CREATE A CHATBOT IN PYTHON

Team Leader: Saravanakkumar D
Team Member-1:Kumarakabilan P
Team Member-2: Veerendran A
Team Member-3: Vijay I
Team Member-4:Saran R

Phase 1: Problem Definition and Design Thinking

The challenge is to create a chatbot in Python that provides exceptional customer service, answering user queries on a website or application. The objective is to deliver high-quality support to users, ensuring a positive user experience and customer satisfaction.

Problem Definition: The challenge is to create a chatbot in Python that provides exceptional customer service, answering user queries on a website or application. The objective is to deliver high-quality support to users, ensuring a positive user experience and customer satisfaction.

Design Thinking:

1. Functionality: Define the scope of the chatbot's abilities, including answering common questions, providing guidance, and directing users to appropriate resources.
2. User Interface: Determine where the chatbot will be integrated (website, app) and design a user-friendly interface for interactions.
3. Natural Language Processing (NLP): Implement NLP techniques to understand and process user input in a conversational manner.
4. Responses: Plan responses that the chatbot will offer, such as accurate answers, suggestions, and assistance.
5. Integration: Decide how the chatbot will be integrated with the website or app.
6. Testing and Improvement: Continuously test and refine the chatbot's performance based on user interactions.

Dataset Link: **https://www.kaggle.com/datasets/grafstor/simple-dialogs-for-chatbot**

1. Functionality:

Define what your chatbot will do:
- Answer common questions.
- Provide guidance.
- Assist with tasks.
- Direct users to resources.

2. User Interface:

- Integrate the chatbot on your website or app.
- Design an inviting chat window.
- Include a welcome message.
- Offer easy start and stop options.

3. Natural Language Processing (NLP):

- Use Python NLP libraries.
- Train the chatbot to understand intent, entities, and sentiment.
- Create a knowledge base of FAQs.

4. Responses:

- Provide accurate answers.
- Suggest relevant solutions.
- Guide users through processes.
- Transition to human support for complex issues.

5. Integration:

- Use chatbot development frameworks.
- Set up communication channels (websockets, REST APIs).
- Ensure data security and privacy compliance.

6. Testing and Improvement:

- Gather user feedback.
- Monitor analytics.
- Update knowledge base and responses & train the chatbot to improve over time.