# Workshop 1

Please perform the following activities and write a report that illustrates you have performed and completed the activities. You need to present your report to your lecturer before the end of the class or at the beginning of week 2 workshop.

**Activity One**

Form a group of maximum 4 students. Settle down on a project topic and write a project proposal.

**Group name:** Group 8

**Group members:** Bao Linh Van, Cao Shaung Li, Yun Qiu Zhu, Vincent Thendean

**Project Proposal: Orientation Chatbot Application**

1. **Project Overview**

The purpose of this project is to develop a web-based chatbot application tailored to enhance the experience and engagement of new students during their orientation program. This chatbot will serve as an interactive digital assistant, providing immediate and accurate responses to questions about the orientation program and fostering connections among new students through suggested games and interactions. By leveraging data provided by the Student Affairs department, the chatbot will deliver an engaging, informative, and seamless experience for students, aligning with the department's goals of improved orientation outcomes and increased student satisfaction.

1. **Objectives**
   * Develop a web-based chatbot capable of answering questions about the orientation program, based on information shared by the Student Affairs department.
   * Promote engagement and interaction among new students by suggesting games and collaborative activities.
   * Create a scalable and user-friendly platform that integrates seamlessly with the existing orientation process.
2. **Features**
   * **Information Support:**
     + Respond to frequently asked questions about the orientation schedule, locations, events, and other key details.
     + Provide immediate clarification on orientation-related policies or procedures.
   * Interactive Suggestions:
     + Recommend games or interactive activities to encourage socialization among students.
     + Suggest team-building exercises or group discussions.
   * Feedback Collection:
     + Collect feedback on the chatbot’s performance to enhance future iterations.
     + Use feedback to improve the orientation program.
3. **Technology Stack**

**Frontend**

Streamlit: A Python library for creating interactive web applications with minimal effort.

Flask: A lightweight WSGI web application framework for serving APIs and handling backend logic.

**Backend**

Programming Language:

Python: A versatile choice for backend development, utilized throughout the stack.

Streamlit: Serves as both the frontend and backend, simplifying application architecture and enabling quick deployment.

**AI/NLP**

Chatbot Service:

Claude API: For natural language processing and generating responses.

Gemini API: For AI embeddings, enhancing contextual understanding.

**Translation:**

Google Translate Library: Imported into Python to translate responses to different languages using the googletrans method.

**Vectorstore:**

FAISS: For efficient similarity search and clustering of embeddings.

LangChain: Facilitates the integration of LLMs (like Gemini) with various data sources and manages conversational flows effectively.

**Prompt Engineering:**

Model Training: Develop effective prompts to enhance the model's understanding and response capabilities.

Query Optimization: Dynamically adjust prompts based on conversation context to improve relevance and accuracy.

**Database**

MongoDB: A NoSQL database used to store orientation information and user feedback.

**Version Control**

Git: For version control, enabling tracking of changes and collaboration.

GitHub: For repository hosting and collaboration, facilitating team workflows and open-source contributions.

**Data and Privacy**

The chatbot will be trained on data provided by the Student Affairs department, including orientation schedules, event descriptions, and FAQs. All data will be handled with strict adherence to privacy regulations, ensuring that student information and interactions remain confidential.

**Activity Two**

Project management.

GITHUB link: <https://github.com/Blynxxxx/CHATBOT_PROJECT>

Bao Linh Van: Tester

Cao Shaung Li: Developer

Yun Qiu Zhu: UX/UI Designer

Vincent Thendean: Developer

# Workshop 2

Please perform the following activities and write a report that illustrates that your team has performed and completed the activities. You need to present your report to your lecturer before the end of the class.

**Activity One**

Interview your target users for feedback on their requirements for your software project.

**Interview**

1) What do you want?

A chatbot that answers student's questions on Orientation

2) What should the chatbot be like?

It has to reply in a natural language.

3) What kind of answers should the chatbot know?

It will summarize information from the orientation pdf

4) How should the chatbot be accessed?

In the orientation site at the bottom

5) How different is information from one orientation to another?

Structure and events remain the same. Only time and place change.

**Activity Two**

Write the requirements document for your project and provide estimation of each user story.

**User Stories**

Story 1:

Title: Gather Orientation Information for Users

Estimate: 3 days

Description: Developers will collect and structure essential orientation information to enable the chatbot to provide accurate and helpful responses to users.

Story 2:

Title: Interact UI to chat with the chatbot

Estimate: 8 days

Description: Users will have a user-friendly interface to interact with the chatbot for seamless communication.

Story 3:

Title: Training the Chatbot on Orientation Information

Estimate: 13 days

Description: Developers will train the chatbot with comprehensive orientation information, enabling it to provide accurate and contextually relevant answers to user inquiries.

Story 4:

Title: Ask question freely

Estimate: 13 days

Description: Users can freely ask any questions, with the chatbot offering relevant orientation answers, including general information when needed.

Story 5:

Title: Multi-Languages

Estimate: 8 days

Description: Users will be able to interact with the chatbot in various languages, enhancing accessibility.

Story 6:

Title: Temporary History

Estimate: 5 days

Description: Users can view their past messages within the same conversation, allowing them to reference previous questions and maintain context throughout the interaction.

Story 7:

Title: Question Options

Estimate: 8 days

Description: Users will see predefined question options at the start, allowing them to choose and receive quick answers.

Story 8:

Title: Give Feedback

Estimate: 5 days

Description: Users can provide feedback or report inaccuracies, helping to improve the chatbot's performance and responses.

# 

# Workshop 3

Please perform the following activities and write a report that illustrates that

your team has performed and completed the activities. You need to present your

report to your lecturer before the end of the class.

**Activity One**

Give priority to each user story and divide your user stories into three iterations.

**Story 1:**

Title: Gather Orientation Information for Users

Estimate: 3 days

Priority: 10

Description: Developers will collect and structure essential orientation information to enable the chatbot to provide accurate and helpful responses to users.

**Story 2:**

Title: Interact UI to chat with the chatbot

Estimate: 8 days

Priority: 10

Description: Users will have a user-friendly interface to interact with the chatbot for seamless communication.

**Story 3:**

Title: Training the Chatbot on Orientation Information

Estimate: 13 days

Priority: 10  
Description: Developers will train the chatbot with comprehensive orientation information, enabling it to provide accurate and contextually relevant answers to user inquiries.

**Story 4:**

Title: Ask question freely

Estimate: 13 days

Priority: 20

Description: Users can freely ask any questions, with the chatbot offering relevant orientation answers, including general information when needed.

**Story 5:**

Title: Multi-Languages

Estimate: 8 days

Priority: 30

Description: Users will be able to interact with the chatbot in various languages, enhancing accessibility.

**Story 6:**

Title: Temporary History

Estimate: 8 days

Priority: 30

Description: Users can view their past messages within the same conversation, allowing them to reference previous questions and maintain context throughout the interaction.

**Story 7:**

Title: Question Options

Estimate: 8 days

Priority: 40

Description: Users will see predefined question options at the start, allowing them to choose and receive quick answers.

**Story 8:**

Title: Give Feedback

Estimate: 5 days

Priority: 50

Description: Users can provide feedback or report inaccuracies, helping to improve the chatbot's performance and responses.

**Committed days: 10 days (2 week 1 iteration)**

**Default velocity: 0.7**

**Members: 4**

**Working days: 10 \* 0.7 \* 4 = 28 days**

**Iteration 1 (On Week 5)**

**Story 1:**

Title: Gather Orientation Information for Users

Estimate: 3 days

Priority: 10

Description: Developers will collect and structure essential orientation information to enable the chatbot to provide accurate and helpful responses to users.

**Story 2:**

Title: Interact UI to chat with the chatbot

Estimate: 8 days

Priority: 10

Description: Users will have a user-friendly interface to interact with the chatbot for seamless communication.

**Story 3:**

Title: Training the Chatbot on Orientation Information

Estimate: 13 days

Priority: 10  
Description: Developers will train the chatbot with comprehensive orientation information, enabling it to provide accurate and contextually relevant answers to user inquiries.

**Iteration 2 (On Week 7)**

**Story 4:**

Title: Ask question freely

Estimate: 13 days

Priority: 20

Description: Users can freely ask any questions, with the chatbot offering relevant orientation answers, including general information when needed.

**Story 5:**

Title: Multi-Languages

Estimate: 8 days

Priority: 30

Description: Users will be able to interact with the chatbot in various languages, enhancing accessibility.

**Story 6:**

Title: Temporary History

Estimate: 5 days

Priority: 30

Description: Users can view their past messages within the same conversation, allowing them to reference previous questions and maintain context throughout the interaction.

**Story 7:**

Title: Question Options

Estimate: 8 days

Priority: 40

Description: Users will see predefined question options at the start, allowing them to choose and receive quick answers.

**Story 8:**

Title: Give Feedback

Estimate: 5 days

Priority: 50

Description: Users can provide feedback or report inaccuracies, helping to improve the chatbot's performance and responses.

**Activity Two**

Implement at least one user story.

**Story 1:**

Title: Gather Orientation Information for Users

Estimate: 3 days

Priority: 10

Description: Developers will collect and structure essential orientation information to enable the chatbot to provide accurate and helpful responses to users.

Google doc link: <https://docs.google.com/document/d/1FX5MxzUNBqr0pUGOT6BbxRi5xiWPnjnWcFvFNCD28EQ/edit?usp=sharing>

# Workshop 4

Please perform the following activities and present your result to your lecturer before the end of the class.

**Activity One**

**Break your user stories into tasks and provide estimation for each task.**

**Story 1:**

Title: Gather Orientation Information for Users

Estimate: 3 days

Priority: 10

Description: Developers will collect and structure essential orientation information to enable the chatbot to provide accurate and helpful responses to users.

**Task:**

1. Collect Orientation Infomation on JCU Official Website (2 days)
2. Reformat all information to feed into LLM (1 day)

**Story 2:**

Title: Interact UI to chat with the chatbot

Estimate: 8 days

Priority: 10

Description: Users will have a user-friendly interface to interact with the chatbot for seamless communication.

**Task:**

1. Create a chatting space for user to type input and see response using Streamlit (1 day)
2. Research on AI API for chatbot OpenAI, GeminiAI, Anthropic Claude AI (5 days)
3. Test and Improve chatbot UI (2 days)

**Story 3:**

Title: Training the Chatbot on Orientation Information

Estimate: 13 days

Priority: 10  
Description: Developers will train the chatbot with comprehensive orientation information, enabling it to provide accurate and contextually relevant answers to user inquiries.

**Task:**

1. Implement the formatted document from local database to the selected LLM (1 day)
2. Extract text from document (1 day)
3. Divide text into chunks (1 day)
4. Convert vector stores and AI embeddings (3 days)
5. Migrate document to Mongo database (3 days)
6. Test and adjust chatbot (5 days)

**Story 4:**

Title: Ask question freely

Estimate: 13 days

Priority: 20

Description: Users can freely ask any questions, with the chatbot offering relevant orientation answers, including general information when needed.

**Task:**

1. Amend the code to allow general AI to give respond if user’s input is not related to the document (3 days)
2. Create prompt (5 days)
3. Test and adjust the chatbot (5 days)

**Story 5:**

Title: Multi-Languages

Estimate: 5 days

Priority: 30

Description: Users will be able to interact with the chatbot in both English and Chinese, enhancing accessibility.

**Task:**

1. Add multi-language option to chatbot language button (1 day)
2. Add google translator to translate message (2 days)
3. Test and adjust the chatbot (2 days)

**Story 6:**

Title: Temporary History

Estimate: 8 days

Priority: 30

Description: Users can view their past messages within the same conversation, allowing them to reference previous questions and maintain context throughout the interaction.

Task:

1. Create temporary history which store users’ past messages (3 days)
2. Implement function where users can refer to past messages for consistent conversation (3 days)
3. Test the chatbot (2 days)

**Story 7:**

Title: Quick Questions

Estimate: 8 days

Priority: 40

Description: Users will see predefined questions at the start, allowing them to choose and receive quick answers.

Task:

1. Add four quick questions to the chatbot (3 days)
2. Adjust the location of the quick question (2 days)
3. Test and modfiy the quick question in the chatbot (3 days)

**Story 8:**

Title: Give Feedback

Estimate: 5 days

Priority: 50

Description: Users can provide feedback or report inaccuracies, helping to improve the chatbot's performance and responses.

Task:

1. Add a place for user to input feedback and a button for submission (1 day)
2. Connect feedback session with Mongo database to store users’ feedback (2 days)
3. Test feedback functionality (2 days)

**Activity Two**

**Develop class diagrams of your project.**

**Story 4:**

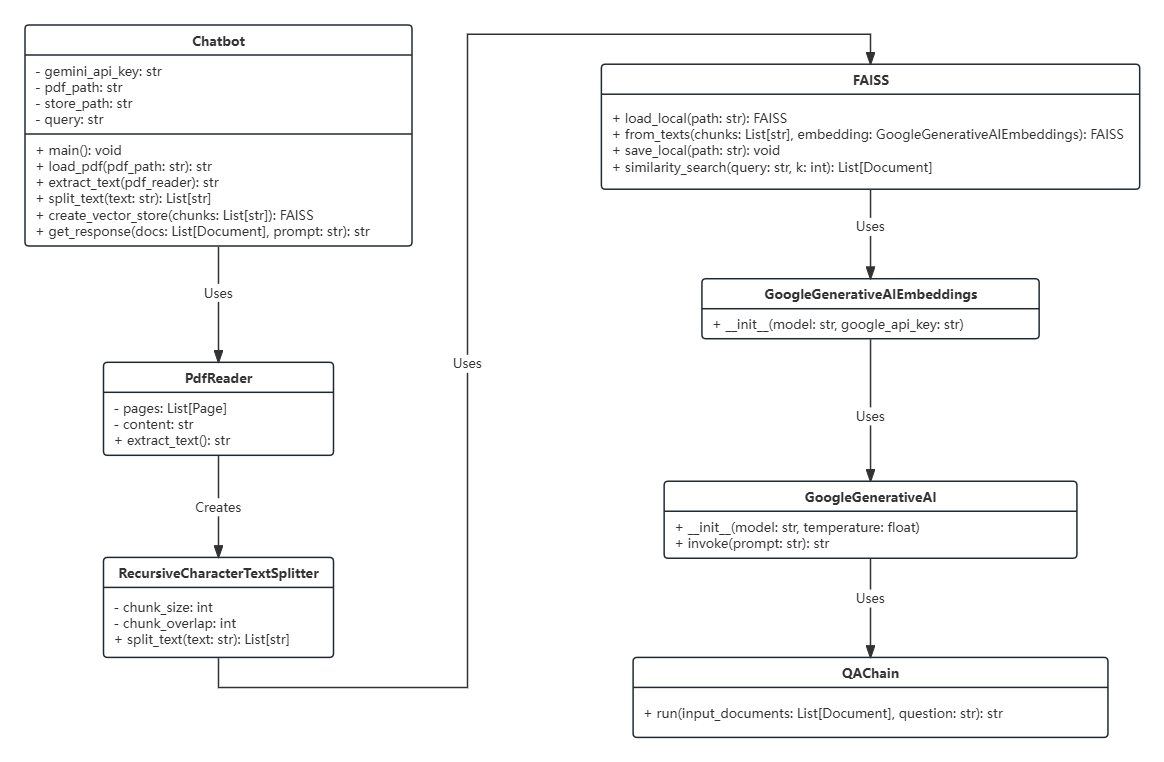
Title: Training the Chatbot on Orientation Information

Estimate: 13 days

Priority: 10  
Description: Developers will train the chatbot with comprehensive orientation information, enabling it to provide accurate and contextually relevant answers to user inquiries.

**Task:**

1. Implement the formatted document from local database to the selected LLM (½ day)
2. Extract text from document (½ day)
3. Divide text into chunks (½ day)
4. Convert vector stores and AI embeddings (3 days)
5. Test if the vector store is created correctly (½ day)
6. Adjust tokens, temperature of chatbot, and text overlaps (½ day)
7. Migrate document to Mongo database (3 days)
8. Add appropriate prompt (½ days)
9. Test if chatbot gives accurate response (1½ day)
10. Adjust the prompt accordingly (1½ days)



**Activity Three**

**Develop at least one sequence diagram to illustrate the key features/operations of your project.**

Title : Obtain precise information

Estimate: 13 days

Priority: 10

Description: A user will be able to obtain relevant information about orientation from the chatbot

TASKS:

**Define chatbot response structure -** 1 day

**Extract orientation schedule from PDF -** 2 days

**Identify the API -** 5 days

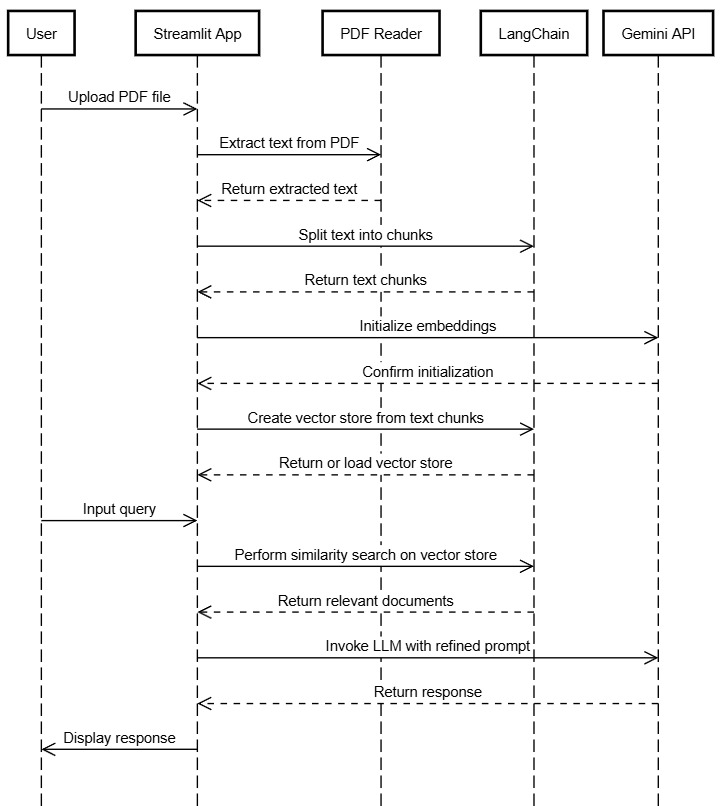
**Use QA by Langchain -** 5 days

**Story 4:**

Title: Training the Chatbot on Orientation Information

Estimate: 13 days

Priority: 10  
Description: Developers will train the chatbot with comprehensive orientation information, enabling it to provide accurate and contextually relevant answers to user inquiries.

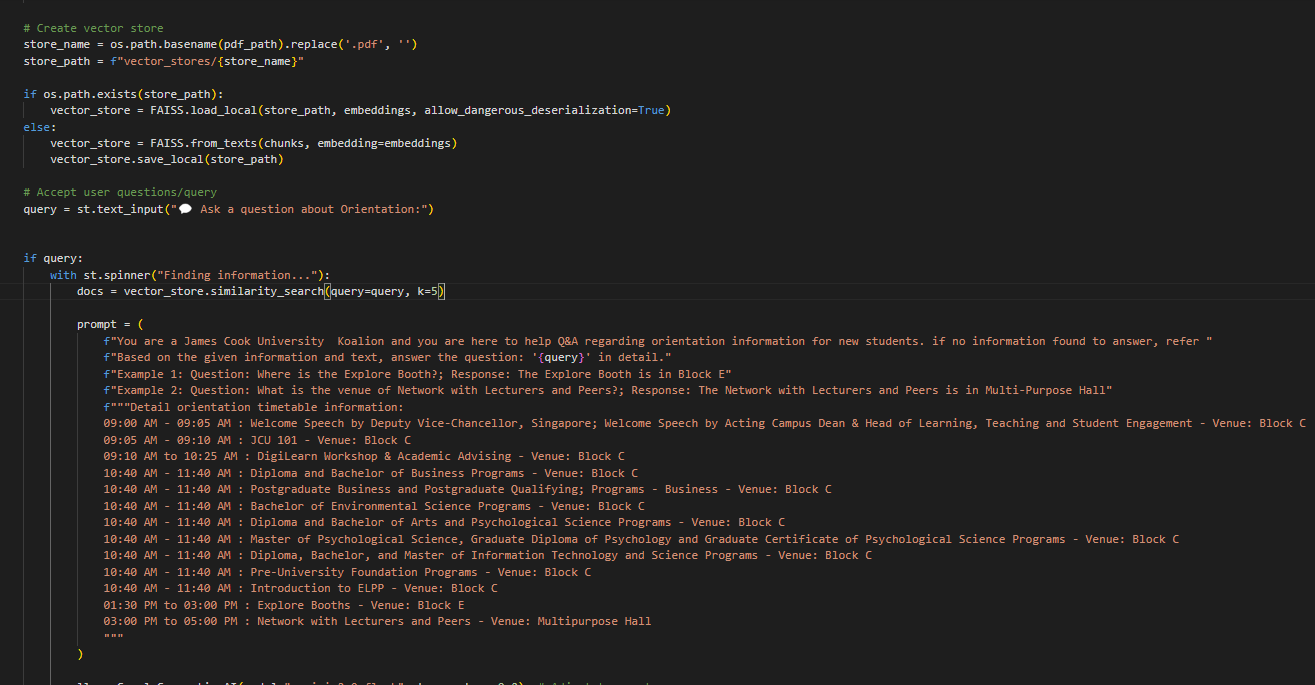
****

**Activity Four**

Write the code to implement your sequence diagrams.

Screenshots of code snippets below:





# 

# Workshop 5

Please perform the following activities and write a report that illustrates you have performed and completed the activities. You need to present your report to your lecturer before the end of the class.

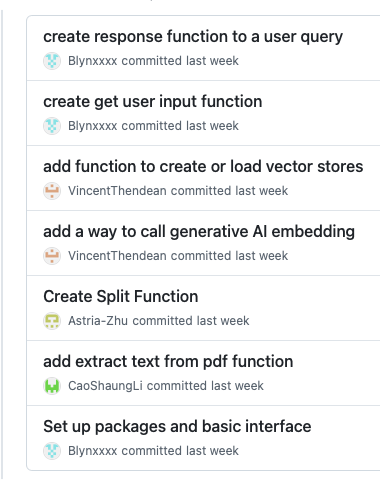
**Activity One**

Check your classes to see whether they satisfy SRP and DRY and list down what you have found.

**Activity Two**

Copy screenshots of your github changes and/or commits.

<https://github.com/Blynxxxx/CHATBOT_PROJECT/commits/main/>



**Activity Three**

Summarize project progress and state any issues you are facing.

**Project progress:**

**Define Chatbot Response Structure** :

* Successfully established the response format for the chatbot, ensuring clarity and user-friendliness in the information provided.

**Extract Orientation Schedule from PDF**:

* Implemented a function to extract text from the specified PDF file containing the orientation schedule.
* Handled errors effectively during the extraction process to enhance robustness.

**Identify the API**:

* Researched and selected the Google Gemini API for integration, ensuring it meets the project's requirements for AI capabilities.

**Use QA by LangChain**:

* Integrated LangChain’s question-answering functionality into the chatbot.
* Configured the system to process user queries and generate responses based on the extracted and indexed information from the orientation schedule.

**Issue we are currently facing:**

* Prompt still needs further improvement for accuracy
* Chatbot still requires modification to answer general questions which are not from the pdf files

# Workshop 6

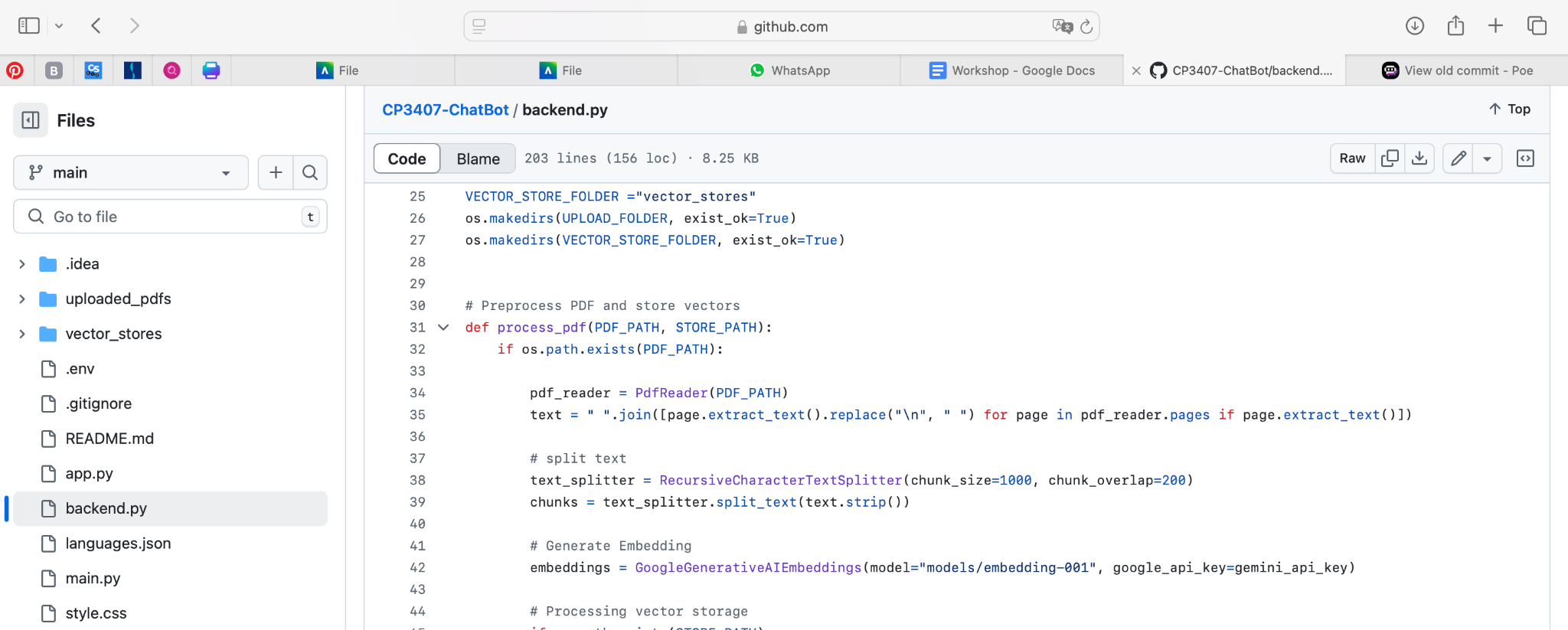
Please perform the following activity and write a report that illustrates you have performed and completed the activity.

You need to present your report to your lecturer before the end of the class.

**Activity One**

Each team member refactors some other member’s code and commits to Github.

Record any conflict that you have encountered when refactoring other team members’ code.

**Refactor app.py into Two Distinct Files: backend.py and main.py then commit to GitHub.**

1. **Create backend.py**:
   * This file will handle all backend processes, including:
     + **PDF Processing**: Implement functionality to convert PDF documents into manageable chunks and vectors for further analysis.
     + **Gemini API Integration**: Set up communication with the Gemini API to retrieve data necessary for the chatbot's operations.
     + **Data Handling**: Include any necessary functions or classes to manage the data flow between the PDF processing and the chatbot.
2. **Create main.py**:
   * This file will focus on the user interface aspects of the application, including:
     + **Responsive Design**: Develop a user-friendly interface that adapts to various screen sizes, ensuring a seamless experience on both desktop and mobile devices.
     + **User Input**: Implement text input fields where users can enter their questions or commands for the chatbot.
     + **Display Responses**: Set up a section to display responses from the chatbot, ensuring that interactions are clear and accessible.
     + **Event Handling**: Include event listeners to manage user interactions, such as submitting questions and receiving answers.

# Workshop 7

Please perform the following activities and write a report that illustrates you have performed and completed the activities. You need to present your report to your lecturer before the end of the class.

**Activity One**

Select at least five user stories from your project. Write at least 3 test cases for each of your chosen user story (reference to page 242 example).

Selected User story:

**Story 3:**

Title: Training the Chatbot on Orientation Information

Estimate: 13 days

Priority: 10  
Description: Developers will train the chatbot with comprehensive orientation information, enabling it to provide accurate and contextually relevant answers to user inquiries.

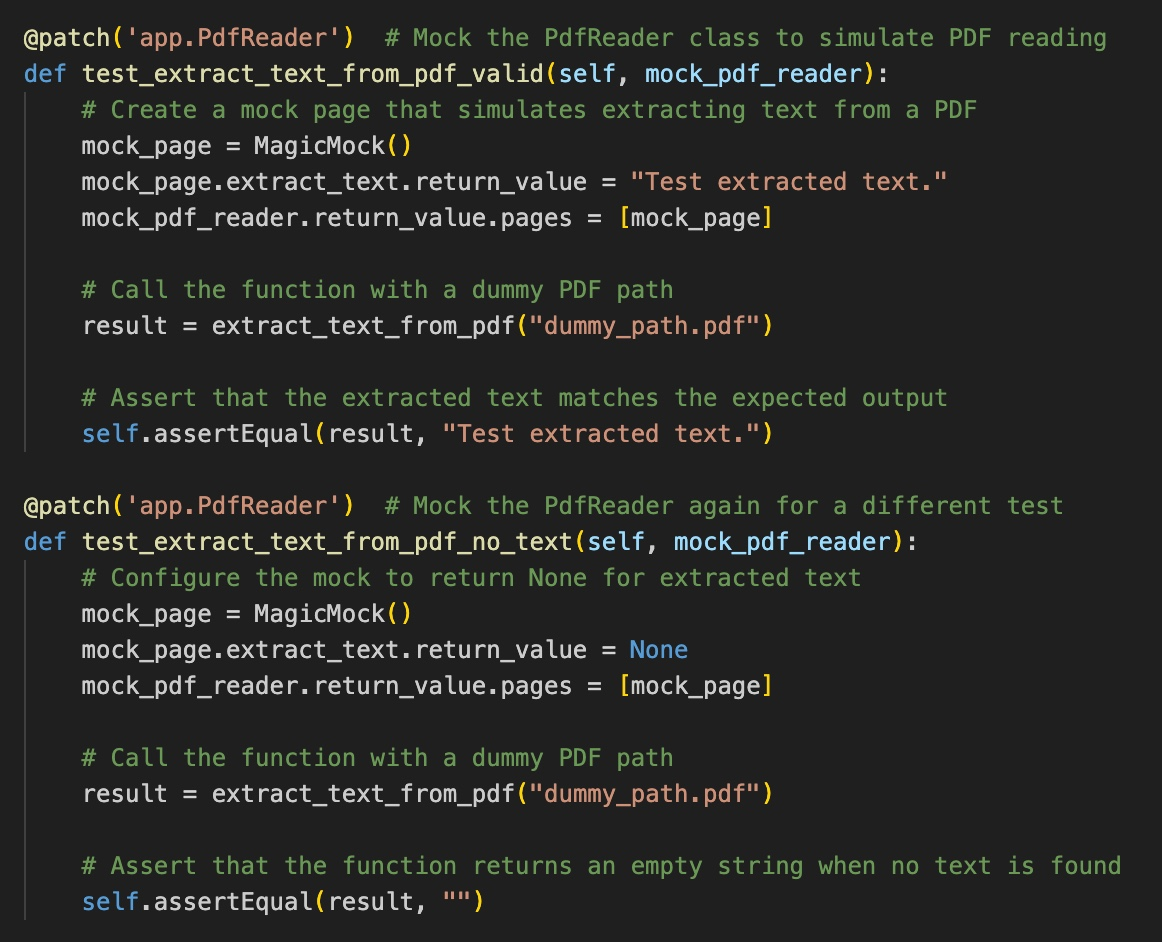
**Task:**

1. Implement the formatted document from local database to the selected LLM (1 day)
2. Extract text from document (1 day)
3. Divide text into chunks (1 day)
4. Convert vector stores and AI embeddings (3 days)
5. Migrate document to Mongo database (3 days)
6. Test and adjust chatbot (5 days)

**Test to extract information from pdf**:

Create a function to extract text from a PDF

Assert that the extracted content is the same with expected content

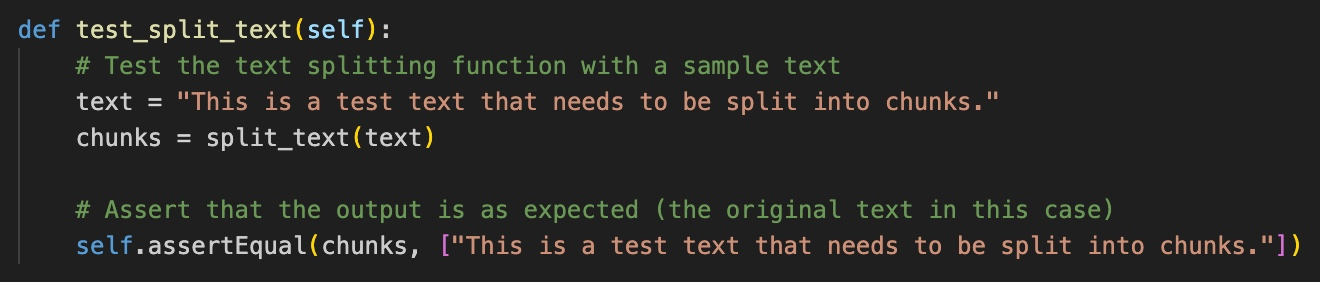


**Test to split text into chunks:**

Create a text variable

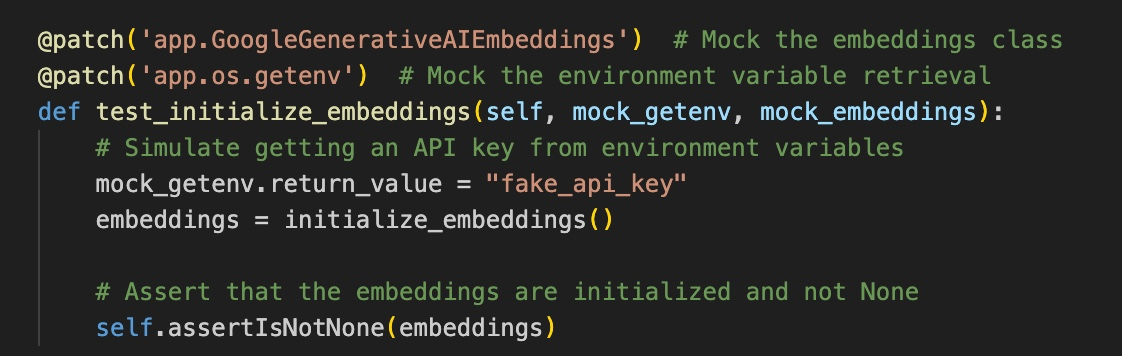
Split the text into chunks

Assert if the chunks result equals the expected result.



**Test to convert chunks to vectors using embeddings:**

Check that the embeddings variable is not empty

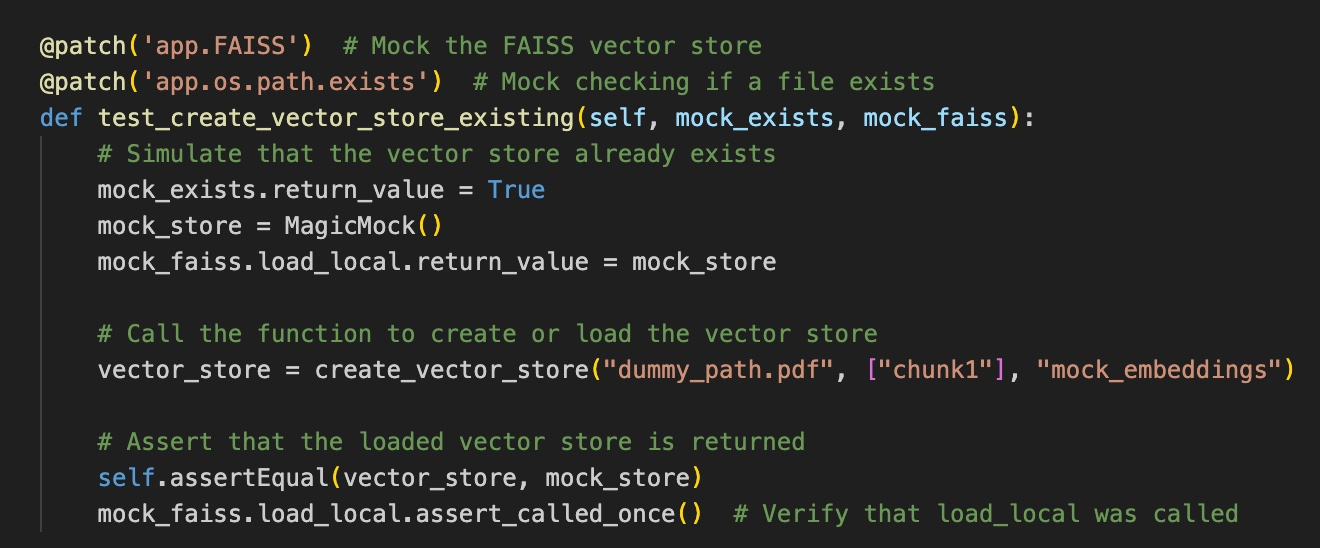


**Test to create vector store 1:**

Simulate that a vector store already exists

Create a vector store by calling create\_vector\_store function and input required parameters

Assert to verify if the loaded vector store is returned

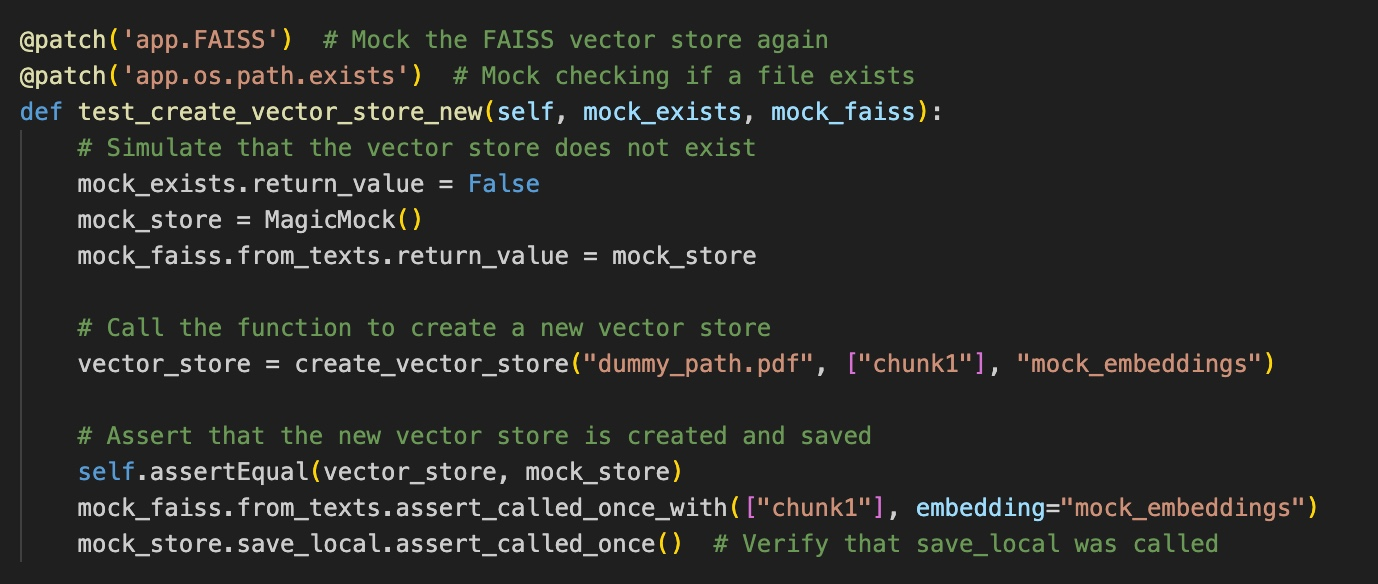


**Test to create vector store 2:**

Simulate that the vector store does not exist

Create vector store by calling create\_vector\_store using the required arguments

Assert to verify if the new vector store is created

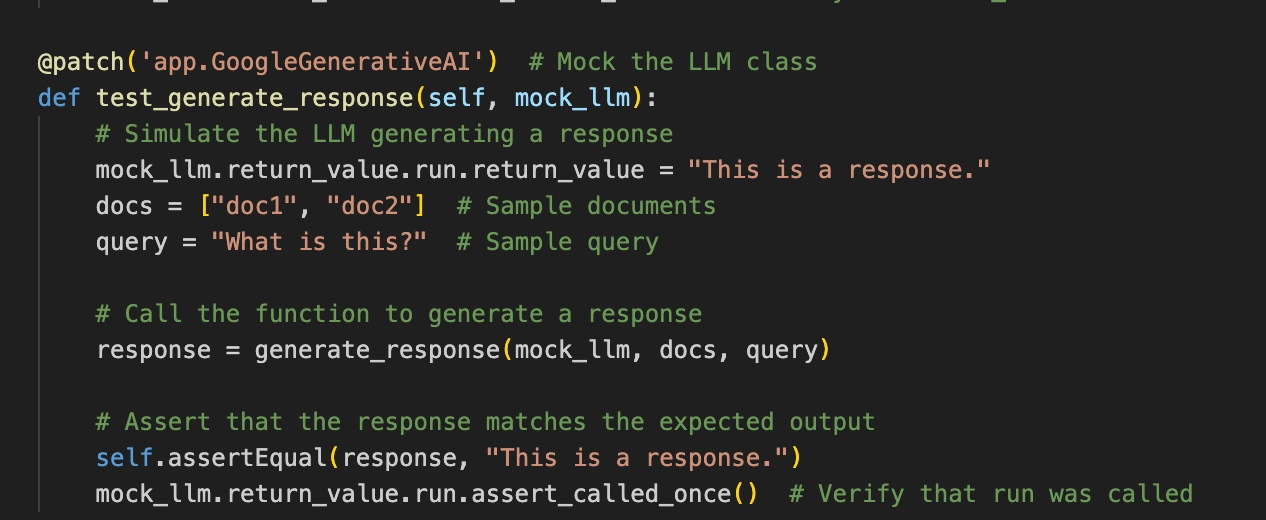


**Test the basic response function**:

Simulate the LLM generating a response

Call the function generate\_response

Assert if response matches the expected output



Activity Two

Implement at least 15 automated tests for your project.

See figures above

# Workshop 8

Please perform the following activities and write a report that illustrates you have performed and completed the activities. You need to present your report to your lecturer before the end of the class.

**Activity One**

Doing research on Strategy patterns. Identify at least one place in your project that has implemented the Strategy pattern.

The **Strategy Pattern** is a behavioral design pattern that enables selecting an algorithm's behavior at runtime. It defines a family of algorithms, encapsulates each one, and makes them interchangeable. This allows clients to vary their behavior without modifying the code that uses them.

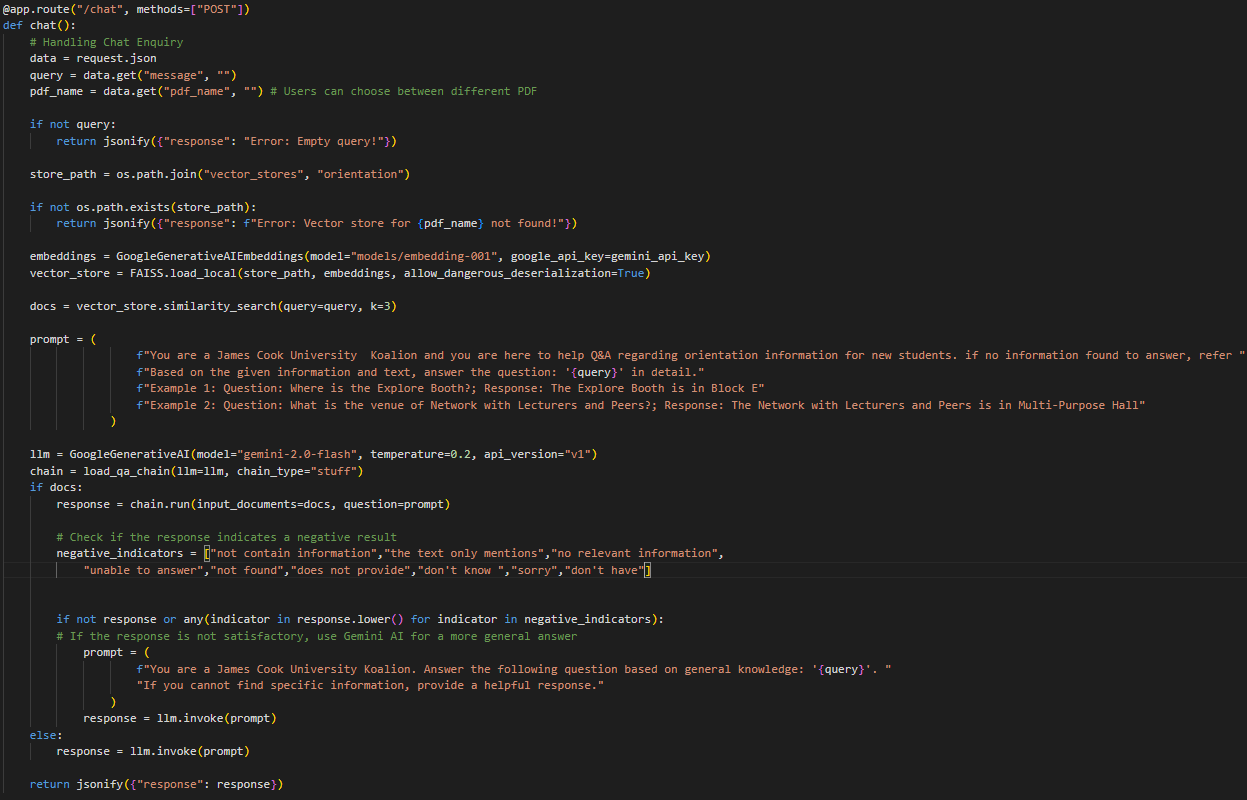
**Key Components:**

1. **Context**: Maintains a reference to a Strategy object and can switch strategies.
2. **Strategy Interface**: An interface common to all concrete strategies.
3. **Concrete Strategies**: Implementations of the Strategy interface, each providing a different algorithm.

**Benefits:**

* Promotes flexibility and reusability by decoupling the algorithm from the client code.
* Simplifies code maintenance and readability.

<https://en.wikipedia.org/wiki/Strategy_pattern>

****

The Strategy pattern is identified in the chat() function. The generate\_response function uses different strategies to generate a response based on conditional statements. If relevant documents are found, the response is generated using the loaded QA chain with the provided context. If no relevant documents are found, a fallback strategy is employed where a more general prompt is sent to the LLM, seeking an answer based on general knowledge.

**Activity Two**

Doing research on Mock Object Framework. Trying to implement mock objects to your project testing.

A **Mock Object Framework** is a testing tool that allows developers to create mock objects for unit testing. These mocks simulate the behavior of real objects in a controlled way, enabling developers to isolate components of their code and test them independently.

**Key Features:**

1. **Isolation**: Mocks allow you to test a specific unit of code without relying on external systems (like databases or APIs).
2. **Behavior Verification**: You can verify that certain methods are called with expected parameters, helping ensure that your code interacts correctly with dependencies.
3. **Simplified Testing**: Mocks can return predefined responses, making it easier to simulate various scenarios (e.g., success, failure).

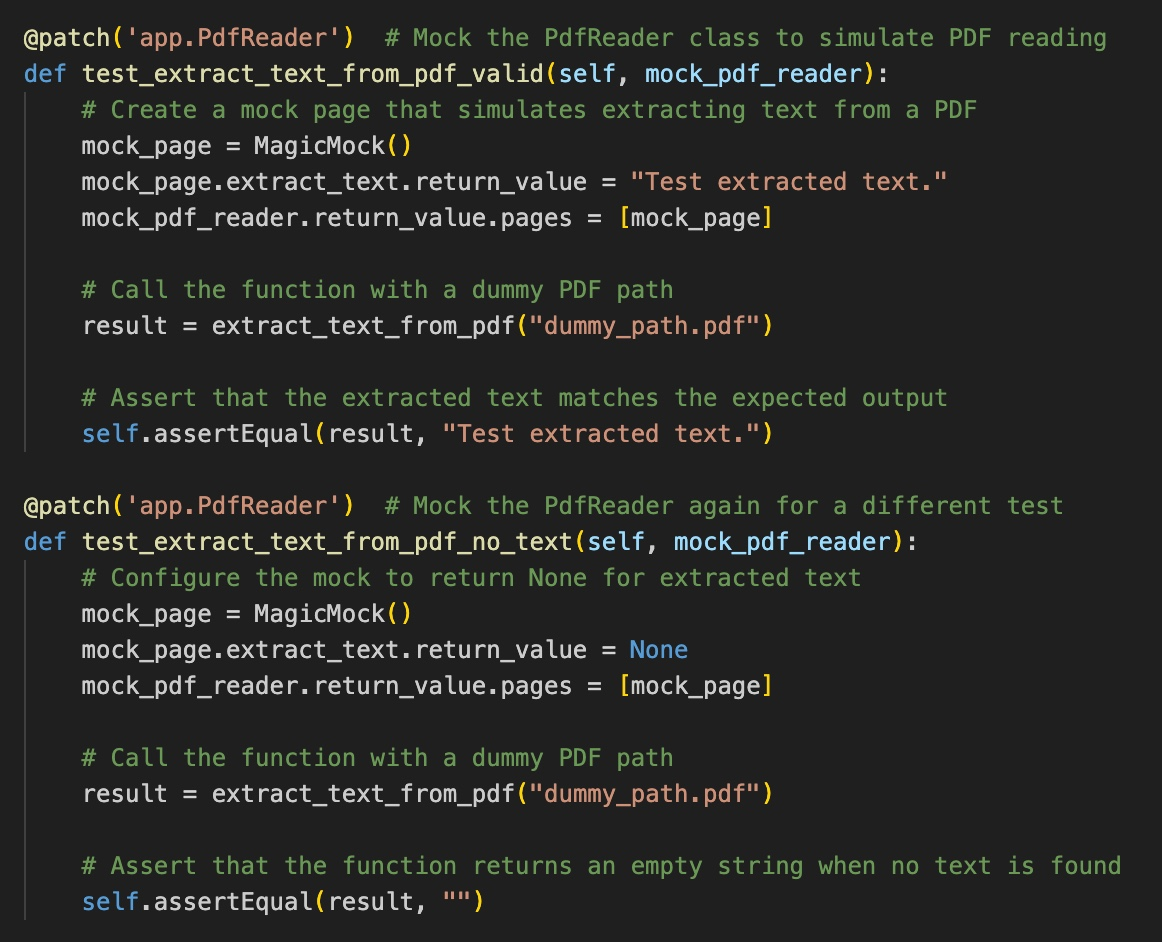
**Popular Mock Frameworks:**

* **Mockito** (Java)
* **unittest.mock** (Python)
* **Jest** (JavaScript)
* **Mocha** (JavaScript)

<https://en.wikipedia.org/wiki/Mock_object>

In our project, we mock the PdfReader class to simulate the PDF reading and create a mock page to test the function: extract\_text\_from\_pdf in two cases:

* If the extracted text match the expected output.
* What is the return when no text is found.



# Workshop 9

Please perform the following activities and write a report that illustrates you have performed and completed the activities. You need to present your report to your lecturer before the end of the class.

**Activity One**

**Write a system testing policy for your project.**

1. Purpose

This policy establishes the framework for conducting system testing on the CHAT BOT project, ensuring it operates correctly and provides accurate responses in both Chinese and English.

2. Scope

The policy covers:

* Functionality of chatbot responses.
* Usability testing in Chinese and English.
* UI and Response time of Chatbot.

3. Objectives

* Verify accurate responses to user inputs.
* Ensure effective use of general AI for unrelated queries.
* Assess user experience and interaction flow.

4. Responsibilities

* **Testing Team**: Execute tests and document findings.
* **Developers**: Fix identified issues based on priority.
* **Project Manager**: Oversee testing processes and compliance.

5. Reporting and Documentation: Findings must be recorded in a bug-tracking system.

6. Review and Updates

This policy will be reviewed regularly to reflect changes in project requirements or testing practices.

**Activity Two**

**Write your detailed system testing plan.**

1. Introduction

The purpose of this system testing plan is to detail the approach and methodology for testing the CHAT BOT project, ensuring that it meets the functional and performance requirements in both Chinese and English.

2. Scope

The testing will cover:

* Functionality of responses to user inputs.
* Usability and user experience in both Chinese and English.
* The User interface and Response time of Chatbot.

3. Objectives

* Verify that the chatbot accurately responds to user inputs in both Chinese and English.
* Ensure appropriate use of general AI for questions not related to the database.
* Assess the overall user experience and interaction flow.

4. Testing Approach

4.1 Manual Testing

Testers will manually input various queries in both languages: Chinese and English

Responses will be assessed for:

* **Accuracy**: Correctness of the information provided.
* **Relevance**: Appropriateness of the response to the query.
* **Coherence**: Logical flow and clarity of responses.
* **Response Time**: Measure the time taken for the chatbot to respond.
* **Display Quality**: Evaluate the user interface and how information is displayed during interaction.

4.2 Test Case Development

Test cases will include:

* Common inquiries related to the database.
* Edge cases with unrelated questions to evaluate general AI responses.
* User experience scenarios to assess conversational flow.

4.3 Performance Testing

* Evaluate the chatbot's response time.

5. Test Environment

* **Hardware**: Standard user devices (PCs).
* **Database**: MongoDB for storing user queries and responses.
* **AI Support**: Internet connection for accessing general AI capabilities.

6. Reporting and Documentation

All findings will be documented in a bug-tracking system, including:

* Description of the issue
* Steps to reproduce
* Expected vs. actual results
* Assigned priority level

7. Priority Assignment

Bugs will be prioritized as follows:

* High Priority: Issues that prevent functionality.
* Medium Priority: Usability issues that do not halt functionality.
* Low Priority: Minor issues with minimal impact on user experience**.**

8. Schedule

* Testing will be conducted in cycles aligned with project milestones, focusing initially on Milestone 1.
* Regular status reviews will be held to assess testing progress and address critical issues.

9. Roles and Responsibilities

* Testing Team: Execute tests and document findings.
* Developers: Address and fix identified issues based on priority.
* Project Manager: Oversee the testing process and ensure adherence to this plan.

10. Review and Updates

* This testing plan will be reviewed and updated regularly to adapt to changes in project requirements or testing methods.

# System Testing Result Documentation

**1. Introduction**

The purpose of this system testing plan is to detail the approach and methodology for testing the CHAT BOT project, ensuring that it meets the functional and performance requirements in both Chinese and English.

**2. Scope**

The testing will cover:

* Functionality of responses to user inputs.
* Usability and user experience in both Chinese and English.
* The User interface and Response time of Chatbot.

**3. Test Summary**

| Test Type | Total Tests | Passed | Failed | Not Executed |
| --- | --- | --- | --- | --- |
| Functional Testing | 20 | 20 | 0 | 0 |
| Performance Testing | 10 | 6 | 3 | 0 |
| Usability Testing | 10 | 10 | 0 | 0 |

**4. Detailed Test Results**

**4.1 Functional Testing Results**

| **Test Case ID** | **Description** | **Expected result** | **Actual result** | **Status** | **Comments** |
| --- | --- | --- | --- | --- | --- |
| TC001  (7 tests) | Construct queries using the information stored in the database. | Accurate response aligned with database | Correct response consistent with the database | Pass |  |
| TC002  (7 tests) | Construct queries using information related to JCU that is not in the database. | Respond to the answer based on information available on the internet about JCU. | Provide a response based on the information found online about JCU. | Pass |  |
| TC003  (6 tests) | Construct general queries not related to JCU. | Respond to the answer based on information available on the internet. | Provide a response based on the information found online. | Pass |  |

**4.2 Performance Testing Results**

| **Test Case ID** | **Description** | **Expected result** | **Actual result** | **Status** | **Comments** |
| --- | --- | --- | --- | --- | --- |
| TC004  (1 test) | Load Handling | Stable under load | No crash, stable | Pass |  |
| TC005  (9 tests) | Response Time | < 10 seconds | sometimes longer than 10 seconds | Fail |  |

**4.3 Usability Testing Results**

| **Test Case ID** | **Description** | **Expected result** | **Actual result** | **Status** | **Comments** |
| --- | --- | --- | --- | --- | --- |
| TC006  (5 Test) | User Interface Layout | Intuitive and easy to navigate, display history and suggest questions | User-friendly layout; the temporary history and suggested questions are displayed effectively. | Pass |  |
| TC007  (5 Tests) | Interaction Flow | Smooth and logical | The responses are ordered chronologically. | Pass |  |

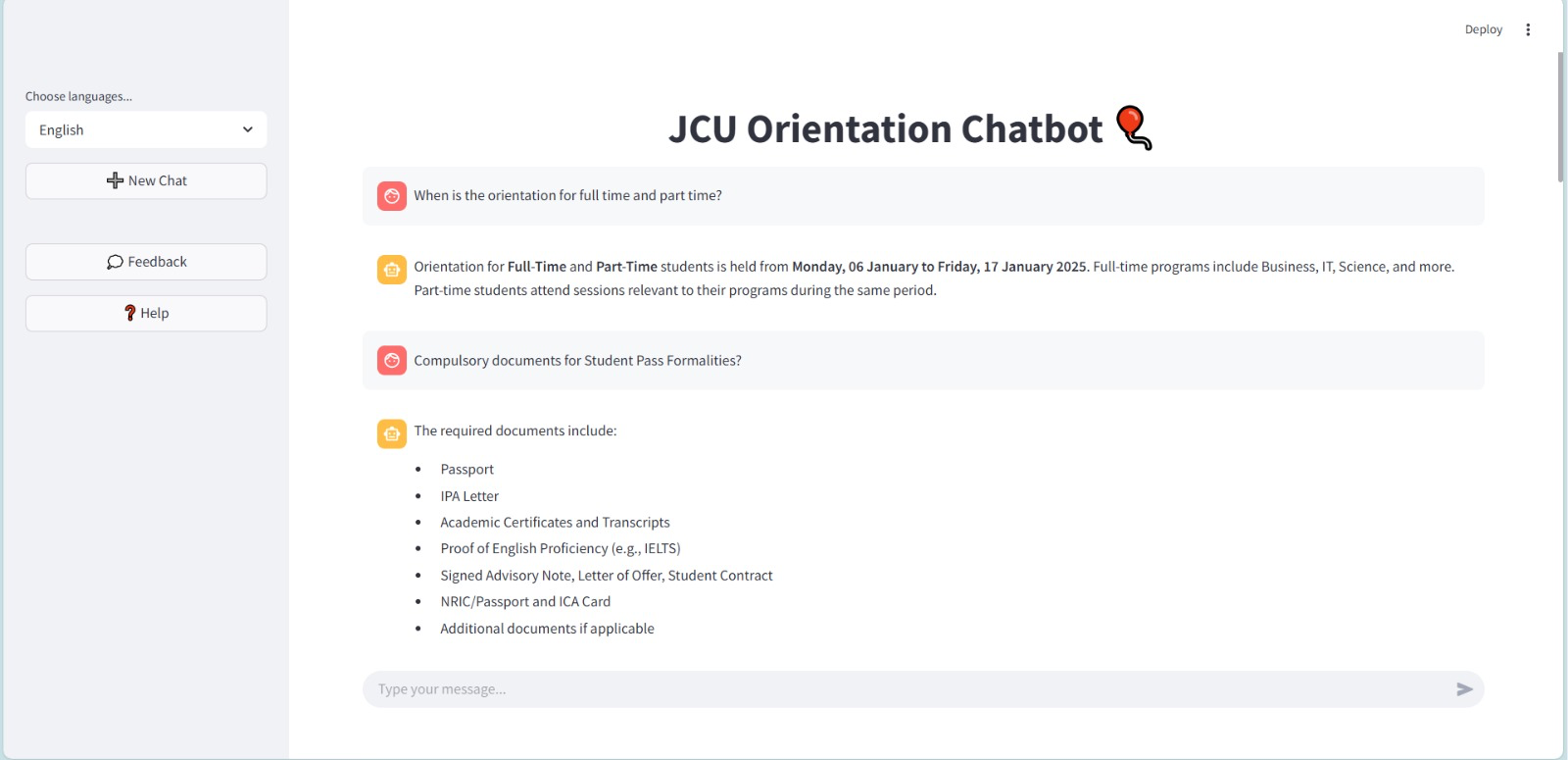
**5. Conclusion**

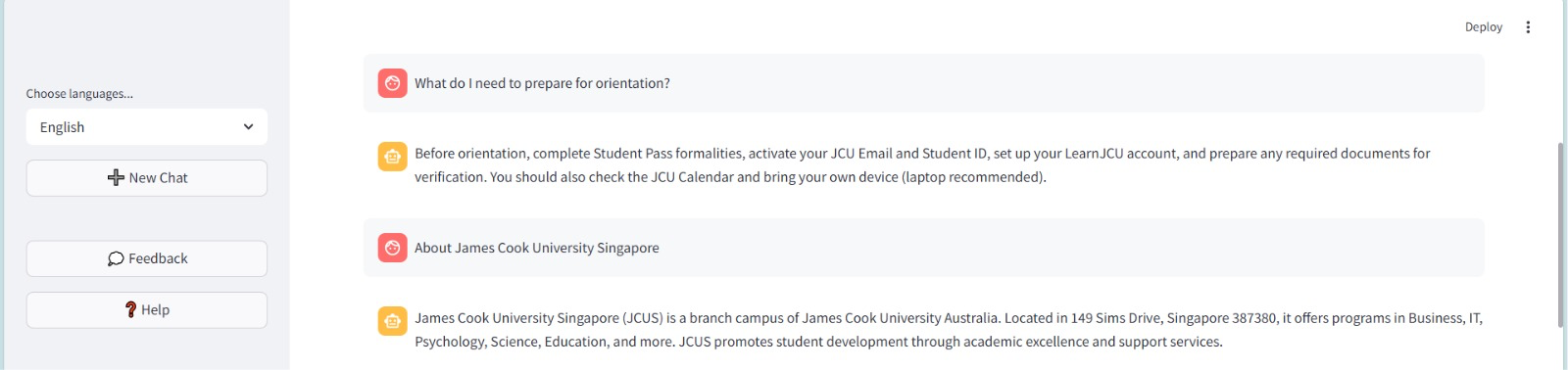
Overall, the chatbot offers a satisfactory user experience in both Chinese and English, featuring a user-friendly interface and reasonable response times. However, the response functionality could be improved, as it sometimes takes over 10 seconds to reply. Despite this, it meets customer needs adequately and is effective for its intended purpose.

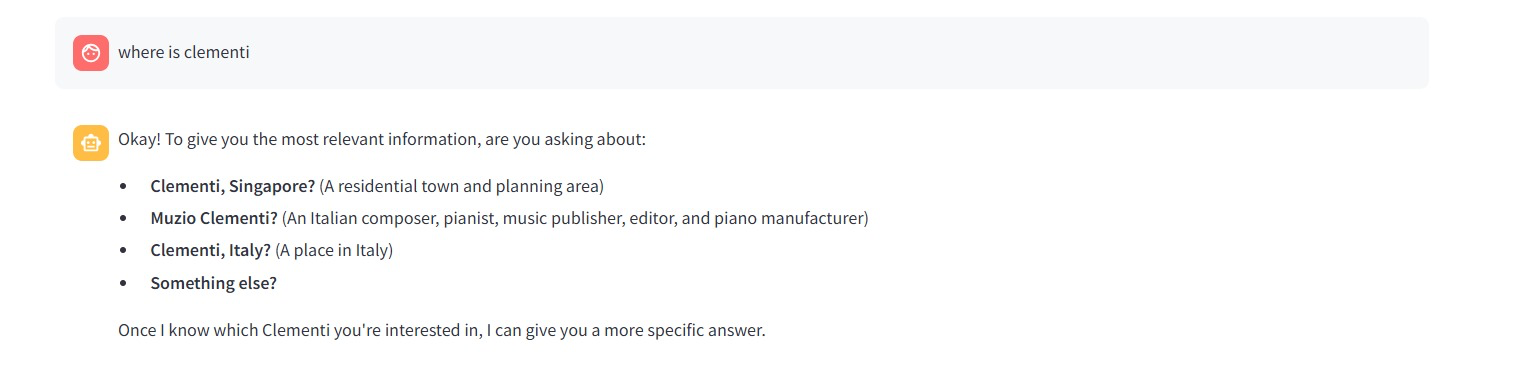
**6. Recommendations**

Concentrate on optimizing response times. Test with multiple users simultaneously.

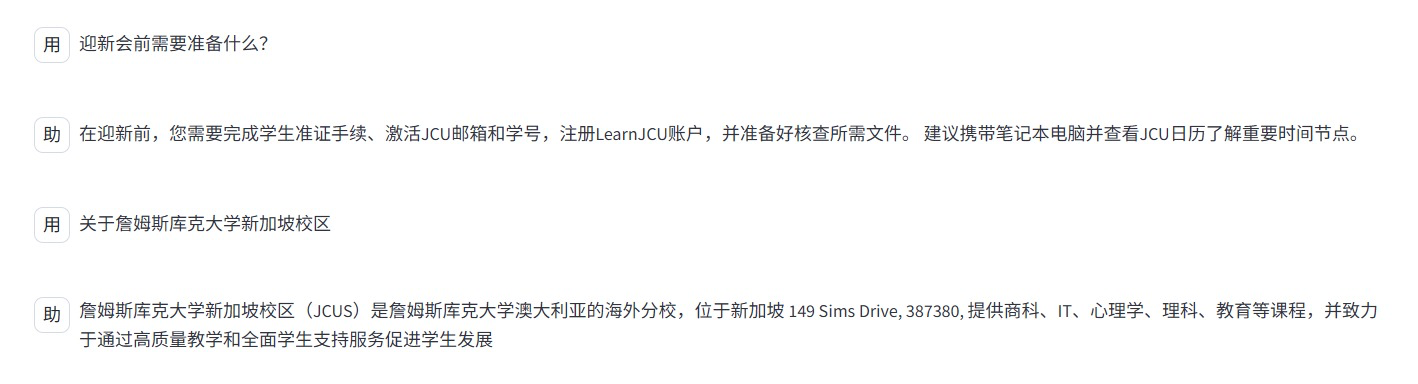
**7. Appendix**

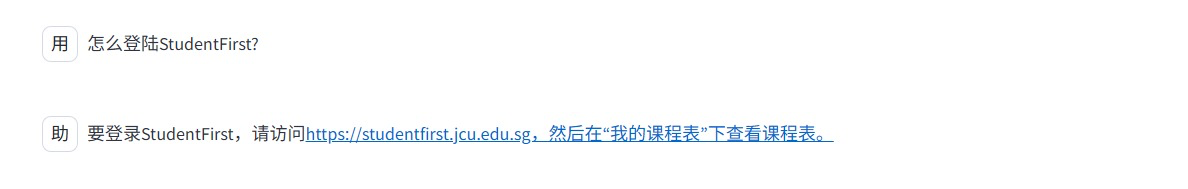
****

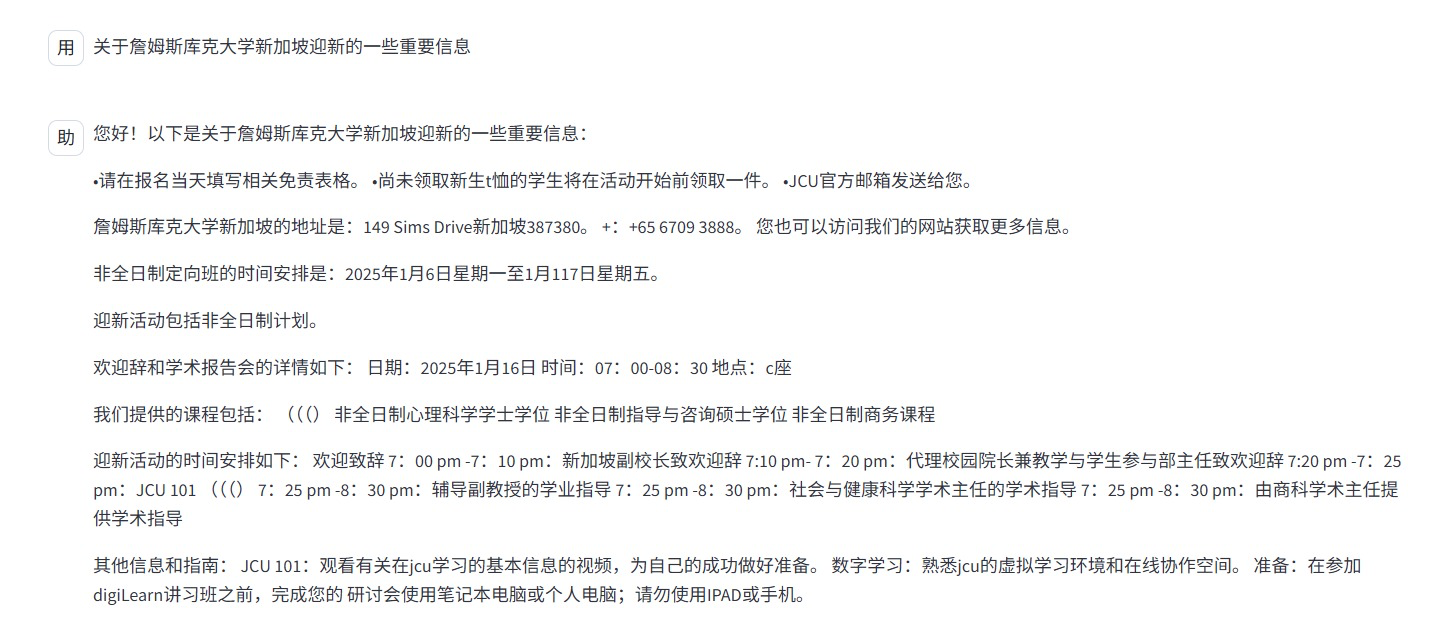
****

****

****

****

****

****