

**CHATBOT-GREVIENCE AND READDRESSAL SYSTEM**

**A PROJECT REPORT**

***Submitted by***

**SYED UMAR FARUQ S (1920106713)**

**VIJAY K (1920106714)**

**VENIKA P I (1920106103)**

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

**IN**

INFORMATION TECHNOLOGY

**SONA COLLEGE OF TECHNOLOGY**

**SALEM-5(Autonomous)**

**NOVEMBER 2023**

**ANNA UNIVERSITY: CHENNAI 600 025**

**BONAFIDE CERTIFICATE**

Certified that this project report “**“CHAT-BOT GREVIENCE AND READDRESSAL SYSTEM ”** is the bonafide work of **“SYED UMAR FARUQ S (1920106713), VIJAY K (1920106714) VENIKA P I (1920106103)”** who carried out the project work under my supervision.

**SIGNATURE**

**SIGNATURE**

Dr. J. Akilandeswari

Professor

Dr.Y.Suresh

Professor

**HEAD OF THE DEPARTMENT**

**SUPERVISOR**

Department of Information Technology

Department of Information Technology

Sona College of Technology

Sona College of Technology

Salem- 636 005.

Salem- 636 005.

Submitted for mini project viva voce examination held on

|  |  |
| --- | --- |
| **INTERNAL EXAMINER** | **EXTERNAL EXAMINER** |

**ACKNOWLEDGEMENT**

First and foremost, we thank to **power of almighty** for showing us inner peace and for all blessings .Special gratitude to our parents, for showing their support and love always.

We express our sincere thanks to chairman **Sri.C.VALLIAPPA** and principal **Dr.S.R.R.SENTHILKUMAR** for providing adequate facilities to complete the project.

We are immensely grateful to our Head of Information Technology Department, **Dr.J.AKILANDESWARI for** the continue encouragement to complete theproject.

We heartfully thank our project coordinator **Dr.Y.SURESH** for all of his support and guidance throughout this project. Your leadership has been instrumental in our success.

We express our heartfelt thanks to our project supervisor **Dr.Y.SURESH** for his valuable guidance and fruitful discussions throughoutthe course of the project work.

We feel proud in sharing this success with all our department Faculty, Staff members and friends who helped directly and indirectly to completing this project successfully.

**ABSTRACT**

The Chatbot-Grievance and Redressal System addresses the escalating volume of user-generated grievances in online platforms by leveraging advanced technologies to streamline the resolution process. This innovative system incorporates a chatbot equipped with natural language processing (NLP) capabilities and machine learning algorithms to autonomously classify user grievances into predefined categories. In recent years, the proliferation of online platforms has led to an exponential increase in user-generated content, including grievances and complaints. Addressing these issues efficiently is crucial for maintaining user satisfaction and organizational reputation. This research presents a novel approach to grievance classification through the implementation of a chatbot-based system. Leveraging natural language processing (NLP) and machine learning techniques, the chatbot autonomously categorizes user grievances into predefined classes, streamlining the resolution process. The underlying architecture integrates seamlessly with existing customer support or grievance resolution systems, ensuring a cohesive and efficient workflow. The system's development involves meticulous data collection and preprocessing, utilizing a diverse and labeled dataset to train the chatbot effectively. Performance metrics such as accuracy, precision, recall, and F1 score are employed to evaluate the model, and user feedback is considered to align the system with user expectations. Additionally, the chatbot is designed for dynamic learning, adapting to evolving patterns in user grievances over time. The project emphasizes ethical considerations, including transparency in decision-making, bias mitigation, and user privacy. Through comprehensive evaluations, the Chatbot-Grievance and Redressal System not only aims to enhance the efficiency of the grievance resolution process but also contributes to the evolving field of automated grievance management in the digital era.

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **CHAPTER NO.** | **TITLE** |  | **PAGE NO.** |
|  | **LIST OF FIGURES** | |  |
|  | **LIST OF ABBREVIATIONS** | |  |
| **1.** | **INTRODUCTION** | | **01** |
|  | 1.1. | Background and statement of the project | 01 |
|  | 1.2. | Project objective and significance | 03 |
| **2.** | **LITERATURE SURVEY** | | **04** |
|  | 2.1. | Previous Grievance Handling Systems | 04 |
|  | 2.2. | Natural Language Processing | 05 |
|  | 2.3. | Machine Learning Algorithms | 07 |
|  | 2.4. User Experience and Interface Design | | 08 | |
|  | 2.5. Case Studies and Success Stories | | 11 | |
| **3.** | **PROJECT DESCRIPTION** | | **13** |
|  | 3.1. | Scope of the Project | 13 |
|  | 3.2. | Data collection and sources | 14 |
|  | 3.3. | Methodology | 15 |
|  | 3.4. | Block Diagram | 17 |
|  | 3.5. | Training and validation | 18 |
| **4.** | **RESULTS** | | **21** |
|  | 4.1. | Characterization results | 21 |
|  | 4.2. | Model performance observation | 23 |
| **5.** | **CONCLUSION** | | **26** |
| **6.** | **APPENDIX** | | **27** |
| **7.** | **REFERENCES** | | **41** |

**LIST OF FIGURES**

|  |  |
| --- | --- |
| **CHAPTER 1** | **Page No** |
| Fig 1.1 Architecture of voice based chatbot | 02 |
| **CHAPTER 2** |  |
| Fig 2.1 NLP Process | 06 |
| Fig 2.2 User Interface model | 10 |
| Fig 2.3 UI/UX Cycle | 10 |
| Fig 2.4User InterfaceFlow diagram | 11 |
| **CHAPTER 3** |  |
| Fig 3.1 Data Collection process | 15 |
| Fig 3.2 Methodology process and Structural diagram | 16 |
| Fig 3.3Structural flow of chatbot Agorithm | 17 |
| Fig 3.4 Chatbot:Architecture , Applications and Design Process | 17 |
| Fig 3.5 Training and Validation Process | 20 |
| **CHAPTER 4** |  |
| Fig 4.1 Outline of chatbot | 21 |
| Fig 4.2 Voice Recognition Interaction | 22 |
| Fig 4.3 Recording is enabled | 22 |
| Fig 4.4 User Interaction with Microphone | 23 |
| Fig 4.5 Chat-Bot Interaction with audible mode | 24 |
| Fig 4.6 Chat-Bot Responsive OpenAI Algorithm | 25 |

**LIST OF ABBREVIATIONS**

**NLP**– Natural Language Processing

**ML** – Machine Learning

**FAQ** – Frequently Asked Questions

**UE** – User Experience

**API** – Application Programming Interface

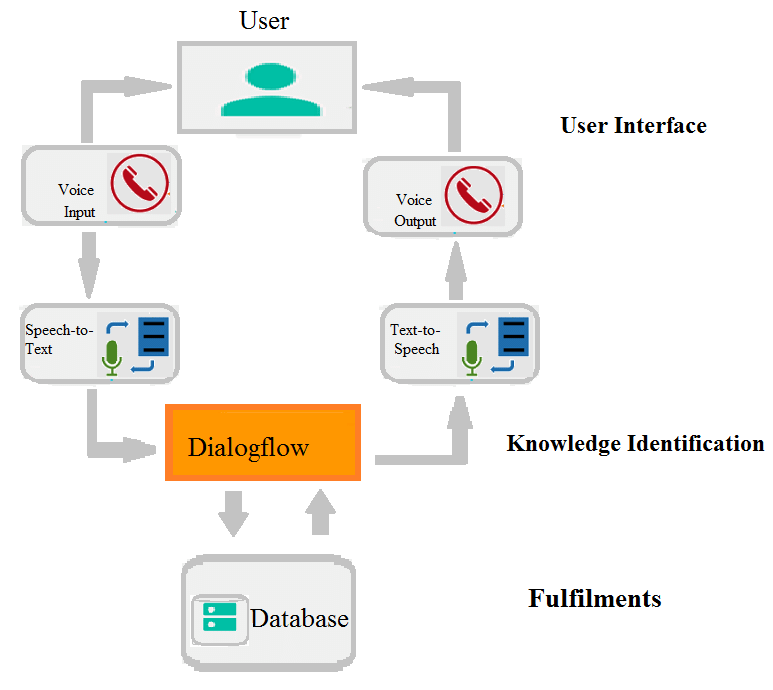
**UI** – User Interface

**CHAPTER 1**

**INTRODUCTION**

**1.1 BACKGROUND AND STATEMENT OF THE PROJECT**

The rapid growth of online platforms has significantly increased the volume of user-generated grievances, necessitating efficient systems for their prompt resolution. In response to this demand, the Chatbot-Based Grievance Classification project seeks to develop an intelligent and automated solution to streamline the grievance handling process. The project's foundation rests on the integration of a sophisticated chatbot, incorporating advanced natural language processing (NLP) techniques and machine learning algorithms. This chatbot is designed to autonomously categorize user grievances into predefined classes, thereby expediting the resolution workflow. The motivation behind this project stems from the recognition of the limitations in traditional grievance management systems and the potential for chatbots to enhance accuracy, efficiency, and user satisfaction. The project aims to create a robust system that not only classifies grievances effectively but also adapts to evolving patterns over time, ensuring a dynamic and responsive approach to user feedback. By addressing these challenges, the Chatbot-Based Grievance Classification project endeavors to contribute to the optimization of customer support processes in the digital age.



# Figure 1.1 Architecture of voice based chatbot

**1.2 PROJECT OBJECTIVE AND SIGNIFICANCE**

The primary objective of the Chatbot-Based Grievance Classification project is to develop an intelligent and automated system that efficiently classifies user grievances in online platforms. By leveraging natural language processing (NLP) techniques and machine learning algorithms, the project aims to empower a chatbot to autonomously categorize diverse user-generated complaints into predefined classes. The ultimate goal is to enhance the efficiency of the grievance resolution process, reducing response times and improving overall user satisfaction.

The significance of this project lies in its potential to revolutionize the way organizations handle and address user grievances in the digital age. With the increasing volume of online interactions, manually processing and categorizing grievances can be time-consuming and prone to errors. The implementation of a chatbot-based system not only offers a scalable solution but also contributes to the optimization of resources in customer support departments. Additionally, by automating the grievance classification process, organizations can allocate human resources more strategically, focusing on complex or high-priority cases that may require personalized attention. This project aligns with the growing demand for innovative technologies that enhance customer service and ensures a more seamless experience for users interacting with online platforms.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 PREVIOUS GRIEVANCE HANDLING SYSTEMS**

In the realm of grievance handling, traditional systems have historically relied on manual processes and human intervention. These conventional approaches often involve dedicated customer support teams tasked with reviewing and categorizing user-generated complaints and feedback. In this manual paradigm, individuals manually sift through a myriad of grievances, attempting to discern patterns and allocate them to relevant categories for subsequent resolution. However, this method has inherent limitations, including the potential for human error, scalability challenges, and increased response times, particularly in the face of a growing volume of user interactions on digital platforms.

As technology advanced, early automated grievance handling systems emerged, integrating basic rule-based algorithms to categorize and address user complaints. These systems, while introducing a level of automation, were often rigid and struggled to adapt to the nuanced and dynamic nature of natural language used in grievances. They were limited by predefined rules and lacked the ability to learn and evolve over time. Moreover, the absence of sophisticated natural language processing (NLP) capabilities hindered their effectiveness in accurately understanding and classifying user sentiments.

In recent years, the advent of machine learning and NLP technologies has significantly transformed the landscape of grievance handling. Modern systems leverage advanced algorithms, such as deep learning models and ensemble techniques, to autonomously classify user grievances based on their content and context. These systems can dynamically learn from data, improving their accuracy and adaptability over time. Additionally, they incorporate sentiment analysis to discern the emotional tone of user messages, providing a more nuanced understanding of grievances. This evolution in grievance handling systems signifies a paradigm shift towards more efficient, automated, and intelligent approaches, aiming to enhance user satisfaction and streamline the resolution process in the ever-expanding digital ecosystem.

**2.2 NATURAL LANGUAGE PROCESSING**

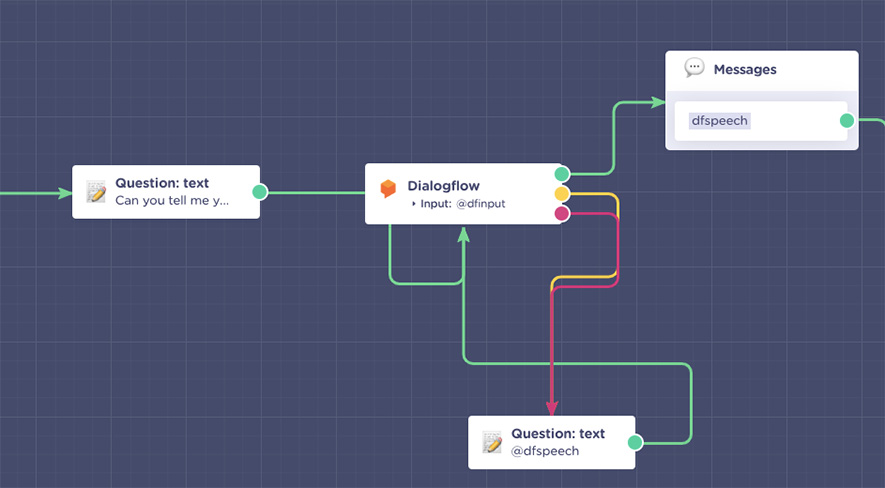
Natural Language Processing (NLP) is a subfield of artificial intelligence (AI) that focuses on enabling machines to understand, interpret, and generate human language. It encompasses a diverse range of techniques and algorithms designed to bridge the gap between human communication and computer understanding. At its core, NLP seeks to empower machines with the ability to comprehend the intricacies of natural language, including syntax, semantics, and pragmatics, in a manner analogous to human linguistic processing.

One of the fundamental challenges in NLP is the ambiguity and complexity inherent in natural language. Human languages are rich in context, idioms, and expressions, making them intricate for machines to decipher. NLP algorithms strive to extract meaningful information from text data, involving tasks such as tokenization, where sentences are broken down into individual words or tokens, and part-of-speech tagging, which assigns grammatical categories to each token.

NLP also includes syntactic and semantic analysis to understand the grammatical structure and meaning of sentences. Sentiment analysis, another crucial aspect of NLP, aims to discern the sentiment or emotional tone expressed in text, facilitating applications such as sentiment-aware customer feedback analysis.

Machine learning plays a pivotal role in NLP, with models trained on vast datasets to recognize patterns and associations within language. Traditional approaches, including rule-based systems, have been supplemented and, in many cases, supplanted by more advanced techniques such as deep learning. Deep neural networks, particularly recurrent neural networks (RNNs) and transformers, have demonstrated remarkable success in tasks like language translation, question answering, and text summarization.

NLP applications are pervasive in our daily lives, from virtual assistants and chatbots that understand and respond to user queries, to language translation services that break down language barriers. As NLP continues to advance, its impact on various industries, including healthcare, finance, and customer service, is expected to grow, ushering in a new era of human-computer interaction characterized by improved language understanding and communication.



**Figure 2.1 NLP Process**

**2.3 MACHINE LEARNING ALGORITHMS**

Machine learning algorithms represent a core component of artificial intelligence, enabling computers to learn patterns and make predictions or decisions without being explicitly programmed. These algorithms are designed to iteratively analyze and interpret data, identifying underlying patterns and relationships. One fundamental distinction in machine learning is between supervised and unsupervised learning.

In supervised learning, algorithms are trained on labeled datasets, where the input data is paired with corresponding output labels. The algorithm learns to map inputs to outputs, making predictions on new, unseen data. Common supervised learning algorithms include linear regression for regression tasks and classification algorithms such as support vector machines, decision trees, and neural networks for classification tasks. Neural networks, particularly deep learning architectures like convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have gained prominence for their ability to handle complex patterns and hierarchical representations in data.

Unsupervised learning, on the other hand, involves training algorithms on unlabeled data, allowing the algorithm to identify patterns and structures independently. Clustering algorithms, such as k-means and hierarchical clustering, group similar data points together based on inherent similarities. Dimensionality reduction techniques, like principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE), aim to simplify complex datasets by reducing the number of features while retaining essential information.

Reinforcement learning is another paradigm where an agent learns to make decisions by interacting with an environment. The agent receives feedback in the form of rewards or penalties, guiding it towards optimal decision-making strategies over time. This approach is particularly relevant in scenarios where an algorithm needs to make a sequence of decisions to achieve a long-term goal.

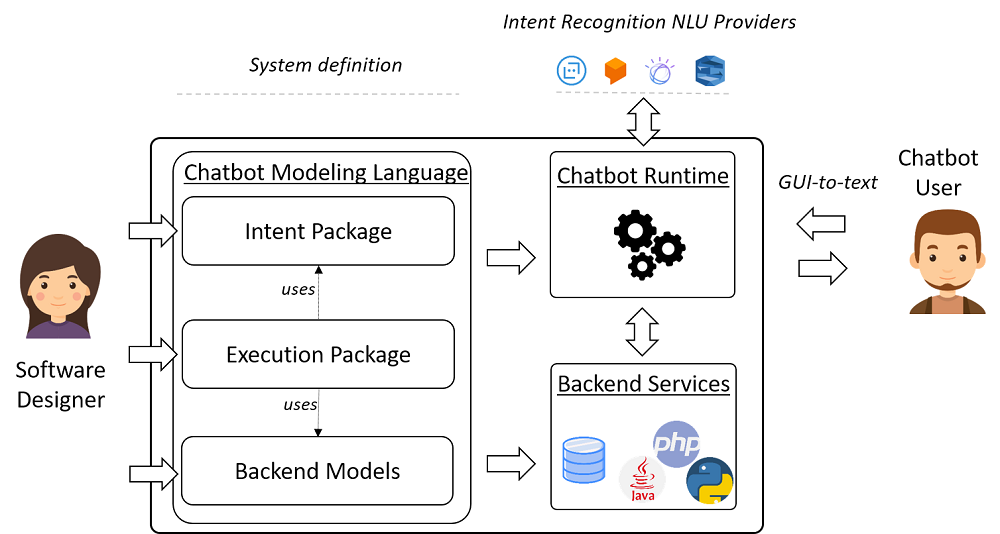
Ensemble learning techniques, like random forests and gradient boosting, combine multiple models to improve overall performance and robustness. These methods leverage the diversity of multiple models to collectively make more accurate predictions.

The choice of machine learning algorithm depends on the nature of the problem, the characteristics of the data, and the desired outcome. As machine learning continues to advance, hybrid models and meta-learning approaches that combine elements of different algorithms are becoming increasingly prevalent, offering solutions to a broader range of complex problems across various domains..

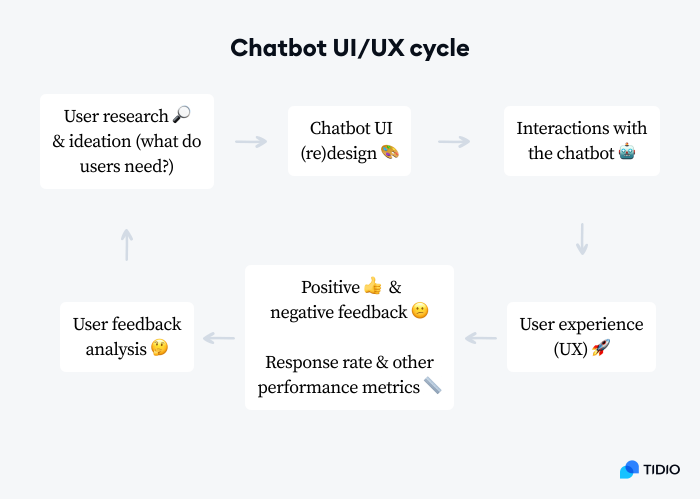
**2.4 USER EXPERIENCE AND INTERFACE**

User experience (UX) and interface design play integral roles in shaping the interaction between users and software systems, ensuring that the experience is intuitive, efficient, and satisfying. The user interface (UI) serves as the point of interaction, while UX encompasses the broader journey, emotions, and perceptions of users throughout their interaction with the system.

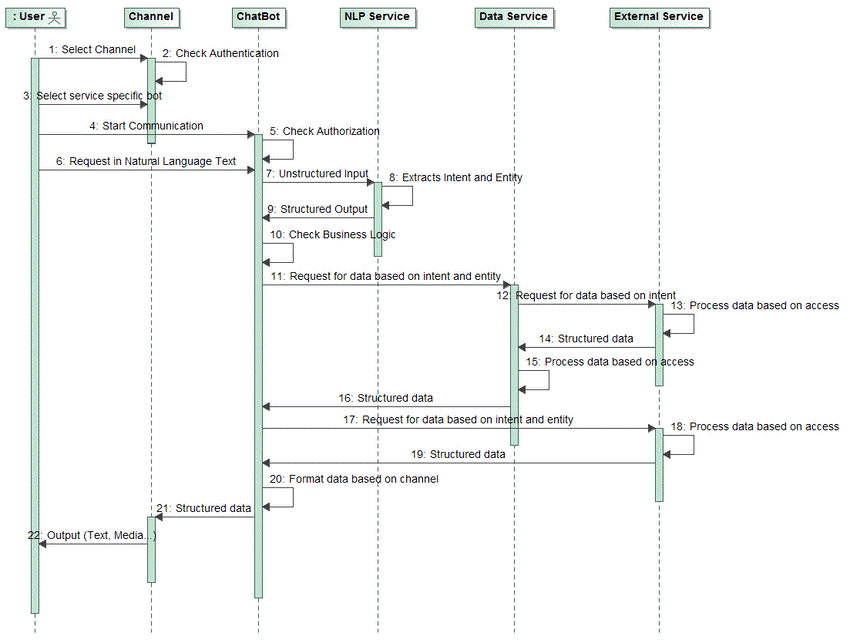
In the context of a chatbot-based grievance classification system, an effective UX begins with a well-designed UI that prioritizes clarity and ease of use. The interface should present a clean and organized layout, guiding users through the grievance submission and resolution process seamlessly. Clear navigation elements, such as well-labeled buttons and menus, contribute to an intuitive flow, reducing the likelihood of user confusion. The design of the UI should align with established usability principles, considering factors such as consistency, where elements behave predictably across the interface, and simplicity, ensuring that users can achieve their goals with minimal effort. Additionally, visual elements, such as color schemes and typography, should be chosen deliberately to enhance readability and maintain a visually appealing design. For a chatbot handling grievances, the conversational aspect of the interface becomes crucial. The chatbot should employ natural language understanding to engage users in a manner that feels human-like and empathetic. Responses should be clear, concise, and contextually relevant, fostering a sense of understanding and trust between the user and the system. User feedback mechanisms, such as acknowledgment messages and progress indicators, contribute to a transparent and informative UX. These elements keep users informed about the status of their grievance, reinforcing a sense of control and engagement throughout the resolution process. Accessibility is another key consideration in UX and interface design. The system should be designed to accommodate users with diverse needs, including those with disabilities. This involves providing alternative text for images, ensuring keyboard navigation, and offering adjustable font sizes. Regular usability testing and user feedback sessions are essential components of refining the UX and interface design. By gathering insights from real users, designers can identify pain points, areas of confusion, and opportunities for improvement, leading to iterative enhancements that align more closely with user expectations and preferences.



**Figure 2.2 User Interface model**



**Figure 2.3 UI/UX CYCLE**



**Figure 2.4 User InterfaceFlow diagram**

**2.5 CASE STUDIES AND SUCCESS STORIES**

One noteworthy case study in the realm of chatbot-based grievance systems involves the implementation by a leading e-commerce platform. Faced with a growing volume of customer grievances related to order tracking, returns, and product inquiries, the company introduced an intelligent chatbot to enhance their customer support capabilities. The chatbot, integrated into their customer service interface, effectively categorized and addressed common issues, providing instant resolutions for straightforward concerns. Users could seamlessly interact with the chatbot to inquire about order status, initiate returns, and receive real-time updates. The success of this implementation was marked by a significant reduction in the resolution time for routine grievances, leading to increased customer satisfaction. The chatbot not only handled straightforward queries but also efficiently escalated complex issues to human agents, ensuring a personalized touch for more intricate problem-solving. The e-commerce platform reported positive feedback from customers regarding the speed and efficiency of grievance resolution, highlighting the success of the chatbot in optimizing the overall customer experience.

In another case, a telecommunications giant deployed a chatbot-based grievance system to enhance its customer support services. The chatbot was designed to understand and categorize a wide array of customer queries, ranging from billing issues to technical troubleshooting. By leveraging natural language processing capabilities, the chatbot accurately identified the context of user messages and provided relevant information or step-by-step guidance for issue resolution. The implementation resulted in a notable reduction in the volume of routine queries reaching human agents, allowing them to focus on more complex problem-solving. Additionally, the chatbot contributed to improved customer engagement by offering instant, 24/7 support, ultimately enhancing the overall efficiency and effectiveness of the telecommunications company's customer service operations.

These case studies underscore the tangible benefits of chatbot-based grievance systems in diverse industries. By automating routine tasks, categorizing grievances, and providing swift resolutions, these systems contribute to streamlined customer support processes, reduced resolution times, and heightened user satisfaction.

**CHAPTER 3**

**PROJECT DESCRIPTION**

**3.1 SCOPE OF THE PROJECT**

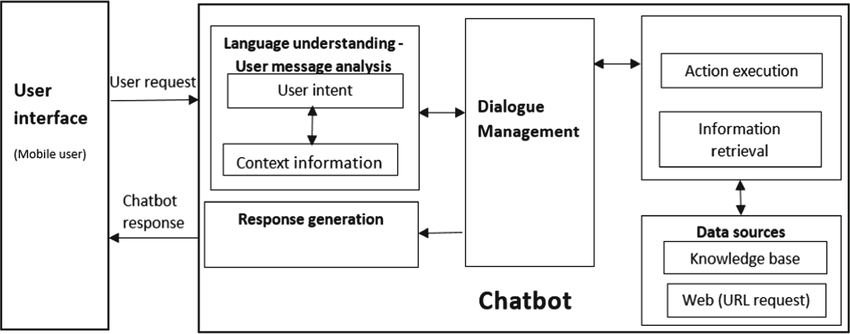
The scope of the Chatbot Grievance System project encompasses the development and implementation of an intelligent and automated platform designed to effectively handle user grievances in a diverse range of domains. The project will focus on creating a chatbot equipped with advanced natural language processing (NLP) capabilities and machine learning algorithms to classify and prioritize user-generated complaints. The system's scope extends to integrating seamlessly with existing customer support or grievance resolution frameworks, ensuring compatibility and a cohesive workflow. The classification process will cover a spectrum of grievance categories, from simple inquiries to more complex issues, enabling the chatbot to autonomously resolve routine matters and escalate intricate cases to human agents when necessary.

Data collection and preprocessing form an integral part of the project's scope, involving the compilation of a diverse and labeled dataset to train the chatbot effectively. The machine learning algorithms employed will be tailored to text classification, ensuring the system's ability to adapt and learn from evolving patterns in user grievances over time. Additionally, the project encompasses the design of a user-friendly interface to facilitate seamless interaction between users and the chatbot, promoting a positive user experience throughout the grievance resolution process.

Ethical considerations, such as transparency in decision-making, bias mitigation, and user privacy, are within the purview of the project's scope. The system will be designed to align with ethical standards, ensuring fair and unbiased treatment of user grievances. Usability testing and feedback collection will contribute to the iterative refinement of the chatbot's performance, addressing potential shortcomings and enhancing its overall effectiveness. The scope extends to the potential integration of adaptive learning mechanisms, allowing the chatbot to dynamically adjust its classification strategies based on evolving linguistic patterns and user behavior. In summary, the project's scope encompasses the comprehensive development of a Chatbot Grievance System, from data collection and model training to interface design and ethical considerations, with the overarching goal of significantly improving the efficiency and user satisfaction in the grievance resolution process.

**3.2 DATA COLLECTION AND SOURCES**

The data collection process for the Chatbot Grievance System involves gathering a diverse and representative dataset of user-generated grievances to train the chatbot effectively. Sources of data may include historical customer support interactions, online complaint forums, and records of previous grievance resolutions. These datasets should encompass a variety of grievance categories, allowing the chatbot to learn and distinguish between different types of user complaints. Data may be collected in the form of text entries, capturing the natural language expressions commonly used by users when expressing their grievances. Additionally, user feedback on past resolutions and interactions can contribute to the creation of a more nuanced and context-aware chatbot. Ethical considerations will be paramount during the data collection process to ensure user privacy and confidentiality. The comprehensive dataset, compiled from various sources, will serve as the foundation for training the machine learning algorithms, enabling the chatbot to understand, categorize, and respond to user grievances with accuracy and relevance.



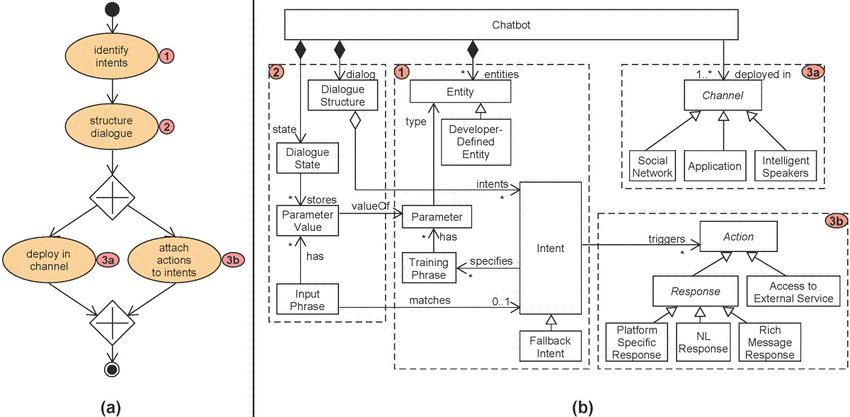
**Figure 3.1 Data Collection process**

**3.3 METHODOLOGY**

The methodology for the Chatbot Grievance System project involves a systematic and phased approach to ensure the successful development and deployment of an efficient and intelligent grievance handling solution. The project will commence with an in-depth analysis of existing grievance resolution processes, both manual and automated, to identify pain points, challenges, and opportunities for improvement. Following this, the project will focus on data collection, involving the compilation of a diverse and labeled dataset that encompasses a wide array of user grievances across different domains. The heart of the methodology lies in the application of natural language processing (NLP) techniques and machine learning algorithms for text classification. The selected algorithms will be trained on the dataset, leveraging supervised learning to enable the chatbot to autonomously categorize user-generated complaints. The training phase will involve iterative refinement to enhance the model's accuracy, efficiency, and adaptability to evolving patterns in user language. Simultaneously, the project will address the user interface design, ensuring a user-friendly and intuitive interaction between users and the chatbot. Usability testing and feedback collection will be integral to refining the interface and optimizing the overall user experience. Ethical considerations will be woven throughout the methodology, incorporating transparency in decision-making, bias mitigation strategies, and robust user privacy protection measures. The system will undergo rigorous testing, evaluating its performance across various grievance categories, and benchmarking against predefined metrics such as accuracy, precision, recall, and user satisfaction.

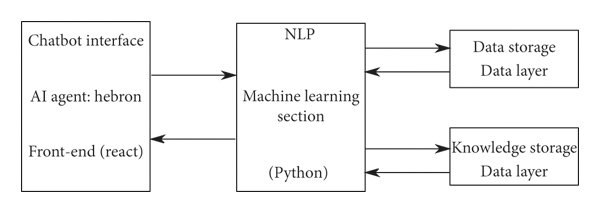
The project's dynamic learning aspect will involve continuous monitoring of the chatbot's performance in real-world scenarios. Adaptive mechanisms will be integrated to allow the system to learn and adjust its classification strategies based on ongoing user interactions and emerging linguistic patterns.

Lastly, the methodology will culminate in the integration of the chatbot into existing customer support or grievance resolution frameworks, ensuring a seamless and cohesive workflow. Through this comprehensive methodology, the Chatbot Grievance System project aims to deliver an innovative, adaptive, and user-centric solution to streamline the grievance resolution process across diverse domains.

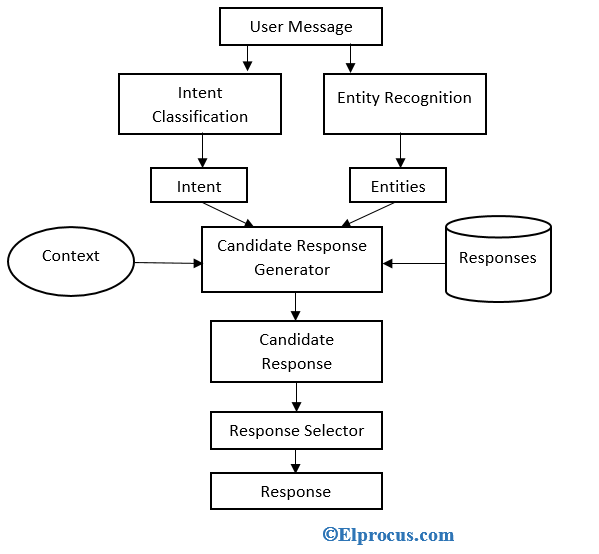


# Figure 3.2 Methodology process and Structural diagram

**3.4 BLOCK DIAGRAM**



**Figure 3.3 Structural flow of chatbot Agorithm**



**Figure 3.4** **[Chatbot : Architecture, Applications and Design Process](https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.elprocus.com%2Fchatbot-design-process-and-its-architecture%2F&psig=AOvVaw2H8WZ2YRkT-HPl5na2ax8I&ust=1700299870323000&source=images&cd=vfe&opi=89978449&ved=0CBIQjhxqFwoTCOi8ycvdyoIDFQAAAAAdAAAAABAH" \t "_blank)**

**3.5 TRAINING AND VALIDATION**

Training and validation are crucial phases in the development of a chatbot for grievance and redressal systems, as they directly impact the bot's effectiveness and user satisfaction. The training process involves preparing the chatbot to understand and respond appropriately to user queries, while validation ensures that the bot performs accurately and reliably in real-world scenarios.

During the training phase, the chatbot is exposed to a diverse dataset that includes examples of user queries and corresponding correct responses. Natural Language Processing (NLP) techniques are employed to teach the chatbot to recognize patterns, understand context, and extract meaningful information from user inputs. This training dataset should encompass a wide range of possible grievances and their resolutions, reflecting the diversity of user concerns. Supervised learning methods are commonly used, where the chatbot learns by generalizing from labeled examples provided during training.

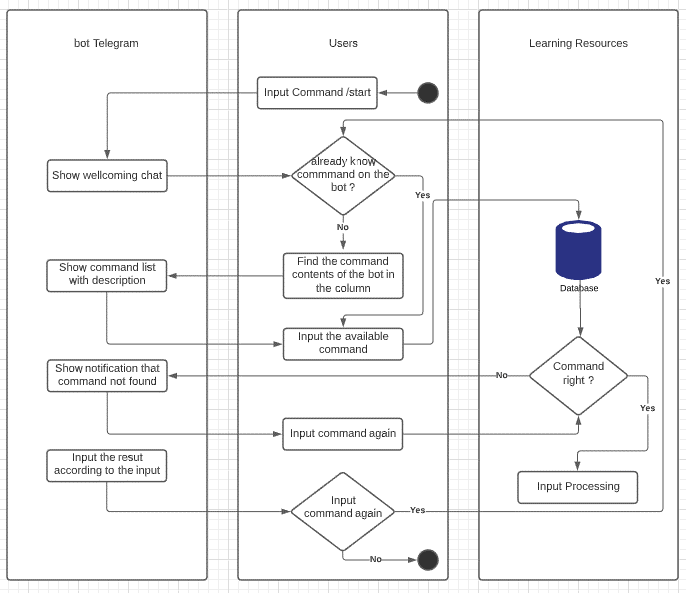
Moreover, the training process involves fine-tuning the chatbot based on continuous feedback. This feedback loop helps the chatbot improve its understanding of nuanced language, colloquial expressions, and evolving user concerns. Regular updates to the training dataset and model parameters enable the chatbot to adapt to changing user behaviors and emerging issues.

Validation is the process of assessing the chatbot's performance to ensure it meets predefined criteria and effectively addresses user grievances. This phase involves rigorous testing under various conditions to identify potential weaknesses and areas for improvement. Different types of validation methods, such as cross-validation and holdout validation, are employed to evaluate the chatbot's generalization capabilities.

During validation, the chatbot is exposed to a set of data that it has not seen during the training phase. This allows developers to assess the bot's ability to handle new, unseen queries and ensures that it doesn't overfit to the training data. Performance metrics, such as accuracy, precision, recall, and F1 score, are measured to quantify the bot's effectiveness.

User acceptance testing is also a critical component of validation, involving real users interacting with the chatbot. This provides valuable insights into how well the chatbot meets user expectations and whether it effectively resolves grievances. User feedback collected during this phase is used to make further improvements, refine the training dataset, and enhance the chatbot's overall performance.

In summary, the training and validation phases are iterative processes that involve continuous refinement of the chatbot's capabilities. Regular updates, feedback mechanisms, and a commitment to ongoing improvement are essential to ensure that the chatbot remains responsive, accurate, and capable of addressing a wide range of user grievances

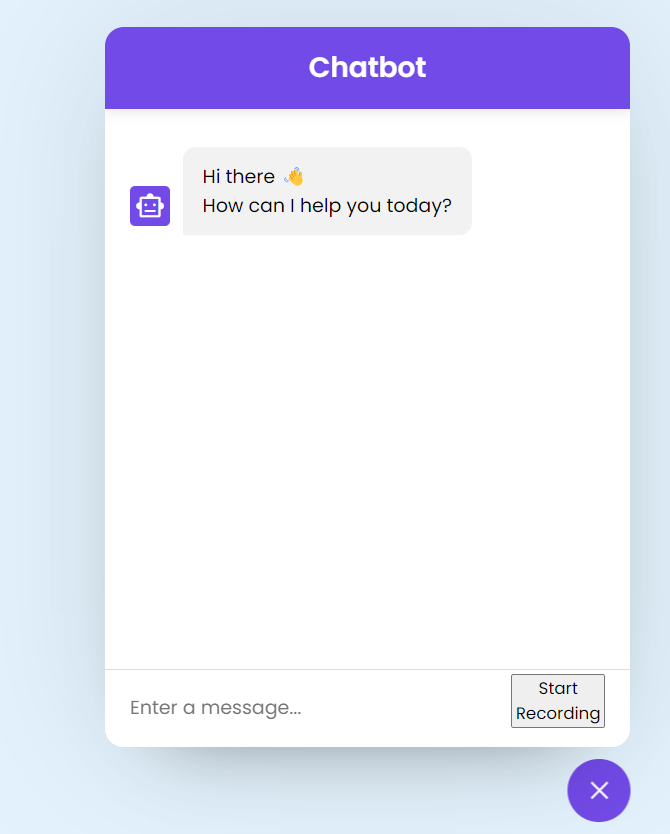


**Figure 3.5**  **[Training and Validation Process](https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.elprocus.com%2Fchatbot-design-process-and-its-architecture%2F&psig=AOvVaw2H8WZ2YRkT-HPl5na2ax8I&ust=1700299870323000&source=images&cd=vfe&opi=89978449&ved=0CBIQjhxqFwoTCOi8ycvdyoIDFQAAAAAdAAAAABAH" \t "_blank)**

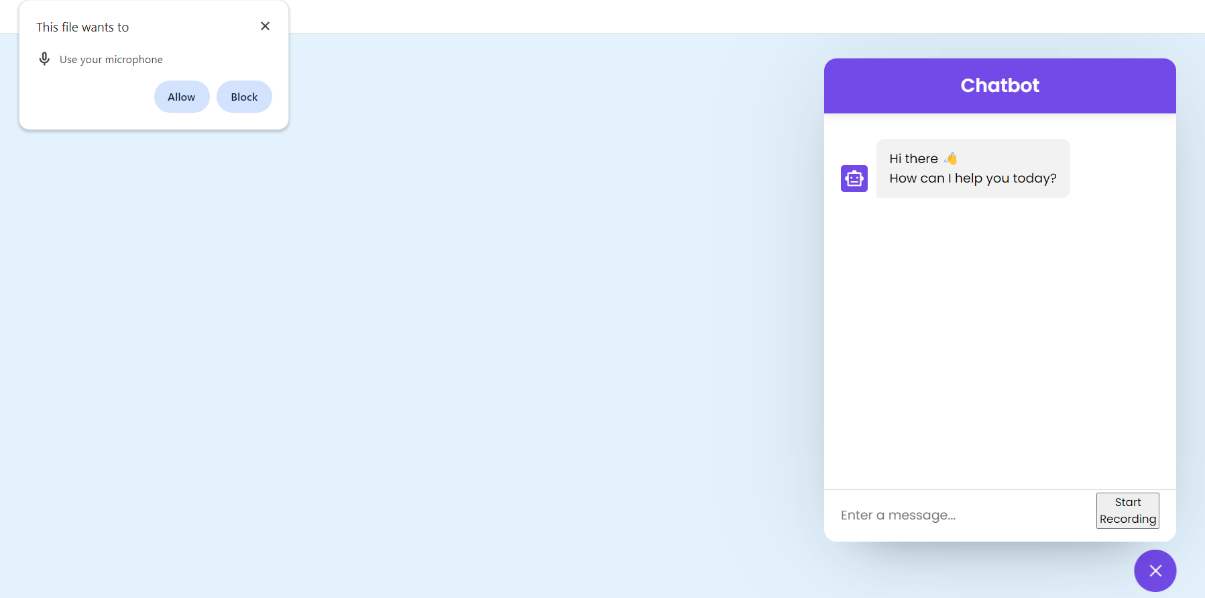
**CHAPTER 4**

**RESULTS**

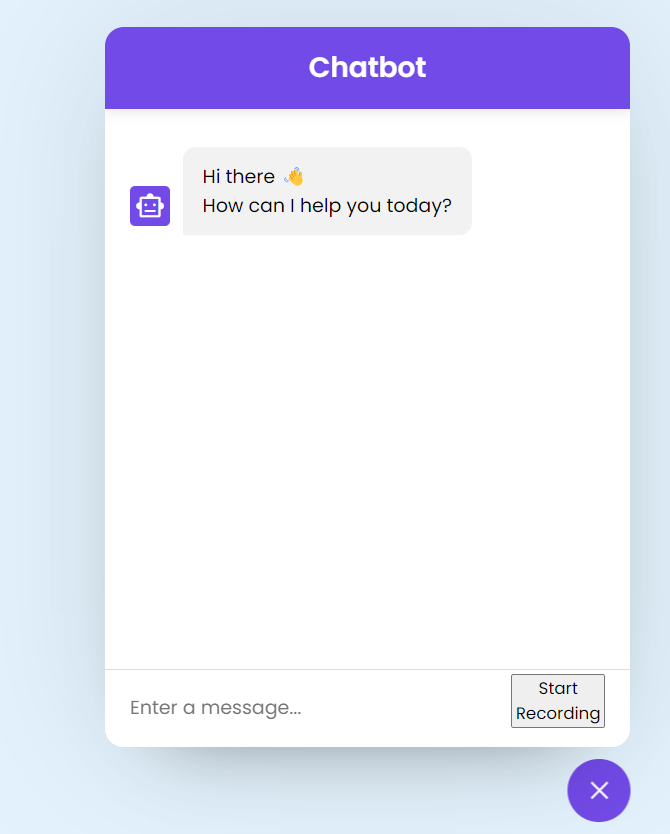
**4.1 CHARACTERIZATION RESULTS**

****

# Figure 4.1 Outline of chatbot

****

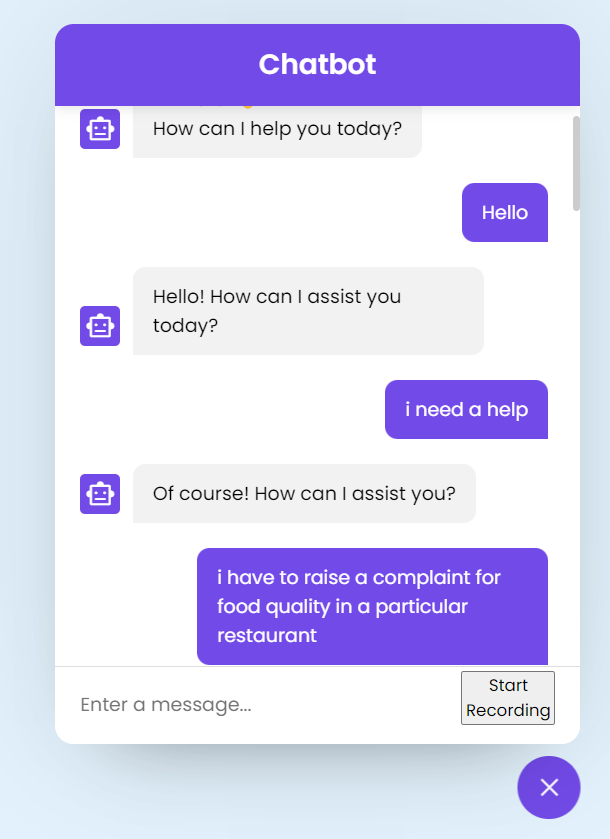
# Figure 4.2 Voice Recognition Interaction

****

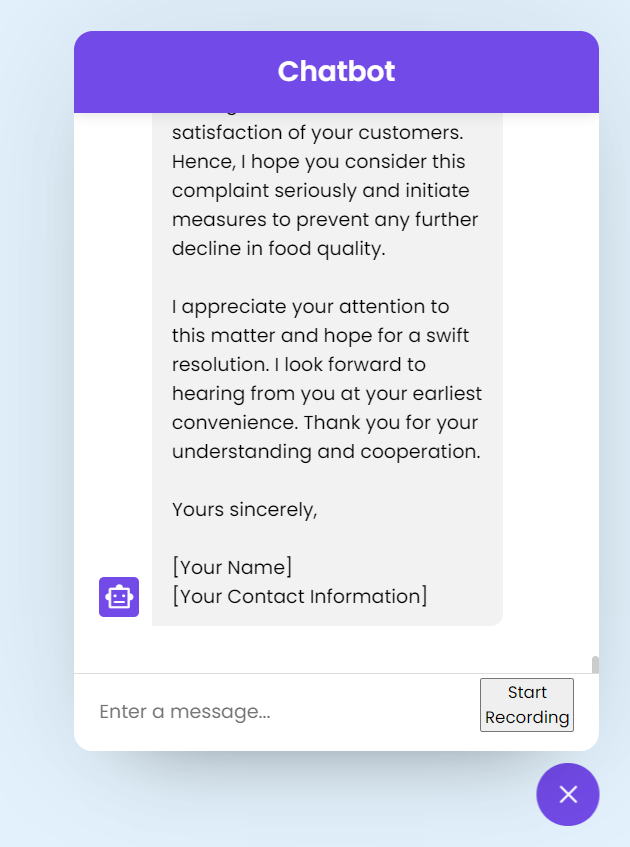
****

# Figure 4.3 Recording is enabled

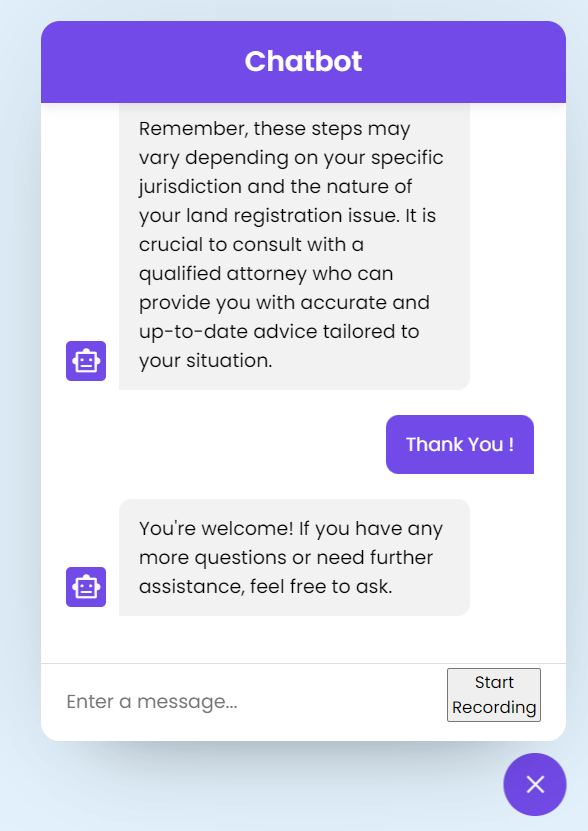
**4.2 MODEL PERFORMANCE OBSERVATIONS**

****

# Figure 4.4 User Interaction with Microphone

****

# Figure 4.5 Chat-Bot Interaction with audible mode

****

# Figure 4.6 Chat-Bot Responsive OpenAI Algorithm

**CHAPTER 5**

**CONCLUSION**

In conclusion, the development and implementation of a chatbot for grievance and redressal systems mark a significant advancement in enhancing user interaction and problem resolution. By leveraging Natural Language Processing (NLP) and machine learning, these chatbots streamline the grievance handling process, offering users a user-friendly and accessible platform to express their concerns. The round-the-clock availability ensures prompt query resolution, while the consistency and objectivity of responses contribute to a standardized service experience. The continuous improvement loop, driven by training, validation, and user feedback, allows the chatbot system to adapt to evolving user needs. As a cost-effective solution, it automates routine queries, enabling human resources to focus on more complex cases. Moreover, prioritizing security and privacy safeguards user information and instills trust in the system. Overall, a well-implemented chatbot for grievance and redressal systems is not just a technological innovation but a strategic investment in improving organizational efficiency, user satisfaction, and the overall quality of service.

**CHAPTER 6**

**APPENDIX**

**CODE (HTML) :**

<!DOCTYPE html>

<!-- Coding By CodingNepal - www.codingnepalweb.com -->

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title>Chatbot in JavaScript | CodingNepal</title>

<link rel="stylesheet" href="style.css">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!-- Google Fonts Link For Icons -->

<link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz,wght,FILL,GRAD@48,400,0,0" />

<link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Material+Symbols+Rounded:opsz,wght,FILL,GRAD@48,400,1,0" />

<script src="script.js" defer></script>

</head>

<body>

<button class="chatbot-toggler">

<span class="material-symbols-rounded">mode\_comment</span>

<span class="material-symbols-outlined">close</span>

</button>

<div class="chatbot">

<header>

<h2>Chatbot</h2>

<span class="close-btn material-symbols-outlined">close</span>

</header>

<ul class="chatbox">

<li class="chat incoming">

<span class="material-symbols-outlined">smart\_toy</span>

<p>Hi there 👋<br>How can I help you today?</p>

</li>

</ul>

<div class="chat-input">

<textarea placeholder="Enter a message..." spellcheck="false" required></textarea>

<span id="send-btn" class="material-symbols-rounded">send</span>

<div id="user-input">

<button class="voice-btn" onclick="startRecording()">Start Recording</button>

</div>

</div>

</div>

</body>

</html>

**CODE (CSS) :**

/\* Import Google font - Poppins \*/

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600&display=swap');

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: "Poppins", sans-serif;

}

body {

background: #E3F2FD;

}

.chatbot-toggler {

position: fixed;

bottom: 30px;

right: 35px;

outline: none;

border: none;

height: 50px;

width: 50px;

display: flex;

cursor: pointer;

align-items: center;

justify-content: center;

border-radius: 50%;

background: #724ae8;

transition: all 0.2s ease;

}

body.show-chatbot .chatbot-toggler {

transform: rotate(90deg);

}

.chatbot-toggler span {

color: #fff;

position: absolute;

}

.chatbot-toggler span:last-child,

body.show-chatbot .chatbot-toggler span:first-child {

opacity: 0;

}

body.show-chatbot .chatbot-toggler span:last-child {

opacity: 1;

}

.chatbot {

position: fixed;

right: 35px;

bottom: 90px;

width: 420px;

background: #fff;

border-radius: 15px;

overflow: hidden;

opacity: 0;

pointer-events: none;

transform: scale(0.5);

transform-origin: bottom right;

box-shadow: 0 0 128px 0 rgba(0,0,0,0.1),

0 32px 64px -48px rgba(0,0,0,0.5);

transition: all 0.1s ease;

}

body.show-chatbot .chatbot {

opacity: 1;

pointer-events: auto;

transform: scale(1);

}

.chatbot header {

padding: 16px 0;

position: relative;

text-align: center;

color: #fff;

background: #724ae8;

box-shadow: 0 2px 10px rgba(0,0,0,0.1);

}

.chatbot header span {

position: absolute;

right: 15px;

top: 50%;

display: none;

cursor: pointer;

transform: translateY(-50%);

}

header h2 {

font-size: 1.4rem;

}

.chatbot .chatbox {

overflow-y: auto;

height: 510px;

padding: 30px 20px 100px;

}

.chatbot :where(.chatbox, textarea)::-webkit-scrollbar {

width: 6px;

}

.chatbot :where(.chatbox, textarea)::-webkit-scrollbar-track {

background: #fff;

border-radius: 25px;

}

.chatbot :where(.chatbox, textarea)::-webkit-scrollbar-thumb {

background: #ccc;

border-radius: 25px;

}

.chatbox .chat {

display: flex;

list-style: none;

}

.chatbox .outgoing {

margin: 20px 0;

justify-content: flex-end;

}

.chatbox .incoming span {

width: 32px;

height: 32px;

color: #fff;

cursor: default;

text-align: center;

line-height: 32px;

align-self: flex-end;

background: #724ae8;

border-radius: 4px;

margin: 0 10px 7px 0;

}

.chatbox .chat p {

white-space: pre-wrap;

padding: 12px 16px;

border-radius: 10px 10px 0 10px;

max-width: 75%;

color: #fff;

font-size: 0.95rem;

background: #724ae8;

}

.chatbox .incoming p {

border-radius: 10px 10px 10px 0;

}

.chatbox .chat p.error {

color: #721c24;

background: #f8d7da;

}

.chatbox .incoming p {

color: #000;

background: #f2f2f2;

}

.chatbot .chat-input {

display: flex;

gap: 5px;

position: absolute;

bottom: 0;

width: 100%;

background: #fff;

padding: 3px 20px;

border-top: 1px solid #ddd;

}

.chat-input textarea {

height: 55px;

width: 100%;

border: none;

outline: none;

resize: none;

max-height: 180px;

padding: 15px 15px 15px 0;

font-size: 0.95rem;

}

.chat-input span {

align-self: flex-end;

color: #724ae8;

cursor: pointer;

height: 55px;

display: flex;

align-items: center;

visibility: hidden;

font-size: 1.35rem;

}

.chat-input textarea:valid ~ span {

visibility: visible;

}

@media (max-width: 490px) {

.chatbot-toggler {

right: 20px;

bottom: 20px;

}

.chatbot {

right: 0;

bottom: 0;

height: 100%;

border-radius: 0;

width: 100%;

}

.chatbot .chatbox {

height: 90%;

padding: 25px 15px 100px;

}

.chatbot .chat-input {

padding: 5px 15px;

}

.chatbot header span {

display: block;

}

}

**CODE (JS) :**

// Create a speech recognition object

const recognition = new (window.SpeechRecognition || window.webkitSpeechRecognition)();

recognition.continuous = false;

const chatbotToggler = document.querySelector(".chatbot-toggler");

const closeBtn = document.querySelector(".close-btn");

const chatbox = document.querySelector(".chatbox");

const chatInput = document.querySelector(".chat-input textarea");

const sendChatBtn = document.querySelector(".chat-input span");

const voiceBtn = document.querySelector(".voice-btn");

let userMessage = null; // Variable to store user's message

const API\_KEY = "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"; // Paste your API key here

const inputInitHeight = chatInput.scrollHeight;

// Function to start microphone recording

function startRecording() {

recognition.start();

console.log('Listening...');

}

// Event listener for speech recognition end

recognition.onend = () => {

// Restart recognition to keep listening

recognition.start();

console.log('Listening...');

};

// Event listener for speech recognition result

recognition.onresult = (event) => {

const transcript = event.results[0][0].transcript;

chatInput.value = transcript;

// Handle the speech result, e.g., sendMicrophoneQuery();

};

// Event listener for speech recognition result

recognition.onresult = (event) => {

const transcript = event.results[0][0].transcript;

chatInput.value = transcript;

};

// Event listener for speech recognition error

recognition.onerror = (event) => {

console.error('Speech recognition error:', event.error);

const errorMessage = "Speech recognition error. Please try again.";

chatbox.appendChild(createChatLi(errorMessage, "incoming"));

chatbox.scrollTo(0, chatbox.scrollHeight);

};

// Function to send microphone query directly

function sendMicrophoneQuery() {

startRecording();

}

// Create a chat <li> element with passed message and className

const createChatLi = (message, className) => {

const chatLi = document.createElement("li");

chatLi.classList.add("chat", `${className}`);

let chatContent = className === "outgoing" ? `<p></p>` : `<span class="material-symbols-outlined">smart\_toy</span><p></p>`;

chatLi.innerHTML = chatContent;

chatLi.querySelector("p").textContent = message;

return chatLi;

}

// Generate response from the chatbot and convert it to audio

const generateResponse = (chatElement) => {

const API\_URL = "https://api.openai.com/v1/chat/completions";

const messageElement = chatElement.querySelector("p");

const requestOptions = {

method: "POST",

headers: {

"Content-Type": "application/json",

"Authorization": `Bearer ${API\_KEY}`

},

body: JSON.stringify({

model: "gpt-3.5-turbo",

messages: [{ role: "user", content: userMessage }],

})

}

fetch(API\_URL, requestOptions)

.then(res => res.json())

.then(data => {

const responseText = data.choices[0].message.content.trim();

// Display the text response in the chatbox

messageElement.textContent = responseText;

// Convert the text response to speech

speakResponse(responseText);

})

.catch(() => {

messageElement.classList.add("error");

messageElement.textContent = "Oops! Something went wrong. Please try again.";

})

.finally(() => chatbox.scrollTo(0, chatbox.scrollHeight));

}

// Function to speak the text response

const speakResponse = (text) => {

const synth = window.speechSynthesis;

const utterance = new SpeechSynthesisUtterance(text);

// Optionally, you can configure additional properties of the utterance

// utterance.volume = 1; // Volume (0 to 1)

// utterance.rate = 1; // Speaking rate (0.1 to 10)

// utterance.pitch = 1; // Pitch (0 to 2)

// Speak the text

synth.speak(utterance);

}

// Handle user input and initiate chat

const handleChat = () => {

userMessage = chatInput.value.trim();

if (!userMessage) return;

chatInput.value = "";

chatInput.style.height = `${inputInitHeight}px`;

chatbox.appendChild(createChatLi(userMessage, "outgoing"));

chatbox.scrollTo(0, chatbox.scrollHeight);

setTimeout(() => {

const incomingChatLi = createChatLi("Thinking...", "incoming");

chatbox.appendChild(incomingChatLi);

chatbox.scrollTo(0, chatbox.scrollHeight);

generateResponse(incomingChatLi);

}, 600);

}

// Event listeners

chatInput.addEventListener("input", () => {

chatInput.style.height = `${inputInitHeight}px`;

chatInput.style.height = `${chatInput.scrollHeight}px`;

});

chatInput.addEventListener("keydown", (e) => {

if (e.key === "Enter" && !e.shiftKey && window.innerWidth > 800) {

e.preventDefault();

handleChat();

}

});

if ('SpeechRecognition' in window || 'webkitSpeechRecognition' in window) {

const voiceBtn = document.querySelector(".voice-btn");

voiceBtn.style.display = 'block';

voiceBtn.addEventListener('click', sendMicrophoneQuery); // Change to sendMicrophoneQuery

} else {

alert('Speech recognition is not supported in this browser.');

}

sendChatBtn.addEventListener("click", handleChat);

closeBtn.addEventListener("click", () => document.body.classList.remove("show-chatbot"));

chatbotToggler.addEventListener("click", () => document.body.classList.toggle("show-chatbot"));

**REFERENCES**

[1] (Member, Ieee), Guilherme Guy De Andrade 1,2022.

[2] Vic (J.R.) Winkler, "Securing the Cloud: Cloud Computer Security Techniques and

Tactics", Syngress ,2022.

[3] Wordstream.com. 2020. 10 Of The Most Innovative Chat bots On The Web. [online]

Available at: <https://www.wordstream.com/blog/ws/2017/10/04/chat bots>

[Accessed 11 September 2022

[4] KBot: A Knowledge Graph-Based Chat bot for Natural Language Understanding Over

Linked Data-2020 AddiAit-Mlouk and Lili Jiang.

[5] E. Adamopoulou and L. Moussiades, "An overview of chat bot technology" in Artificial

Intelligence Applications and Innovations, Cham, Switzerland:Springer, pp. 373-383,

2020.

[6] J. Jiang and N. Ahuja, "Response quality in human-chat bot collaborative systems",

Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., pp. 1545-1548, Jul. 2020.

[7] J. Feine, S. Morana and A. Maedche, "A chat bot response generation system", Proc.

Conf. Mensch Comput. (MuC), pp. 333-341, Sep. 2020.ISSN: 2583-7346

DOI: https://doi-ds.org/doilink/06.2023-19862138/0023062023, Volume-1, Issue-3, pp. 65-71

International Journal of Advances in Engineering Architecture Science and Technology

www.ijaeast.com The Palm Series 71

[8] K. Deepika, V. Tilekya, J. Mamatha and T. Subetha, "Jollity chat bot—A contextual AI

assistant", Proc. 3rd Int. Conf. Smart Syst. Inventive Technol. (ICSSIT), pp. 1196-1200,

Aug. 2020.

[9] M. Luca, A. Montresor, C. Caprini and D. Miorandi, "An architecture-independent data

model for managing information generated by human-chat bot interactions", Proc. 8th

Int. Conf. Model-Driven Eng. Softw. Develop., pp. 344-351, 2020.

[10] K. Kvale, O. A. Sell, S. Hodnebrog and A. Følstad, "Improving conversations: Lessons

learnt from manual analysis of chat bot dialogues", Proc. Int. Workshop Chat bot Res.

Design, pp. 187-200, 2019.