**Phase-2 Submission Template**

**Student Name:** R.Jaichandran

**Register Number:** 511323104035

**Institution:** kingston engineering college

**Department:** computer science and engineering

**Date of Submission:** / /

**Github repository link:** [**https://github.com/jaichandran1234455/ameen**](https://github.com/jaichandran1234455/ameen)

### **1.Problem Statement**

 **Customer Support Challenges**: Customers often face long wait times, inconsistent support quality, and limited availability from traditional customer service teams.

 **24/7 Availability**: Businesses struggle to provide round-the-clock assistance to users, resulting in missed opportunities and customer frustration.

 **High Operational Costs**: Relying on human agents for routine queries increases operational costs, as well as the workload on customer support teams.

 **Scalability Issues**: As customer interactions grow, businesses find it difficult to scale support without compromising quality or efficiency.

### **2. Project Objectives**

 **24/7 Availability:** Provide round-the-clock automated support to handle customer inquiries at any time, reducing response delays.

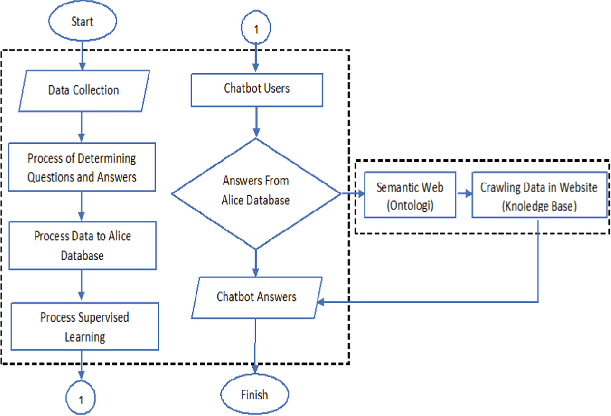
 **Automate Common Queries**: Automate responses for frequently asked questions, order tracking, product information, and basic troubleshooting to reduce the load on human agents.

 **Improve Response Time:** Deliver immediate, accurate, and consistent responses to customer queries, enhancing user satisfaction.

** Seamless Escalation:** Implement an intelligent escalation system that transfers complex issues to human agents with context, ensuring smooth handovers.

 **Personalized Interactions:** Utilize customer data and previous interactions to offer personalized support, improving the user experience.

### **3. Flowchart of the Project Workflow**



### **4. Data Description**

.  **Customer Interaction Data**: Chat logs, conversation histories, and support tickets containing real-time user queries and responses.

 **Intent Data**: Labeled data indicating the intent behind each customer query (e.g., order tracking, product inquiry, technical support).

 **Entity Data:** Key pieces of information extracted from conversations such as product names, order numbers, customer names, and dates.

 **Frequently Asked Questions (FAQs)**: A knowledge base containing common queries and their corresponding answers to help train the chatbot.

 **User Profile Data:** Information like customer demographics, purchase history, and preferences, which help personalize interactions.

### **5. Data Preprocessing**

. ** Text Cleaning:** Remove unnecessary characters (e.g., special symbols, punctuation, and extra spaces), correct spelling errors, and standardize formatting.

 **Tokenization:** Break down customer queries into smaller components like words or phrases (tokens) for easier analysis.

 **Lowercasing:** Convert all text to lowercase to ensure uniformity and avoid discrepancies in word matching.

 **Removing Stop Words:** Eliminate common, non-informative words (e.g., "the", "and", "is") that don't add value to intent recognition.

 **Lemmatization/Stemming:** Reduce words to their root form (e.g., “running” to “run”) to enhance the chatbot's ability to understand variations of words.

### 

### **6. Exploratory Data Analysis (EDA)**

 **Query Frequency Analysis**: Identify the most common types of queries or intents users ask, helping to prioritize chatbot training for high-demand topics.

** Intent Distribution:** Visualize the distribution of different intents (e.g., product inquiry, order status, technical support) to understand the areas that require more focus in chatbot responses.

 **Entity Occurrence:** Analyze the frequency and types of entities (e.g., product names, order IDs, locations) to ensure the chatbot is trained to recognize and handle key information accurately.

 **Sentiment Analysis:** Perform sentiment analysis to determine the emotional tone (positive, negative, neutral) of customer interactions, providing insights into customer satisfaction and areas for improvement.

 **Response Time Analysis**: Measure the average response time from the chatbot and human agents, identifying bottlenecks or inefficiencies in the customer support process.

### **7. Feature Engineering**

.  **Intent Classification Features:** Extract features such as keywords, phrases, or sentence structure to help the chatbot classify the user's intent accurately (e.g., order tracking, product inquiry).

 **Entity Recognition Features:** Identify and extract entities like product names, order numbers, dates, and locations, which are essential for understanding the user’s specific request.

** Contextual Features**: Include historical conversation data or session context (previous queries and responses) to maintain continuity and improve understanding of ongoing conversations.

 **User Profile Features:** Incorporate information from customer profiles (e.g., past interactions, purchase history, preferences) to personalize chatbot responses and provide more relevant support.

 **Sentiment Features:** Extract sentiment-related features (positive, negative, neutral) from the user’s query to tailor responses accordingly and improve the user experience.

### 

### **8. Model Building**

** Data Collection:** Gather and preprocess relevant data, including labeled user queries, intents, entities, and conversation logs for training the model.

 **Intent Classification Model:** Develop and train a model (e.g., using machine learning algorithms like SVM, Random Forest, or neural networks) to classify user intents based on their queries.

 **Entity Recognition Model**: Use named entity recognition (NER) techniques to identify key entities (such as product names, order numbers, and dates) in user queries.

 **Text Preprocessing:** Apply NLP techniques such as tokenization, lemmatization, and stop word removal to process text and prepare it for model training.

 **Embedding Layer:** Use pre-trained embeddings like Word2Vec, GloVe, or BERT to transform text data into numerical vectors that capture semantic meanings for improved model understanding.

### 

### **9. Visualization of Results & Model Insights**

 **Confusion Matrix:** Visualize the performance of the intent classification model by showing true positives, false positives, true negatives, and false negatives for each intent.

 **Accuracy, Precision, Recall, F1-Score:** Plot these evaluation metrics for different intents to assess model performance and balance across various categories.

** Intent Distribution Bar Chart:** Display the frequency of different intents in the dataset to identify the most common customer requests and adjust model training accordingly.

 **Entity Detection Performance:** Use bar graphs or pie charts to visualize how accurately the chatbot is identifying various entities (e.g., product names, order numbers) in customer queries.

 **Response Time Distribution:** Visualize the distribution of response times to evaluate the efficiency of the chatbot and identify any delays or bottlenecks.

 **Sentiment Analysis Graphs:** Create sentiment trend charts or pie charts to track customer sentiment (positive, negative, neutral) throughout interactions, helping assess customer satisfaction.

### **10. Tools and Technologies Used**

### Github

### Jupyter

### Colab

### VS code

### **11. Team Members and Contributions**

* + **Data cleaning** - B.Harish
  + **EDA**  - D.Iniya roshan
  + **Feature engineering**  - R.Jaichandran
  + **Model development** – P.Bharath
  + **Documentation and reporting** -B.Harish