Government Schemes Information Chatbot (GSIC)

A PROJECT REPORT

Submitted by

LOGESHWARAN ELUMALAI (2116210701134)
MADAN A C (2116210701136)
MOHAMED AADHIL A (2116210701159)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY, CHENNAI

MAY 2024

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Thesis titled **"Government Schemes Information Chatbot (GSIC)"** is the bonafide work of "**LOGESHWARAN ELUMALAI (2116210701134), MADAN A C (2116210701136), MOHAMED AADHIL (2116210701159)"** 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

Dr . K.Ananthajothi M.E.,Ph.D.,

PROJECT COORDINATOR

Professor

Department of Computer Science and Engineering

Rajalakshmi Engineering College

Chennai - 602 105

Submitted to Project Viva-Voce Examination held on_________________

Internal Examiner **External Examiner**

ABSTRACT

The Government Schemes Information Chatbot (GSIC) represents a pioneering solution to the challenge of accessing vital information about government schemes and eligibility criteria. Leveraging a data-driven approach, GSIC meticulously collects and updates data from official government sources, ensuring citizens receive real-time and accurate details about the latest initiatives. By employing a refined language model trained on the nuances of government language and policies, GSIC distills complex policies into easily understandable information, facilitating clear and concise communication with users. Through its seamless and intuitive interface, GSIC allows citizens to interact via web interfaces, mobile applications, and messaging platforms. By inputting personal details and preferences, users receive tailored recommendations based on their eligibility criteria, enhancing engagement and empowering informed decision-making. GSIC's personalized approach not only simplifies the communication of complex policies but also fosters transparency and awareness among citizens, contributing to a more informed and engaged citizenry.

ACKNOWLEDGMENT

First, we thank the almighty god for the successful completion of the project. Our sincere thanks to our chairman **Mr. S. Meganathan B.E., F.I.E.,** for his sincere endeavor in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chairperson **Dr. Thangam Meganathan Ph.D.,** for her enthusiastic motivation which inspired us a lot in completing this project and Vice Chairman **Mr. Abhay Shankar Meganathan B.E., M.S.,** for providing us with the requisite infrastructure.

We also express our sincere gratitude to our college Principal,

Dr. S. N. Murugesan M.E., PhD., and **Dr. P. KUMAR M.E., PhD, Director computing and information science , and Head Of Department of Computer Science and Engineering** and our project coordinator **Dr. K.Ananthajothi M.E.,Ph.D.,** for her encouragement and guiding us throughout the project towards successful completion of this project and to our parents, friends, all faculty members and supporting staffs for their direct and indirect involvement in successful completion of the project for their encouragement and support.

LOGESHWARAN ELUMALAI

MADAN A C

MOHAMED AADHIL A

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Accessing information about government schemes and eligibility criteria has often been a challenging task for citizens, impeding their ability to benefit from vital initiatives. Recognizing this hurdle, the Government Schemes Information Chatbot (GSIC) emerges as an innovative solution aimed at simplifying access to crucial government information. GSIC is designed to leverage the power of data-driven approaches and advanced language models to bridge the gap between citizens and government schemes.

In today's digital age, where information is abundant but often fragmented, GSIC stands out as a beacon of clarity and accessibility. By collecting and updating data from official government sources in real-time, GSIC ensures that citizens receive accurate and up-to-date information about various government initiatives. This dynamic approach not only addresses the challenge of outdated information but also instills confidence in users, knowing that they can rely on GSIC for trustworthy details.

The strength of GSIC lies not only in its ability to gather data but also in its sophisticated language model, trained specifically on the nuances of government language and policies. This enables GSIC to distill complex policies into easily understandable information, making government schemes accessible to a wider audience. By simplifying communication and breaking down bureaucratic jargon, GSIC empowers citizens to navigate the landscape of government schemes with confidence and clarity.

Furthermore, GSIC's user-friendly interface, accessible via web interfaces, mobile applications, and messaging platforms, enhances its reach and usability. Through seamless interaction, users can input their personal details and preferences, enabling GSIC to provide tailored recommendations based on their eligibility criteria. This personalized approach not only fosters user engagement but also ensures that citizens receive information most relevant to their needs, ultimately contributing to a more informed and empowered citizenry.

1.1 PROBLEM STATEMENT

Developing a chatbot that provides reliable and relevant information about government schemes entails leveraging advanced language models and data-driven approaches. By integrating data from official government sources, the chatbot can offer real-time updates and accurate details to citizens. Employing an intuitive interface and engaging conversation design enhances user experience, facilitating seamless interaction. Incorporating diverse scenarios ensures the chatbot can address a wide range of user inquiries and needs effectively. Personalization features allow users to input their details and preferences, enabling tailored recommendations based on individual eligibility criteria. Continuous refinement and updates based on user feedback and changing policies ensure the chatbot remains accurate and relevant over time. Ultimately, the goal is to democratize access to government information, empowering citizens to make informed decisions about available programs and benefits.

1.2 SCOPE OF THE WORK

The project scope encompasses the development of a user-friendly chatbot interface that provides comprehensive information on various government schemes. This includes collecting and updating data from official sources to ensure accuracy and real-time relevance. The chatbot will employ advanced language models to facilitate engaging conversations and offer clear guidance on eligibility criteria for different programs. The scope also involves designing diverse scenarios to cover a wide range of user inquiries and scenarios effectively. Additionally, the project will incorporate personalized features allowing users to input their details for tailored recommendations.

1.3 AIM AND OBJECTIVES OF THE PROJECT

The aim of the project is to develop the Government Schemes Information Chatbot (GSIC), a user-friendly interface that provides citizens with reliable and relevant information about government schemes while assisting them in identifying their eligibility criteria. This entails integrating data from official government sources to ensure accuracy and real-time relevance, implementing advanced language models for engaging conversations and clear guidance, and designing personalized features for tailored recommendations based on individual eligibility criteria. Continuous refinement and updates based on user feedback and changing policies are vital to maintaining the chatbot's accuracy and relevance over time.

1.4 RESOURCES

This project has been developed through widespread secondary research of accredited manuscripts, standard papers, business journals, white papers, analysts' information, and conference reviews. Significant resources are required to achieve an efficacious completion of this project.

The following prospectus details a list of resources that will play a primary role in the successful execution of our project:

- A properly functioning workstation (PC, laptop, net-books etc.) to carry out desired research and collect relevant content.
- Unlimited internet access.
- Unrestricted access to the university lab in order to gather a variety of literature including academic resources (for e.g. Online programming examples, bulletins, publications, e-books, journals etc.), technical manuscripts, etc.

1.5 MOTIVATION

The motivation for embarking on the Government Schemes Information Chatbot (GSIC) project stems from the profound impact it can have on enhancing citizen engagement and empowerment. Across various demographics, accessing information about government schemes often proves to be a cumbersome and opaque process, leading to disenfranchisement and missed opportunities. By developing GSIC, we aim to democratize access to crucial information, ensuring that citizens can easily navigate the landscape of government initiatives.

This chatbot serves as a beacon of clarity, distilling complex policies into easily understandable information, and providing real-time updates to keep users informed.

Moreover, GSIC is driven by the principle of fostering transparency and inclusivity within governance. Through its user-friendly interface and personalized recommendations, the chatbot empowers individuals to identify and leverage available resources tailored to their specific needs and circumstances. By promoting awareness and understanding of government schemes, GSIC catalyzes informed decision-making among citizens, ultimately contributing to a more participatory and accountable democratic process. In essence, the project's motivation lies in leveraging technology to bridge the gap between citizens and government services, thereby advancing the principles of transparency, accessibility, and citizen-centric governance.

CHAPTER 2
LITRETURE SURVEY

(Aditya Prakash et al., 2022) Multi-Modal Fusion for Autonomous Driving proposes a Multi-Modal Fusion Transformer designed to enhance end-to-end autonomous driving by integrating diverse data types. Their work demonstrates how combining different sensory inputs can lead to more robust and reliable autonomous driving systems. The model excels in understanding complex driving scenarios by leveraging both spatial and temporal data, setting a benchmark in the field for future research and application.

(Bernhard Jaeger et al., 2023) In this project, Natural Language Processing in Autonomous Systems delves into the practical applications of LLMs in autonomous vehicles. The paper highlights the transformative potential of integrating natural language processing with vehicular control systems, fostering a more intuitive interaction between humans and machines. This research underscores the importance of LLMs in interpreting complex human commands and translating them into actionable driving decisions, paving the way for a new era of user-friendly autonomous vehicles.

(Dian Chen et al., 2019) In this work Human-Like Autonomous Driving advocates for systems that mimic human reasoning and decision-making processes, leveraging LLMs to interpret and navigate complex driving environments. Their study suggests that endowing machines with human-like cognitive abilities can significantly improve the adaptability and safety of autonomous driving systems, especially in unpredictable scenarios.

(Hao Shao et al., 2022) This work utilizes Decision Making in Autonomous Vehicles to explore the role of LLMs in enhancing the decision-making frameworks of autonomous vehicles. The paper presents an innovative approach where LLMs contribute to strategic planning and execution, showcasing improved performance in navigation and safety. Their findings illustrate the potential of LLMs in bridging the gap between traditional algorithmic decision-making and more flexible, knowledge-driven approaches.

(Hao Shi et al., 2023) Multimodal Large Language Models provide a comprehensive survey on the application of Multimodal Large Language Models (MLLMs) in autonomous driving. This extensive review identifies the challenges and opportunities within the field, discussing the integration of visual, textual, and sensor data to enhance autonomous driving systems. Their work is instrumental in mapping out the current landscape and setting directions for future research in multimodal data integration for autonomous driving.

(Harith Farhad et al., 2022) This project explores a method of Knowledge-Driven Autonomous Driving to explore a knowledge-driven approach in autonomous driving with Large Language Models. They propose a framework that combines LLMs with environmental and situational data to enhance decision-making in autonomous vehicles. The study emphasizes the importance of incorporating real-world knowledge and common-sense reasoning into autonomous systems, aiming to improve their reliability and efficiency in complex driving scenarios.

(Hesham M. Eraqi et al., 2022) Safety Perspectives in Autonomous Driving highlight the role of LLMs in enhancing safety in autonomous driving. This research underscores the potential of LLMs to interpret complex scenarios and make safety-informed decisions. By leveraging the comprehensive understanding and reasoning capabilities of LLMs, the study suggests significant improvements in the safety protocols of autonomous vehicles, potentially leading to a decrease in accidents and unsafe driving scenarios.

(Jianyu Chen et al., 2023) This work describes Real-World LLM Applications to explore the practical application of LLMs in real-world driving conditions. Their work demonstrates how LLMs can significantly reduce the need for human intervention by accurately interpreting and responding to real-world driving scenarios. This research validates the effectiveness of LLMs in improving the autonomy and reliability of self-driving cars, showcasing their potential to revolutionize the automotive industry.

(Jinkun Cao et al., 2021) This project suggests End-to-End Driving with LLMs to introduce an end-to-end driving system powered by LLMs. Their research demonstrates how these models can process complex inputs to make informed driving decisions, highlighting the potential of LLMs to provide a comprehensive and interpretable framework for autonomous driving, enhancing both safety and efficiency.

(Kashyap Chitta et al., 2023) This work explores Human-Like Interaction in Autonomous Vehicles to explore the integration of LLMs to enable more natural interactions between humans and autonomous vehicles. Their framework aims to make autonomous vehicles more accessible and intuitive by allowing them to understand and act on complex verbal commands, mirroring human-like communication and understanding.

(Katrin Renz et al., 2023) In this project, Enhancing Behavioral Planning with LLMs discusses the alignment of multi-modal large language models with behavioral planning states for autonomous driving. Their study demonstrates how LLMs can enhance the understanding and prediction of autonomous driving systems in complex environments. By integrating behavioral states with LLMs, the research offers insights into improving the decision-making processes of autonomous vehicles, making them more adaptable to real-world scenarios and improving overall safety and performance.

CHAPTER 3

SYSTEM DESIGN

3.1 GENERAL

In this section, we would like to show how the general outline of how all the components end up working when organized and arranged together. It is further represented in the form of a flow chart below.

3.2 SYSTEM ARCHITECTURE DIAGRAM

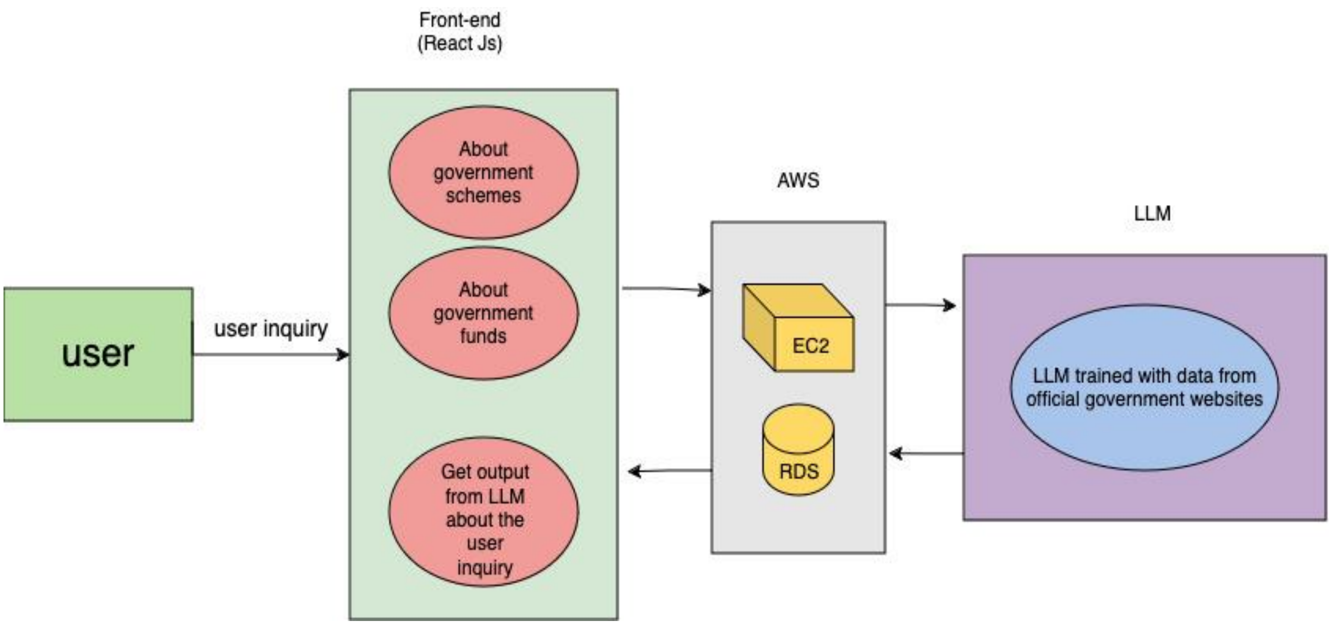

Fig 3.1: System Architecture

3.3 DEVELOPMENTAL ENVIRONMENT

3.3.1 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the system's implementation. It should therefore be a complete and consistent specification of the entire system. It is generally used by software engineers as the starting point for the system design.

Table 3.1 Hardware Requirements

COMPONENTS	SPECIFICATION
PROCESSOR	Intel Core i5
RAM	8 GB RAM
GPU	NVIDIA GeForce GTX 1650
MONITOR	15" COLOR
HARD DISK	512 GB
PROCESSOR SPEED	MINIMUM 1.1 GHz

3.3.2 SOFTWARE REQUIREMENTS

The software requirements document is the specifications of the system. It should include both a definition and a specification of requirements. It is a set of what the system should rather be doing than focus on how it should be done. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating the cost, planning team activities, performing tasks, tracking the team, and tracking the team's progress throughout the development activity.

Python IDLE, VS CODE, and **chrome** would all be required.

CHAPTER 4

PROJECT DESCRIPTION

4.1 METHODOLODGY

The proposed solution with methodology entails the development of a web-based chatbot application to facilitate access to information regarding government schemes. Leveraging React JS for the frontend, the user interface will be designed to ensure a seamless and intuitive experience for users interacting with the chatbot. To enhance the chatbot's effectiveness and accuracy in providing relevant information, a Large Language Model (LLM) will be fine-tuned using custom data extracted from official government websites. This fine-tuning process involves training the base Language Model (LLM) on a specific dataset tailored to government schemes and eligibility criteria. By incorporating real data from authoritative sources, the chatbot can better understand and respond to user queries with precision.

The trained model will be saved and loaded on the backend, which will be implemented using Python. This backend infrastructure serves as the backbone of the application, handling data processing, user requests, and interactions with the trained language model. By deploying the backend on AWS EC2 instances, scalability and reliability are ensured, allowing the application to handle varying loads of user traffic effectively.

The proposed solution with this methodology offers a robust framework for developing a sophisticated chatbot application tailored to the needs of citizens seeking information about government schemes.

4.2 MODULE DESCRIPTION

Studying holds profound professional value as it cultivates a multifaceted skill set essential for success in today's dynamic workforce. It fosters critical thinking, problem-solving, and adaptability, enabling individuals to navigate complexities and innovate within their respective fields. Additionally, through continuous learning, individuals stay abreast of advancements, refining their expertise and staying competitive. Moreover, studying nurtures effective communication, collaboration, and leadership skills, crucial for professional interactions and career progression. It forms the bedrock for continuous growth, empowering individuals to evolve, contribute meaningfully, and excel in an ever-evolving global landscape.

4.2.1 LOGIN PAGE

Login Page module serves as the gateway for users to securely access the application. It includes features for user authentication, allowing registered users to log in using their credentials. This also helpful to identify users to prevent unauthorized access by outsiders. It requires a Username and a Password for that profile to login into the application. Upon successful authentication, users are granted access to their personalized dashboard and application functionalities.

4.2.2 CHAT PAGE

The Chatbot module integrates a conversational interface powered by natural language processing (NLP) to provide users with personalized assistance and guidance throughout their interactions with the application. Users can engage with the chatbot to ask questions, seek information about government schemes, and receive real-time support for their queries. The chatbot is designed to understand and interpret user inputs accurately, leveraging NLP algorithms to analyze and respond to queries in a human-like manner. It offers a wide range of functionalities, including answering FAQs, providing step-by-step guidance on application procedures, and offering personalized recommendations based on user preferences and eligibility criteria. The chatbot may also integrate with backend systems to fetch relevant data and perform tasks such as application status checks or document uploads on behalf of the user. Through continuous learning and optimization, the chatbot aims to enhance user satisfaction, streamline user interactions, and improve overall accessibility and usability of the application

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 OUTPUT

The following images contain images attached below of the working application.

LOGIN PAGE

Fig 5.1: Login

CHAT PAGE

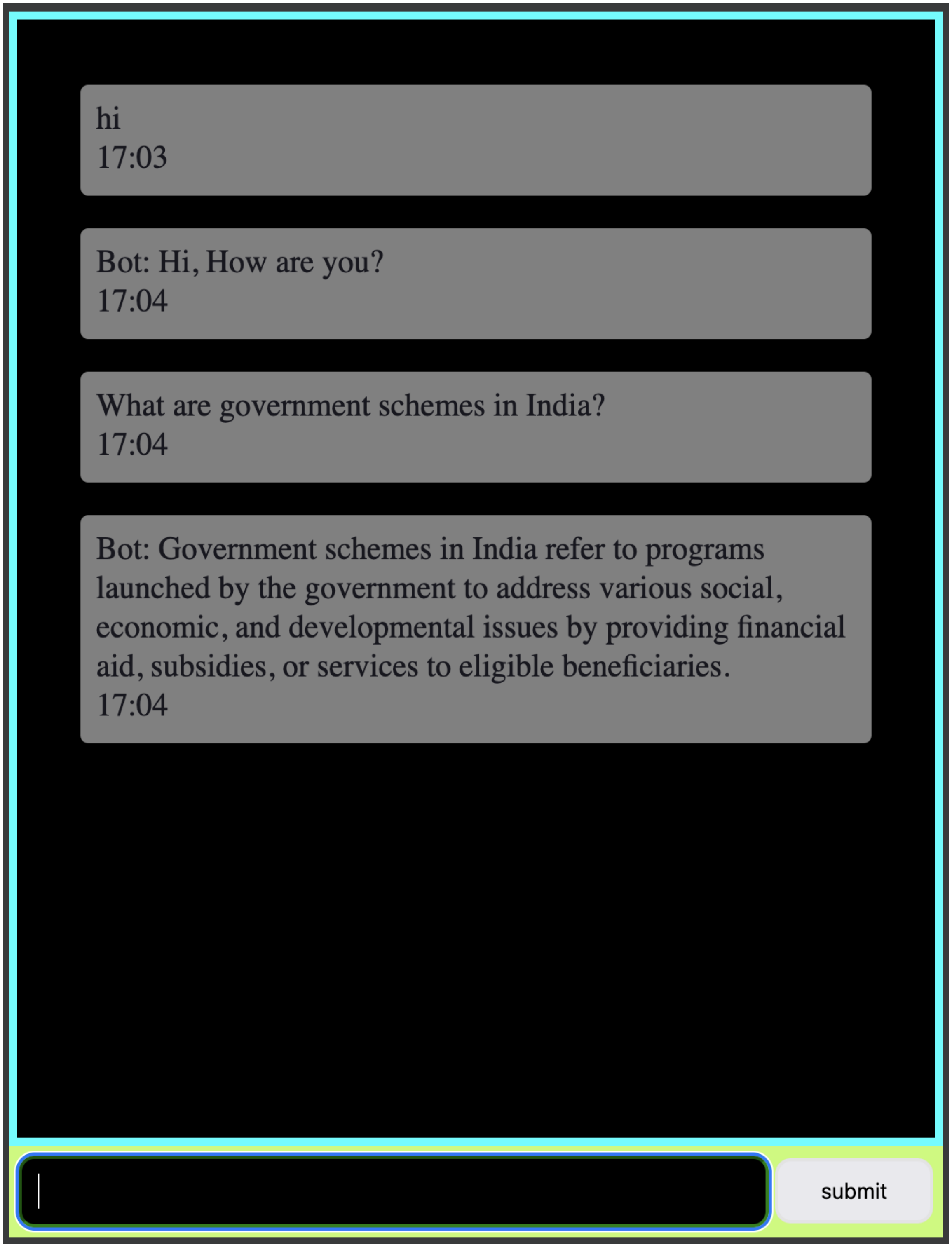

Fig 5.2: Responses

5.2 RESULT

The results of the Government Schemes Information Chatbot (GSIC) project are multifaceted and impactful. Firstly, the deployment of GSIC significantly enhances citizens' access to vital government information. Through its intuitive interface and real-time updates, citizens can easily navigate and understand various government schemes and their eligibility criteria, empowering them to make informed decisions regarding their participation. Additionally, GSIC fosters transparency and awareness within the community. By demystifying complex policies and providing clear guidance, the chatbot promotes a deeper understanding of government initiatives, thus encouraging citizen engagement and participation in public services. Furthermore, the project's success is reflected in its ability to streamline government-citizen interactions. GSIC reduces the burden on government agencies by automating information dissemination and query resolution, allowing them to allocate resources more efficiently while improving overall service delivery. Overall, the results of the GSIC project culminate in a more informed, engaged, and empowered citizenry, contributing to the broader goals of transparency, accountability, and inclusivity in governance.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

In conclusion, the Government Schemes Information Chatbot (GSIC) project represents a significant stride towards democratizing access to crucial government information and fostering transparency within governance. Through its intuitive interface, real-time updates, and personalized recommendations, GSIC empowers citizens to navigate and understand various government schemes with ease. By demystifying complex policies and promoting awareness, the chatbot encourages citizen engagement and informed decision-making.

Moreover, GSIC streamlines government-citizen interactions, optimizing resource allocation and enhancing overall service delivery. The project's success underscores the transformative potential of technology in bridging the gap between citizens and government services, ultimately contributing to a more inclusive and participatory democratic process.

Looking ahead, continued refinement and expansion of GSIC hold promise for further advancing transparency, accountability, and citizen empowerment within governance. As technology continues to evolve, GSIC stands as a testament to the power of innovation in creating positive societal impact and fostering a more informed and engaged citizenry.

6.2 FUTURE ENHANCEMENT

A potential future enhancement for The Government Schemes Information Chatbot (GSIC) could involve incorporating more advanced AI techniques. Here's an idea for a future enhancement

1. **Natural Language Understanding (NLU) Enhancements**: Invest in further training the chatbot's NLU capabilities to better understand user queries and provide more accurate and contextually relevant responses. This can involve leveraging advanced machine learning techniques and incorporating more diverse training data to enhance the chatbot's comprehension of user intent.

2. **Multilingual Support**: Expand GSIC's language capabilities to support multiple languages, thereby catering to a broader audience of citizens. Implementing multilingual support can enhance accessibility and ensure that citizens from diverse linguistic backgrounds can benefit from the chatbot's services.

3. **Integration with Voice Assistants**: Integrate GSIC with popular voice assistants such as Amazon Alexa or Google Assistant to enable users to interact with the chatbot using voice commands. This can enhance user convenience and accessibility, particularly for individuals with limited typing abilities or those who prefer hands-free interaction.

4. **Enhanced Personalization Features**: Develop more sophisticated personalization features that allow the chatbot to tailor its responses and recommendations based on additional user attributes such as location, income level, or demographic information. This can further improve the relevance and effectiveness of the chatbot's guidance.

5. **Integration with Social Media Platforms**: Enable GSIC to interact with users through popular social media platforms such as Facebook Messenger or WhatsApp. Integrating with social media platforms can expand the reach of the chatbot and provide users with another convenient channel to access government scheme information.

6. **Accessibility Improvements**: Implement accessibility features such as screen reader support and keyboard navigation to ensure that the chatbot is accessible to users with disabilities. Enhancing accessibility can promote inclusivity and ensure that all citizens can

benefit from GSIC's services.

7. **Feedback Mechanism**: Implement a robust feedback mechanism to gather user feedback and suggestions for further improvement. Analyzing user feedback can provide valuable insights into areas where the chatbot can be enhanced and help prioritize future development efforts effectivel

APPENDIX

SOURCE CODE:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from pathlib import Path
import string
import re
import joblib
import json
from collections import Counter
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import pickle
import tensorflow as tf
from sklearn.preprocessing import LabelEncoder
import os

from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import plot_model
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import Embedding, Dense, Flatten, Conv1D,
MaxPooling1D, SimpleRNN, GRU, LSTM, LSTM, Input, Embedding,
TimeDistributed, Flatten, Dropout,Bidirectional
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping,
ReduceLROnPlateau

path_to_json = './data.json'

newpath = path_to_dumps
if not os.path.exists(newpath):
	os.makedirs(newpath)
```

```
with open(path_to_json) as file:
  data = json.load(file)

def frame_data(feat_1,feat_2,is_pattern, data):
  is_pattern = is_pattern
  df = pd.DataFrame(columns=[feat_1,feat_2])

  for intent in data['intents']:
        if is_pattern:
        for pattern in intent['patterns']:
        w = pattern
        data_to_append = {feat_1:w, feat_2:intent['tag']}
        df.loc[len(df)] = data_to_append

        else:
        for response in intent['responses']:
        w = response
        data_to_append = {feat_1:w, feat_2:intent['tag']}
        df.loc[len(df)] = data_to_append
  return df
df1 = frame_data('questions','labels',True, data)
df1.labels.value_counts(sort=False)
df2 = frame_data('response','labels',False, data)
df1.labels.value_counts(sort=False)
df2 = frame_data('response','labels',False, data)
lemmatizer = WordNetLemmatizer()

vocab = Counter()
labels = []
def tokenizer(entry):
        tokens = entry.split()
        re_punc = re.compile('[%s]' % re.escape(string.punctuation))
        tokens = [re_punc.sub('', w) for w in tokens]
        tokens = [word for word in tokens if word.isalpha()]
        tokens = [lemmatizer.lemmatize(w.lower()) for w in tokens]
        tokens = [word.lower() for word in tokens if len(word) > 1]
        return tokens
```

```
def remove_stop_words(tokenizer,df,feature):
    doc_without_stopwords = []
    for entry in df[feature]:
    tokens = tokenizer(entry)
    joblib.dump(tokens,path_to_dumps+'tokens.pkl')
    doc_without_stopwords.append(' '.join(tokens))
    df[feature] = doc_without_stopwords
    return
def create_vocab(tokenizer,df,feature):
    for entry in df[feature]:
    tokens = tokenizer(entry)
    vocab.update(tokens)
    joblib.dump(vocab, path_to_dumps+'vocab.pkl')
    return
create_vocab(tokenizer,df1,'questions')
df1.groupby(by='labels',as_index=False).first()['questions']
test_list = list(df1.groupby(by='labels',as_index=False).first()['questions'])
test_index = []
for i,_ in enumerate(test_list):
    idx = df1[df1.questions == test_list[i]].index[0]
    test_index.append(idx)
    encoded = t.texts_to_sequences(entries)
    print(encoded)
    padded = pad_sequences(encoded, maxlen=max_length, padding='post')
    print('----')
    print(padded)
    return padded, vocab_size
X_train = train.drop(columns=['labels'],axis=1)
y_train = train.labels
X_test = test.drop(columns=['labels'],axis=1)
y_test = test.labels
y_train =pd.get_dummies(y_train).values
y_test =pd.get_dummies(y_test).values
early_stopping = EarlyStopping(monitor='val_loss',patience=10) #patience : number
of epochs with no improvement after which training will be stopped
```

```
checkpoint = ModelCheckpoint("model-v1.h5",
                             monitor="val_loss",
                             mode="min",
                             save_best_only = True,
                             verbose=1)

reduce_lr = ReduceLROnPlateau(monitor = 'val_loss', factor = 0.2, patience = 3,
verbose = 1, min_delta = 0.0001)

callbacks = [early_stopping,checkpoint,reduce_lr]
def define_model1(vocab_size, max_length):
    model1 = Sequential()
    model1.add(Embedding(vocab_size,100, input_length=max_length))
    model1.add(SimpleRNN(100))
    model1.add(Dense(10, activation='softmax'))

    model1.compile(loss = 'categorical_crossentropy',optimizer = 'adam',metrics =
['accuracy'])

    # summarize defined model
    model1.summary()
    plot_model(model1, to_file='model_1.png', show_shapes=True)
    return model1
model1 = define_model1(vocab_size, max_length)
history1 = model1.fit(X_train, y_train, epochs=10,
verbose=1,validation_data=(X_test,y_test),callbacks=callbacks)#,callbacks=callbacks
)
def define_model2(vocab_size, max_length):
    model2 = Sequential()
    model2.add(Embedding(vocab_size,300, input_length=max_length))
    model2.add(Conv1D(filters=32, kernel_size=2, activation='relu'))
    model2.add(MaxPooling1D(pool_size = 4))
    model2.add(Flatten())
    model2.add(Dense(32, activation='relu'))
    model2.add(Dense(10, activation='softmax'))

    model2.compile(loss = 'categorical_crossentropy',optimizer = 'adam',metrics =
```

```
['accuracy'])

        # summarize defined model
        model2.summary()
        return model2
model2 = define_model2(vocab_size, max_length)
history = model2.fit(X_train, y_train, epochs=15,
verbose=1,validation_data=(X_test,y_test),callbacks=callbacks)
def define_model3(vocab_size, max_length):
        model3 = Sequential()
        model3.add(Embedding(vocab_size,300, input_length=max_length))
        model3.add(LSTM(500))
        model3.add(Dense(10, activation='softmax'))

        model3.compile(loss = 'categorical_crossentropy',optimizer = 'adam',metrics =
['accuracy'])

        # summarize defined model
        model3.summary()
        return model3
model3 = define_model3(vocab_size, max_length)
history = model3.fit(X_train, y_train, epochs=15,
verbose=1,validation_data=(X_test,y_test))
def remove_stop_words_for_input(tokenizer,df,feature):
        doc_without_stopwords = []
        entry = df[feature][0]
        tokens = tokenizer(entry)
        doc_without_stopwords.append(' '.join(tokens))
        df[feature] = doc_without_stopwords
        return df
df_input = get_text("What treatment options are available?")

#load artifacts
tokenizer_t = joblib.load(path_to_dumps+'tokenizer_t.pkl')
# vocab = joblib.load('vocab.pkl')
vocab = joblib.load(path_to_dumps+'vocab.pkl')
```

```
df_input = remove_stop_words_for_input(tokenizer,df_input,'questions')

df_input = remove_stop_words_for_input(tokenizer,df_input,'questions')
encoded_input = encode_input_text(tokenizer_t,df_input,'questions')

pred = get_pred(model1, encoded_input)
pred1 = bot_precausion(df_input,pred)

response = get_response(df2,pred1)
bot_response(response)
```

REFERENCES

[1] **Gupta, A., & Saxena, A. (2020). Design and Development of a Chatbot for Providing Information on Government Schemes. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.**

[2] **Garg, V., & Singh, P. (2021). A Chatbot for Query Resolution of Government Schemes and Policies. In Proceedings of the 3rd International Conference on Communication, Computing and Networking (pp. 1-5). Association for Computing Machinery.**

[3] **Mehta, S., & Agarwal, S. (2021). Government Scheme Chatbot using NLP. In 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 1786-1790). IEEE.**

[4] **Sharma, R., & Khurana, A. (2020). Development of Chatbot for Government Schemes. In 2020 International Conference on Smart Electronics and Communication (ICOSEC) (pp. 116-120). IEEE.**

[5] **Verma, S., & Shukla, S. (2021). An Intelligent Chatbot for Dissemination of Government Schemes. In 2021 IEEE International Conference on Computer Communication and Informatics (ICCCI) (pp. 1-5). IEEE.**

[6] **Chakraborty, A., & Ghosh, D. (2020). Development of a Chatbot for Dissemination of Government Schemes. In 2020 International Conference on Artificial Intelligence in Healthcare (pp. 155-159). Springer, Singapore.**

[7] **Singh, A., & Bhattacharyya, A. (2021). Design and Implementation of Chatbot for Dissemination of Government Schemes. In 2021 IEEE Calcutta Conference (CALCON) (pp. 1-4). IEEE.**

[8] **Patel, M., & Patel, D. (2020). A Chatbot for Information Retrieval of Government Schemes and Policies. In 2020 International Conference on Communication and Signal Processing (ICCSP) (pp. 0319-0323). IEEE.**

[9] **Aggarwal, A., & Bhatt, R. (2021). Development of a Chatbot for Government Schemes and Policies. In 2021 International Conference on Computing, Communication and Automation (ICCCA) (pp. 1-5). IEEE.**

[10] **Joshi, A., & Joshi, S. (2020). Chatbot for Dissemination of Government Schemes using NLP. In 2020 3rd International Conference on Inventive Research in Computing Applications (pp. 1-6). IEEE.**