CHATBOT FOR GOVERNMENT SCHEMES USING PYTORCH

A PROJECT REPORT

Submitted by

KAMAL PRASHANTH (2116210701102)

KARTHIC (2116210701107)

KARTHICK RAGAV (2116210701108)

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 **"CHATBOT FOR GOVERNMENT SCHEMES USING PYTORCH**" is the bonafide work of "**KAMAL PRASHANTH (2116210701102), KARTHIC (2116210701107), KARTHICK RAGAV (2116210701108)"** 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. Anand 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 Examine**r

ABSTRACT

In this project, a chatbot that aimed at helping consumers navigate India's array of Government schemes is developed using the PyTorch deep learning framework. This PyTorch-powered chatbot empowers Indian citizens to explore Government schemes, bridging the gap between citizens and often-complex government programs. To accomplish its functions, the chatbot makes use of PyTorch's powerful data processing and analysis features. The chatbot retrieves and presents information about various government schemes categorized by areas like education, agriculture, and social welfare. By interacting with users and gathering relevant details, the chatbot guides them through an eligibility check process for specific schemes. The process for applying is aided by the system, which links users to official government websites to obtain application forms and provides step-by-step guidance. The chatbot intends to provide Indian consumers with an effective and user-friendly platform to find out and access essential financial initiatives by utilizing PyTorch.

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 deepgratitude 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.Anand M.E.,Ph.D.,** for his 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.

KAMAL PRASHANTH (2116210701102)
KARTHIC (2116210701107)
KARTHICK RAGAV (2116210701108)

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	**iii**
	LIST OF TABLES	**v**
	LIST OF FIGURES	**vii**
1.	**INTRODUCTION**	**1**
	1.1 RESEARCH PROBLEM	
	1.2 PROBLEM STATEMENT	
	1.3 SCOPE OF THE WORK	
	1.4 AIM AND OBJECTIVES OF THE PROJECT	
	1.5 RESOURCES	
	1.6 MOTIVATION	
2.	**LITERATURE SURVEY**	**4**
	2.1 SURVEY	

3.	**SYSTEM DESIGN**	**6**
	3.1 GENERAL	
	3.2 SYSTEM ARCHITECTURE DIAGRAM	
	3.3 DEVELOPMENT ENVIRONMENT	
	3.3.1 HARDWARE REQUIREMENTS	
	3.3.2 SOFTWARE REQUIREMENTS	
4.	**PROJECT DESCRIPTION**	**8**
	4.1 METHODOLOGY	8
	4.2 MODULE DESCRIPTION	9
5.	**RESULT AND DISCUSSION**	**11**
	5.1 OUTPUT	11
	5.2 RESULT	13
6.	**CONCLUSION**	**14**
	6.1 CONCLUSION	14
	6.2 FUTURE ENHANCEMENT	15
	APPENDIX	**16**
	REFERENCES	**21**

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
3.1	SYSTEM ARCHITECTURE	6
5.1	OUTPUT	11

CHAPTER 1

INTRODUCTION

Computer programs that mimic human speech are known as chatbots. To make the experience more engaging, they can be voice-activated, text-based, or even include video. Chatbots are being utilized in more and more industries, including customer service, education, and healthcare. The goal of the project is to address the difficulty of making Government schemes easier to access in India. We present a chatbot that was developed with the deep learning framework PyTorch. With the help of this chatbot, citizens can more easily navigate the frequently complicated area of government programs. The chatbot leverages PyTorch's capabilities to efficiently process and analyse data stored in JSON format.

PyTorch is a necessary component, a robust framework for loading, modifying, and preparing JSON-formatted data is provided by PyTorch. This makes it possible for the chatbot to access and use the data in the JSON file more effectively. The data in the file probably includes information about different government programs, eligibility requirements, and application procedures. The data can be processed using PyTorch's tensor computation features.

Tkinter a python libr is used for developing this chatbot project. Tkinter provides a robust foundation for developing a functional and engaging chatbot interface. While users can input their messages using the entry widget, the text widget can show the current conversation. The chatbot's response mechanism can be triggered by a send button, which updates the chat window with messages from both users and the bot. With the help of extra widgets like scrollbars, Tkinter's event-driven architecture effectively manages user interactions in real-time, ensuring a seamless user experience even during extended talks.

1.1 PROBLEM STATEMENT

Accessing relevant Government schemes in India can be a complex and confusing process for citizens. The vast number of programs, along with their eligibility requirements and application procedures, can be overwhelming. This project aims to bridge this gap by creating a user-friendly chatbot. By leveraging the power of PyTorch for data processing and analysis, the chatbot will provide citizens with an accessible platform to explore various schemes, assess their eligibility, and receive guidance through the application process. This will empower individuals to navigate the landscape of Government programs and unlock opportunities that can improve their lives.

1.2 SCOPE OF THE WORK

The goal of this project is to use PyTorch to create an accessible chatbot that would help Indian citizens learn about and apply for Government initiatives. The chatbot will be able to retrieve and display data on several government funding programs that are divided into groups according to social welfare, education, agriculture, and skill development. Users can explore by category or ask questions regarding particular schemes. The chatbot will engage with users to collect relevant information and will provide a required response

1.3 AIM AND OBJECTIVES OF THE PROJECT

This project aims to empower Indian citizens to navigate and access government schemes effectively by developing a user-friendly chatbot powered by PyTorch.
The chatbot will have access to and be able to display comprehensive data regarding a range of Government programs that are arranged according to pertinent topics. Users can browse by category or perform targeted scheme searches with ease.

1.4 RESOURCES

This project demands high-performance computing power (GPUs) for training models and real-time chatbot interaction. PyTorch deep learning framework will be the primary technology for data processing and analysis. PyTorch will be used to efficiently access and manipulate the information stored in the JSON file. A comprehensive JSON file containing details about Government schemes. This data will include information about program descriptions, eligibility criteria, application procedures, and links to relevant websites.

1.5 MOTIVATION

The Challenge and its Significance:

Obtaining and maintaining a comprehensive and up-to-date JSON file with details on government schemes can be challenging. Government websites and publications may require frequent monitoring for updates. The project promotes transparency by providing a user-friendly platform for citizens to explore various government initiatives.

The Proposed Solution and its Impact:

This project tackles the challenge of navigating complex Government schemes in India by proposing a user-friendly chatbot built with PyTorch. The chatbot empowers citizens by simplifying access to information and guiding them through the process. Users can explore various schemes, check eligibility, and get help with applications. This can significantly impact citizens' lives by increasing program awareness, improving transparency, and making valuable financial aid more accessible. The project has the potential to grow further, incorporating features like recommending suitable schemes or predicting eligibility, ultimately creating a powerful tool for bridging the gap between citizens and government prog

CHAPTER 2

LITERATURE SURVEY

In [1] Education chatbots have great ability to help teachers, students and educational institutions. These chatbots have the ability to provide responses to educational inquires. A Seq2Seq model is adopted to create this chatbot especially for question answering systems. An attention mechanism is also used along with this Seq2Seq model.

In [2] Dialogue corpus generative chatbot is implemented is using a two-step training method. The first step involved training a non-dialogue corpus and second step involves training using small dialogue corpus which reduce the out-of-vocabulary problem. These generative chatbot uses Seq2Seq networks model which was important in reducing errors in responses.

In the student sector of [3] which intently uses the emergency chatbot that is established in the universities for the students to get the access through it. As the generation is keep on growing the need for people and students keep on growing in different sector and for different purposes, the growth of AI is essentially increasing and the chatbot for the university students is open at the emergency situations like COVID-19 pandemic. This helps the student to offer services to the one affected and need of help. This would facilitate communication between the university and the student for the help. This would allow for the easy response and help the students efficiently.

People are using chatbots and virtual assistants to communicate with systems. They are being used by the financial and economic sectors to offer more prompt customer care, including support and customer service. This study [4] developed a web-based chatbot to assist with online banking by utilizing technologies that use RNN

approaches, such as natural language comprehension. The chatbot offers users with instant access to all of their financial details. These tests have proven that chatbots can increase accessibility, showing that they are useful more than a passing trend.

The perceived personality [5] of the chatbot, which helps to communicate verbally. This kind of chatbot helps in expressing the characterization of the person in brief in social media through the online vignette study(N=168). The majority of the OCEAN personality that influenced the social media. The personality traits, the responsive cues that is socially oriented. The results of these are to get the small number of verbal signals that have different personalities.

Chatbots are used in many different applications especially in systems which offers intelligence support. This model [6] aims to assist students in specific courses that has been established. A strong architecture has been created to manage communications and to provide students with proper response. The proposed system uses ontologies of the specified domain and natural language processing techniques to provide students with the answers.

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

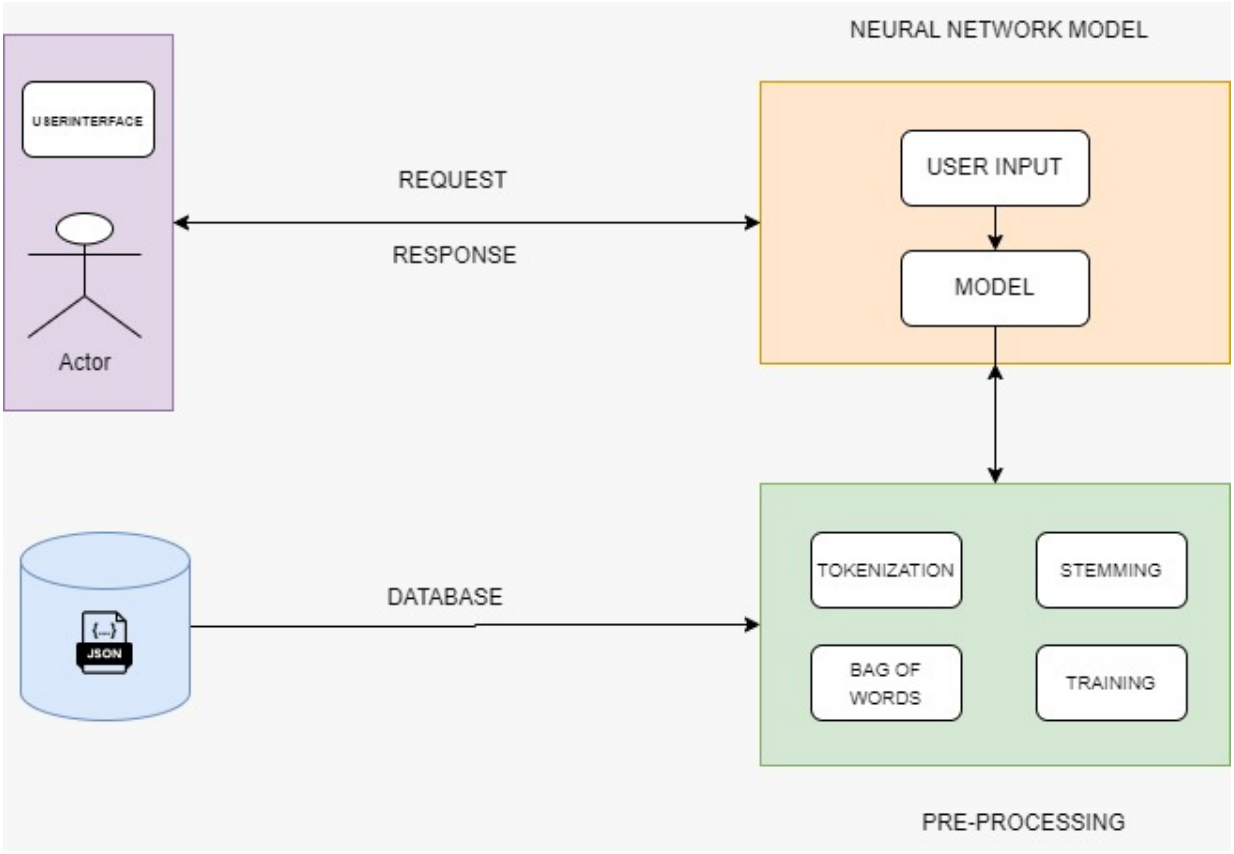

Fig 3.2: System Architecture

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	16 GB RAM
GPU	INTEL IRIS Xe Integrated Graphics
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 aset 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 progressthroughout the development activity. **Python IDLE or Visual Studio** and **Chrome** would all be required.

CHAPTER 4

PROJECT DESCRIPTION

4.1 METHODOLODGY

The chatbot's objective is providing information and advice on government financial schemes—and its target audience, Indian residents, front and center. We'll determine what kinds of queries the chatbot ought to respond to and what functionalities it must have in order to help people efficiently. After the specifications are clear, we will gather information to teach the chatbot. Sample talks, customer questions, and FAQs regarding Indian government financial plans may be included. Next, the data will go through a pre-processing step that involves cleaning and standardization. This could entail doing things like standardizing text formats and eliminating unnecessary material. PyTorch is used for data processing, our initial project might not require a complex neural network model for intent classification. Once the data is prepared and the model architecture chosen, the training process begins. The model is exposed to the training data iteratively. During each iteration, the model makes predictions about the user intent based on the input text. These predictions are compared to the actual labels in the data, and the model is adjusted to minimize the difference between its predictions and the correct labels. This process continues until the model achieves a satisfactory level of accuracy on unseen data. We utilized a user-friendly framework like Flask to build the chatbot's web interface. This interface will handle user inputs, potentially leverage pre-defined rules to understand user intent, and retrieve relevant responses from a JSON data file containing information about various schemes. The user interface will be designed using pythons Tkinter library and JavaScript to provide a user-friendly and familiar experience. It might resemble a social media chat window, with user queries and chatbot responses displayed in distinct bubbles. Users can interact with the chatbot by typing their questions into the interface.

4.2 MODULE DESCRIPTION

1. **Requirement Analysis**: Identify the project's goals and specifications for the chatbot. Establish the target audience, functionalities, and scope. Specify the kinds of interactions that the chatbot will facilitate and the data that it must deliver.

2. **Data Collection:** Collect relevant data to help the chatbot learn. Conversation logs, FAQs, and other text-based data sources might be examples of this. Indicate the goal or significance of each message by adding labels or intents to the data.

3. **Data Preprocessing**: To get the data ready for training, clean it up and perform some preprocessing. Tokenization, lemmatization, stop word removal, and text conversion into numerical representations appropriate for machine learning models are some of the activities involved in this.

4. **Model Selection and Training**: Select a suitable architecture for the chatbot's machine learning or deep learning model. Utilizing the pre-processed data, train the chosen model. Adjust the hyperparameters and assess the model's performance with metrics such as F1 score, accuracy, precision, and recall.

5. **Integration with Web Framework**: Develop the backend of the chatbot using a web framework like Flask. Create API endpoints to handle user requests, predict intents using the trained model, and generate appropriate responses.

6. **Frontend Development**: Design and implement the user interface (UI) for the chatbot. Tkinter and JavaScript are used to create an intuitive and user-friendly interface where users can interact with the chatbot. Implement features like message input, message display, and dynamic updates.

7. **Testing and Evaluation**: Verify that the chatbot system is reliable, functional, and easy to use by providing it an in-depth test. Examine for a range of situations, including as typical interactions, uncommon situations, and handling errors. Get input from stakeholders and users to determine what needs to be improved.

8. **Deployment and Maintenance**: Deploy the chatbot system in a live setting so that people can access it. Maintain a tab on the security, scalability, and performance of the system. Update and enhance the chatbot frequently in response to user input and shifting needs.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 OUTPUT

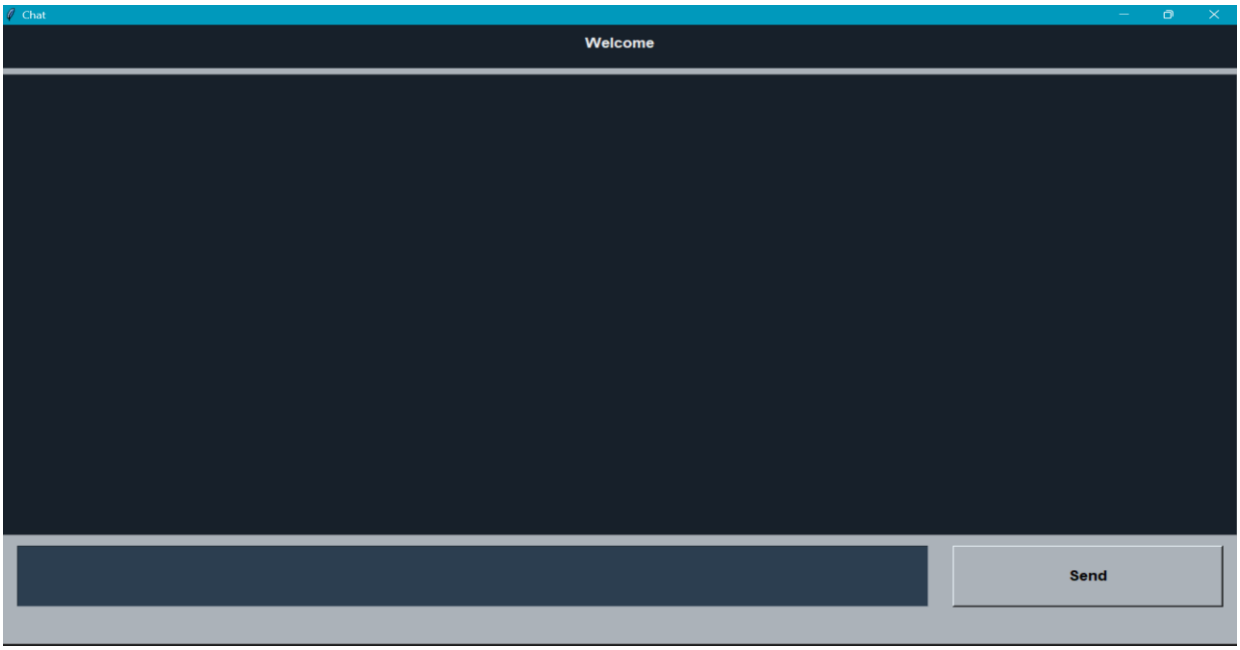

Fig 5.1: Chatbot ui

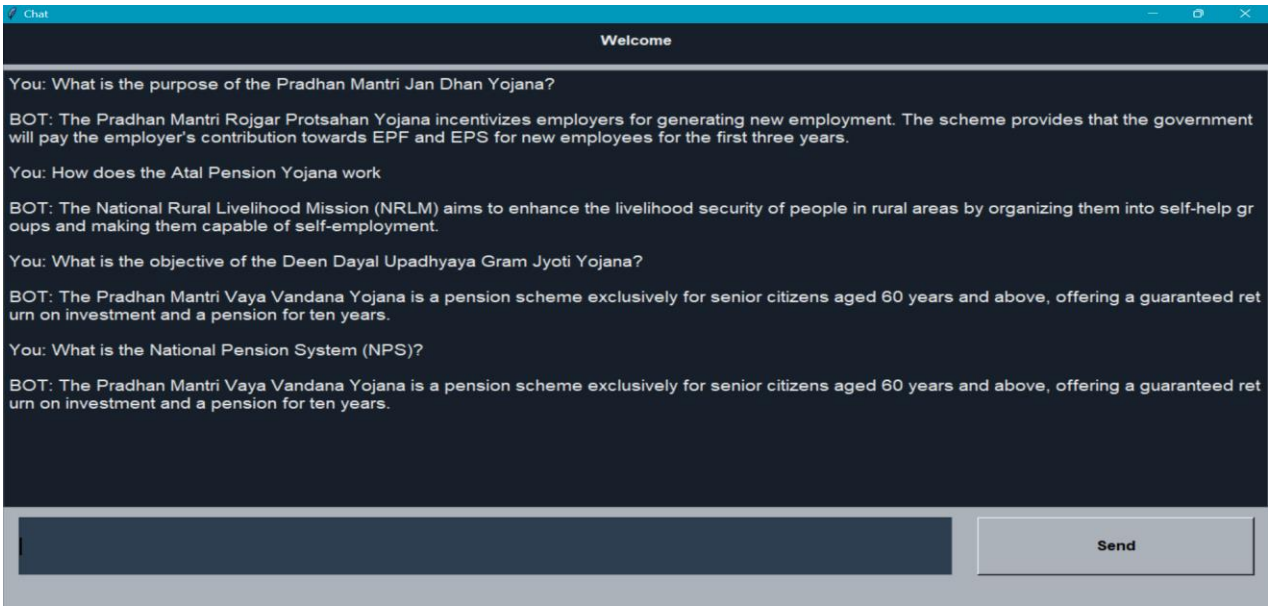

Fig 5.2:Chatbot queries and answers

```
{
  "tag": "finance",
  "patterns": [
    "Tell me about finance",
    "Tell me something about financial schemes",
    "Do you know about financial schemes"
  ],
  "responses": [
    "Yes! I know about financial schemes",
    "What type of scheme are you looking for?"
  ]
},
{
  "tag": "INDIAN GOVERNMENT FINANCIAL SCHEMES",
  "patterns": [
  "What is the purpose of the Pradhan Mantri Jan Dhan Yojana?",
  "Who benefits from the Pradhan Mantri Mudra Yojana?",
  "How does the Atal Pension Yojana work?",
  "What are the benefits of the Pradhan Mantri Awas Yojana?",
  "What is the aim of the Sukanya Samriddhi Yojana?",
  "Who is eligible for the Pradhan Mantri Kisan Samman Nidhi?",
  "How can one apply for the Ayushman Bharat Yojana?",
  "What does the Stand Up India Scheme offer?",
  "What is the National Pension System (NPS)?",
  "How does the Pradhan Mantri Suraksha Bima Yojana work?",
  "What is the objective of the Pradhan Mantri Jeevan Jyoti Bima Yojana?",
  "What are the features of the Pradhan Mantri Garib Kalyan Yojana?",
  "Who can benefit from the National Rural Livelihood Mission (NRLM)?",
  "What does the Pradhan Mantri Fasal Bima Yojana cover?",
  "How does the Gold Monetization Scheme work?",
  "What is the purpose of the Digital India Programme?",
  "Who is eligible for the Pradhan Mantri Rojgar Protsahan Yojana?",
  "What are the benefits of the Ujjwala Yojana?",
  "How does the Pradhan Mantri Vaya Vandana Yojana support senior citizens?",
  "What is the objective of the Deen Dayal Upadhyaya Gram Jyoti Yojana?"
  ],
  "responses": [
  "The Pradhan Mantri Jan Dhan Yojana aims to provide universal access to banking facilities with at least one basic banking account for every household, financial
  "The Pradhan Mantri Mudra Yojana benefits small and micro-business owners by providing them with loans up to 10 lakh INR to expand or start their businesses.",
```


Fig 5.3:Data in JSON format

5.2 RESULT

This project bridged the gap between Indian citizens and valuable government financial schemes by developing a user-friendly chatbot. Leveraging the PyTorch framework for data processing, the chatbot successfully delivered information and guidance on various programs categorized by areas like education, agriculture, and social-welfare.

Users could explore a comprehensive database of government schemes through the chatbot. By simply typing their queries or browsing by category, they gained access to detailed program descriptions, eligibility criteria, and relevant benefits. This eliminated the need for navigating complex government websites or struggling to understand official documents.

The chatbot performs well, handling user messages and producing responses with quickness. When faced with a range of user inputs, the trained model responds quickly and without noticeable lag. Performance, however, could differ based on outside variables like server load and user query complexity. To sum up, the web application for chatbots effectively offers an interactive interface that allows consumers to communicate with the chatbot. The application's strong capabilities, scalability, performance, and user-friendly design make it an invaluable tool for resolving user inquiries and improving overall user experience.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

In conclusion, this project successfully developed a user-friendly chatbot that empowers Indian citizens to navigate the landscape of government financial schemes. By leveraging PyTorch for data processing, the chatbot provides a comprehensive information retrieval system and guides users through the application process. This not only improves access to valuable programs but also promotes transparency and simplifies citizen interaction with government initiatives. The initial focus on a rule-based system proved effective, and the project lays the groundwork for future enhancements that could incorporate machine learning for personalized recommendations and even explore eligibility prediction. The potential impact on improving lives and financial well-being is significant, making this project a valuable contribution to bridging the gap between citizens and government support. Through user interaction, the chatbot guided individuals through an eligibility assessment process for specific schemes. Additionally, it presented step-by-step instructions for completing applications and linked users directly to official government websites for forms and further resources. This eliminated confusion and frustration often associated with applying for government programs. his project has the potential to significantly impact the lives of Indian citizens. By simplifying access to information and assisting with the application process, the chatbot empowers individuals to explore and potentially benefit from government programs designed to improve their financial well-being. With ongoing development and future iterations, this chatbot has the potential to become a powerful tool for bridging the gap between citizens and government initiatives.

6.2 FUTURE ENHANCEMENT

We can develop a recommendation system powered by machine learning to suggest suitable government schemes based on user profiles and program details. This could personalize the user experience and connect citizens with programs that best align with their needs. Explore the potential of using machine learning models to predict a user's eligibility for certain schemes. This would require careful consideration of data privacy, fairness, and potential biases in the data. Transparency and clear communication about the limitations of such prediction would be critical.

Use NLP approaches to help the chatbot comprehend user inquiries more deeply and provide responses that seem natural. This can entail dealing with slang, synonyms, and several ways to phrase the same query. To promote a more organic and interesting dialogue with users, use elements like follow-up questions and clarification prompts we can use conversational flow.

Expanding the chatbot's linguistic vocabulary to accommodate users who would rather communicate in languages spoken throughout India. Integrate the chatbot with voice assistant systems to allow people with visual impairments to engage hands-free. Establish a feedback system to gather opinions and recommendations from users for ongoing enhancements to the chatbot's features and answers By implementing these enhancements, the chatbot can evolve into a powerful and user-friendly tool, empowering Indian citizens to navigate the landscape of government financial schemes with greater ease and achieve better access to programs that can improve their lives. Working together with stakeholders from each step along the project would guarantee that the application stays up-to-date flexible, and open to changing to user demands and market trends.

APPENDIX

SOURCE CODE:

App.py

```
from tkinter import *
from chat import get_response, bot_name

BG_GRAY = "#ABB2B9"
BG_COLOR = "#17202A"
TEXT_COLOR = "#EAECEE"

FONT = "Helvetica 14"
FONT_BOLD = "Helvetica 13 bold"

class ChatApplication:

    def __init__(self):
        self.window = Tk()
        self._setup_main_window()

    def run(self):
        self.window.mainloop()

    def _setup_main_window(self):
        self.window.title("Chat")
        self.window.resizable(width=True, height=True)
        self.window.configure(width=470, height=550, bg=BG_COLOR)

        # head label
        head_label = Label(self.window, bg=BG_COLOR, fg=TEXT_COLOR,
                           text="Welcome", font=FONT_BOLD, pady=10)
        head_label.place(relwidth=1)

        # tiny divider
        line = Label(self.window, width=450, bg=BG_GRAY)
        line.place(relwidth=1, rely=0.07, relheight=0.012)

        # text widget
        self.text_widget = Text(self.window, width=20, height=2, bg=BG_COLOR,
```

```
        fg=TEXT_COLOR,
                          font=FONT, padx=5, pady=5)
        self.text_widget.place(relheight=0.745, relwidth=1, rely=0.08)
        self.text_widget.configure(cursor="arrow", state=DISABLED)

        # scroll bar
        scrollbar = Scrollbar(self.text_widget)
        scrollbar.place(relheight=1, relx=0.974)
        scrollbar.configure(command=self.text_widget.yview)

        # bottom label
        bottom_label = Label(self.window, bg=BG_GRAY, height=80)
        bottom_label.place(relwidth=1, rely=0.825)

        # message entry box
        self.msg_entry = Entry(bottom_label, bg="#2C3E50", fg=TEXT_COLOR,

        font=FONT)
        self.msg_entry.place(relwidth=0.74, relheight=0.06, rely=0.008, relx=0.011)
        self.msg_entry.focus()
        self.msg_entry.bind("<Return>", self._on_enter_pressed)

        # send button
        send_button = Button(bottom_label, text="Send", font=FONT_BOLD,
width=20, bg=BG_GRAY,
                             command=lambda: self._on_enter_pressed(None))
        send_button.place(relx=0.77, rely=0.008, relheight=0.06, relwidth=0.22)

    def _on_enter_pressed(self, event):
        msg = self.msg_entry.get()
        self._insert_message(msg, "You")

    def _insert_message(self, msg, sender):
        if not msg:
            return

        self.msg_entry.delete(0, END)
        msg1 = f"{sender}: {msg}\n\n"
        self.text_widget.configure(state=NORMAL)
        self.text_widget.insert(END, msg1)
```

```
        self.text_widget.configure(state=DISABLED)

        msg2 = f"{bot_name}: {get_response(msg)}\n\n"
        self.text_widget.configure(state=NORMAL)
        self.text_widget.insert(END, msg2)
        self.text_widget.configure(state=DISABLED)

        self.text_widget.see(END)


if __name__ == "__main__":
    app = ChatApplication()
    app.run()
```

Chat.py

```
import torch
import torch.nn as nn


class NeuralNet(nn.Module):
    def __init__(self, input_size, hidden_size, num_classes):
        super(NeuralNet, self).__init__()
        self.l1 = nn.Linear(input_size, hidden_size)
        self.l2 = nn.Linear(hidden_size, hidden_size)
        self.l3 = nn.Linear(hidden_size, num_classes)
        self.relu = nn.ReLU()

    def forward(self, x):
        out = self.l1(x)
        out = self.relu(out)
        out = self.l2(out)
        out = self.relu(out)
        out = self.l3(out)
        # no activation and no softmax at the end
        return out
import random
import json

import torch
```

```
from model import NeuralNet
from nltk_utils import bag_of_words, tokenize

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

with open('intents.json', 'r') as json_data:
    intents = json.load(json_data)

FILE = "data.pth"
data = torch.load(FILE)

input_size = data["input_size"]
hidden_size = data["hidden_size"]
output_size = data["output_size"]
all_words = data['all_words']
tags = data['tags']
model_state = data["model_state"]

model = NeuralNet(input_size, hidden_size, output_size).to(device)
model.load_state_dict(model_state)
model.eval()

bot_name = "BOT"

def get_response(msg):
    sentence = tokenize(msg)
    X = bag_of_words(sentence, all_words)
    X = X.reshape(1, X.shape[0])
    X = torch.from_numpy(X).to(device)

    output = model(X)
    _, predicted = torch.max(output, dim=1)

    tag = tags[predicted.item()]

    probs = torch.softmax(output, dim=1)
    prob = probs[0][predicted.item()]
    if prob.item() > 0.75:
        for intent in intents['intents']:
            if tag == intent["tag"]:
```

```
                return random.choice(intent['responses'])

    return "I do not understand..."


if __name__ == "__main__":
    print("Let's chat! (type 'quit' to exit)")
    while True:
        # sentence = "do you use credit cards?"
        sentence = input("You: ")
        if sentence == "quit":
            break

        resp = get_response(sentence)
        print(resp)
```

REFERENCES

[1] N. N. Khin and K. M. Soe, Question answering based university chatbot using sequence to sequence model. 2020, doi: 10.1109/o-cocosda50338.2020.9295021.

[2] J. Kim, H.-G. Lee, H. Kim, Y. Lee, and Y.-G. Kim, Two-Step Training and Mixed Encoding-Decoding for Implementing a Generative Chatbot with a Small Dialogue Corpus. 2018. doi: 10.18653/v1/w18-6707.

[3] M. E. Cortés-Cediel, A. Segura-Tinoco, I. Cantador, and M. P. R. Bolívar, "Trends and challenges of e-government chatbots: Advances in exploring open government data and citizen participation content," Government Information Quarterly, vol. 40, no. 4, p. 101877, Oct. 2023, doi: 10.1016/j.giq.2023.101877.

[4] D. S. Doherty and K. Curran, "Chatbots for online banking services," Web Intelligence, vol. 17, no. 4, pp. 327–342, Dec. 2019, doi: 10.3233/web-190422.

[5] Y. Li et al., "Multi-source Seq2seq guided by knowledge for Chinese healthcare consultation," Journal of Biomedical Informatics, vol. 117, p. 103727, May 2021, doi: 10.1016/j.jbi.2021.103727.

[6] F. Colace, M. De Santo, M. Lombardi, F. Pascale, A. Pietrosanto, and S. Lemma, "Chatbot for E-Learning: A case of study," International Journal of Mechanical Engineering and Robotics Research, pp. 528–533, Jan. 2018, doi: 10.18178/ijmerr.7.5.528-533.

.