

Product Backlog

1. Chat UI Component Implementation

- **Description:** Develop the basic chat interface with a chat window, input field, and toggle button.
- **Acceptance Criteria:**
 - The chat window renders correctly.
 - User input is captured and displayed in the UI.
 - Chat window can be toggled (open/minimized).

2. Chat Controller and Communication Layer

- **Description:** Implement the Chat Controller with AJAX and/or WebSocket integration to forward user queries from the UI to the Chat API.
- **Acceptance Criteria:**
 - Input validation is performed before sending requests.
 - User queries are forwarded via the appropriate protocol.
 - Real-time updates are received and rendered.

3. Chat API Endpoint Development

- **Description:** Create the REST/WebSocket endpoints for the Chat API that will serve as a bridge between the frontend and the chatbot service.
- **Acceptance Criteria:**
 - Endpoints are secure (HTTPS) and documented.
 - Endpoints reliably forward queries and return responses.
 - API error handling is implemented.

4. Chat Processing Component (Backend) Implementation

- **Description:** Build the Chat Processor with its internal classes to process incoming chat queries using the LLM backend.
- **Acceptance Criteria:**
 - Incoming queries are parsed and interpreted.
 - The appropriate context is retrieved.
 - Responses are generated and forwarded back to the API.

5. Document Context Manager Component Implementation

- **Description:** Develop the system to manage pre-indexed documents.
- **Acceptance Criteria:**
 - Documents are stored and indexed correctly.
 - The query engine can retrieve relevant documents based on input.
 - Integration with Chat Processing Component is verified.

6. End-to-End Integration and Testing

- **Description:** Integrate the frontend and backend components and run end-to-end tests to ensure the entire chat system functions as expected.
- **Acceptance Criteria:**
 - User messages flow seamlessly from UI to backend response.
 - Comprehensive tests validate user interactions and API responses.

7. User Interface Enhancements and Error Handling

- **Description:** Improve the chat interface with clear error messages, loading indicators, and responsive design.
- **Acceptance Criteria:**
 - UI feedback is provided for each action (e.g., loading, error states).
 - Usability tests show a clear and intuitive user experience.

First Sprint (Sprint 1) Backlog

Goal: Establish the foundational chat system components that allow for user input capture and communication with a basic API endpoint.

Sprint 1 Items

1. Initial Chat API Endpoint

- Build a simple REST endpoint that accepts user input.
- Return a mock response for testing purposes.
- Set up logging and basic error handling.

2. Integration of Chat UI and Chat API

- Connect the chat controller to the API.

- Verify that user queries from the UI are received by the API.
- Display the mock responses on the UI.

3. Unit Testing and Basic Documentation

- Write tests for the chat controller logic.

4. LLM-choice-of-the-day

- Find out what the best LLM for our contextual needs is.
- Test out prompts, limits, understanding and contextual shape-up of the LLM.
- Start making different prompts for basic types and subtypes of questions.

DoD

Satisfactory responses from the LLM, finding the best one for our use-case. Testing working out fine without any issue with the API or perceived user experience.

Sprint 1 Timeline

- **Planning & Design:** 4 day
- **Development:** 5 days
- **Testing & Documentation:** 1 day
- **Sprint Review & Retrospective:** 1 day

Total Duration: 11 days