Activity 1:

User Story 1: Ask Orientation Questions

Test Cases:

TC1.1: Verify predefined questions (e.g., "Where is Block C?") trigger accurate responses within 3 seconds.

Steps: Click a predefined question button in **index.html**.

Expected: Chatbot returns correct venue details (e.g., "Block C") and HTTP 200.

TC1.2: Validate response accuracy against Student Affairs data updates.

Steps: Update the training data in **app.py** and ask, "When is the Medical Check-Up?"

Expected: Chatbot reflects the updated schedule (e.g., "6–7 January 2025").

TC1.3: Test real-time query handling.

Steps: Type "What is the dress code for the Welcome Session?"

Expected: Response includes "Smart Casual (No shorts or slippers)" within 3 seconds.

User Story 2: Multilingual Support for International Students

Test Cases:

TC2.1: Verify Chinese input translation.

Steps: Send "体检时间" (Medical Check-Up time) via API.

Expected: Chatbot responds in English with "Medical Check-Up: 6–7 January 2025."

TC2.2: Test language dropdown functionality.

Steps: Select "Chinese" in **index.html** and ask a question.

Expected: Response is translated to Chinese.

TC2.3: Validate context retention after translation.

Steps: Ask "Where is Block C?" in Chinese, then ask "What events are there?" in English.

Expected: Chatbot references Block C in the second response.

User Story 3: Scalability and Load Testing

Test Cases:

TC3.1: Simulate 300 concurrent users accessing **/chat** endpoint.

Steps: Use JMeter to send 300 requests simultaneously.

Expected: Average response time ≤ 3 seconds, no server crashes.

TC3.2: Test database connection under load.

Steps: Run load test while querying "Orientation schedule."

Expected: All responses are consistent and accurate.

TC3.3: Validate error handling during peak traffic.

Steps: Overload the system with 500 users.

Expected: Graceful degradation (e.g., HTTP 503 for excess requests).

User Story 4: Basic Input Format Validation

Test Cases:

TC4.1: Test empty input rejection.

Steps: Send empty JSON to **/chat**.

Expected: HTTP 400 error with "No question provided."

TC4.2: Validate input length limit (500 characters).

Steps: Send a 600-character question.

Expected: HTTP 400 error with "Input exceeds 500 characters."

TC4.3: Check sanitization of special characters.

Steps: Send **<script>alert('test')</script>**.

Expected: Sanitized response without executing the script.

User Story 5: Prompt Engineering for Optimized Responses

Test Cases:

TC5.1: Test prompt template version control.

Steps: Roll back to a previous prompt version.

Expected: Chatbot uses the older template for responses.

TC5.2: Validate sandbox testing of new prompts.

Steps: Deploy a test prompt asking, "What is JCU 101?"

Expected: Response matches the sandbox-defined answer.

TC5.3: Verify documentation completeness.

Steps: Check if prompt engineering guidelines are documented.

Expected: A PDF/Confluence page exists with templates and best practices.

Activity 2:

Testing Environment：

The tests are written using pytest and executed against the Flask application defined in app.py. All tests utilize Flask's test client for making HTTP requests to the /chat endpoint.

Test Case Summary：

Test ID	Description	Expected Result
TC1.1	Test predefined question about Block C	Response contains 'Block C' and HTTP 200
TC1.2	Test Medical Check-Up response	Response includes dates like '06' or '07'
TC1.3	Check dress code info	Response includes 'Smart Casual'

TC2.1	Chinese translation query	Response includes 'Medical Check-Up' in English
TC2.2	Test context retention across languages	Response references Block C or event info
TC3.1	Run 10 rapid requests (simulate load)	All responses return HTTP 200
TC4.1	Empty input test	Returns HTTP 400 and error message
TC4.2	Input length exceeds 500 characters	Returns HTTP 400 or 200 with logic
TC4.3	Special characters sanitization	Script tags are not echoed back
TC5.1	Ask about orientation events	Response mentions 'event' or 'Welcome'
TC5.2	Ask about JCU orientation	Response mentions 'JCU' or 'Orientation'
TC6.1	Simple greeting response	Non-empty message returned
TC6.2	Response returned within 3 seconds	Time taken < 3.0s
TC6.3	Welcome bag collection query	Mentions 'welcome' or 'bag'
TC6.4	Multiple sequential question response check	All responses valid and in context

Conclusion：

All 15 automated tests were implemented to cover functional, input validation, multilingual handling, prompt logic, and performance aspects of the chatbot. These tests provide confidence in the system’s reliability, scalability, and correctness in handling orientation-related queries for JCU.