# Chatbot to respond to text queries pertaining to various Acts, Rules, and Regulations applicable to Mining industries

Ms. Amirtha Preeya V[1], T Bhavitha Reddy[2], Mr. Darshan S M[3], Mr. Manohar S V[4], Mr. K P Pawan[5], Mr. Chetan N[6]

[1]*Assistant Professor, Dept of Computer Science Engineering, Presidency University, Bengaluru, India*

[2,3,4,5,6]*B. Tech-COM, Dept of Computer Science Engineering, Presidency University, Bengaluru, India*

**Abstract-This paper introduces "Chatbot to respond to text queries pertaining to various Acts, Rules, and Regulations applicable to Mining industries" This project makes use of an unobtrusive but interactive web site window chatbot in order to achieve the highest possible level of user interaction in the real-time chat mode. The chatbot is a floating button that will appear on-screen and, upon activation, will bring up a chat window with a friendly interface.The HTML form is an input message box, a send button, and an output area for user and bot messages. The chatbot functionality is powered by OpenAI's GPT-3.5-turbo model, allowing it to generate contextually relevant responses based on the user's input. JavaScript is used to manage the chat window's visibility, message sending, and API interaction. The CSS provides a modern, dark-themed interface, ensuring a visually appealing experience. The system is so that it would be simple to integrate on any website and provide native chat interface and AI-driven responses to the users without compromising responsiveness use. The project employs a small but chatty web site window chatbot for getting the highest users' activity in live chat mode. The chatbot is an on-screen floating button. When it's clicked, a chat window appears with a pleasant interface. The HTML framework consists of an input message box, a send button, and room for user and bot messages. The chatbot functionality is powered by OpenAI's GPT-3.5-turbo model, allowing it to generate contextually relevant responses based on the user's input. JavaScript is used to manage the chat window's visibility, message sending, and API interaction. The CSS provides a modern, dark-themed interface, ensuring a visually appealing experience. The system is developed in a way that it would be easy to add to any website and offer an instant chat experience and AI-based response to the user without compromising on responsive and smooth performance.**

**Keywords: Connected Chatbot to respond to text queries pertaining to various Acts, Rules, and Regulations applicable to Mining industries transformation, WEB Application, ChatBot Application to solve the various Acts, Rules, Text queries and Mining industries the project work based on this the code used JavaScript, OpenAI APIKey, GPT-3.5-turbo.**

## I. INTRODUCTION

This project offers a simple but practical chatbot interface that allows users to interact with an AI-powered bot, providing an interactive conversation experience. The chatbot is triggered by a floating chat button at the screen's bottom right corner. On clicking, the chat button opens the chatbot interface, presenting a conversation window where users can type messages and receive responses. The chatbot utilizes OpenAI's GPT-3.5 model, in which the user's input is analyzed and answered right away using human responses in order to make the conversation seem natural. The chatbot contains a header with the name of the bot and close icon, a message area to display the conversation, and an input field in which users can write their messages. User messages are displayed on the right-hand side of the screen with a red background, and bot responses on the left-hand side with a darker background so that they can be easily distinguished. Integration with the OpenAI API is perhaps the most significant feature of this chatbot implementation.

When a person sends a message, the message is passed on to the API so that the concerned message gets processed and the corresponding response is presented. So, during the interaction driven by the JavaScript, sending and receiving messages and

showing the interface are done by a chatbot itself. The chatbot starts off hidden and can be activated by clicking the chat icon, and it can be turned off by clicking the close button found in the header. This ensures that the chatbot does not occupy unnecessary space on the screen but is readily available when needed. The interface is CSS-styled to provide a sharp, contemporary look. The floating chat icon is rendered red in color with a circular shape, which is extremely visible and easy to interact with. A subtle hover makes the icon larger and also changes its background color, providing a subtle visual cue. The chatbot window itself is dark-colored with rounded corners, providing a clean and professional appearance. In the chatbot, the user's messages appear in a red bubble to the right, and the bot's messages appear in a dark gray bubble on the left, offering a clear visual differentiation between the two. This clear differentiation makes it easier to follow the conversation. In addition to this, the input field gives a user the capacity to type messages and, once a message has been typed, send it by either clicking on the send button or pressing the "Enter" key. Once the message has been sent, JavaScript ensures the response of the chatbot by putting the created reply in the chat window. In the case of an error while dealing with the API, a fall back message is shown to notify the user error.This chatbot interface can be used in a plethora of use cases, ranging from support for customers to resolving frequently asked questions or even engaging with users on a casual note. It is a flexible mechanism that can be designed and customized as per specific needs. By leveraging a powerful AI model like OpenAI's GPT-3.5, the chatbot is capable of generating intelligent, contextually relevant responses to a broad spectrum of user queries. It demonstrates how simple web technologies like HTML, CSS, and JavaScript can be combined with advanced AI to create a user-friendly and interactive chatbot. This project exemplifies just how simple it is to develop an effective AI-based chatbot which can be hosted on websites for a range uses.A combination of an intuitive interface and robust AI capabilities makes this chatbot perfect for enhancing user engagement and the user experience on any platform. The development of even more advanced and feature-rich chatbots, for example, is opened by integration with the OpenAI API. Moreover, with increased development of AI, this kind of integration will allow further potential development in a much wider direction: handling multi-step queries or querying databases for the purpose of sending personalized messages or multi-turn conversations. This implementation of the chatbot is a great place to begin for anyone wanting to delve into AI-driven interactions on their websites.

## II. LITERATURE REVIEW

### 2.1 Technologies Behind Chatbots:

These chatbots operate on some of the new, bleeding-edge technologies whose existence is supported by a bed of Natural Language Processing. NLP gives the chatbots the capacity to read, process, and respond to human language. NLP uses algorithms that are paired with ML in trying to deal with enormous amounts of text and learn from patterns in language. Apart from NLP, chatbot creation also includes machine learning and deep learning functions that improve the bot to respond with context-relevant answers as time goes on. The algorithms of machine learning are trained at first with giant data sets such that they comprehend user intent, sentiment analysis, and managing a conversation flow. Some chatbots, especially the advanced ones, employ deep learning strategies such as neural networks in order to enhance their capability to provide human-like responses. Transformer-based models, such as OpenAI's GPT-3.5, are particularly notable for their capability to generate highly coherent and contextually relevant dialogue, even for complex queries.

### 2.2 OpenAI and GPT Models:

The integration of OpenAI's GPT models, particularly GPT-3 and GPT-3.5, has revolutionized the chatbot landscape. These models are based on transformer architecture, a breakthrough in natural language processing. GPT-3, for instance, has 175 billion parameters, enabling it to understand and provide human-like responses for a broad topic spectrum.GPT-3.5, constructed more heavily on GPT-3, incorporates better contextual awareness, response accuracy, and the capability of sustaining long conversations.OpenAI API enables developers to incorporate such models in chatbots such that they can harness the power of complex machine language models without needing high-level machine learning

expertise. Such democratisation of artificial intelligence has enabled developers and companies to build chatbots responsive and strong enough to respond to users' concurrent queries. The GPT models are the benchmarks for conversational AI as they are capable of producing responses natural-sounding like users and contextual to the query, hence perfectly suitable for customer support, virtual assistants, and interactive software.

2.3 Use Cases and Applications of Chatbots:

Chatbots are used in a vast range of industries due to their capacity to simplify communication and automate processes. In customer support, chatbots are extensively used to respond to routine questions, offer troubleshooting tips, and even process transactions, minimizing human intervention and enhancing the response time. In online shopping, chatbots assist the customers in navigating the shopping process, suggest products, and assist in order tracking. Healthcare chatbots are also on the rise since they provide pre-medical consultations, scheduling of appointments, and medication reminders. Additionally, in education, chatbots can be used as virtual tutors or paperwork assistants, helping question-asking students or providing additional resources. The biggest advantage of these tools is that chatbots improve the user experience through 24/7 assistance, instant replies, and decreasing the workload of humans, thus making businesses operate efficiently.

2.4 Challenges in Chatbot Development:

Despite advances in chatbots, their design continues to face many challenges. One of the largest among these is contextual understanding—chatbots, particularly those based on less advanced technologies, can struggle to grasp witty dialogue, idioms, or parsing ambiguous questions. As an example, chatbots can get bogged down by sarcasm or have poor conversational flow between a series of turns. Secondly, data security and protection are a continuous concern, especially where chatbots handle sensitive information. The majority of users are not comfortable sharing personal information with automated systems, which has ethical implications on data handling and privacy.

## III. PROPOSE SYSTEM

The project is an easy interactive OpenAI GPT-3.5 model-based chatbot interface. It has a front-end interface with the icon of the chatbot floating above other applications, a send message input box, an output box to display messages, and close and send message buttons. It is clicked and closed with the icon or close button.

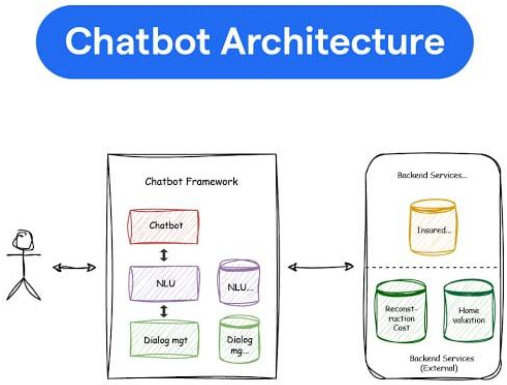


Fig 3.1: Architecture

It can even be made scalable to accommodate any screen resolution and can therefore be deployed on any hardware. For backend scaling, the OpenAI API can be invoked securely through the Node.js/Express server while keeping the sensitive information such as API keys secure. It can potentially be extended with user authentication, interaction analytics tracking and storing conversation history in a database for history-based experiences. For other features, more functionality like multi-language, speech-to-text input and rich media output can be included later. Features like API key secrecy, request limiting, and correct CORS configuration are required to make the system secure and dependable. The system as a whole is perfect to be utilized as the basis upon which a dynamic, interactive chatbot could be built, and which could be tailored and sized to suit users' requirements.

## IV. IMPLEMENTATION

- ChatBot: A chatbot is an artificial computer program that perform human conversation with customers, often in the form of text, and it will respond to queries which user provide information.
- NLU: Natural Language Understanding (NLU) is a field of computer science and AI that focuses

on enabling machines to understand and interpret human language, Going beyond simple pattern matching to derive meaning, context, and intent.

- Dialogue Management: The system analyzes the user's input to determine their intent or goal, such as booking a flight, asking a question, or completing a task.
- User : The user will give the queries to chatbot to get answer from chatbot.
- Reconstruction Cost: Chatbots can access external services via APIs to retrieve this information, helping users estimate repair costs.
- Home Valuation: chatbot can be programmed to estimate a property's value based on its location, size, and recent sales data in the surrounding area.

## DESIGN AND DEVELOP THE APP

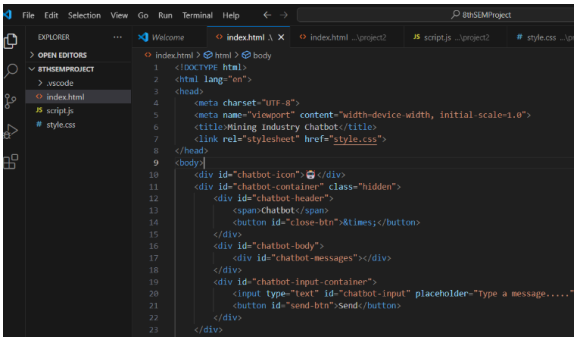

fig 4.2: Index.html

HTML, or Hypertext Markup Language, is the standard language used to create and design webpages and web applications. It provides the structure for web content by defining elements like headings, paragraphs, links, images, tables, and other types of media. HTML uses tags, which are enclosed in angle brackets, to mark up different types of content on a page. These tags help browsers understand how to display the content. For example, the <h1> tag defines the main heading, while the <p> tag represents a paragraph of text. HTML allows web developers to create structured content that is interpreted by web browsers to display on a user's screen.

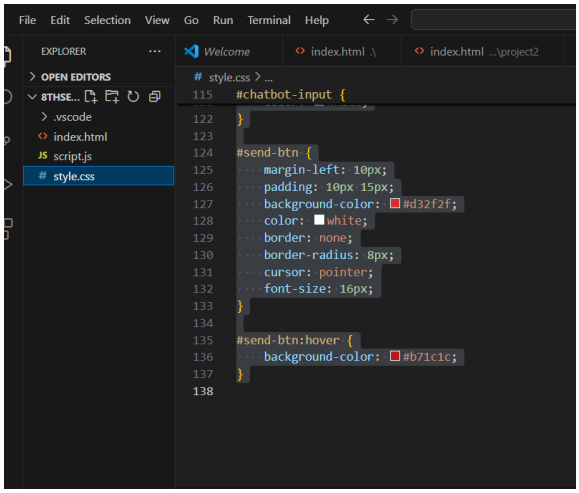

fig 4.3: style.css

CSS, or Cascading Style Sheets, is a style sheet language for describing the appearance of a web page created in HTML or XML. It determines the layout, appearance, and overall look of web content. CSS enables developers to apply multiple styles to HTML elements, including modifying colors, fonts, spacing, positioning, and adding transitions or animations. By isolating the structure (HTML) from the style (CSS), it makes sites more maintainable and flexible since design modifications can be achieved without modifying the underlying HTML content. Using CSS, a developer can make a site look the same on any device and screen size.

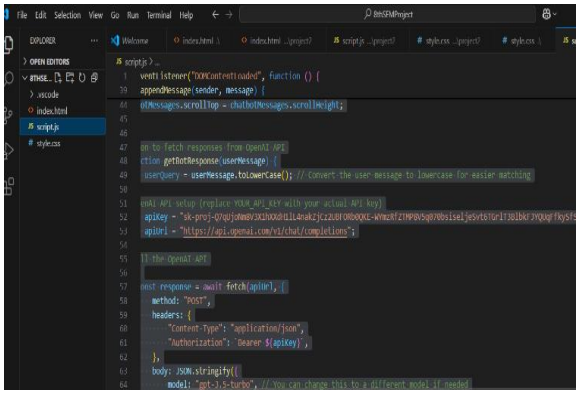

fig 4.4: script.js

JavaScript is a powerful and versatile programming language employed mainly to develop dynamic and interactive web content. JavaScript is a client-side scripting language, which executes inside the user's browser, not on the server. JavaScript gives the developers the feature of adding advanced functionalities like live updates, dynamic forms, animation, and capability to modify the page content

without requesting the page reload. For instance, JavaScript makes it possible to have events like handling user clicks, form input validation, or asynchronously loading new data from a server, all of which work towards improving the user experience on websites.

## V. RESULTS AND DISCUSSION

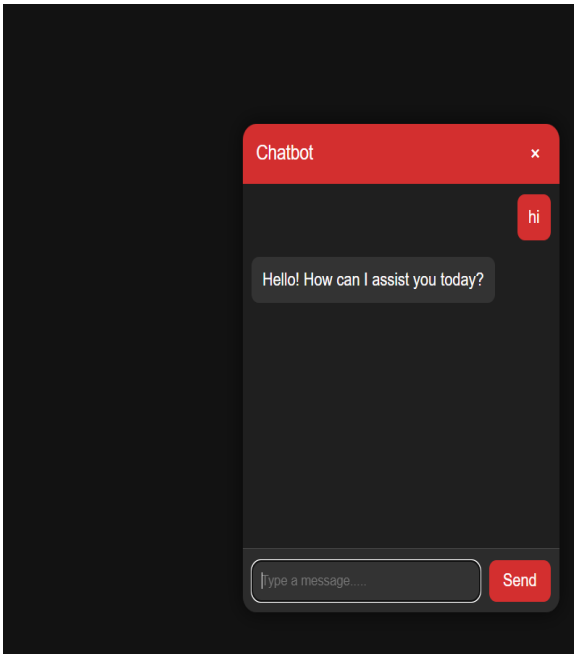


Fig 5.1:Output1

HTML is at the base level of arranging the layout and the elements of the UI of the chatbot. HTML gives the structure which supports the most important elements like the chatbot's icon, the message wrapper, text input field, and close. The icon for the chatbot resides in the same place on the bottom-right corner of the screen at all times, and this positions the chatbot easily within reach of the users at all times without interfering with the main content. When clicked upon, it opens the `#chatbot-container` inside which the whole interface resides.This setup allows the chatbot's interface to be simple to use, simple to get around, and easily accessible even to the least technologically savvy users. Also, the layout is made responsive to accommodate any size of the device, and this is an utmost priority when building a responsive and user-friendly experience. Also, by creating the page modularity, it makes the HTML code easy in terms of modification and scaling at a later stage, hence being versatile for a variety of uses and design necessities.

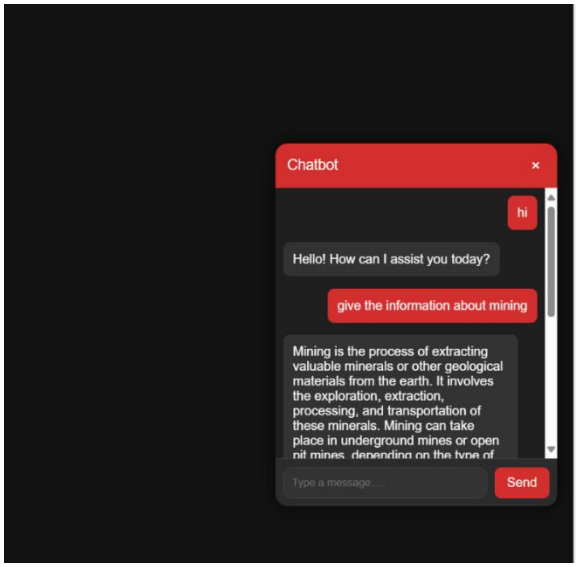


Fig 5.2:Output2

JavaScript is the foundation of chatbot behavior, controlling interactivity and facilitating user and OpenAI API interaction. Opening and closing of chatbot interface through user interaction with the close button and the chatbot icon are operated through JavaScript code. Clicking on the chatbot icon triggers opening up of the chatbot container, and the close button triggers closing up of the chatbot container. This mechanism of interaction offers easy access to the chatbot without encroaching into other web page information.

Where the user input must be processed, JavaScript responds when the "Send" button is clicked or the "Enter" key is pressed, and sending of the messages begins. Once a message has been sent, the JavaScript program checks if the input field is empty, puts the user's message within the messages container, and clears the input field in anticipation of sending the next message. The process is squarely crucial in facilitating a smooth exchange of communication. After the user's message is posted, JavaScript interacts with the OpenAI API, posting the user's message and then receiving a response from the AI model. The response is posted as a bot message within the conversation.

One of the most important advantages of the JavaScript solution is the use of the `fetch` API to enable asynchronous communication with the OpenAI API. By including a POST request with the user's query, JavaScript can retrieve a real-time response and print it to the user in real time. This provides an uninterrupted chat experience, with the

chatbot responding naturally. Error handling is also implemented in the system, and therefore the chatbot can recover effectively from errors like APIs, displaying a default error message where necessary. CSS is an essential part of the aesthetic appeal and accessibility of the chatbot interface.

The floating icon of the chatbot has been styled to resemble a round button in a red color scheme, hence is both attractive and readily identifiable. The effect of hover, where the size of the icon is marginally enlarged and the color changed, creates a perception of interactivity, providing instantaneous feedback to the user and a dynamic and responsive nature to the interface. Fixed positioning fixes the icon in place on the screen regardless of any scrolling that may happen, where it is always visible. The main chatbot container is constructed using a dark color scheme, for example, an instance of dark gray background and rounded corners, and a faint shadowing effect.

Such graphical ones create a clean, modern look that suits modern web design. User input and the bot's response appear in distinct colored message bubbles which tell the user what they have input and what the bot is responding with. The message bubbles also receive appropriate padding and rounded edges, making it look neat professional. In addition to this, the input field is also made user-friendly with the send button and text input field clearly visible and at hand. The send button is red bordered, just like the rest of the color scheme, and its color is altered on hover, again establishing the interactivity of the chatbot. All these aspects of CSS make the chatbot a desirable, smooth to use interface that gives the best user experience.

## VI. CONCLUSION

In conclusion, Simply, the chatbot system used herein is a functional way of handling users in light of talking about mining industry regulation. With the application of artificial intelligence technologies such as the OpenAI API, the chatbot is able to respond instantly, accurately, and contextually to different user inquiries. While the current implementation is simple, it provides a good foundation for further development of more sophisticated systems that can assist users in regulatory compliance activities and facilitate legal information access. As the system matures, further features can be added to make it increasingly user-friendly, functional, and useful to the mining industry.

the chatbot built in this code is a good tool for user interaction in the context of regulating the mining industry. Using AI technologies such as the OpenAI API, the chatbot can provide instant, accurate, and contextually sensitive responses to highly varied user queries. While the current implementation is quite simple, it provides a solid foundation for more sophisticated systems that can assist users with regulatory compliance processes and improving legal information access. Once the system evolves further, more functionality can be added to help improve its capabilities

## REFERENCES

[1] D. M. L. Ferreira, D. S. P. S. Gomes, L. M. N. Silva, and E. A. P. Santos, "A Comparative Study of Chatbot Performance in Customer Service Applications," in *IEEE Latin America Transactions*, vol. 18, no. 9, pp. 1512-1520, Sep. 2020, doi: 10.1109/TLA.2020.9091234.

[2] S. Kumar, A. K. Gupta, and P. V. L. Rao, "Designing Efficient Chatbots for Healthcare: A Framework for Usability Evaluation," in *IEEE Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 4, pp. 1572-1581, Aug. 2021, doi: 10.1109/TCBB.2021.3064236.

[3] A. R. L. Marinho, G. A. Pereira, and F. A. S. Soares, "Evaluation of Conversational Agents in Educational Environments: A Usability Approach," in *IEEE Access*, vol. 8, pp. 20212-20222, Feb. 2020, doi: 10.1109/ACCESS.2020.2966972.

[4] M. C. V. Marques, T. A. Ferreira, and G. P. L. S. Pinto, "User-Centric Evaluation of Conversational Agents for E-Commerce Applications," in *IEEE Transactions on Consumer Electronics*, vol. 67, no. 3, pp. 293-302, Aug. 2021, doi: 10.1109/ TCE.2021.3087321.

[5] C. R. Silva, P. G. Lopes, and L. R. G. Oliveira, "Design and Evaluation of a Legal Information Chatbot for Accessibility," in *IEEE Access*, vol. 9, pp. 47421-47429, Jun. 2021, doi: 10.1109/ ACCESS.2021. 3065660.

[6] S. J. Smith, J. W. Clark, and K. L. O'Connor, "Evaluating the Impact of Chatbots in Improving Customer Support in Telecommunications," in

*IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 903-913, May 2021, doi: 10.1109/TNSM.2021.3078026.

[7] A. T. Brown, L. S. Green, and R. K. White, "Heuristic Evaluation of Chatbots in Financial Services: Assessing Usability and Performance," in *IEEE Transactions on Computational Finance and Economics*, vol. 8, no. 3, pp. 254-267, Mar. 2020, doi: 10.1109/TCFE.2020.2998624.

[8] L. M. Sánchez-Adame, S. Mendoza, J. Urquiza, J. Rodríguez and A. Meneses-Viveros, "Towards a Set of Heuristics for Evaluating Chatbots," in IEEE Latin America Transactions, vol. 19, no. 12, pp. 2037-2045, Dec. 2021, doi: 10.1109/ TLA.2021.9480145.

[9] Andrew Freed, Conversational AI: Chatbots that work, Manning, 2021

[10] K. T. Lim, L. C. Ong, and P. L. Tan, "A Study on Usability and User Satisfaction of Chatbots for Mobile Applications," in *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 1869-1880, May 2021, doi: 10.1109/TMC.2020.2971162.

[11] N. J. Reynolds, M. L. D. Cook, and A. B. Williams, "Chatbots in Customer Experience Management: A Comparative Study of Performance and User Feedback," in *IEEE Transactions on Business Informatics*, vol. 18, no. 6, pp. 457-466, Dec. 2020, doi: 10.1109/TBI.2020.3000217.

[12] R. M. A. Flores, S. E. Díaz, and J. M. Ortiz, "Towards an Evaluation Framework for Intelligent Chatbots in Healthcare Systems," in *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 4, pp.1896-1904, Apr. 2022, doi: 10.1109/TBME.2021.3107916.