Cloud Counselage

GIFT A CAREER FOUNDATION

SYSTEM DESIGN SPECIFICATION

Artificial Intelligence

AI Chatbot

Created By:	Snehil Sharma	**Approved By:**	Eesha Kiran Patro
Created On:	26-Aug-2023	**Approved On:**	26-Aug-2023

Page left blank intentionally

INDEX

1 PURPOSE

This document is created based on the requirement specification document. The purpose of this System Design Specification (SDS) Document is to break down the project into components to describe in detail what the purpose of each component is and how it will be implemented. The SDS will also serve as a tool for verification and validation of the final product.

2 PROJECT SCOPE

The scope of the AI Chatbot includes its distinct features, its benefits, and its limitations. The system's distinct features allow it to build a unified Chatbot that is linked to Facebook, Instagram, LinkedIn, Twitter, YouTube, Email, and the Website as sources where the users can communicate to Cloud Counselage Pvt. Ltd. and gift a career foundation for our Industry Academia Community Program. FAQs & their answers are fed into the chatbot backend. Using Python, Flask, PyPDF2, HTML, CSS, JavaScript, jinja2 template, OpenAI API, and ChimeraGPT API, the system enables the user to efficiently interact with the chatbot through both speech-driven and text-based conversations, addressing queries and providing instant support for the Industry Academia Community Program.

3 SYSTEM OVERVIEW

This section will provide an outline of the various components and subsystems of AI Chatbot.

The Unified AI Chatbot designed for Cloud Counselage's Industry Academia Community (IAC) Program is a sophisticated system that seamlessly integrates with multiple platforms to provide users with comprehensive program-related information and support. The system is comprised of various components and subsystems that collectively enable efficient communication, engagement, and interaction between the chatbot and its users.

1. User Interface:

- The User Interface (UI) serves as the front-end of the chatbot, enabling users to input queries and interact with the system.
- The UI includes both text-based input fields and speech recognition functionality for versatile communication.

2. Platform Integration Subsystem:

- This subsystem facilitates integration with various platforms, including Facebook, Instagram, LinkedIn, Twitter, YouTube, Email, and the Website.
- Each platform is linked to the chatbot, allowing users to access the chatbot's services through their preferred channels.

3. Natural Language Processing (NLP) Subsystem:

- The NLP subsystem is responsible for processing user inputs, extracting intent, and generating contextually relevant responses.
- It utilizes the OpenAI GPT-3.5 Turbo model for accurate and coherent text-based responses.

4. Speech Recognition Subsystem:

- This subsystem converts user speech inputs into text using the implemented speech recognition mechanism.
- It enables users to interact with the chatbot using spoken language in addition to text.

5. FAQ Knowledge Base Subsystem:

- The system maintains a repository of frequently asked questions (FAQs) and their corresponding answers related to the IAC Program.
- The knowledge base is utilized by the NLP subsystem to provide accurate responses to user queries.

6. Response Generation Subsystem:

- The response generation subsystem processes user queries, consults the FAQ knowledge base, and generates appropriate responses.
- It ensures contextually accurate and informative answers are provided to users.

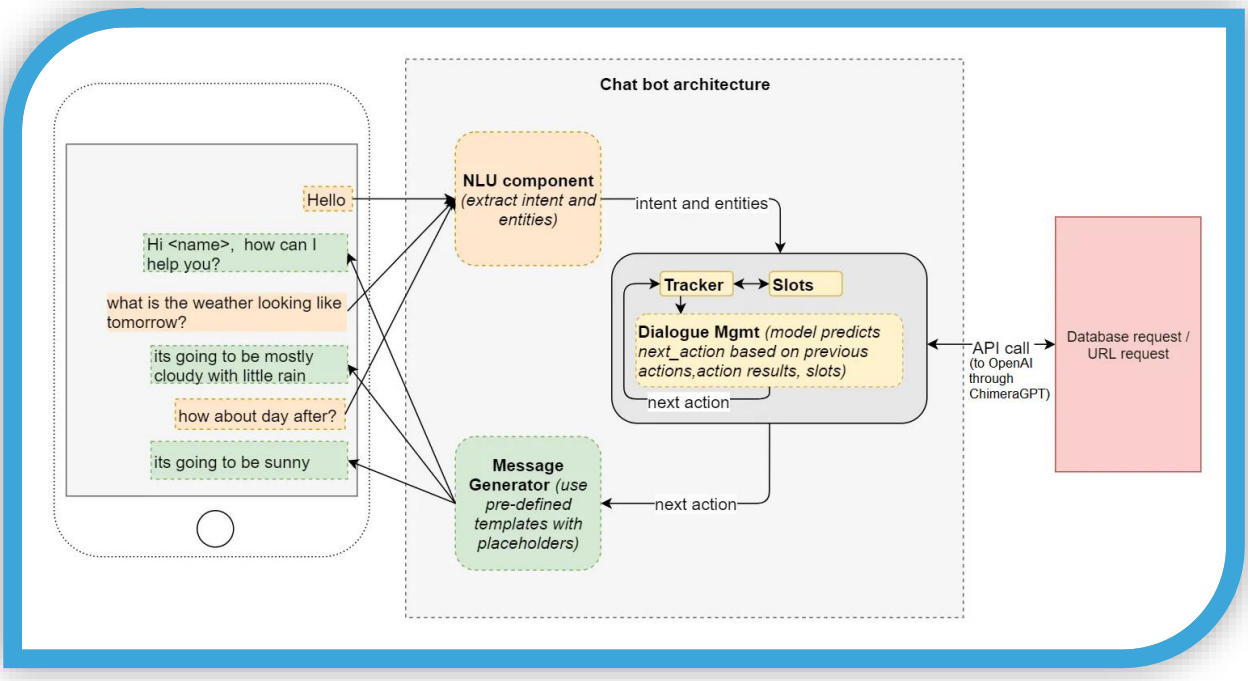
7. Multi-Platform Output Subsystem:

- This subsystem manages the distribution of responses generated by the chatbot to the respective platforms.
- It formats and delivers responses tailored to the unique interfaces and requirements of each platform.

9. User Engagement Subsystem:

- The user engagement subsystem employs interactive prompts and personalized responses to maintain user interest and engagement.
- It enhances user satisfaction and encourages continued interaction.

The Unified AI Chatbot's various components and subsystems collaborate to create a seamless and user-centric experience for Cloud Counselage's Industry Academia Community Program participants. By leveraging AI, platform integration, and user-friendly interfaces, the system effectively addresses user queries, disseminates program information, and fosters engagement within the IAC community.



4 DESIGN CONSIDERATIONS

This section describes requirements, assumptions and dependencies to be addressed to devise a complete design solution.

4.1 Requirements

Hardware Requirements

1. Computer: Modern laptop/desktop with multi-core processor (Intel i5 or higher) and at least 8GB RAM.
2. Storage: Minimum 256GB SSD for development files and datasets.
3. GPU: Optional but recommended for faster training (NVIDIA GeForce GTX or RTX series).
4. Internet Connection: Stable and reasonably fast for downloading libraries and testing.
5. Microphone: Required for speech recognition functionality.
6. Mobile Devices: Consider for mobile deployment.

Software Requirements

1. Operating System: Windows, macOS, or Linux (Ubuntu, CentOS, etc.).
2. Python: Version 3.9 or higher.
3. Development Environment: Text editor or IDE (e.g., Visual Studio Code).
4. NLP Libraries: NLTK or SpaCy.
5. ML and DL Libraries: TensorFlow, PyTorch, or Scikit-learn.
6. Speech Recognition: SpeechRecognition or Google Speech-to-Text API.
7. Web Framework: Flask.
8. Social Media: Facebook, Instagram, Twitter, LinkedIn, Youtube, Email.
9. Version Control: Git (Github).
10. Deployment Platform: Cloud platform (Google Cloud Platform).
11. Generative AI API: OpenAI, ChimeraGPT

4.2 Assumptions

1. Sufficient resources and expertise for chatbot development.
2. Availability of FAQs and their accurate answers.
3. Social media APIs remain accessible and functional.
4. Adequate user engagement with the chatbot.
5. Speech recognition technology works effectively.

4.3 Dependencies

1. OpenAI API
2. ChameraGPT API
3. Google Cloud Platform (GCP)

5 SYSTEM ARCHITECTURE

The software system architecture refers to the logical organization of a distributed system into software components. It defines how components of a software system are assembled, their relationship and communication between them. It serves as a blueprint for software application and development basis for developer team. An effective architecture serves as the conceptual glue that holds every phase of the project together for all of its stakeholders, enabling agility, time and cost savings, and early identification of design risks.

The Software architecture:

- Defines structure of a system
- Defines behaviour of a system
- Defines component relationship
- Defines communication structure
- Balances stakeholder's needs
- Influences team structure
- Focuses on significant elements
- Captures early design decisions

Below some important characteristics which are commonly considered are explained.

Operational Architecture Characteristics:

- Availability
- Performance
- Reliability
- Low fault tolerance
- Scalability

Structural Architecture Characteristics:

- Configurability
- Extensibility
- Supportability
- Portability
- Maintainability

Cross-Cutting Architecture Characteristics:

- Accessibility
- Security
- Usability
- Privacy
- Feasibility

5.1 Architectural Strategies

The architectural design of the Unified AI Chatbot system for Cloud Counselage's Industry Academia Community (IAC) Program incorporates a set of major components that collectively enable seamless interaction, integration with multiple platforms, and efficient response generation. The following architectural strategies outline the key components of the system:

1. User Interface (UI) Layer:

- Provides an intuitive and user-friendly interface for users to interact with the chatbot.
- Includes both text-based input fields and speech recognition functionality for versatile communication.

2. Platform Integration Layer:

- Facilitates integration with various platforms, such as Facebook, Instagram, LinkedIn, Twitter, YouTube, Email, and the Website.
- Enables users to access the chatbot's services through their preferred channels.

3. Natural Language Processing (NLP) Layer:

- Processes user inputs and extracts intent using the OpenAI GPT-3.5 Turbo model.
- Generates contextually relevant responses by leveraging the language model's capabilities.

4. Speech Recognition Layer:

- Converts user speech inputs into text using the implemented speech recognition mechanism.
- Enhances user engagement by enabling voice-based interactions.

5. Knowledge Base Layer:

- Maintains a repository of FAQs and their corresponding answers related to the IAC Program.
- Supports accurate response generation by providing relevant information.

6. Response Generation Layer:

- Processes user queries, consults the knowledge base, and generates appropriate responses.
- Ensures contextually accurate and informative answers are provided to users.

7. Multi-Platform Output Layer:

- Manages the distribution of responses to the respective platforms.
- Adapts responses to suit the unique interfaces and requirements of each platform.

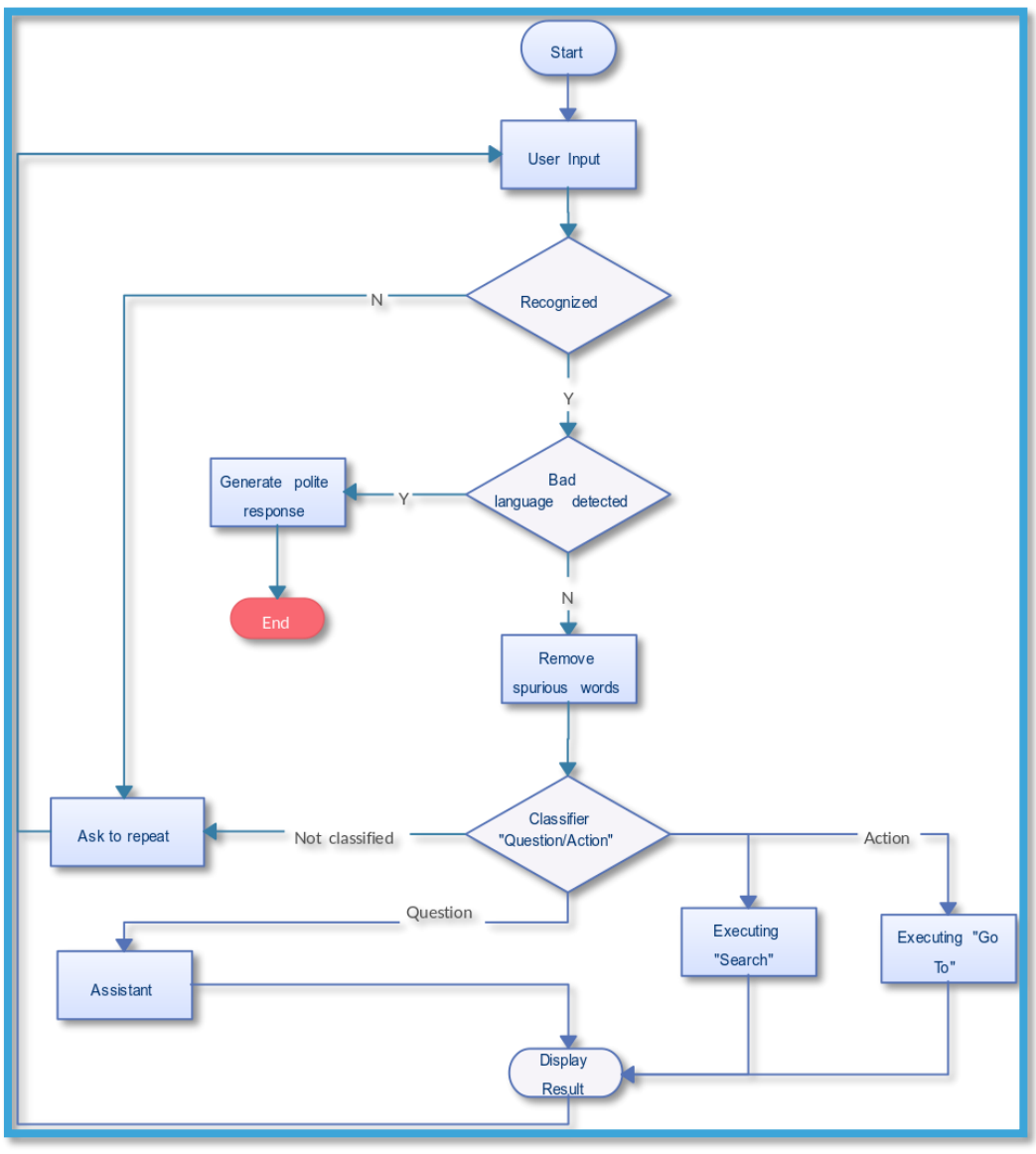
9. User Engagement Layer:

- Utilizes interactive prompts and personalized responses to maintain user interest.
- Enhances user satisfaction and encourages continued interaction.

10. System Management Layer:

- Monitors system performance, user interactions, and platform integrations.
- Supports debugging, error handling, and system maintenance.

5.2 Structure & Relationships



6 DETAILED DESCRIPTION OF COMPONENTS

For detailed description of the components, please refer **Appendix A – Detailed Description of Components**

The below template will be used to specify the details of all the components

Table 1: Detailed Design Specification Template

Identification	The unique name for the component and the location of the component in the system.
Type	A module, a subprogram, a form, a data file, a control procedure, a class, etc.
Purpose	Function and performance requirements implemented by the design component, including derived requirements. Derived requirements are not explicitly stated in the SRS - but are implied or adjunct to formally stated SDS requirements.
Subordinates	The internal structure of the component, the constituents of the component, and the functional requirements satisfied by each part.
Dependencies	How the component's function and performance relate to other components. How this component is used by other components. The other components that use this component. Interaction details such as timing, interaction conditions (such as order of execution and data sharing), and responsibility for creation, duplication, use, storage, and elimination of components.
Interfaces	Detailed description of all external or internal interfaces as well as of any mechanism for communicating through messages, parameters, or common data areas. All error messages and error codes should be identified. All screen formats, interactive messages, and other user interface components (originally defined in the SRS) should be given here.
Resources	A complete description of all resources (hardware or software) external to the component but required to carry out its functions.
Processing	A full description of the functions presented in the Function subsection. Pseudocode can be used to document algorithms, equations, and logic.
Data	For the data internal to the component, describes the representation method, initial values, use, semantics, and format.

7 INTEGRATIONS

1. OpenAI & Chimera GPT API Integration:

- Utilization of the OpenAI GPT-3.5 Turbo model for natural language processing and response generation.
- Integration of the OpenAI API for sending queries and receiving AI-generated responses.

2. Speech Recognition Integration:

- Implementation of speech recognition mechanisms to convert user speech into text.
- Integration of the speech recognition module within the chatbot's UI.

3. User Interface (UI) Integration:

- Development of user interface components using HTML, CSS, and JavaScript.
- Integration of the UI components with the backend to capture user inputs and display responses.

4. Multi-Platform Output Integration:

- Formatting and tailoring responses for each platform's interface requirements.
- Integration of the output module to distribute responses to respective platforms.

5. Documentation and Reporting Integration:

- Integration of reporting and logging mechanisms to track system performance and user interactions.
- Generation of reports based on user engagement, feedback, and usage statistics.

6. Google Cloud Platform (GCP) Services Integration:

- Integration with cloud hosting services to deploy the chatbot application.
- Utilization of cloud storage for storing and retrieving data, including FAQs and user interactions.

7. User Engagement Enhancement Integration:

- Implementation of interactive prompts and responses to maintain user engagement.
- Integration of engagement strategies to encourage continued interaction.

8 APPENDICES

8.1 Appendix A – Detailed Description of Components

1. User Interface (UI) Component

Identification	• **Name:** User Interface (UI) Component • **Location:** Front-end of the Unified AI Chatbot system
Type	Interface module
Purpose	• **Function Requirements:** Facilitate user interaction and input to the chatbot system. • **Performance Requirements:** Provide an intuitive and user-friendly interface for both text and speech interactions.
Subordinates	• **Text Input Field:** Allows users to type queries. • **Speech Recognition Interface:** Enables users to interact using spoken language.
Dependencies	• The UI Component's function is closely related to the NLP Component for processing user inputs and generating responses. • The UI Component is used by the Platform Integration Component to transmit user queries and receive responses. • Interaction details include the sequence of UI input and NLP processing, as well as timing considerations for speech recognition.
Interfaces	• **External Interfaces:** Platform-specific UI designs for Facebook, Instagram, LinkedIn, Twitter, YouTube, Email, and the Website. • **Internal Interfaces:** Interaction with the NLP Component to send user inputs for processing and receive generated responses.
Resources	• Hardware resources include display devices for presenting UI elements. • Software resources include HTML, CSS, JavaScript, and Web Speech API for speech recognition.
Processing	• Display UI components according to platform-specific designs. • Capture user text inputs from the input field. • Process speech inputs using the Web Speech API. • Interface with the NLP Component to send user inputs and retrieve responses.
Data	• **Text Input Field:** Uses standard HTML text input elements. • **Speech Recognition:** Utilizes the Web Speech API for capturing and processing spoken language. • **Initial Values:** Clear input fields when starting a new interaction. • **Semantics:** Text inputs are used to formulate user queries, while speech inputs are converted to text for processing. • **Format:** Text input data is in plain text format, and speech input data is in audio format.

2. Natural Language Processing (NLP) Component

Identification	• **Name:** Natural Language Processing (NLP) Module • **Location:** Integrated within the Unified AI Chatbot's backend architecture.
Type	Module
Purpose	• **Function:** The NLP Module is responsible for processing user inputs, extracting intent, and generating contextually relevant responses using the OpenAI GPT-3.5 Turbo model. • **Performance Requirements:** The NLP Module is expected to provide accurate and coherent responses to user queries while maintaining efficient processing times. It should handle a wide range of user input variations and support meaningful interactions. • **Derived Requirements:** The NLP Module's accuracy and relevance contribute to user satisfaction and the overall success of the chatbot's engagement.
Subordinates	• **Model Integration Submodule:** Responsible for integrating the OpenAI GPT-3.5 Turbo model through ChimeraGPT within the chatbot's architecture. • **Input Processing Submodule:** Handles preprocessing of user inputs, ensuring optimal formatting and compatibility with the model. • **Response Generation Submodule:** Utilizes the model's outputs to generate meaningful and relevant responses to user queries.
Dependencies	• The NLP Module's function relies on the availability of the OpenAI GPT-3.5 Turbo model and its accurate processing of text inputs. • The responses generated by the NLP Module are used by the Multi-Platform Output Component to distribute responses to respective platforms. • User interactions with the NLP Module influence the overall user engagement, shaping the success of the User Engagement Component.
Interfaces	• **External Interfaces:** The NLP Module communicates with the OpenAI GPT-3.5 Turbo model by OpenAI API through ChimeraGPT API. • **Internal Interfaces:** The NLP Module interfaces with the Input Processing Submodule to receive pre-processed user inputs. • **Error Messages:** In case of NLP-related errors or limitations, the system may generate error messages indicating the inability to provide satisfactory responses.
Resources	• **Hardware:** The NLP Module requires an internet-connected environment to interact with the OpenAI API and the deployed chatbot application. • **Software:** The OpenAI GPT-3.5 Turbo model and the OpenAI API serve as essential software resources for the NLP Module's functioning.
Processing	• The NLP Module employs the OpenAI GPT-3.5 Turbo model to process user inputs. • User inputs are pre-processed to ensure they conform to the model's input format and to optimize the generation of meaningful responses.
Data	• **Model Input:** User queries are transformed into appropriate input format for the OpenAI GPT-3.5 Turbo model. • **Model Output:** The model's generated responses are processed further to ensure coherent and contextually relevant answers. • **Semantics:** The NLP Module aims to capture the semantics and context of user queries, enabling accurate response generation.

3. Speech Recognition Component

Identification	• **Component Name:** Speech Recognition Module • **Location:** Integrated within the User Interface Layer of the Unified AI Chatbot System.
Type	Module
Purpose	• **Function:** Convert user speech inputs into text for processing by the Natural Language Processing (NLP) subsystem. • **Performance Requirements:** Achieve accurate speech-to-text conversion with minimal latency. • **Derived Requirements:** Ensure compatibility with a wide range of accents, dialects, and speech patterns.
Subordinates	• **Audio Input Interface:** Captures user speech through the device's microphone. • **Speech-to-Text Conversion Algorithm:** Converts audio input into corresponding text representations.
Dependencies	• **Function & Performance:** Accurate speech recognition is crucial for generating meaningful text inputs for subsequent processing by the NLP subsystem. • **Interaction with NLP Subsystem:** The converted text is used as input for the NLP subsystem for generating contextually accurate responses.
Interfaces	• **Audio Input Interface:** Receives audio signals from the device's microphone. • **Speech-to-Text Conversion Algorithm:** Accepts audio input and produces text output. • **NLP Subsystem Interface:** Provides the converted text to the NLP subsystem for further processing.
Resources	• **Hardware:** Device's microphone for audio input. • **Software:** Speech recognition libraries and algorithms.
Processing	• The speech recognition module captures audio input from the device's microphone. • It employs the speech-to-text conversion algorithm, which involves signal processing techniques to translate audio signals into corresponding text characters.
Data	• **Representation Method:** Audio signals are transformed into a series of text characters based on recognized phonetic patterns. • **Initial Values:** Starts the audio capture and conversion process upon user initiation. • **Use:** The converted text serves as input for the NLP subsystem. • **Semantics:** The text representation aims to preserve the content and context of the user's spoken words. • **Format:** The output text follows a structured format, enabling seamless integration with subsequent subsystems.

4. FAQ Knowledge Base Component

Identification	• **Component Name:** FAQ Knowledge Base • **Location:** Centralized repository within the system architecture.
Type	**Data File:** Manages structured storage of frequently asked questions (FAQs) and corresponding answers.
Purpose	• **Function:** To provide a comprehensive repository of FAQs related to the Industry Academia Community (IAC) Program. • **Performance Requirements:** Efficient storage, retrieval, and update of FAQs. Fast query response time.
Subordinates	• **Constituents:** Individual FAQ entries, each consisting of a question and its corresponding answer. • **Functional Requirements:** Storage, indexing, retrieval, and update mechanisms for FAQs.
Dependencies	• **Function and Performance:** Central to the system's capability to generate contextually accurate responses to user queries. • **Use by Other Components:** Utilized by the Response Generation Component to fetch relevant information for generating responses. • **Interaction Details:** Retrieval of FAQs occurs in real-time based on user queries. No creation, duplication, or storage responsibility.
Interfaces	• **External Interfaces:** None. • **Internal Interfaces:** Accessed by the Response Generation Component for information retrieval. • **Communication:** APIs or methods to query and retrieve specific FAQs based on user queries. • **Error Handling:** Appropriate error messages if an FAQ is not found.
Resources	• **Hardware:** None, as it is a data storage component. • **Software:** Storage mechanisms, databases, or structured file systems.
Processing	• Functions: Storage, indexing, retrieval, and update of FAQ entries.
Data	• **Representation:** Each FAQ entry represented as a pair of question and answer. • **Initial Values:** FAQs are initially populated during system setup or through manual entry. • **Use and Semantics:** Utilized to provide accurate and relevant answers to user queries. • **Format:** Structured format with distinct question and answer fields.

5. Response Generation Component

Identification	• **Component Name:** Response Generation • **Location:** Central component within the chatbot system architecture.
Type	Module
Purpose	The Response Generation component is responsible for creating contextually accurate and coherent responses to user queries. It transforms the processed user inputs, extracted intents, and data from the FAQ knowledge base into well-structured and informative answers that address user inquiries.
Subordinates	• **Algorithmic Response Formulation Subordinate:** Responsible for generating responses by combining NLP model outputs, FAQ knowledge, and interactive prompts. • **FAQ Data Retrieval Subordinate:** Manages the retrieval of relevant FAQ data based on user intents and queries.
Dependencies	• **Interaction with NLP Component:** Receives processed inputs from the NLP component, which include extracted intents and user context. • **Interaction with FAQ Knowledge Base Component:** Accesses the FAQ data repository to retrieve relevant information. • **Interaction with Multi-Platform Output Component:** Delivers the generated responses to be formatted for platform-specific interfaces.
Interfaces	• **External Interfaces:** None. • **Internal Interfaces:** Interfaces with the NLP component to receive processed inputs and with the FAQ Knowledge Base component to retrieve FAQ data. • **Error Messages and Codes:** Error messages related to unavailability of FAQ data, incomplete information, and any other errors encountered during response generation.
Resources	• **External Resources:** None • **Internal Resources:** Utilizes the OpenAI GPT-3.5 Turbo model and the FAQ Knowledge Base to formulate responses.
Processing	• Utilizes algorithmic logic to combine information from the NLP component, the FAQ Knowledge Base, and any interactive prompts to generate contextually relevant responses. This process involves structuring sentences, ensuring coherence, and adding informative details to answers.
Data	• **Data Representation Method:** Utilizes structured data from the FAQ Knowledge Base, NLP model outputs, and user interaction context. • **Initial Values:** None • **Use:** To generate responses to user queries and provide information. • **Semantics and Format:** Generates text-based responses in natural language, ensuring contextual accuracy and coherence.