AN INTELLIGENT CHATBOT FOR GOVERNMENT SCHEMES

A MINI PROJECT REPORT

Submitted by

NITHISH S (2116221801502)

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY IN

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

ANNA UNIVERSITY, CHENNAI

NOV 2024

ANNA UNIVERSITY, CHENNAI 600025

BONAFIDE CERTIFICATE

Certified that this Report titled "**AN INTELLIGENT CHATBOT FOR GOVERNMENT SCHEMES**" is the bonafide work oF **NITHISH S (2116221801502)** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Dr.J.M.Gnanasekar M.E.,Ph.D.,	**Mr. SURESH KUMAR, M.E.,(Ph.D).,**
Professor and Head	Professor,
Department of Artificial intelligence and Data Science	Department of Artificial intelligence and Data Science
Rajalakshmi Engineering College	Rajalakshmi Engineering CollegeChennai-
Chennai-602 105	Chennai-602 105

Submitted for the project viva-voce examination held on_______________.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman Mr. S. MEGANATHAN, B.E, F.I.E., our Vice Chairman Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S., and our respected Chairperson Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D., for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to Dr. S.N. MURUGESAN, M.E., Ph.D., our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to Dr. J. M. GNANASEKAR., M.E., Ph.D.,Head of the Department, Professor and Head of the Department of Artificial Intelligence and Data Science for his guidance and encouragement throughout the project work. We are glad to express our sincere thanks and regards to our supervisor Mr. S. SURESH KUMAR, M.E., (Ph.D) Professor, Department of Artificial Intelligence and Data Science and coordinator, Dr. P. INDIRA PRIYA, M.E., Ph.D., Professor, Department of Artificial Intelligence and Data Science, Rajalakshmi Engineering College for their valuable guidance throughout the course of the project.

Finally, we express our thanks for all teaching, non-teaching, faculty and our parents for helping us with the necessary guidance during the time of our project.

ABSTRACT

With the exponential growth of information available online, citizens face increasing challenges in accessing timely and reliable data on various government initiatives. Education schemes in India, aimed at promoting literacy, skill development, and accessibility for all, are no exception. To bridge this information gap, we have developed an intelligent Question and Answer (Q&A) chatbot dedicated to providing accurate, relevant, and prompt responses to queries about Indian government education schemes. This chatbot harnesses the power of Language Learning Models (LLMs) and fine-tuned BERT (Bidirectional Encoder Representations from Transformers) to comprehend and generate contextually accurate responses. Through Python's efficient handling of data and model integration, this system not only ensures high performance but also aligns with the diverse linguistic and regional needs of users across the country.

The Q&A chatbot employs LoRA (Low-Rank Adaptation) for fine-tuning the BERT model, making it adept at handling specific educational scheme queries. LoRA helps adapt the BERT model with minimal computational overhead, thus enhancing its accuracy in understanding nuanced questions related to government education initiatives. By incorporating domain-specific data during the fine-tuning phase, the model achieves a higher degree of relevance in its responses, providing tailored and insightful answers that address user inquiries effectively. This approach makes the chatbot an efficient tool for both general users seeking information on eligibility, application processes, and benefits, as well as for researchers and policymakers requiring detailed insights into the impacts of these schemes.

The user interface, designed with HTML, CSS, and JavaScript, provides a seamless and intuitive experience for users. The front-end is integrated with Django, a powerful web framework, which facilitates smooth interaction between the user and the chatbot, ensuring rapid and coherent data exchange. The website's interface is built to be responsive, supporting accessibility across multiple devices, from mobile phones to desktops.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	I
	LIST OF TABLES	II
	LIST OF FIGURES	III
1	**INTRODUCTION**	
	1.1 GENERAL	1
	1.2 NEED FOR THE STUDY	2
	1.3 OBJECTIVES OF THE STUDY	3
	1.4 OVERVIEW OF THE PROJECT	4
2	**REVIEW OF LITERATURE**	
	2.1 INTRODUCTION	6
	2.2 LITERATURE OVERVIEW	9
3	**SYSTEM OVERVIEW**	
	3.1 EXISTING SYSTEM	10
	3.2 PROPOSED SYSTEM	11
	3.3 FEASIBILITY STUDY	12
4	**SYSTEM REQUIREMENTS**	
	4.1 SOFTWARE REQUIREMENTS	13
5	**SYSTEM DESIGN**	
	5.1 SYSTEM ARCHITECTURE	14
	5.2 MODULE DESCRIPTION	16
	5.2.1 BUILDING THE BERT MODEL	16
	5.2.2 MODULE SELECTION AND FINE TUNING	18

	5.2.3 QUESTION MATCHING AND RESPONSE GENERATION	19
	5.2.4 WEB INTERFACE MODULE	20
6	**RESULT AND DISCUSSION**	
	6.1 RESULT AND DISCUSSION	21
7	**CONCLUSION AND FUTURE ENHANCEMENT**	
	7.1 CONCLUSION	22
	7.2 FUTURE ENHANCEMENT	23
	APPENDIX	
	A1.1 SAMPLE CODE	25
	A1.2 SCREENSHOTS	26
	REFERENCES	27

LIST OF TABLES

Table no	Table Name	Page No
1	Literature Review	9

LIST OF FIGURES

Figure No	Figure Name	Page No
5.1	System Architecture	14
5.2.1	Building BERT Module	16
5.2.2	Model selection and fine tuning	18
5.2.3	Question matching and response generation	19
5.2.4	Frontend Integration	20
7.1	Building Model	25
7.2	Dataset	25
7.3	Response for Input	26
7.4	Chatbot	28

CHAPTER 1

INTRODUCTION

1.1 GENERAL

The development of an intelligent AI chatbot focused on government education schemes is a promising initiative aimed at making essential information more accessible to citizens. Leveraging advancements in Natural Language Processing (NLP), this chatbot uses a combination of the BERT (Bidirectional Encoder Representations from Transformers) model and Retrieval-Augmented Generation (RAG) techniques on the backend. BERT, developed by Google, enables the chatbot to understand and respond accurately to user queries by processing language in context, both from previous and following words. This capability allows the chatbot to interpret questions about education schemes, retrieve relevant details, and respond in a user-friendly way.

RAG, which combines pre-trained transformers with retrieval capabilities, enhances the chatbot's accuracy and relevance by drawing from a large database of government documents, policies, and FAQs. Unlike traditional search systems, this chatbot doesn't just pull up links; it provides clear and concise answers, as it can generate responses based on the retrieved content, making it particularly useful for users unfamiliar with official language or complex terminologies. Users can ask about eligibility, benefits, or specific scheme application processes, and the chatbot will respond with tailored, relevant information. This makes it a valuable tool for citizens looking for quick, reliable information without navigating complex government websites.

The frontend of the chatbot interface is developed using HTML, CSS, and JavaScript, providing a clean and intuitive experience. HTML structures the content, CSS adds styling for a visually appealing layout, and JavaScript enables interactivity, allowing users to type questions and view responses in real-time. The design focuses on simplicity and usability, ensuring that users from various backgrounds, even those with minimal technological expertise, can easily interact with the chatbot. The web interface allows for responsive interaction, adapting to different devices and screen sizes, making it accessible on both mobile and desktop platforms.

1.2 NEED FOR THE STUDY

The study for developing an intelligent chatbot focused on government education schemes is driven by the need to improve accessibility, accuracy, and user experience when citizens seek information on educational programs. Traditional methods of disseminating information—such as static government websites, brochures, or help centers—can be time-consuming and confusing, especially for users unfamiliar with official language or digital navigation. This chatbot project seeks to address these issues by providing a single, intelligent platform that can answer questions accurately and in real-time, allowing citizens to access relevant information easily. This project could simplify the process for users and reduce their dependence on customer service representatives or extensive online searches.

One of the main drivers for this study is the need to enhance awareness and participation in government education schemes. Many citizens are unaware of the benefits and opportunities provided by these programs due to the lack of easily accessible information. By developing a chatbot with the ability to answer questions based on real government data, the project aims to make it easier for users to learn about available schemes, eligibility requirements, application processes, and benefits. This approach not only helps individuals directly but also serves the broader goal of promoting educational equity and social mobility by enabling more people to take advantage of government-supported opportunities.

Moreover, the study is essential for leveraging modern advancements in Natural Language Processing (NLP) and Machine Learning (ML) to optimize public service delivery. By using BERT and RAG models, the chatbot can understand complex queries, retrieve relevant information accurately, and generate human-like responses, providing a more engaging and useful interaction than traditional search engines. This use of AI can set a precedent for future government digital services, showing how intelligent, user-centered solutions can simplify public interactions with government bodies.

Key Needs Driving This Study

1. Accessibility: Providing an easy-to-use, intuitive platform for citizens to access information on education schemes without complex navigation.

2. Awareness and Participation: Encouraging greater engagement with government programs by making relevant information readily available.

3. Efficiency in Information Retrieval: Reducing the time and effort required for users to find accurate information through advanced NLP techniques.

4. Enhanced User Experience: Offering a conversational, human-like interaction that guides users through their queries and enhances the overall experience.

1.3 OBJECTIVES OF THE STUDY

The primary objective of this study is to develop an intelligent chatbot that simplifies access to government education scheme information, using advanced NLP models to deliver accurate, user-friendly responses. This chatbot aims to bridge the gap between citizens and government programs by making complex information easy to understand, thus promoting greater awareness, participation, and equity in educational opportunities.

1. Develop a platform that allows citizens to access information about educational schemes effortlessly. By simplifying navigation and enabling real-time responses, the chatbot aims to reduce barriers for users, especially those with limited technical expertise.

2. Increase the public's knowledge and engagement with government schemes by offering detailed, easy-to-access information. The chatbot's capabilities encourage users to explore and benefit from programs they may not have known about previously.

3. Use NLP models like BERT and RAG to provide relevant responses quickly, ensuring that users spend less time searching for accurate information. This objective aims to enhance efficiency and reduce frustration among citizens seeking specific details.

4. Ensure that the chatbot's language and interface cater to a wide range of users, including those with different educational and linguistic backgrounds. This inclusivity aims to make government resources accessible to all, supporting social and educational equity.

5. Design the chatbot for a conversational, user-friendly interaction that mimics a human conversation, enhancing engagement. This objective is focused on creating a comfortable experience that guides users smoothly through their queries.

6. Demonstrate the potential of AI-driven solutions in improving public service delivery, setting an example for other government platforms. By showcasing the chatbot’s effectiveness, this study could influence future developments in government-citizen digital interactions.

1.4 OVERVIEW OF THE PROJECT

It Is dedicated to providing citizens with easy access to information about government education schemes. By leveraging advanced Natural Language Processing (NLP) techniques, specifically BERT and Retrieval-Augmented Generation (RAG), the chatbot offers accurate and contextually relevant answers in a conversational format. The goal is to streamline the process of finding, understanding, and utilizing information about educational programs, thereby promoting greater awareness and participation among citizens.

Key Features:

1. **Purpose and Motivation**: The project seeks to improve citizen access to government education scheme information by using an intelligent chatbot. Motivated by the need to bridge the information gap, it simplifies complex data into understandable, personalized responses.

2. **Technological Foundation**: This chatbot is built on BERT and RAG NLP models, allowing it to understand and retrieve relevant information accurately. These advanced models enable the chatbot to interpret user queries contextually, enhancing the reliability of responses.

3. **User-Centered Interface**: The frontend is designed with HTML, CSS, and JavaScript to offer an intuitive, visually appealing, and responsive experience. This ensures that users from various backgrounds can interact seamlessly with the chatbot on both desktop and mobile devices.

4. **Impact Goals**: The project aims to enhance public awareness and engagement with educational schemes, supporting social inclusion and equity.

Workflow:

- User Initiates Query: The user visits the chatbot interface on the website and types a question related to government education schemes (e.g., "How do I apply for a scholarship?").
- Query Processing: The chatbot processes the input using NLP models (BERT) to understand the context of the question and identify key details like scheme name, eligibility, or application process.
- Information Retrieval: The chatbot uses Retrieval-Augmented Generation (RAG) to search the relevant database or documents for the most accurate information based on the user's query.
- 4. Generate Response: The chatbot generates a response using the retrieved information and presents it in a simple, easy-to-understand format for the user.
- 5. User Follow-Up: If the user requires further clarification or has additional questions, they can ask follow-up queries, and the chatbot will continue to provide answers based on the same process.

CHAPTER 2

REVIEW OF LITERATURE

2.1 INTRODUCTION

An intelligent chatbot has been developed to provide citizens with reliable and relevant information about various government education schemes, utilizing advanced Natural Language Processing (NLP) techniques in its backend. The chatbot leverages the BERT (Bidirectional Encoder Representations from Transformers) model, which excels in understanding the context of words within sentences. By fine-tuning the BERT model specifically for government education schemes, the chatbot is able to process and understand a wide range of user queries, providing accurate and context-aware responses. Additionally, the backend integrates Retrieval-Augmented Generation (RAG) architecture, which combines the generative capabilities of models like GPT with a retrieval system that pulls relevant data from a knowledge base. This allows the chatbot not only to generate responses but also to retrieve factual information from an up-to-date database of government education schemes, ensuring users receive precise and accurate details. On the frontend, a user-friendly webpage has been created using HTML, CSS, and JavaScript, providing an intuitive and responsive interface for users to easily interact with the chatbot. Through this system, citizens can ask questions, receive informative answers, and gain access to a variety of government educational opportunities, ultimately helping them make informed decisions about the schemes available to them.

S. No	Author Name	Paper Title	Description	Journal	Volume/ Year
1	Jin K. Kim	ChatGPT and large language model (LLM) chatbots	Performed a scoping review of available literature to understand the current state of LLM use in medicine and to provide a guideline for future utilization in academia.	IEEE	2021
2	Samuel Kernan Freire	Conversational Assistants in Knowledge-Intensive Contexts: An Evaluation of BERT versus Intent-based Systems	Conversational Assistants (CA) are increasingly supporting human workers in knowledge management. Traditionally, CAs respond in specific ways to predefined user intents and conversation patterns.	IEEE	2022

3	Sumit Kumar Dam	A Complete Survey on BERT AI Chatbots	Conversational agents, often referred to as AI chatbots, rely heavily on such data to train large language models (LLMs) and generate new content (knowledge) in response to user prompts.	IEEE	2022
4	Babymol Kurian	Automating the Development of Task-oriented LLM-based Chatbots	GovInfoHub' presents a pioneering solution to provide citizens with real-time and accessible information regarding government schemes.	IEEE	2020

2.2 LITERATURE REVIEW

The development of intelligent chatbots for providing information on government education schemes has gained significant attention in recent years, with numerous studies and advancements focusing on utilizing Natural Language Processing (NLP) models to improve accessibility and user engagement. One of the core advancements in this domain is the use of transformer-based models, such as BERT (Bidirectional Encoder Representations from Transformers), which has revolutionized NLP by enabling better understanding of context in text, outperforming traditional models in tasks such as question answering and information retrieval. Researchers have explored various methods of fine-tuning BERT for specific domains, such as education, to enhance its ability to answer queries accurately and in a contextually relevant manner. Furthermore, the integration of Retrieval-Augmented Generation (RAG) models has been found to further improve chatbot performance by combining the strengths of information retrieval with generative models, allowing chatbots to not only generate responses but also access a large, external knowledge base, thus providing real-time, fact-based answers. In the context of government education schemes, chatbots powered by RAG and BERT have shown promising results in delivering personalized, accurate information to citizens about available opportunities. The frontend development of these systems, often utilizing web technologies like HTML, CSS, and JavaScript, ensures an intuitive and responsive interface that improves user experience. Studies also emphasize the importance of building user-friendly, interactive interfaces to facilitate easy communication between citizens and the chatbot, thus ensuring that the information is accessible to a wide range of users.

CHAPTER 3

SYSTEM OVERVIEW

3.1 EXISTING SYSTEM

Existing systems for providing information about government education schemes typically rely on static web pages, documents, or customer service helplines, where users manually search for information or call for assistance. While these systems can offer detailed content, they often lack the interactivity and personalization that users increasingly expect. Static websites provide large amounts of information, but they can overwhelm users with data that is not tailored to their specific needs. Moreover, the lack of real-time updates and complex navigation can make finding relevant information cumbersome, especially for individuals unfamiliar with the schemes or those who are not proficient in navigating through dense text-based resources.

With the rise of intelligent systems, some government initiatives have adopted chatbots to assist users in retrieving information about education schemes. These chatbots typically use rule-based systems, where responses are pre-programmed to answer specific queries. While this approach can work for basic information, it is limited in flexibility and fails to account for the nuances and variety of queries users might pose. For example, if the user's query is not directly covered by the predefined rules or if the language used is different from what was anticipated, the chatbot may provide irrelevant or unhelpful responses. Additionally, many chatbots lack the capability to retrieve up-to-date information, making them less reliable for providing accurate and current details about government schemes.

Recent advancements, however, have led to the integration of more sophisticated technologies such as machine learning and Natural Language Processing (NLP) models like BERT and GPT in government service applications. These intelligent chatbots, utilizing the BERT model for better understanding of language and context, offer more dynamic and adaptive responses. The addition of Retrieval-Augmented Generation (RAG) allows these systems to not only generate text-based answers but also retrieve data from extensive, real-time knowledge bases. This enables a more personalized and accurate experience, as the chatbot can pull the latest information regarding specific education schemes based on the user's query. By leveraging advanced models and real-time data retrieval, these systems can better serve citizens, helping them navigate through available education opportunities with ease and precision.

3.2 PROPOSED SYSTEM

The proposed system aims to enhance the accessibility and efficiency of providing information about government education schemes by leveraging advanced machine learning models and an intuitive web interface. The system will utilize BERT (Bidirectional Encoder Representations from Transformers), a state-of-the-art NLP model, to understand and process complex user queries. By fine-tuning BERT specifically for government education schemes, the system can deliver more accurate, context-aware responses, offering tailored information to users based on their specific needs.

In addition to BERT, the system will incorporate Retrieval-Augmented Generation (RAG) architecture. This combination of information retrieval and generative capabilities allows the chatbot to pull real-time data from a centralized knowledge base of government education schemes, ensuring that users receive the most up-to-date and accurate information available. RAG enhances the chatbot’s ability to not only generate informative responses but also retrieve factual details that are crucial for answering a wide range of questions.

The proposed system will feature a user-friendly web interface built using HTML, CSS, and JavaScript, ensuring a seamless and interactive experience for users. The chatbot will be easily accessible through the webpage, allowing users to ask questions, explore available schemes, and receive real-time assistance in an intuitive, conversational format. The web interface will be designed to be responsive and accessible to a broad audience, including those with limited technical expertise.

3.3 FEASIBILITY STUDY

Key to this feasibility study is easy to scale and update, providing a user-friendly experience for a broad audience. Economically, it is cost-effective due to the use of open-source models and cloud infrastructure, while legally and ethically, it ensures compliance with data protection regulations. Overall, the system is a viable solution for improving access to government education scheme information.

1. **Technical Feasibility**: The proposed system is technically feasible due to the availability of advanced Natural Language Processing (NLP) models like BERT and Retrieval-Augmented Generation (RAG), which have been proven effective in understanding complex user queries and retrieving relevant information. These models can be integrated with a web-based frontend using HTML, CSS, and JavaScript, which are widely used and well-supported technologies. The system can also be hosted on cloud platforms to ensure scalability and accessibility for a large user base, making the technical implementation highly feasible.
2. **Operational Feasibility**: The system is operationally feasible as it can be easily deployed within the existing government infrastructure. Many government departments already maintain centralized databases of education schemes, which can be integrated into the system's knowledge base. Additionally, the chatbot can be trained and updated with new information as required, ensuring its responses remain relevant and accurate. The web interface is simple and user-friendly, making it accessible to a broad range of citizens, including those with limited technical skills, enhancing its operational feasibility.
3. **Economic Feasibility**: From an economic perspective, the system is cost-effective because it leverages open-source NLP models like BERT and existing web technologies, reducing the need for expensive proprietary software. The costs associated with development primarily include the initial setup and integration of the knowledge base, as well as maintenance and periodic updates.

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 SOFTWARE REQUIREMENT

1. Operating System: Windows 10/11

2. Programming Languages:

Python3: Python is required for developing the core functionality, including machine learning models and video processing. Python libraries such as TensorFlow, Keras, and OpenCV will be used.

3. Web Development:

Flask: Flask, a lightweight Python web framework, is used for building the backend of the web application. It handles routing, form submissions, and communication between the frontend and backend.

HTML5, CSS3, and JavaScript: HTML is essential for structuring the web page, while CSS provides styling to ensure a professional user interface. JavaScript adds interactivity and dynamic content, such as displaying pie charts or video results.

4. Machine Learning Libraries:

TensorFlow/Keras: These deep learning libraries are essential for training and running the human action recognition model that detects student engagement and behavior in the videos.

BERT Model:

The BERT model (Bidirectional Encoder Representations from Transformers) is used in chatbots to understand the context of user queries by analyzing the relationships between words in a sentence.

5. Data Processing: NumPy and Pandas These libraries are essential for handling and processing numerical data, such as managing predictions and preparing data .

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

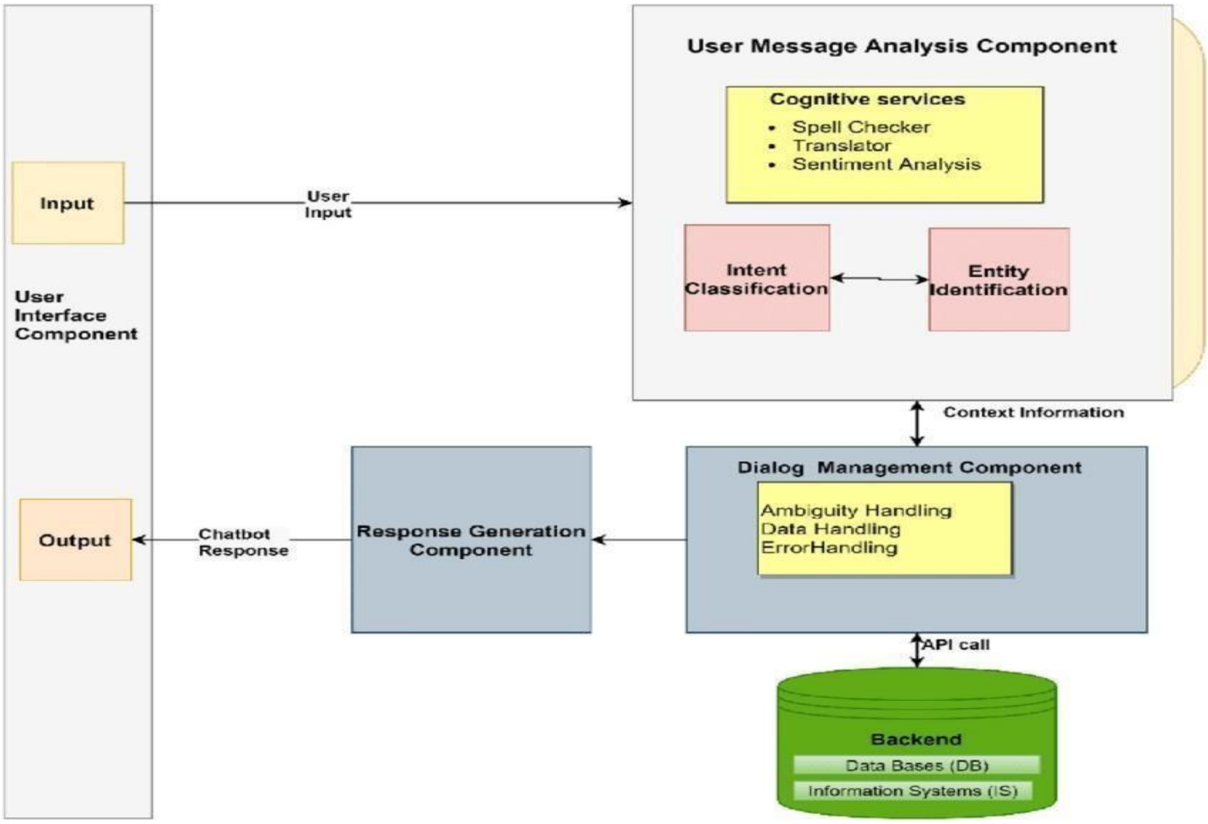


Fig 5.1 System Architecture

The chatbot system designed for answering queries related to education schemes is divided into two major components: the frontend (user interface) and the backend (query processing). The frontend acts as an interface through which users can interact with the system, asking questions about various education schemes, such as scholarship opportunities, loan schemes, or government policies. On the backend, the system leverages advanced Natural Language Processing (NLP) techniques, specifically the BERT (Bidirectional Encoder Representations from Transformers) model, to process these user queries and retrieve relevant responses from a database of education schemes.

The frontend of the chatbot is typically a web or mobile interface that allows users to input their questions. This can be through a text input box where users type their queries, such as "What are the eligibility criteria for the XYZ Scholarship?" or "How can I apply for the ABC Loan scheme?". The frontend is designed to be user-friendly, providing features like suggestion buttons, autocomplete for common queries, and quick replies. It also displays responses generated by the backend, which may include text, links to official pages, or downloadable resources.

Once the user submits a query from the frontend, it is passed to the backend for processing. Here, the query undergoes several stages. The primary task of the backend is to analyze and interpret the user's question accurately. This is where the BERT model comes into play. The BERT model is fine-tuned with a domain-specific corpus related to education schemes. It performs two key functions: understanding the context of the user query and matching the query with the most relevant education scheme from the database.

BERT's ability to understand the semantic meaning of a query (including nuances like intent, entities, and context) helps it generate precise answers. For instance, when a user asks about eligibility criteria, BERT can identify that the question pertains to a specific scheme's requirements and retrieve the relevant data.BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained transformer-based model that has shown remarkable performance in NLP tasks. In this architecture, BERT is fine-tuned with education-related datasets, which include detailed information about various education schemes, such as eligibility, benefits, application processes, etc. When the user submits a question, BERT analyzes the query to identify the key entities (e.g., scholarship name, eligibility criteria) and the intent behind the question (e.g., asking for steps to apply or the eligibility conditions). The model then ranks and retrieves the most relevant pieces of information from the database, returning an accurate and contextually appropriate response.

Once the backend processes the user's query, the response is delivered to the frontend, where it is displayed to the user. The system may also include additional features such as providing follow-up questions or clarifications. For example, if the user asks for the steps to apply for a scholarship, the chatbot might offer a follow-up question like "Do you need help with the application form?" to further assist the user. Additionally, the chatbot may ask for user feedback on the response quality, helping to fine-tune the model over time. If the BERT model is uncertain or does not find an exact match, the system can escalate the query to a human operator for further assistance.

This architecture provides an efficient and scalable solution for answering user queries related to education schemes using a combination of modern chatbot technologies and the BERT model. By utilizing a powerful NLP model like BERT in the backend, the system is capable of understanding complex user queries and delivering relevant, context-aware responses, enhancing the user experience and accessibility of information related to education schemes.

5.2 MODULE DESCRIPTION

5.2.1 BUILDING THE BERT MODEL:

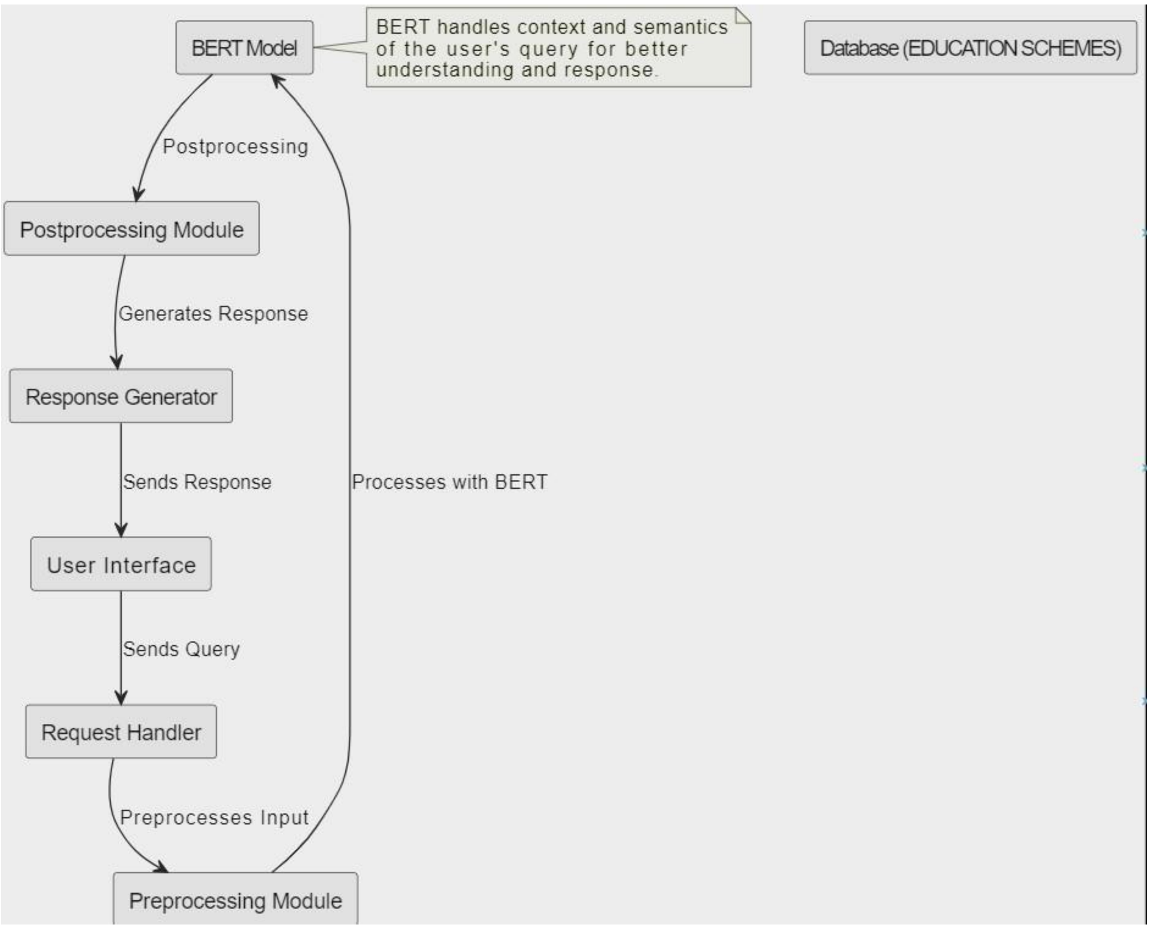


Fig 5.2.1 BERT Model

User sends a query:

The user interface sends a query from the front-end to the backend (Request Handler).

Request Handler:

The request handler routes the query to the preprocessing module, where the input is cleaned and formatted.

Preprocessing:

The preprocessing module prepares the input text for the BERT model by tokenizing and normalizing it.

BERT processes the input:

BERT takes the prepared input and processes it to understand the meaning and intent behind the user's query.

Post Processing:

Once BERT has processed the input, post processing takes place to refine the model’s output.

Response Generation:

The response generator formulates a final answer and sends it back to the UI.

Optional Database:

If needed, the chatbot may query a database for additional information to enhance the response, especially in domain-specific scenarios.

5.2.2 Model Selection and Fine-Tuning:

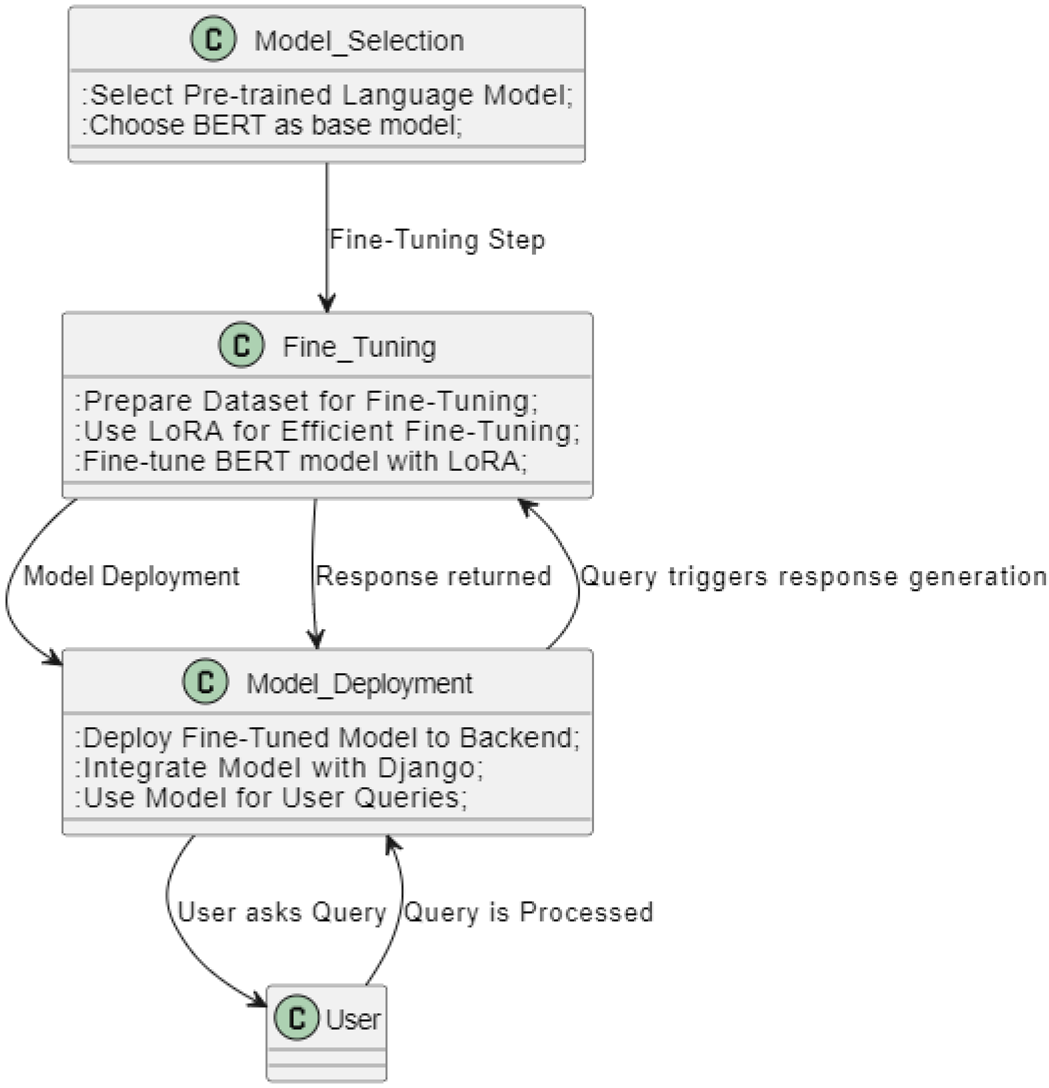


Fig 5.2.2 Model Selection

1. Pre-trained Model Selection: Choose a pre-trained language model like BERT, which is capable of understanding complex language patterns and context.

2. Domain-Specific Fine-Tuning: Fine-tune the selected model using domain-specific data (government education schemes) to adapt it to the required task.

3. Low-Rank Adaptation (LoRA): Use LoRA to efficiently fine-tune the model by modifying only a small number of parameters, making it more computationally efficient for the specific task.

4. Data Preprocessing for Fine-Tuning: Clean and structure the educational scheme data to ensure it is well-suited for fine-tuning the model for better accuracy in responses.
5. Evaluation and Optimization: Continuously evaluate the fine-tuned model's performance with sample queries to ensure its relevance and accuracy in answering user questions about schemes.

5.2.3 Question Matching and Response Generation:

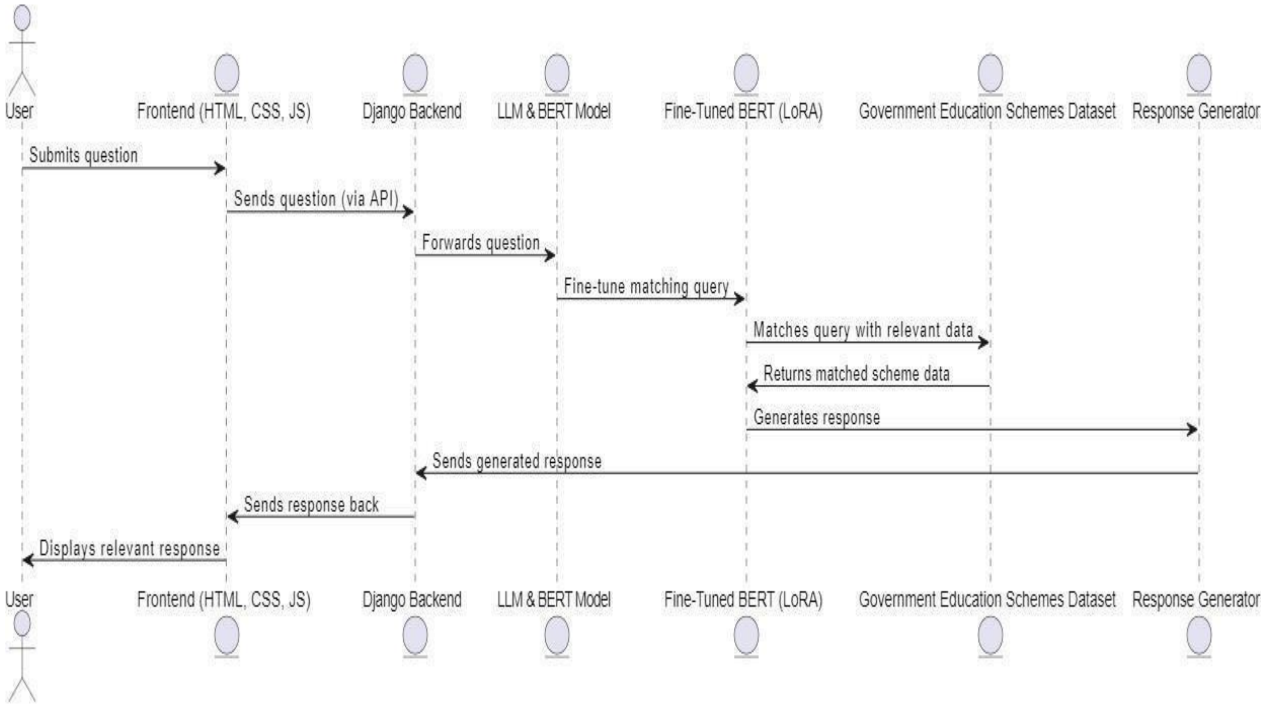


Fig 5.2.3 Response Generation

1. User Query Handling:

- The process of receiving and processing a user's question through the frontend interface.

2. Natural Language Understanding (NLU):

- The model interprets the user's question by understanding the intent, entities, and context within the input text.

3. Semantic Search:

- Matching the input question with the most relevant data from the dataset by evaluating semantic similarity rather than simple keyword matching.

4. Contextual Response Generation:

- Generating a relevant answer by leveraging the fine-tuned model (e.g., BERT) to provide responses based on the contextual understanding of the question.

5. Answer Extraction:

- Selecting the most accurate and informative response from the preprocessed dataset that best answers the user's query about the government schemes.

5.2.4 Frontend Interface Module:

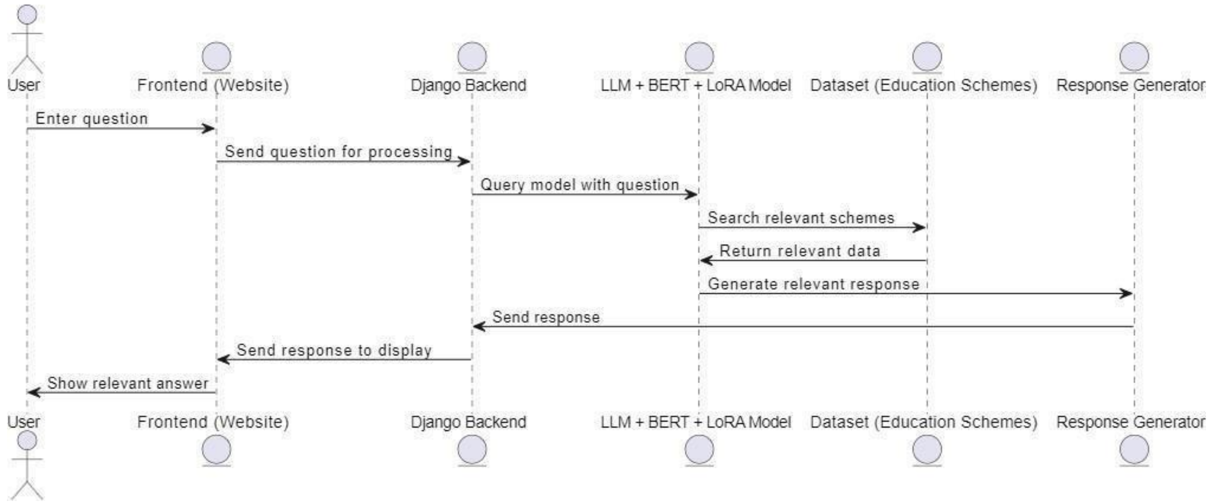


Fig 5.2.4 Frontend Interface

1. User Interface (UI) Design: Creating a responsive, user-friendly chat interface using HTML, CSS, and JavaScript to ensure smooth interaction.
2. Chatbot Input/Output: Developing input fields for users to submit questions and dynamic areas to display responses from the chatbot.
3. JavaScript for Real-time Interaction: Using JavaScript to capture user input and update the page in real time, ensuring a seamless chat experience.
4. API Communication: Integrating Django APIs to send user queries to the backend, process them through the model, and return relevant answers.

5. Responsive Design: Ensuring the website adapts well to different devices and screen sizes, providing a consistent experience across desktop and mobile platforms.

CHAPTER 6

RESULT AND DISCUSSION

6.1 Result and Discussion

1. Effective Natural Language Understanding with BERT

The BERT model ensures high accuracy in understanding user queries by considering the full context of the input, leading to more precise and meaningful responses. This bidirectional processing capability allows the chatbot to better interpret and respond to complex, nuanced, or ambiguous questions compared to traditional models.

2.Preprocessing and Postprocessing Enhance Model Efficiency

The preprocessing module cleans and formats input data, while post processing refines the output, ensuring that the chatbot generates responses that are both accurate and readable. Proper input and output handling significantly improves the chatbot's ability to deal with different query types and ensures the clarity of the responses.

3. Scalability with Optional Database Integration

The optional database integration allows the chatbot to scale and retrieve relevant information for domain-specific queries, offering personalized and informed responses. A database enhances the chatbot's ability to handle more complex queries and provide relevant data without relying solely on the BERT model's pre-trained knowledge.

4. Streamlined Backend Workflow for Improved User Experience

The well-organized backend workflow, from query reception to response generation, minimizes latency, ensuring the chatbot provides timely answers.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

The development of an intelligent Q&A chatbot for government education schemes in India represents a significant step forward in enhancing accessibility to crucial information for citizens. By leveraging the power of a Large Language Model (LLM) and the BERT model, the chatbot is capable of understanding and interpreting complex questions, even those that may be phrased differently from the dataset. BERT, with its strong contextual understanding, ensures that the chatbot can identify the intent behind the user's queries with a high degree of accuracy. The use of LoRA (Low-Rank Adaptation) for fine-tuning allows for efficient model customization, ensuring that the chatbot is specifically tuned to provide relevant, accurate, and up-to-date information related to various education schemes in India, catering to a wide range of inquiries such as eligibility, application processes, deadlines, and scheme benefits.

The integration of Python as the core language for the model implementation ensures smooth communication between the machine learning algorithms and the backend infrastructure. The LoRA technique further enhances the model's ability to generalize across diverse queries while minimizing the computational overhead, making the solution scalable and responsive. Once a user interacts with the frontend, which is built using HTML, CSS, and JavaScript, the question is passed to the backend, where Django facilitates the communication between the frontend and the machine learning models. Django serves as the framework that processes the input, invokes the fine-tuned models, and delivers an appropriate response from the scheme dataset, ensuring the user receives a seamless and informative experience.

The overall workflow guarantees that the chatbot provides highly relevant and personalized responses. By matching the user's query with relevant entries in the dataset, the chatbot offers concise and accurate answers, even in the case of complex or ambiguous questions. This system empowers citizens by enabling quick and easy access to vital information about government schemes, thereby improving transparency and efficiency. It also reduces the dependency on manual inquiry systems, making information dissemination more efficient and accessible to a broader audience. Ultimately, the integration of LLM, BERT, LoRA, and Django culminates in a robust, intelligent system capable of enhancing public awareness and engagement with government education programs.

7.2 FUTURE ENHANCEMENT:

For the intelligent Q&A chatbot you've developed, the workflow can be enhanced in several ways to improve the user experience, response accuracy, and overall system performance. Below are some suggestions for future enhancements:

1. Improved Model Fine-Tuning

- Use Domain-Specific Data: For better accuracy, fine-tune the model specifically on data that is directly related to government education schemes. This could involve scraping official government websites, PDF documents, or relevant government publications.

2. Contextual Understanding

- Context-Aware Responses: Implementing a contextual conversation mechanism can improve how the chatbot remembers previous questions and can respond in a more human-like, relevant manner. This involves maintaining context for multi-turn conversations.

3. Data Enrichment

- Multilingual Support: Since India is a multilingual country, incorporating support for multiple languages (e.g., Hindi, Tamil, Bengali) will make the chatbot accessible to a wider audience. This can be done by leveraging multilingual pre-trained models.

4. Advanced Query Handling

- Intent Detection & Slot Filling: Use BERT for intent detection (identifying the user's request type) and slot filling (extracting important entities like scheme name, eligibility, etc.). This will help the system understand the question better and extract key details for a more precise answer.

5. Backend Enhancements

- Dynamic Dataset Update: Build a system where government data is automatically scraped and updated periodically, allowing the model to respond with new and revised information as it becomes available.

6. User Experience (Frontend) Enhancements

- Personalization: Offer users a way to create profiles so the chatbot can provide tailored information based on user preferences, location, or previous interactions.

7. Advanced Analytics & Feedback Loop

- User Feedback System: Allow users to rate the responses and provide feedback on the relevance and accuracy of answers. This data can be used to fine-tune the model further.

8. Security and Privacy Considerations

- Data Privacy: Ensure the chatbot complies with data privacy regulations, especially when handling sensitive information such as personal details. Implement anonymization and encryption techniques where necessary.

9. Scalability

- Cloud Integration: As your system grows, integrating with cloud services (AWS, Google Cloud, etc.) can ensure scalability, availability, and improved performance. Use services like Amazon Comprehend or Google Dialog Flow to offload some NLP tasks.

APPENDIX

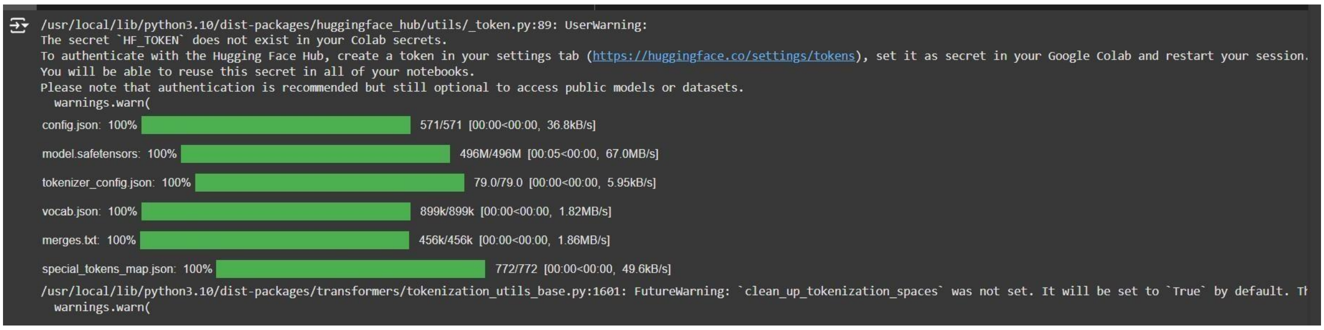
from transformers import

AutoTokenizer,AutoModelForQuestionAnswering,pipeline

model=AutoModelForQuestionAnswering.from_pretrained("deepset/roberta-base-sq u ad2")

tokenizer=AutoTokenizer.from_pretrained("deepset/roberta-base-squad2")

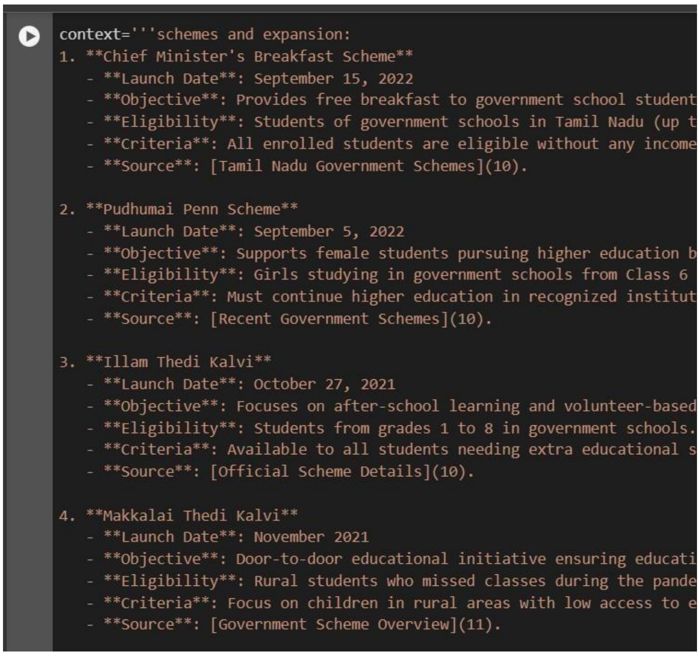
Output of the Model Building:



```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:89: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
config.json: 100% 571/571 [00:00<00:00, 36.8kB/s]
model.safetensors: 100% 496M/496M [00:05<00:00, 67.0MB/s]
tokenizer_config.json: 100% 79.0/79.0 [00:00<00:00, 5.95kB/s]
vocab.json: 100% 899k/899k [00:00<00:00, 1.82MB/s]
merges.txt: 100% 456k/456k [00:00<00:00, 1.86MB/s]
special_tokens_map.json: 100% 772/772 [00:00<00:00, 49.6kB/s]
/usr/local/lib/python3.10/dist-packages/transformers/tokenization_utils_base.py:1601: FutureWarning: `clean_up_tokenization_spaces` was not set. It will be set to `True` by default. Th
  warnings.warn(
```

Fig 7.1 BERT Extraction

A1.2 UPLOAD DATA:



```
context='''schemes and expansion:
1. **Chief Minister's Breakfast Scheme**
   - **Launch Date**: September 15, 2022
   - **Objective**: Provides free breakfast to government school student
   - **Eligibility**: Students of government schools in Tamil Nadu (up t
   - **Criteria**: All enrolled students are eligible without any income
   - **Source**: [Tamil Nadu Government Schemes](10).

2. **Pudhumai Penn Scheme**
   - **Launch Date**: September 5, 2022
   - **Objective**: Supports female students pursuing higher education b
   - **Eligibility**: Girls studying in government schools from Class 6
   - **Criteria**: Must continue higher education in recognized institut
   - **Source**: [Recent Government Schemes](10).

3. **Illam Thedi Kalvi**
   - **Launch Date**: October 27, 2021
   - **Objective**: Focuses on after-school learning and volunteer-based
   - **Eligibility**: Students from grades 1 to 8 in government schools.
   - **Criteria**: Available to all students needing extra educational s
   - **Source**: [Official Scheme Details](10).

4. **Makkalai Thedi Kalvi**
   - **Launch Date**: November 2021
   - **Objective**: Door-to-door educational initiative ensuring educati
   - **Eligibility**: Rural students who missed classes during the pande
   - **Criteria**: Focus on children in rural areas with low access to e
   - **Source**: [Government Scheme Overview](11).
```

Fig 7.2 Upload data

```
ask=pipeline('question-answering',model=model,tokenizer=tokenizer)

result=ask(question="which scheme provide uniform and footwear?",context=context)
```

Output of Response:

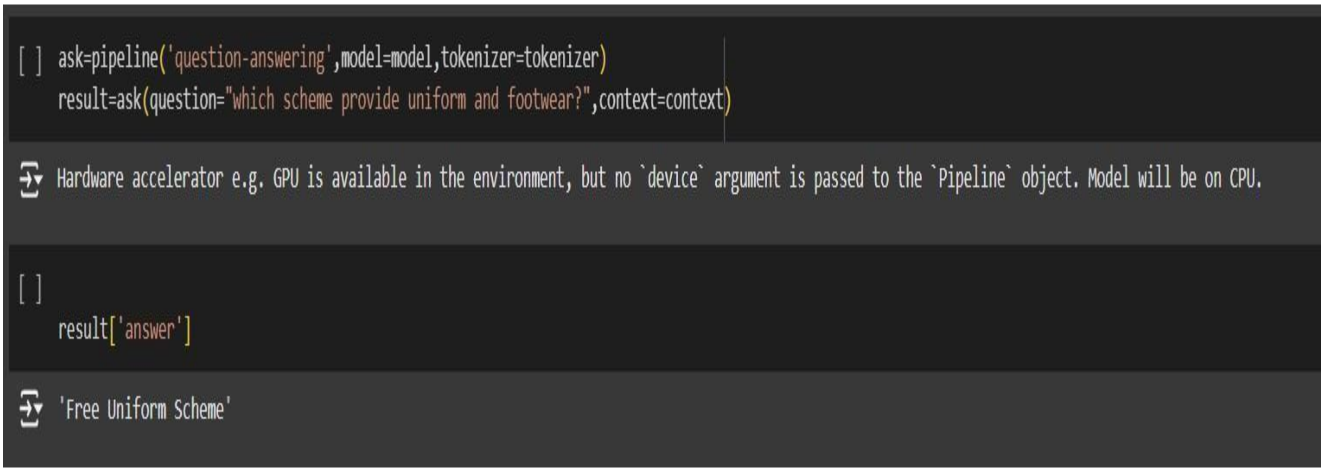


Fig 7.3 Response Generation

A1.3 Web interface sample code:

HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Video Upload</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">
    <script>
        function showLoading() {
            document.getElementById('loading').style.display = 'flex';
        }
    </script>
</head>
<body>
```

```html
    <div class="container">
        <h1>Upload Classroom Video for Analysis</h1>
        <form action="/upload" method="post" enctype="multipart/form-data"
onsubmit="showLoading()">
            <input type="file" name="file" accept="video/*" required>
            <button type="submit">Upload and Analyze</button>
        </form>
    </div>
    <div id="loading" class="loading-overlay" style="display:none;">
        <img src="{{ url_for('static', filename='loading.gif') }}" alt="Loading...">
        <p>Analyzing the video. Please wait...</p>
    </div>
</body>
</html>
```

CSS:

```css
.container:hover {
    transform: translateY(-5px);
    box-shadow: 0 10px 30px rgba(0, 0, 0, 0.2);
}
h1, h2 {
    color: #007bff;
    text-align:
    center;
}
h1 {
    margin-bottom: 20px;
}
```

```
    margin: 20px 0 10px;
}
button {
    padding: 10px
    20px; font-size:
    16px; color: white;
    background-color:
    #007bff; border: none;
    border-radius:
    5px; cursor:
    pointer;
    transition: background-color 0.3s ease, transform 0.2s ease;
}
```

Output of the Web Interface:

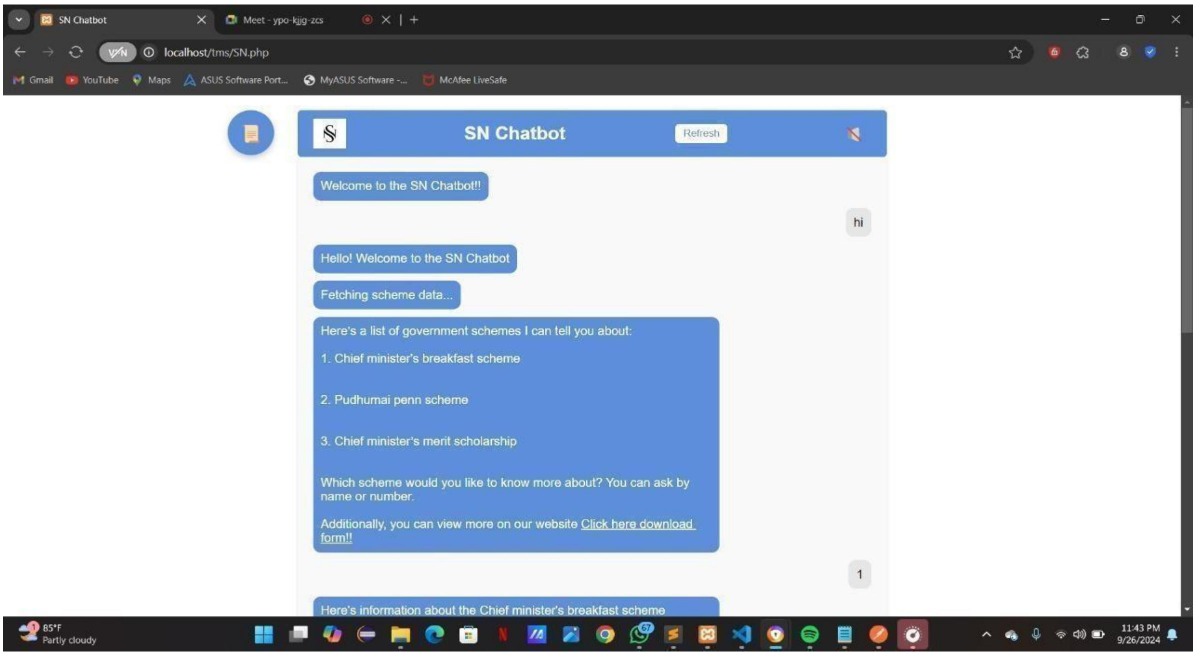


Fig 7.4 Chatbot

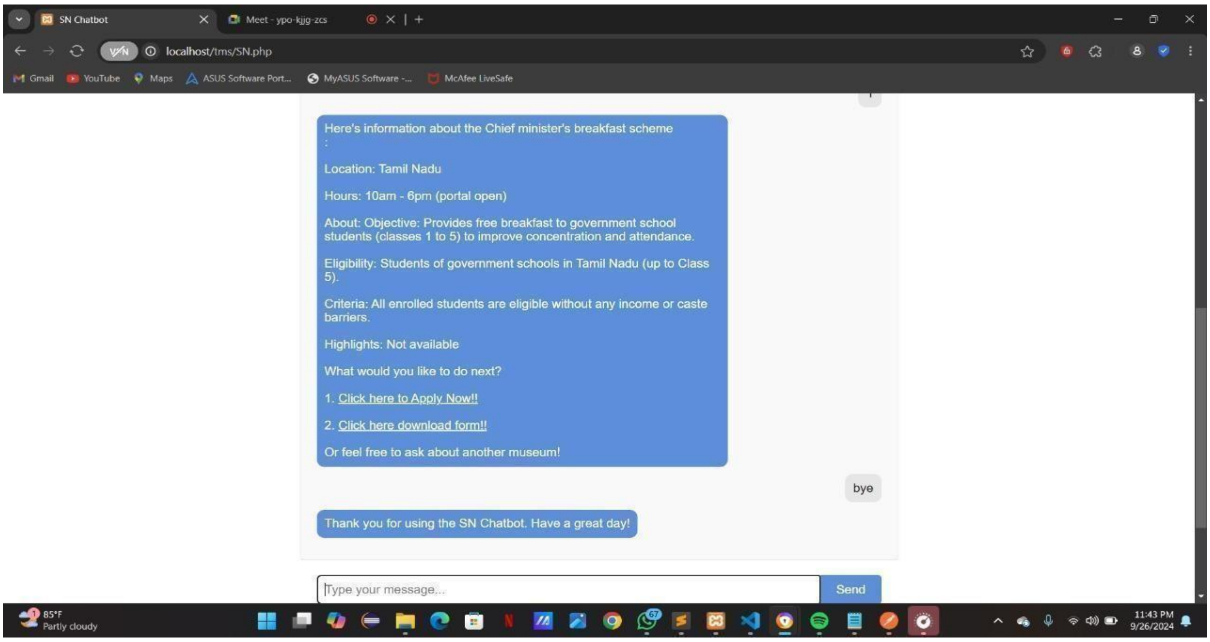


Fig 7.5 Chatbot