

Student Name: THIVESHAN G

Register Number: 410723104089

Institution: Dhanalakshmi college of engineering

Department: BE computer science and engineering

Date of Submission: 30.04.2025

1.Problem Statement

Revolutionizing customer support with an intelligent chatbot for automated assistance

2.Objectives of the Project

To develop and implement an intelligent AI-powered chatbot that automates customer support by accurately understanding and responding to user queries, providing instant, 24/7 assistance, reducing response times, minimizing support costs, and enhancing overall customer satisfaction through seamless integration with existing support systems.

3.Scope of the Project

☐ In-Scope Activities:

- *Design and development of an AI chatbot using NLP for automated customer support.*
- *Integration with existing support systems (CRM, ticketing platforms, knowledge bases).*
- *Implementation of conversational flows for handling FAQs, basic troubleshooting, and transactional queries.*
- *Deployment on selected platforms (e.g., website, mobile app, WhatsApp, Facebook Messenger).*
- *Logging and analysis of user interactions for continuous learning and improvement.*
- *Dashboard for performance analytics (response time, resolution rate, user satisfaction).*
- *Escalation mechanism to transfer queries to human agents when necessary.*

☐ Technology Scope:

- *Use of AI/ML technologies (e.g., GPT, Dialog flow, Rasa) for natural language understanding.*
- *Backend systems for managing data and logic.*
- *Secure handling of user data in compliance with data privacy regulations (e.g., GDPR).*

☐ Out of Scope (Initial Phase):

- *Complex technical support requiring expert diagnosis or hands-on intervention.*
- *Full replacement of human agents (hybrid approach only).*

**Voice-based chatbot systems (unless specified for future phases).
Support in niche languages unless prioritized.*

4.Data Sources

a. Customer Support Logs

- *Historical chat or email transcripts.*
- *Helpdesk tickets from platforms like Zendesk, Freshdesk, or Salesforce.*
- **Purpose:** *Train the model to understand common queries and responses.*

b. FAQs and Knowledge Base Articles

- *Existing company FAQs.*
- *Internal documentation or public help articles.*
- **Purpose:** *Build the chatbot's initial knowledge base.*

c. User Interaction Logs

- *Website interactions, previous bot chats, or call transcripts (if available).*
- **Purpose:** *Understand user intent, navigation flow, and problem areas.*

d. Public Datasets

These are useful for pre-training or augmenting your domain-specific data.

- **Customer Support on Kaggle** (e.g., Customer Support on Twitter)
- **Ubuntu Dialogue Corpus** – *For technical support-related dialogues.*
- **M Ulti WOZ** – *A large-scale multi-domain wizard-of-oz dataset.*

- **DSTC (Dialog State Tracking Challenge)** – Useful for task-oriented dialog systems.

2. Real-Time Data Sources (for chatbot responses and updates):

- **CRM systems:** For retrieving/updating user data or ticket info.
- **Live Knowledge Bases:** For the most up-to-date answers (e.g., articles, product manuals).
- **APIs:** Company databases, shipping status, product availability, etc.
- **User Session Info:** Real-time browsing behaviour, purchase history, or preferences.

5.High-Level Methodolog

1. Requirement Gathering & Analysis

- Identify customer pain points and support use cases.
- Gather data sources: FAQs, support tickets, chat logs, CRM inputs.
- Define success metrics (e.g., response time reduction, resolution rate).

2. Design Phase

- Create user personas and define conversational flows.
- Design chatbot architecture (NLP engine, integration points, databases).
- Prepare UI/UX mock-ups for chatbot interface (if needed).

3. Development Phase

- Train NLP models using collected support data and predefined intents.
- Implement the chatbot engine using tools like Dialog flow, Rasa, or GPT APIs.

- *Set up backend infrastructure for managing sessions, APIs, and logs.*
- *Integrate with CRM/ticketing systems and company knowledge base.*

4. Testing Phase

- *Perform unit testing of chatbot modules.*
- *Conduct end-to-end testing across supported platforms.*
- *Run user acceptance testing (UAT) with real-world scenarios.*

5. Deployment

- *Deploy chatbot on live channels (website, app, or messaging platforms).*
- *Monitor real-time performance and user feedback.*
- *Ensure fallback mechanisms to human agents work smoothly.*

6. Monitoring & Continuous Improvement

- *Collect and analyse interaction data and KPIs.*
- *Fine-tune models and update conversation flows based on feedback.*
- *Add new features (multi-language support, voice input, sentiment analysis).*

6.Tools and Technologies

1. Natural Language Processing (NLP) & AI Platforms

- **OpenAI GPT (ChatGPT API)** – *For advanced conversational AI and context handling.*
- **Google Dialog flow** – *For intent recognition and dialog management.*
- **Rasa** – *Open-source framework for custom chatbot development.*
- **Microsoft Bot Framework** – *For enterprise-grade chatbot development.*

2. Programming Languages

- **Python** – For backend logic, NLP integration, and ML model training.
- **JavaScript/TypeScript** – For frontend integration or Node.js-based bots.
- **HTML/CSS** – For embedding chatbot UI into websites or apps.

3. Backend & Infrastructure

- **Node.js / Flask / Django** – To manage chatbot logic and API interactions.
- **Express.js** – For building RESTful APIs.
- **Firebase** – For real-time database and quick prototyping.

4. Database & Storage

- **MongoDB** – NoSQL database for storing user sessions, logs, and chatbot data.
- **PostgreSQL / MySQL** – For relational data storage (user profiles, ticket records).
- **Redis** – For caching user sessions and context.

5. Integrations & APIs

- **CRM Systems:** Salesforce, HubSpot, or Zoho CRM
- **Ticketing Systems:** Zendesk, Freshdesk, Jira
- **Messaging Platforms:** WhatsApp Business API, Facebook Messenger, Slack, Telegram
- **Webhook & REST APIs:** For integration with internal systems

6. DevOps & Hosting

- **Cloud Platforms:** AWS, Google Cloud Platform (GCP), Microsoft Azure

- **Containerization:** *Docker for scalable deployment*
- **CI/CD Tools:** *GitHub Actions, Jenkins, or GitLab CI for automated deployment*

7. Monitoring & Analytics

- **Google Analytics / Firebase Analytics** – *For tracking usage.*
- **Bot press Analytics** – *Built-in chatbot analytics.*
- **Custom Dashboards:** *Built using Grafana, Kibana, or Power BI*

8. Project & Collaboration Tools

- **Jira / Trello** – *Project management and task tracking.*
- **Figma** – *UI/UX design for chatbot interface.*
- **Postman** – *For API testing.*

7.Team Members and Roles

*** Team Leader : Pragadeesh A**

**Data collecting and Coding*

*** Team Member : Thiveshan G**

**Report writing and documentation*

*** Team Member : Sanjai Ram N**

** Data cleaning and preprocessing*

*** Team Member : Sabarisan D**

** EDA and vitalisation*