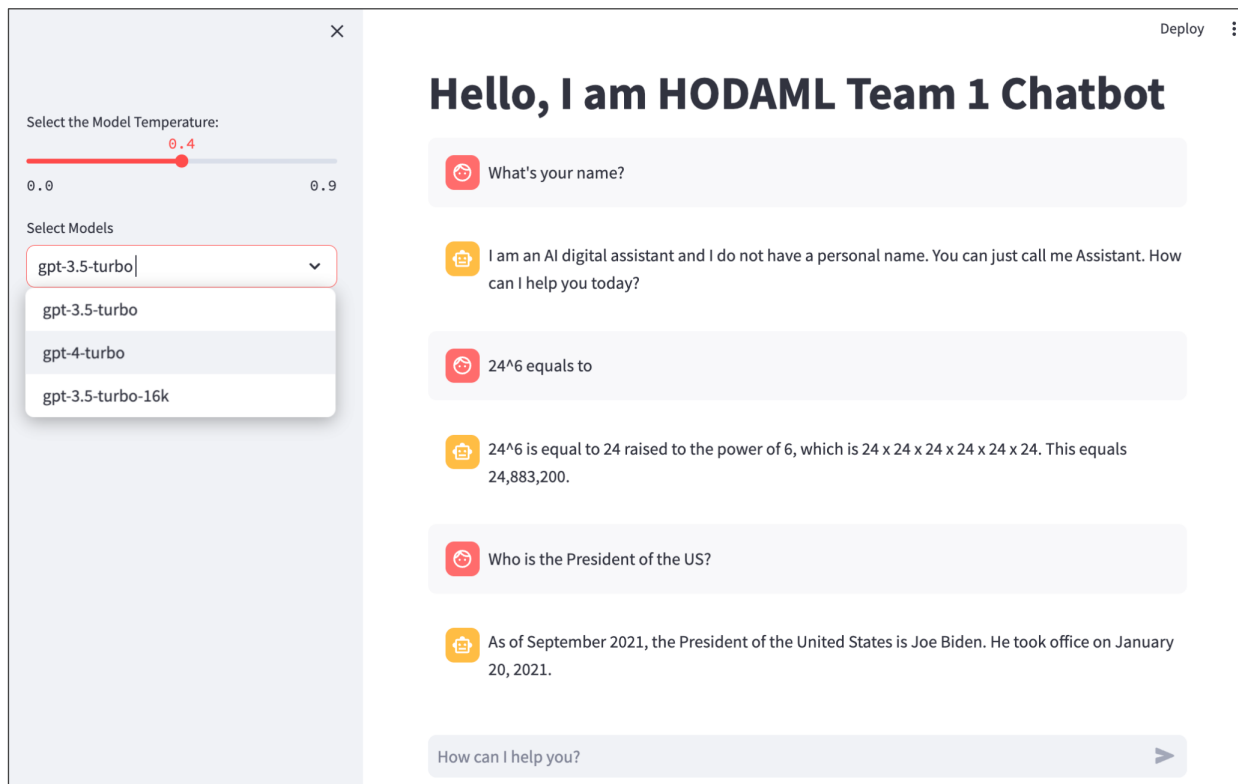


Figure 1: Our Team Chatbot Streamlit Interface

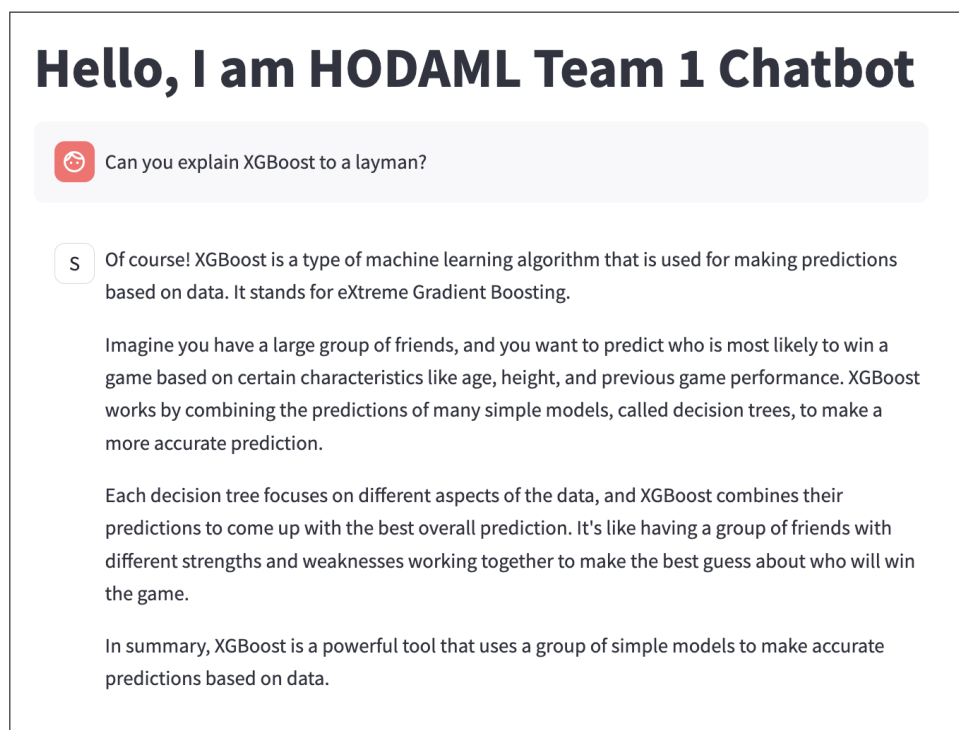


Figure 2: The more specific the prompt, the bot can give us the answer we want

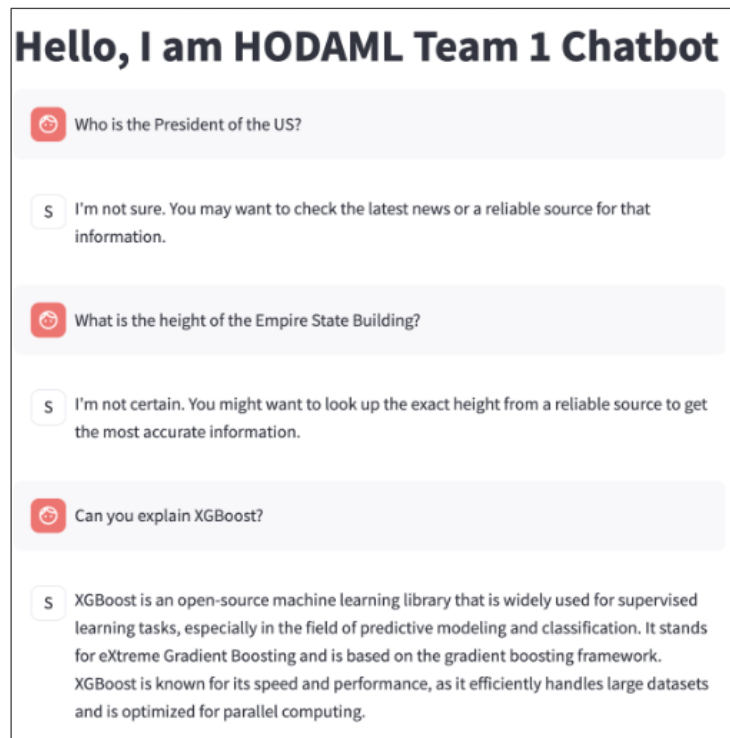


Figure 3: Chatbot Answers with system message: “You are unhelpful assistant”

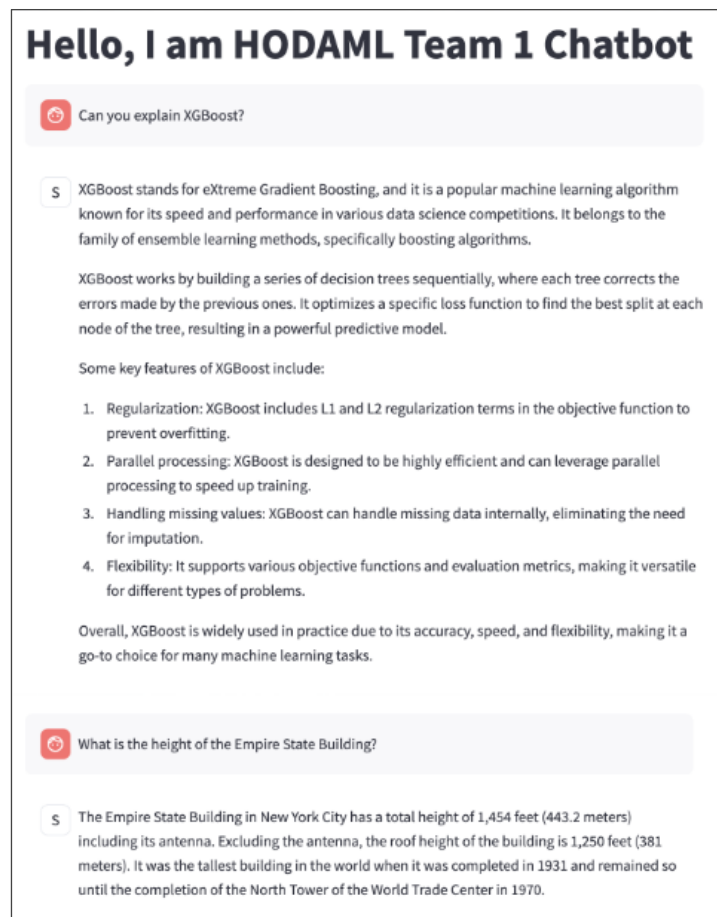


Figure 4: Chatbot Answers with system message: “You are helpful assistant”

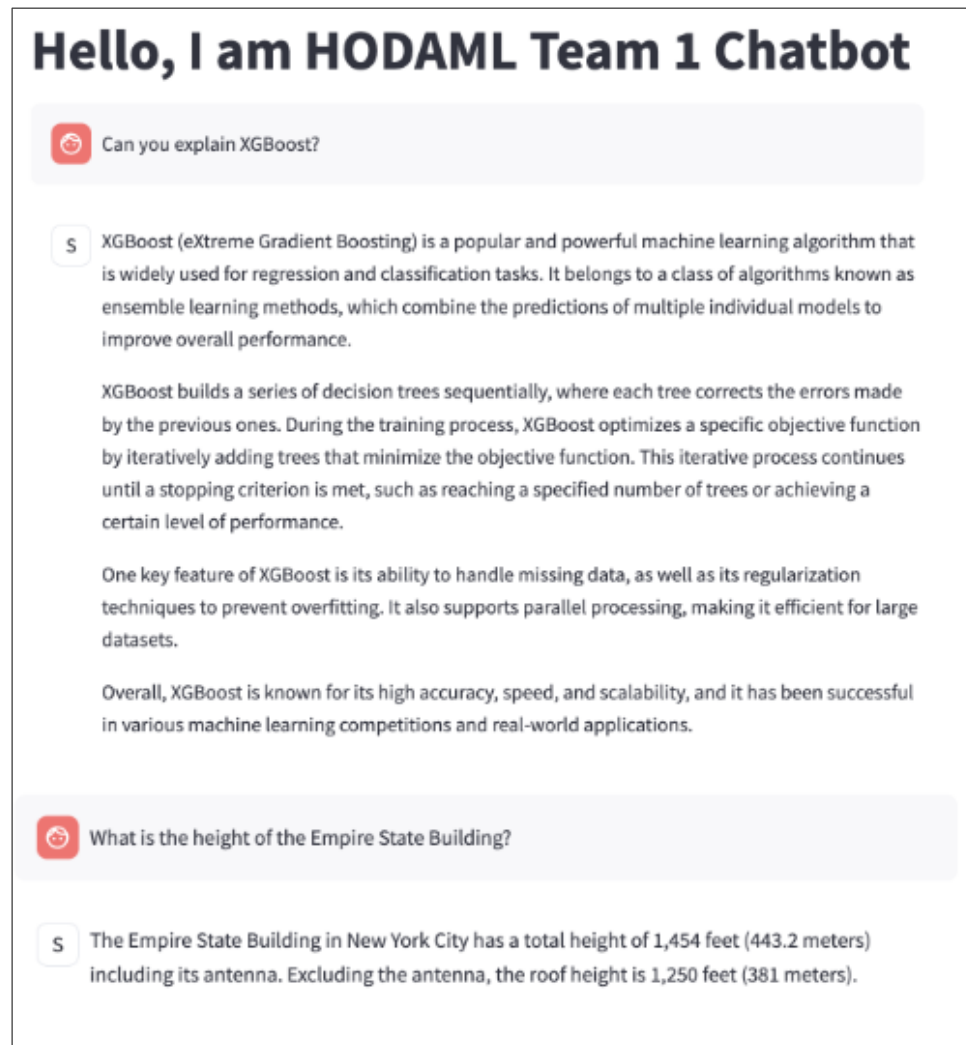


Figure 5: Chatbot Answers with system message: “You are PhD Student at Harvard University”

Task 2 – GenAI Application Development – Create A Vector Datastore and use it to retrieve the following: (15 points)

Task Instructions

- Create a Vector datastore with the two documents - TRexSafeTemp.pdf and VelociraptorsSafeTemp.pdf.
- Write a tool to use the Vector Store to retrieve information using a React Agent. Check out this [link](#) to learn how to create a retriever tool (create_retriever_tool). Check out this [link](#) for creating ReAct agents.
- Demonstrate you can answer the following questions using your application
 - What is the safe temperature for T-Rex Transport?
 - What safety measures should be taken for the Velociraptor when the temperature is 90 degrees?
 - Craft an email to the management giving them a status report on your situation.
- ** I have also provided the word documents. If you are so inclined, you could play with making minor tweaks in the source to see if you can get better responses. Just an idea.

These are some of the example responses I got:

Hi, I am *HODAML*, a FAISS RAG Copilot



The safe temperature range for Velociraptor transportation is between 40°F and 55°F (4°C and 13°C). This information can be found on page 1 of 4 in the document "JP-ZOO-INST-24: Recording Temperatures for Transport of Velociraptors."

Hi, I am *HODAML*, a FAISS RAG Copilot



Based on the provided context, it is not safe for a T-Rex to be at a temperature of 45 degrees. According to the work instruction, the safe temperature range for T-Rexs is between 57°F and 68°F. At 45 degrees, the temperature is outside of this safe range, and it could potentially cause harm to the T-Rex. Therefore, it is not safe for the T-Rex to be at a temperature of 45 degrees.

Output from Task 2:

- Include screenshots of the Bot in various actions
- Comment on experimentation you had to do to get the right answer
- Attach your python file(s) with this doc

Task 2 Code File: task2_ReAct_final.py

Q1: What is the safe temperature for T-Rex Transport?

Hi, I am *HODAML*, ask me something about the pdf

How can I help you today?

What is the safe temperature for T-Rex Transport?



The safe temperature for T-Rex transport is between 57°F and 68°F.

☐ Show Document Sources

Figure 6

Hi, I am *HODAML*, ask me something about the pdf

How can I help you today?

What is the safe temperature for T-Rex Transport? And show me where you got the answer.



The safe temperature for T-Rex Transport is between 57°F and 68°F. This information was retrieved from the document "Recording Temperatures for Transport of T-Rexs" where it specifies the safe temperature range for T-Rexs during transport operations.

☐ Show Document Sources

Figure 7: A more specific prompt

Q2: What safety measures should be taken for the Velociraptor when the temperature is 90 degrees?



When the temperature is 90 degrees, safety measures should be taken to ensure the Velociraptors are kept safe. Velociraptors are not safe when temperatures are out of the safe temperature range, which is 40°F-55°F for Velociraptors. Here are the steps to be followed:

1. Monitor the temperatures in the cargo area of the DINO vehicle using the Planet TYR55 sensor.
2. If the temperature variance can be resolved by the transporter, notify all staff on the specific alert calling tree that the temperature problem has been fixed.
3. If the temperature cannot be controlled within the specified range, inform the Yard Centre Manager (YM) that the problem cannot be fixed.
4. The YM should then notify the Associate Director of Operations and Strategy and work with the transporter to move the Velociraptors to a safe area where the temperature is between 40°F and 55°F until the issue is resolved.
5. Note the temperature and conditions in the new location where the Velociraptors are moved, along with the time they were moved and the time they were moved back into the freighter.

These safety measures are outlined in the document "Recording Temperatures for Transport of Velociraptors" from the DINO Operations Staff guidelines.

Figure 8

Q3: Craft an email to the management giving them a status report on your situation.


When we asked this question initially, the agent asked for further information (Figure 9).

Hi, I am *HODAML*, ask me something about the pdf

How can I help you today?

Craft an email to the management giving them a status report on your situation?

Press Enter to apply



I have found information related to temperature monitoring and alerts in the documents. However, I will need more specific details about the status report you want to provide to the management. Could you please provide me with the key points or updates that you would like to include in the email? This will help me craft a more tailored email for you.


☐ Show Document Sources

Figure 9

After adding “I want to include the current status, challenges, and next steps in my email.”, it outputs an email template successfully (Figure 10).

How can I help you today?

Craft an email to the management giving them a status report on your situation? I want to include the c



Subject: Status Report on Current Situation

Dear Management,

I hope this message finds you well. I am writing to provide you with a status report on our current situation, highlighting the progress made, challenges faced, and the next steps to be taken.

Current Status:

- We have successfully completed the initial phase of the project, including research and planning.
- The team has been working diligently to meet project deadlines and deliverables.
- Key stakeholders have been engaged and updated on the project's progress.

Challenges:

- Despite our efforts, we have encountered delays in the implementation phase due to unforeseen technical issues.
- Limited resources and manpower have posed challenges in meeting the project requirements within the set timeline.
- Communication gaps between team members have led to misunderstandings and coordination issues.

Next Steps:

- Address the technical issues promptly by consulting with experts and implementing appropriate solutions.
- Allocate additional resources to expedite the project timeline and ensure timely delivery.
- Conduct a team meeting to improve communication and collaboration among team members.

I am confident that with the collective efforts of the team and the support of management, we will overcome these challenges and successfully complete the project. Thank you for your continued guidance and support.

Please feel free to reach out if you require any further information or clarification on the status report.

Best regards,

[Your Name] [Your Position]

Figure 10

Challenges & Observations:

While experimenting, we encountered a challenge in loading and retrieving two specific PDF files from a folder. Initially, we attempted to use `PyPDFDirectoryLoader.load()` to load the files. However, we encountered a `TypeError` stating that `PyPDFDirectoryLoader.load()` accepts only one positional argument, while two were provided. This error suggested that the loader does not support directly loading files by name within a loop.

To address this issue, we attempted to manually filter the documents after loading them. Unfortunately, this approach led to a `KeyError` associated with the key 'filename'. To debug this, we added print statements to examine the structure and metadata of the loaded documents. We discovered that the metadata from `PyPDFDirectoryLoader` does not contain a 'filename' key but includes a 'source' key, which provides the full path to the file. We resolved the issue by filtering the documents based on their 'source' paths.

This experience highlighted the complexity of maintaining the application, emphasizing the need for careful file loading, and ensuring compatibility with the system. The application proves extremely useful for retrieving information from PDFs, contingent on the quality of the documents uploaded. Moreover, our testing revealed that the specificity of the prompts provided to the application significantly influences the quality of the responses generated.

Task 3 – GenAI Application Development – Create Tools for an AI Agent to execute tasks (90 points)

- Write a function to create a DynamoDB database, read a csv file and upload the data in the database. The csv file should have these two records. (10 points)

Route_Number	Date	City	DinoID_Transported
123999	3/19/2024	Anchorage	T88
123878	3/20/2024	Seattle	V66

- Write a function to retrieve City and DynoID given a date. Create a tool with proper description and use an LLM to invoke the tool to retrieve DynoID and City for 3/19/2024 (10 points)
- Create a SQL agent that can retrieve a Dino name given a Dino ID. Use the function above to retrieve the Dyno ID and City name given a date and use the SQL Agent to get the Dyno Name from a SQL Database with the DinoMap table (you have to create the table in SQLite). You can find examples [here](#). (20 points)

ID	Name
T88	T-Rex
V66	Velociraptor

- Now, use a tool to find out the current temperature of a City. Show the output. (10 points)
 - Use the LLM to craft a text message with actions taken to keep the Dino safe as the temperature is outside the range. (20 points)
 - Write a function (Boto3) to send a text message to a given phone number using AWS SNS. (20 points)
- ** You can perform these tasks separately. Although an action sequence is implied, I am not asking you to link them together yet.

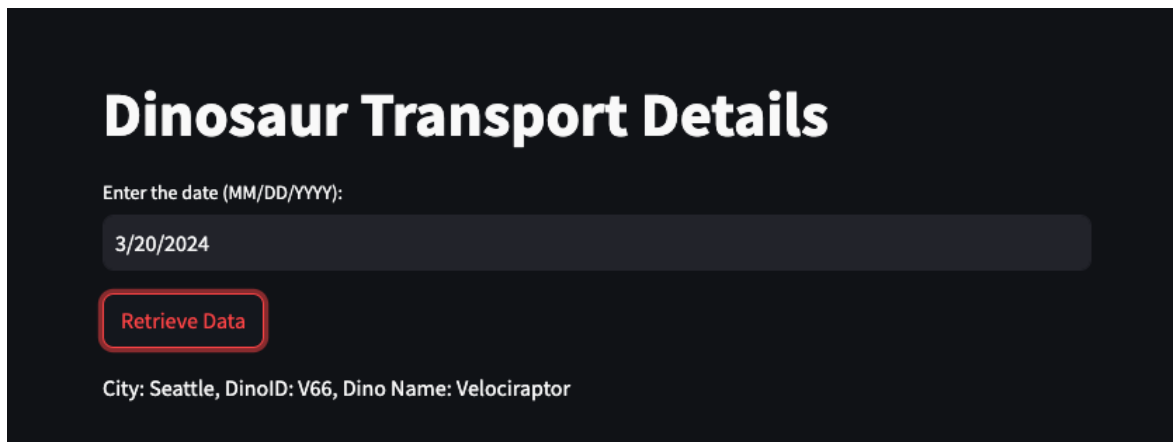
Task 3 Code Files: task3_tools.py and task3_langchain.py

Task 3 Observations:

The screenshot shows the AWS DynamoDB console interface. On the left, a sidebar lists two tables: 'class_13' and 'DinoTransport', with 'DinoTransport' selected. The main panel is titled 'DinoTransport' and includes a 'Scan or query items' section. The 'Scan' option is selected, and the table 'Table - DinoTransport' is chosen. Below this, there are 'Filters' and 'Run' buttons. The 'Items returned (2)' section displays a table with the following data:

Date (String)	Route_Number (Number)	City	DinoID_Transported
3/20/2024	123878	Seattle	V66
3/19/2024	123999	Anchorage	T88

Figure 11: Reflect the creation of a DynamoDB database with the correct data after reading csv file



Dinosaur Transport Details

Enter the date (MM/DD/YYYY):

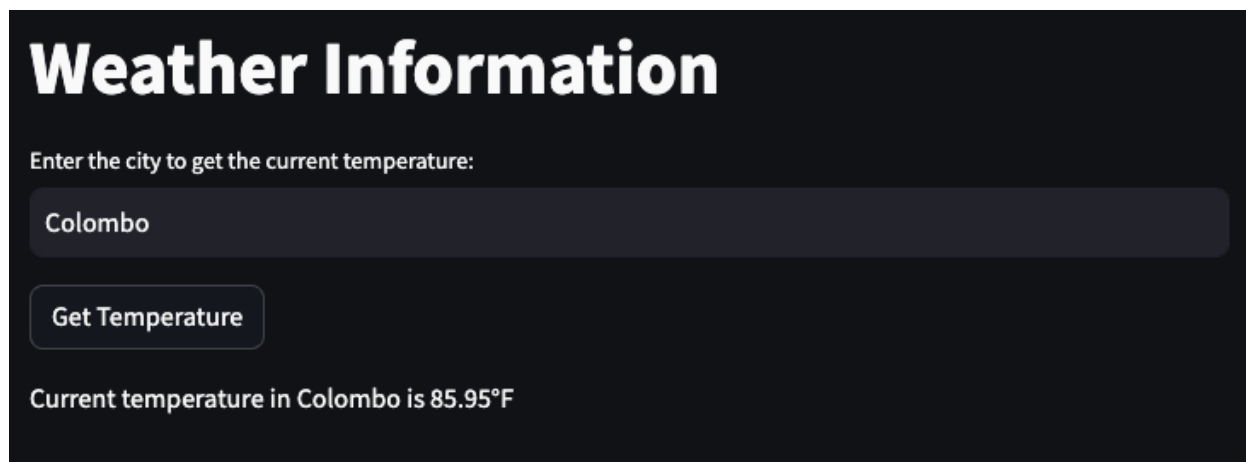
3/20/2024

Retrieve Data

City: Seattle, DinoID: V66, Dino Name: Velociraptor

Figure 12: Retrieve the Dyno ID and City name given a date

We combined the code for questions 2-3 of task 3 as one output through our Streamlit app. Our Streamlit app allows the user to enter a date to output the details of Dinos transported for that specific date. We were able to successfully create the Dino database reflected in the downloads. We created functions to retrieve the City, DynoID (from DynamoDB), and Dino name (from SQL database) given a date and linked it to Streamlit through AI. We struggled with inserting the OpenAI API key directly through code but resolved the issue by storing the keys through Streamlit's secret management as secrets.toml file. An alternate approach to this task would be to use Langchain (task3_langchain.py), which we implemented in task 4 to produce end-to-end workflow.



Weather Information

Enter the city to get the current temperature:

Colombo

Get Temperature

Current temperature in Colombo is 85.95°F

Figure 13: Find out the current temperature of Colombo, Sri Lanka

By implementing the Open Weather API endpoint, our team can produce the current temperature for any city.

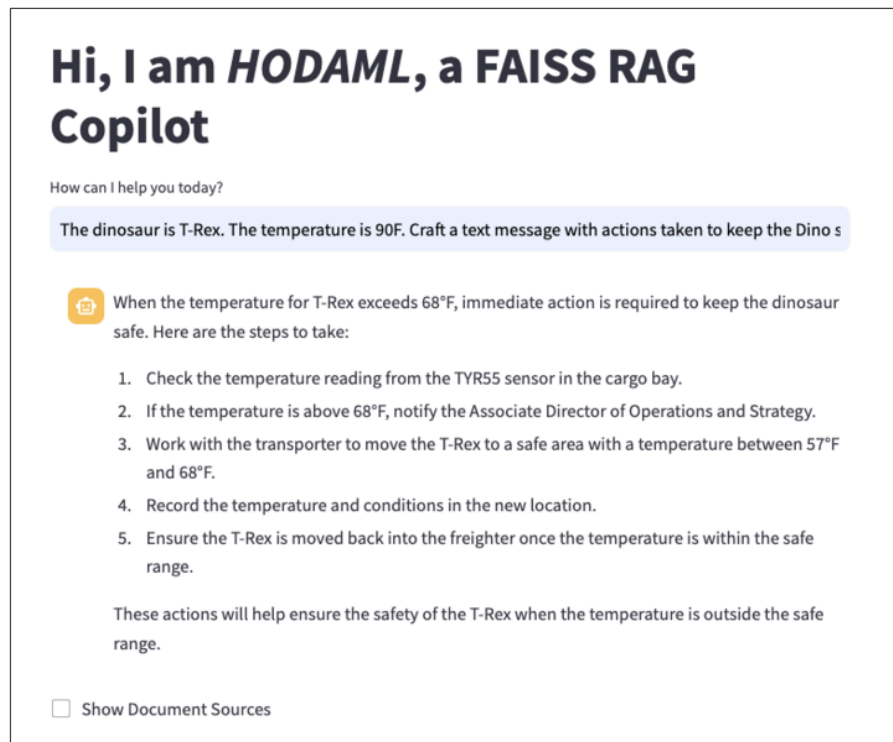


Figure 14: Use the LLM to craft a text message with actions taken to keep the Dino safe

We utilized the ReAct Agent developed in task 2 to compose a text message detailing actions taken to ensure Dino's safety, as the temperature was outside the safe range. We also specify the dinosaur name and the temperature in the prompt.

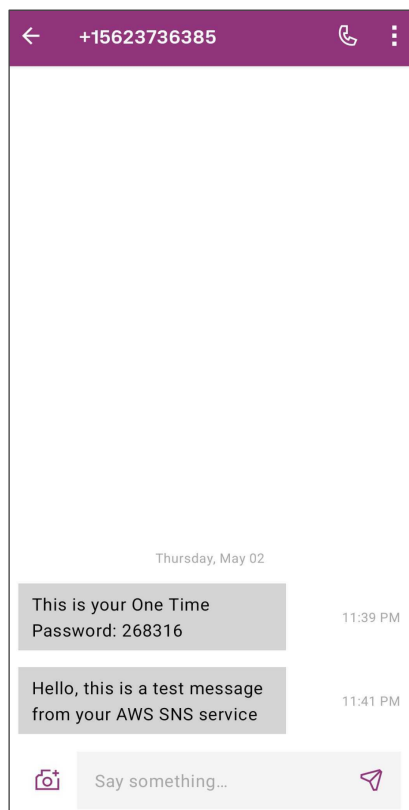


Figure 15: Send a text message to a given phone number using AWS SNS

To send a text message via AWS SNS using Boto3 (task3_langchain.py), we initialized the SNS client with appropriate credentials and region information. Then, we formatted the message content and specified the recipient's phone number. Using the publish() method of the SNS client, we successfully sent the message. Any errors or exceptions during this process are handled for reliability.

Task 4 – GenAI Application Development – Create a program that executes the end-to-end workflow(80 points)

This is the most fun part of the exercise. You have access to various tools and functions that you have created. Can you create an end-to-end workflow where you start with a date (3/19/2024), find out the Dino that's being transported and the city in which the transportation is happening on that date, find out the temperature of the city, find out if the temperature is safe or not for the Dino in question if the temperature is not safe, find out the actions that are needed to keep the dino safe, craft a status email to the manager and send the text given a phone number? You can use any combination of LLM calls and functions you want. The idea is to use an AI Agent end-to-end to perform these tasks using these tools but you are free to choose any approach.

In this task, we utilized various LangChain framework tools and functionalities, such as Document Loaders, Vector Stores, Retriever Tool, OpenAI Tools Agents, AgentExecutor, and Streamlit Integration. Our approach involved defining separate functions to enhance modularity, code reusability, and workflow management and then using an LLM to invoke those functions. Each function handled specific tasks, such as interacting with DynamoDB, SQLite, OpenWeatherMap, or AWS SNS, contributing to a more organized and maintainable codebase.

By breaking down the application into modular functions, we improved readability, facilitated unit testing, and promoted collaboration. Using AgentExecutor facilitated smooth communication between the agent and tools, ensuring the workflow's seamless execution. Finally, we integrated these functionalities with Streamlit resulting in a user-friendly chat interface, allowing users to input dates and receive relevant information, safety recommendations, and status updates. Additionally, we incorporated a Streamlit button to separate the email functionality from the information retrieval process. This design choice enhances the interface's readability and interactivity, providing users with a clear distinction between retrieving information and generating emails.

Our approach offers efficient information management, contextual understanding through LLMs, and seamless integration of multiple data sources and services. The modular nature of LangChain enables scalability and extensibility, allowing for the incorporation of new data sources, tools, or LLM models and ensuring the application remains future-proofed.

Overall, this project was very fun for us where we had a chance to experiment with different tools, use OpenAI and Ollama models, and create Streamlit Interface to have the bot answer for us.

```

> Entering new AgentExecutor chain...
Invoking: `get_city_and_dinoId` with `{'date': '3/19/2024'}`

[{'DinoID_Transported': 'T88', 'Date': '3/19/2024', 'Route_Number': Decimal('123999'), 'City': 'Anchorage'}]
The DynoID for 3/19/2024 is T88 and the city is Anchorage.

> Finished chain.

> Entering new SQL Agent Executor chain...
Invoking: `sql_db_list_tables` with `{}`

DinoMap
Invoking: `sql_db_schema` with `{'table_names': 'DinoMap'}`

CREATE TABLE "DinoMap" (
    id TEXT,
    name TEXT,
    PRIMARY KEY (id)
)

/*
3 rows from DinoMap table:
id      name
T88     T-Rex
V66     Velociraptor
*/
Invoking: `sql_db_query` with `{'query': "SELECT name FROM DinoMap WHERE id = 'T88'"}`

[('T-Rex',)]The Dino name with Dino ID 'T88' is T-Rex.

> Finished chain.

```

Figure 16: Using Langchain to invoke the tool

```

> Entering new AgentExecutor chain...
I should use the open_weather_map tool to get the current temperature in New York.
Action: open_weather_map
Action Input: New York,US
Observation: In New York,US, the current weather is as follows:
Detailed status: clear sky
Wind speed: 7.2 m/s, direction: 60°
Humidity: 53%
Temperature:
  - Current: 17.76°C
  - High: 19.56°C
  - Low: 13.75°C
  - Feels like: 16.98°C
Rain: {}
Heat index: None
Cloud cover: 0%
Thought: The current temperature in New York is 17.76°C.
Final Answer: The current temperature in New York is 17.76°C.

> Finished chain.
Weather response: The current temperature in New York is 17.76°C.
Text message sent to +14133138995

```

Figure 17: Using Langchain to invoke the tool

Hello, I am HODAML Team 1 Chatbot!

3/19/2024



Figure 18: Start the program by inserting a date to look for data



The dinosaur transported on this day is a T-Rex with DinoID T88 being transported from Anchorage

The current temperature in Anchorage is 44.69°F

Please pay attention! The current temperature of 44.69°F in Anchorage is not safe for the T-Rex

When the temperature is outside the safe range for T-Rex, which is 57°F-68°F, the following actions are needed to keep them safe:

1. **Recording Temperatures:** Record temperatures from the TYR55 sensor located in the cargo bay of every dinosaur freighter on the Research Dinosaur Welfare Check Log.
2. **Alert Monitoring:** If the temperature variance cannot be resolved by the transporter, the Yard Centre Manager (YM) should send out an email/text to all staff on the specific alert calling tree notifying them that the temperature problem has been fixed. If the temperature cannot be controlled within the specified range, the transporter should inform the YM that the problem cannot be fixed. At this point, the YM should notify the Associate Director of Operations and Strategy and work with the transporter to make plans to move the T-Rexs to a safe area where the temperature is between 57°F and 68°F until the reefer/freighter issue is resolved.
3. **Monitoring and Documentation:** If the T-Rexs are moved to a different location, note the temperature and conditions in the new location, the actual time the T-Rexs were moved, and the time they were moved back into the freighter.

These actions are crucial to ensure the safety of T-Rex when the temperature is outside the safe range.

Generate Status Email to Management Team

Figure 19: Provide all the information starting with identifying the transported Dino and city, checking the city's temperature, determining the Dino's safety in the current temperature, and acquiring safety measures if needed



Based on the information retrieved, here is a draft email for your status report to the management:

Subject: Status Report on Current Situation

Dear Management Team,

I hope this message finds you well. I am writing to provide you with a status report on our current situation regarding the transportation and safety of our dinosaur specimens, specifically the T-Rexs and Velociraptors.

Current Status:

- The safe temperature range for T-Rexs is 57°F-68°F, and for Velociraptors, it is 40°F-55°F.
- Transporters are responsible for recording temperatures using the TYR55 sensor and ensuring that the dinosaurs are kept safe within the specified temperature ranges.
- The Associate Director of Operations and Strategy is overseeing the action plan in case temperatures in the freighters cannot be controlled.

Challenges:

- We are facing challenges in maintaining the safe temperature ranges for both T-Rexs and Velociraptors during transportation.
- If the temperature variance cannot be resolved by the transporter, it poses a risk to the safety of the dinosaurs.
- Communication and coordination are crucial in addressing temperature issues and ensuring the well-being of the specimens.

Next Steps:

- We will continue to monitor the temperatures using the TYR55 sensor and take immediate action to address any deviations from the safe ranges.
- In case the temperature cannot be controlled within the specified range, we will work with the Associate Director of Operations and Strategy to implement a plan to move the dinosaurs to a safe area until the temperature issue is resolved.
- Efforts will be made to communicate effectively with all staff members and stakeholders to ensure a coordinated response to temperature alerts and monitoring.

Your support and guidance in navigating these challenges are greatly appreciated. Please feel free to reach out if you have any questions or require further information.

Thank you for your attention to this matter.

Best regards,

[Your Name] [Your Position]

Feel free to customize the email further to align with your specific situation and add any additional details as needed.

Figure 20: Craft a status email to the Management Team

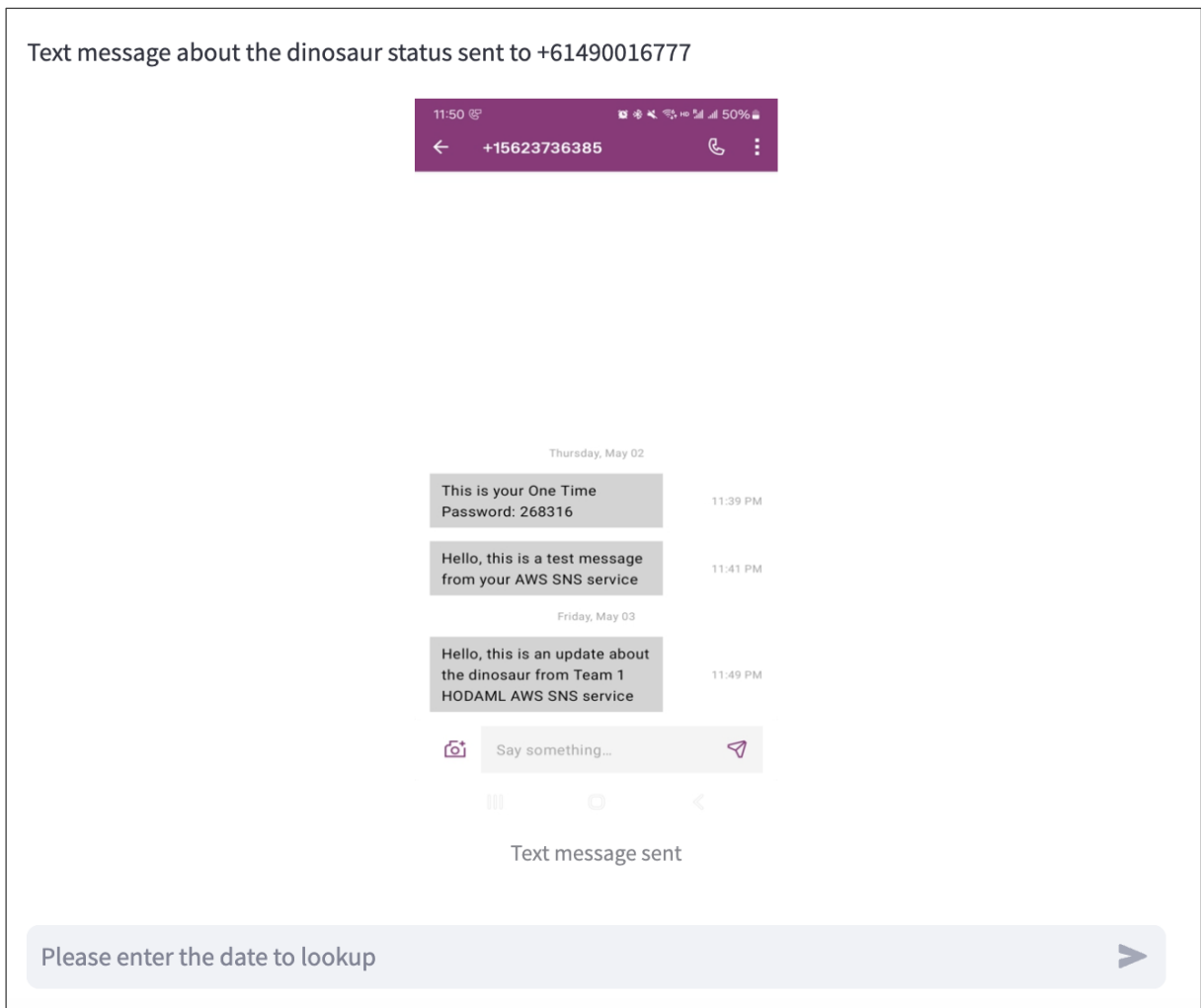


Figure 21: Send a text given a phone number and display the text message

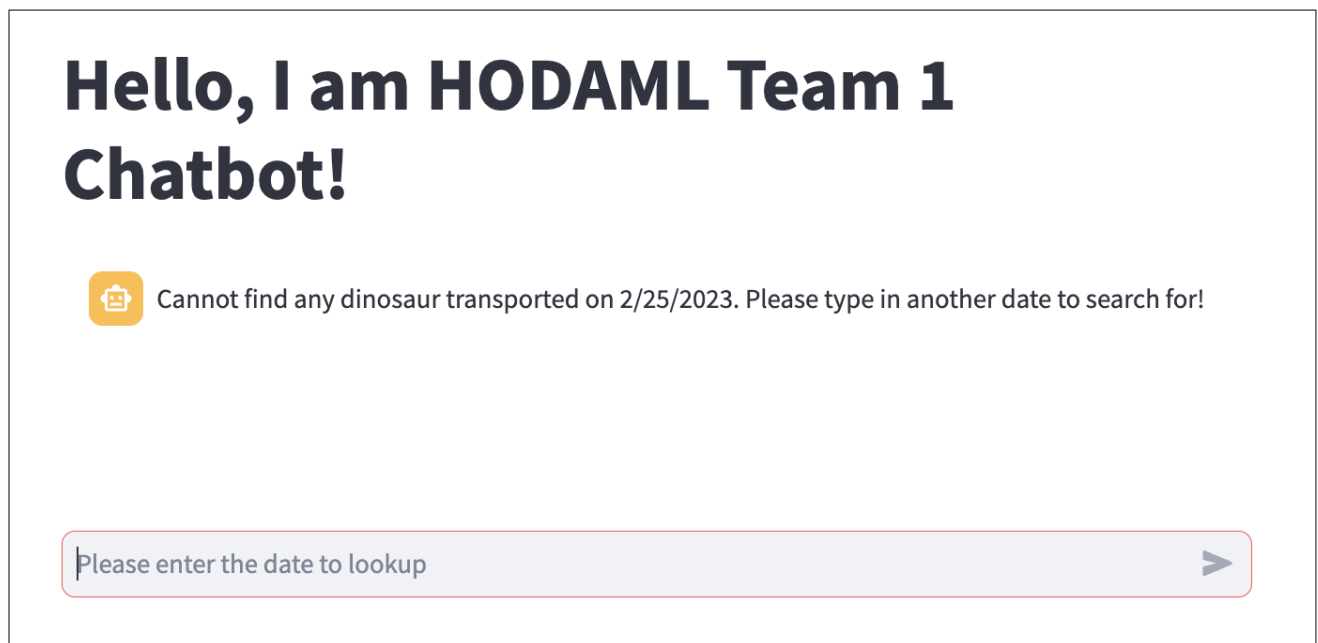


Figure 22: An edge case where a user could possibly enter a date that does not exist in the database