**Phase-1 Submission Template**

**Student Name:** Gopiraja M

**Register Number:** 620123104023

**Institution:** AVS Engineering College

**Department:** BE.Computer Science and Engineering

**Date of Submission:** 26-04-2025

# 1.Problem Statement

**Problem:**

Many organizations face challenges in delivering efficient and timely customer support, often due to limitations in resources, scalability, and availability. This can lead to prolonged response times, increased costs, and a lack of 24/7 support, ultimately impacting customer satisfaction and loyalty

**Why it matters:**

Inadequate customer support can lead to customer dissatisfaction and lost revenue. With significant investments in support staff, optimizing support operations is crucial for business success.

**Solution:**

An intelligent chatbot can answer common questions instantly, work all day and night, and reduce costs—making customer support faster, cheaper, and more reliable.

# 2.Objectives of the Project

**Aim:**

To develop an intelligent chatbot that automates customer support, providing instant, accurate, and 24/7 assistance to improve customer experience and reduce business costs.

**Objectives:**

1. Design a user-friendly chat interface for customers to interact with the chatbot easily.

2. Develop a chatbot using Python and NLP to understand and respond to common customer queries.

3. Ensure 24/7 automated assistance by integrating the chatbot into a web platform.

4. Reduce the workload on human support agents by handling routine questions automatically.

5. Improve response speed and consistency to enhance customer satisfaction.

# 3.Scope of the Project

The scope defines what the chatbot will do and the boundaries of the project:

* **Objective**: To develop and implement an AI-powered chatbot capable of providing 24/7 automated assistance to customers, reducing human agent workload, improving response times, and increasing customer satisfaction.
* **Target Users**: Customers seeking support (pre-sales and post-sales), support agents (as chatbot backup), and business admins.
* **Platforms**: Web applications

**Key Features**

**1. Natural Language Processing (NLP)**

* Understands user queries in conversational language
* Detects intent and extracts relevant entities

**2. Automated Responses**

* Answers FAQs (returns, shipping, pricing, etc.)
* Handles basic troubleshooting
* Provides order tracking and account info (via integration)

**3. Contextual Memory**

* Remembers recent conversation history
* Maintains user context across sessions (if authenticated)

**Limitations**

* **Scope of Understanding**: May struggle with complex, ambiguous, or emotional queries without training.
* **Dependency on Training Data**: Accuracy depends on quality and volume of training data.
* **Limited Personalization (initially)**: Advanced personalization may require extensive data integrations.
* **Not a Full Replacement**: Chatbot cannot entirely replace human agents, especially in sensitive or nuanced scenarios.
* **Multilingual Limitations**: Only English supported initially; expanding to other languages requires NLP retraining.

# 4.Data Sources

For this chatbot project, a manually created set of intents and responses is used as the primary data source. This dataset includes:

* User input patterns (e.g., “Hello”, “I need help”, “Thank you”)
* Corresponding intent tags (e.g., greeting, help, thanks)
* Bot responses for each intent

**Optional Enhancements:**

* If desired, the chatbot can also be trained on:
* Historical customer support chat logs
* FAQs from the company website
* Live user feedback and interaction data

**Dataset Description**

For this project, the chatbot uses a synthetic dataset that is manually created. It contains a set of intents, with each intent including:

* Tag (e.g., greeting, help, thanks)
* Patterns: Sample user inputs that the chatbot might receive
* Responses: Predefined replies the chatbot can give

# 5.High-Level Methodology

**1. Data Collection**

* Technique: Manual creation of a synthetic dataset.
* Content: Includes intents (tags), user input patterns, and bot responses.
* Format: JSON file (e.g., intents.json).

**2. Data Cleaning**

* Tokenization: Break user messages into words using NLTK.
* Lemmatization: Reduce words to their base form (e.g., “running” to “run”).
* Lowercasing: Normalize all text to lowercase.
* Stopword Removal (optional): Common words (e.g., “is”, “the”) can be removed to focus on important terms.

**3. Exploratory Data Analysis (EDA)**

Objective: Understand the diversity of intents and patterns.

Steps:

* Count number of intents.
* Analyze most frequent patterns and keywords.
* Visualize intent distribution (e.g., bar chart showing number of patterns per intent).

**4. Feature Engineering**

Input Features: Cleaned and tokenized user input.

Techniques:

* Convert words into Bag of Words (BoW) or TF-IDF vectors.
* Use one-hot encoding or integer labels for intent tags.
* These features are used to help the chatbot understand the meaning of the message.

**5. Model Building**

* For basic chatbot:

Use rule-based matching or keyword pattern detection.

* For intelligent chatbot:

Build a simple intent classification model using:

Scikit-learn (e.g., Naive Bayes, SVM)

**6. Model Evaluation**

* Use accuracy score, confusion matrix, or F1-score to measure how well the model predicts the right intent.
* Split data into training and testing sets if using ML/DL.
* Test the bot with real-world queries to ensure quality responses.

**7. Visualization**

* Use matplotlib or seaborn to show:
* Intent frequency
* Model performance metrics

**8. Deployment**

* Use Flask to build a simple web server.
* Connect the chatbot logic to a frontend (HTML/JS).

# 6.Tools and Technologies

**1. Programming Language:**

* Python

Commonly used libraries include:

* NLTK for text processing
* Flask for web integration
* random, json for basic logic and data handling

**2. Notebook/IDE:**

* Jupyter Notebook:

Used for initial data preparation, NLP preprocessing, and testing model logic.

* Visual Studio Code (VS Code)
* Ideal for full project development and file management (especially Flask app and HTML/JS files).

# 7.Team Members and Roles

**Barath** - Develop the chatbot logic using python.

Implements natural language processing(NPL)with library like NLTK.

Builds and integrates the backend using flask.

**Harish** - Prepares and cleans the dataset (intents,patterns,responses).

Perform feature engineering and builds intent classification models.

Design and develops the user interface using HTML, CSS and Javascript.

Analysis chatbot performance and suggests improvements.

**Balaji** - Design and develops the user interface using HTML, CSS and Javascript.

Connect the frontend with backend APIs.

Ensure the chatbot is responsive and user-friendly.

**Gopiraja** – Coordinate communicate between team members.

Ensure the project stays on track and meets objectives.

Oversees the project timeline and deliverable.