

1. Ask Orientation Questions

As a new student,

I want to ask the chatbot questions about orientation schedules, resources, and campus facilities,

So that I can quickly find relevant information without searching multiple sources.

- **Acceptance Criteria:**

- User can type or click on a predefined question to ask about orientation details.
- The chatbot provides accurate, up-to-date responses within 3 seconds.
- The chatbot references the latest Student Affairs data.

Estimate: 5 story points

Time: 5 Days

2. Multilingual Support for International Students

As a non-English speaking student,

I want to interact with the chatbot in my native language (e.g., Chinese, Hindi),

So that I can understand orientation details without language barriers.

- **Acceptance Criteria:**

- Integrate translation APIs (e.g., Google Translate) into app.py to handle multilingual inputs.
- Ensure translated responses retain context from the original training data.
- Add a language selection dropdown in index.html.
- Validate accuracy of translations through test cases (e.g., “体检时间” → “Medical Check-Up time”).

Estimate: 6 story points

Time: 10 Days

3. Scalability and Load Testing

As a developer/architect,

I want to ensure the system can handle 300 concurrent users,

So that I can it remains responsive during peak orientation periods.

- **Acceptance Criteria:**

- Perform load tests simulating 300 concurrent users.
- System maintains an average response time of under 3 seconds.

Estimate: 8 story points

Time: 10 Days

4. Basic Input Format Validation

As a system administrator,

I want to ensure user inputs are non-empty and within reasonable length,

So that I can the chatbot does not process invalid or abusive queries.

- **Acceptance Criteria:**

- In app.py, enforce a maximum input length of 500 characters for /chat requests.
- Reject empty inputs in both app.py (HTTP 400) and test.py (prompt re-entry).
- Add error messages for invalid inputs (e.g., "Please enter a non-empty question").

Estimate: 1 story point

Time: 5 Days

5. Prompt Engineering for Optimized Chatbot Responses

As a developer/administrator,

I want to create, refine, and manage prompt templates used by the chatbot's NLP or language model,

So that I can the chatbot's responses remain contextually accurate, consistent, and aligned with Student Affairs guidelines.

- **Acceptance Criteria:**

- Administrators/developers can define prompt templates that structure how the chatbot formulates responses.
- Prompt templates can be version-controlled, allowing rollback to previous configurations.
- Changes to prompts can be tested in a sandbox environment before being deployed to production.
- The chatbot's answers reflect improved accuracy, tone, and style based on refined prompts, as validated by test conversations.
- Documentation exists for how to create and maintain prompt templates, including best practices and guidelines for prompt engineering.

Estimate: 5 story points

Time: 15 Days