**Title: Add an AI chatbot to the Chat Activity**

**OUTLINE**

- About Me
- Technical Statement
- Expectation
- Technical Problem
- Architectural Flow
- Further Enhancement
- TimeLine
- Conclusion

**About Me**

**Full Name:** Jonathan Atiene
**Email and GitHub Username:** bemijonathan@gmail.com
**Your first language:** English
**Location and Timezone:** England

**Intro:** I am a seasoned software engineer with over 5 years of experience, making products. I am currently an Msc Student studying Artificial Intelligence at De Montfort university, my last role was a software engineering role at coderabbit, this team uses LLMs to review code on github, gitlab and even collect context information from jira or other issue tracking software based on the comments on the PR, we also had a chatbot experience where the review bot responds to messages from users with in the context of the Pull Request. My experience there has mostly enriched me with knowledge on how to handle LLMs api integrations in a cost effective approach and handling common but tricky road blocks such as rate limiting prompt versioning and ai agent set up e.t.c.

I have not had a previous contribution to sugar labs but I hope my previous experience and expertise with LLM integrations have been able to convince you that I can be the best man for this job, here are some of my open source code base where i did a bit of prompt engineering code-critique-ai and woot

**Technical Statement : Add an AI chatbot to the Chat Activity**

**Description**
The Chat Activity is an interactive chat program. The goal of this project is to add a chatbot that can be engaged as part of the conversation. The chatbot must be tuned to match the age of our learners, which skews younger than the typical chatbots, so some tuning may be necessary.

**Project Length: 175 hours**

**Expectations:**

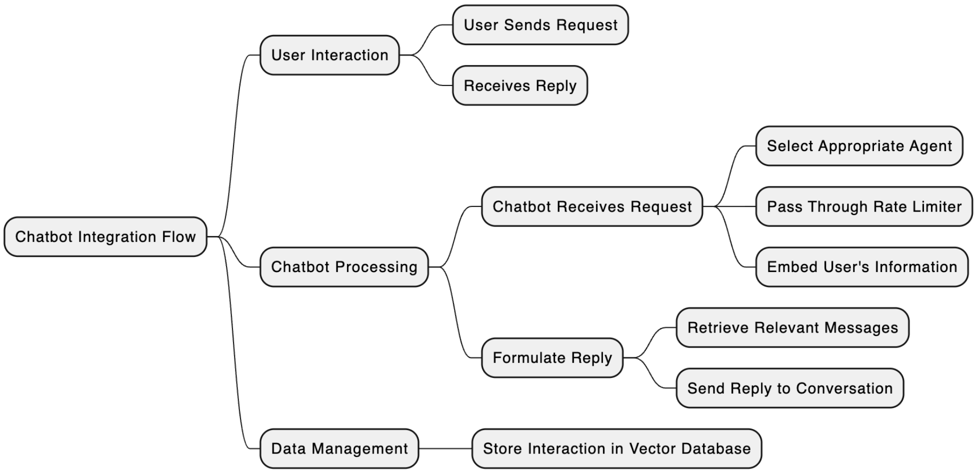
- Create a chatbot program that can participate in conversations.
- Age Prompt tuning that customizes the chatbot to align with the interests and language level of younger learners.
- Safe Interaction the prompts should ensure that the chatbots are safe, respectful and appropriate for the audience.

**Deliverables**

- Integrate the Chatbot into the existing chat codebase
- Customize the chat for younger ones: Here we will introduce multiple bots for each of the age range that will be determined by a simple fuzzy logic for determining the age range of the greater number of participants in call.
- Ensure safe and appropriate interactions
- Balance discussion with Educational value
- Improve user interface.
- Optimize for performance
- Ensure security.

- **Technology Stack**
  - LLM model (gemini ai, Claude, Open ai)
  - Vector Database (pine cone, postgres)
  - Rate Limiters
  - Transformers for fine tuning
  - Python

**Architectural Flow**



**Further Enhancement**

Advanced Chatbot Capabilities
- Implement machine learning to adapt to user behavior and preferences.
- Multilingual Support

User Experience Improvements Add voice recognition and speech-to-text capabilities. Safety and Moderation User Reporting System

Performance Optimization - Scalability Enhancements, Optimize the backend for handling a higher number of simultaneous users efficient caching mechanisms Optimize API calls and database queries.

Chatbot Interaction Analytics Develop a dashboard to monitor chatbot interactions and user engagement.

**Timeline:**

**Week 1: Initialization and Planning**
Get familiar with the existing Chat codebase.
Set up the development environment and tools.
Fix pending issues to gain more familiarity with the codebase.
Understand the entire architecture as regards to sugar labs.
Finalize project plan and milestones.

**Weeks 2-3: Basic Chatbot Integration and Queue Setup**
Add Tests to the codebase.
Create the core chat bot, and the
Implement a message queue system to handle incoming requests efficiently.
Develop the agent selection mechanism.

**Weeks 4-5: Rate Limiter and User Information Embedding**
Integrate a rate limiter to manage request load and prevent overloading.
Implement user information embedding in the chat flow for personalized interactions.

**Weeks 6-7: Database Integration**
Store chat interactions in the vector database, ensuring data is queued and processed efficiently and orderly, context awareness and prompt optimizations.
Develop the system for retrieving messages from the queue.
Implement response formulation based on the retrieved messages.

**Weeks 8: Testing Optimization & Documentation**
Perform comprehensive testing, including stress tests focusing on queue management.

Document the development process, particularly the queue system implementation and vector databases.