

# Database Project (Fall 2024)

## Homework #9 (50pts, Due date: Dec 3)

Student ID: \_\_\_\_2018310773\_\_\_\_

Student Name: \_\_\_\_임승재\_\_\_\_

**Instruction:** In this homework, we provide a jupyter notebook file (DBP\_Homework9.ipynb). You should follow the instructions in these documents. Only the provided code's '*EDIT HERE*' sections must be edited. You must *DELETE* your *GROQ\_API\_KEY* before submitting.

**Submission Guide:** Submit two files as follows:

- DBP\_Homework9\_StudentID\_StudentName.zip
- DBP\_Homework9\_StudentID\_StudentName.ipynb
- DBP\_Homework9\_StudentID\_StudentName.pdf

**1. [20pts] You should create and interact with a chatbot for 5 conversations.** The chatbot recommends a destination you may want to travel to. The conversations must include the initial and final messages provided below. You must have a total of 5 conversations, including these two messages. (Note: Ensure to use LCEL grammar.)

**Your questions at the beginning and end must be:**

**Initial message:** Hi, my name is {your\_name}. Can you recommend a place to travel in the {country\_to\_travel}? Please answer in 20 to 100 characters.

...

**Final message:** Please summarize all previous messages.

**(a) [15pts] Write code to construct a chain for the chatbot, and capture the code to display it.**

**[Answer]**

Enter your code here (captured image or string).

```

### EDIT HERE ###

prompt = ChatPromptTemplate(
    messages=[
        SystemMessagePromptTemplate.from_template("You are a nice chatbot having a
conversation with a human."),
        MessagesPlaceholder(variable_name="chat_history"),
        HumanMessagePromptTemplate.from_template("{question}")
    ]
) # Use `ChatPromptTemplate`
m = ConversationBufferMemory(
    memory_key="chat_history",
    return_messages=True
) # Use `ConversationaBufferMemory`
c = (
    {
        "question": RunnablePassthrough(),
        "chat_history": RunnableLambda(m.load_memory_variables) |
itemgetter("chat_history")
    }
    | prompt
    | llm
) # Hint: LCEL

#####

def pipeline(inputs_, c, m):
    ### EDIT HERE ###
    inputs = inputs_
    response = c.invoke(inputs)
    m.save_context(inputs, {"output": response.content})
    #####
    print(response.content)

```

(b) [5pts] Conduct a conversation with the chatbot and display the results of executing `print_all_chat_history()`.

[\[Answer\]](#)

[Enter your result here \(captured image or string\).](#)

```

#turn: 1
Hi, my name is Seungjae Rim. Can you recommend a place to travel in the USA? Please
answer in 20 to 100 characters.
#turn: 2
Nice to meet you, Seungjae! Consider visiting Grand Canyon, Arizona - breathtaking views
and stunning natural beauty.
#turn: 3

```

Actually, I've already been there. Can you recommend somewhere else?

#turn: 4

You've been to the Grand Canyon. How about Yellowstone National Park, Wyoming - geysers, hot springs, and diverse wildlife?

#turn: 5

Please tell me the representative symbols of Yellowstone National Park.

#turn: 6

Yellowstone National Park's representative symbols include:

1. The American Bison (also known as the American Buffalo)
2. The Grizzly Bear
3. The Bald Eagle
4. The Cutthroat Trout (a native fish species)
5. The Pronghorn Antelope

#turn: 7

Are there any places in the city worth sightseeing other than natural scenery?

#turn: 8

Besides natural scenery, Yellowstone has some fascinating man-made attractions.

1. The Old Faithful Inn: A historic hotel built in 1904, offering a glimpse into the park's early days.
2. The Mammoth Hot Springs Hotel: A charming hotel with a rich history, built in 1937.
3. The Grant Village Visitor Center: A hub for park information and exhibits showcasing the region's geology and wildlife.
4. The Museum of the National Park Ranger: A unique museum showcasing the history of park rangers and their work in Yellowstone.

These places offer a glimpse into the park's human history and culture.

#turn: 9

Please summarize all previous messages.

#turn: 10

Here's a summary of our conversation:

- I recommended visiting Grand Canyon, Arizona, but you mentioned you've already been there.
- I suggested visiting Yellowstone National Park, Wyoming, as an alternative.
- I listed the representative symbols of Yellowstone National Park, including the American Bison, Grizzly Bear, Bald Eagle, Cutthroat Trout, and Pronghorn Antelope.
- I mentioned some man-made attractions in the park worth sightseeing, such as the Old Faithful Inn, Mammoth Hot Springs Hotel, Grant Village Visitor Center, and the Museum of the National Park Ranger.

**2. [30pts] Implement a chatbot using multi-chain.** You need to configure the three chains for the following three questions:

**Questions:**

1. What is the hobby of {name}? ({name}'s information: {information})
2. What is a good place for the hobby {hobby}?
3. Translate the phrase '{location}' to {language}.

**Requirements:**

1. For {name}, please use your name.
2. For {information}, include more than three details about your hobbies.

**[Answer]**

Enter your code and result here. You must show your result (captured image or string).

**Code:**

**### EDIT HERE ###**

```
prompt1 = ChatPromptTemplate.from_template("What is the hobby of {name}? ({name}'s information: {information})")
```

```
chain1 = prompt1 | llm | StrOutputParser()
```

```
prompt2 = ChatPromptTemplate.from_template("What is a good place for the hobby {hobby}?")
```

```
chain2 = {"hobby": chain1} | prompt2 | llm | StrOutputParser()
```

```
prompt3 = ChatPromptTemplate.from_template("Translate the phrase '{location}' to {language}.")
```

```
chain3 = (  
    {"location": chain2, "language": itemgetter("language")}  
    | prompt3  
    | llm  
    | StrOutputParser()  
)
```

**#####**

```
def recommend_location_in_language(input_data) -> str:
```

```
    """
```

**Arguments:**

**input\_data (dict):** A dictionary containing input variables

**Returns:**

**response (str):** A string containing the final recommendation translated into the target language.

```
    """
```

**### EDIT HERE ###**

```
response = chain3.invoke(input_data)
```

**#####**

```
return response
```

```
result = recommend_location_in_language({"name": "Seungjae", "information": "1.I like to play table tennis.\n2. I like playing video games\n3. I like watching livestreaming", "language": "French"})
```

**Result:**

En fonction des loisirs de Seungjae, un bon endroit pour qu'il puisse s'engager dans ces activités pourrait être :

1. Un club de tennis de table local ou un centre communautaire où il peut jouer avec d'autres amateurs.
2. Un salon de jeux vidéo ou un équipement de jeu de son ami à la maison où il peut jouer à des jeux vidéo.
3. Des plateformes en ligne telles que Twitch ou YouTube Live où il peut regarder ses streamers préférés et s'engager avec leurs communautés.

Alternativement, un bon endroit pour Seungjae pourrait être un centre de divertissement polyvalent qui offre une combinaison de ces activités, telle que :

- Un complexe sportif avec une salle de tennis de table et des zones de jeux.
- Un café de jeux vidéo ou un centre de réalité virtuelle de jeux vidéo.
- Un centre communautaire avec un salon de jeux vidéo et une zone sportive.

Finalement, le meilleur endroit pour Seungjae dépendrait de ses préférences personnelles et des activités spécifiques qu'il aime le plus.