

SOCIETY SUPPORT NAVIGATOR USING CHATBOT

Submitted in partial fulfilment of the required for the award of

Masters of Computer Science

by

SANTHOSH S

REGISTER NO (42734021)

DEEMED UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF SCIENCE AND HUMANITIES

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A++” by NAAC

JEPPIAR NAGAR, RAJIV GANDHI SALAI, CHENNAI 600 119

APRIL 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

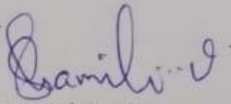
Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

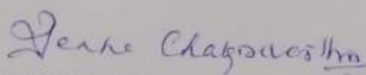
DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

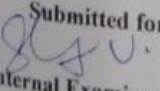
This is to certify that this Project Report is the bonafide work of **SANTHOSH S (42734021)** who carried out the project entitled **"SOCITY SUPPORT NAVIGATOR USING CHATBOT"** under our supervision from JANUARY 2024 TO APRIL 2024.

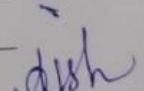

Internal Guide

Ms. SHAMILI V M.Sc., Assistant Professor


DEAN AND HEAD OF THE DEPARTMENT

Dr. REKHA CHAKRAVARTHI M.E., Ph.D.

Submitted for Viva voce examination held on _____

Internal Examiner

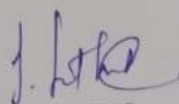

External Examiner

DECLARATION

I am, SANTHOSH S (427340S1) hereby declare that the project report entitled "SOCIETY SUPPORT NAVIGATOR USING CHATBOT" done by me under the guidance of MS. SHAMILI V M.Sc., Assistant Professor, Department of Computer Science at SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY, Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai-600 119 is submitted in partial fulfilment of the requirements for the award of Master of Science degree in Computer Science .

DATE: 20/04/2024

PLACE: Chennai


SANTHOSH S

Signature of the Candidate

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. REKHA CHAKRAVARTHI M.E., Ph.D. Dean of Arts and Science** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my project guide **MS. SHAMILI V M.Sc., Assistant Professor**, Department of Computer Science for their valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff member of the Department of Computer Science who were helpful in many ways for the completion of the project

SANTHOSH S

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

In contemporary society, individuals often find themselves grappling with a myriad of challenges spanning from mental health concerns to financial instability and beyond. Amidst these complexities, the process of navigating the labyrinthine network of support services can prove to be an overwhelming and bewildering task. Many individuals either struggle to locate the appropriate resources or simply forego seeking assistance altogether due to the perceived difficulties associated with accessing support.

Recognizing the pressing need for a streamlined solution to this pervasive issue, we propose the concept of a Society Support Navigator (SSN) powered by chatbot technology. The SSN represents a novel approach to facilitating access to support services by harnessing the capabilities of artificial intelligence (AI) and natural language processing (NLP) to create an intuitive and user-friendly platform.

At its core, the Society Support Navigator aims to serve as a virtual guide, offering personalized assistance to individuals seeking support. Through interactive conversations, users can articulate their needs and preferences in a conversational manner, enabling the chatbot to provide tailored recommendations and information regarding relevant support services. By integrating a comprehensive database of resources spanning various sectors and geographical locations, the SSN ensures that users have access to a wide array of support options tailored to their unique circumstances.

Moreover, the Society Support Navigator leverages machine learning algorithms to continuously refine its recommendations based on user feedback and interactions, thereby enhancing its efficacy and relevance over time. This adaptive approach not only ensures that users receive the most pertinent assistance but also fosters a dynamic and responsive system capable of addressing evolving societal need.

CHAPTER 2

LITRATURE SURVEY

TITLE: Interacting with educational chatbot: a systematic review

AUTHOR: Hatice YILDIZ DURAK

YEAR: 2022

Chatbots hold the promise of revolutionizing education by engaging learners, personalizing learning activities, supporting educators, and developing deep insight into learners' behavior. However, there is a lack of studies that analyze the recent evidence-based chatbot learner interaction design techniques applied in education.

TITLE: Integrating chatbots in education: insights from the chatbot – human interaction satisfaction model

AUTHOR: Jose Belda-Medina & Vendula Kokoskova

YEAR: 2023

Recent advances in Artificial Intelligence (AI) have paved the way for the integration of text-based and voice-enabled chatbots as adaptive virtual tutors in education. Despite the increasing use of AI-powered chatbots in language learning, there is a lack of studies exploring the attitudes and perceptions of teachers and students towards these intelligent tutors.

TITLE: Conversational agent-based guidance: examining the effect of chatbot usage frequency and satisfaction on visual design self-efficacy engagement.

AUTHOR: Julio Cesar Duarte

YEAR: 2022

Journal of Education and Information Technologies (EAIT) is a platform for the range of debates and issues in the intersection of Information Technology and Education fields. It probes the use of computing to improve education and learning in a variety of settings, platforms and environments.

TITLE: Chatbot optimization using Sentiment Analysis and Timeline Navigation

AUTHOR: Wagner da silva maciel sodre

YEAR:2023

chatbot or conversational agent is a software that can interact or "chat" with a human user using a natural language, like English, for instance. Since the first chatbot developed, many have been created but most of their problems still persist, like providing the right answer to the user and user acceptance itself. Considering such facts, in this work, we present a chatbot-building framework that considers the use of sentiment analysis and tree timelines to provide a better chatbot answer.

TITLE : Modeling University ICT Services and Solutions using an Artificial Intelligence Chatbot

AUTHOR: Nwankwo godson, Dr Nwobodo, Prof

Okafor

YEAR: 2024

Modeling university ICT services and solutions using AI Chatbot is building an AI-powered virtual agent to represent the human staff in providing similar services and solutions to ICT users online via Facebook Messenger and Web Demo platforms to enhance the accessibility of the ICT services. This study aims to model artificial intelligence (AI) for university information and communication technology (ICT) services and solutions delivery by providing responses to questions asked by ICT service users. The modeled Chatbot was implemented to solve the challenges that traditional ICT staff in universities faced, such as high volume of requests, long wait times, numerous repeated questions, duty hours constraints, and inability to provide personalized support to ICT users at all times due to human natural capacities. To address these challenges, universities need to explore the use of AI Chatbot as a means of providing more efficient, personalized, and accessible ICT support services

2.1 EXISTING SYSTEM

Currently, chatbots are being increasingly explored in education, showcasing potential in engaging learners, tailoring learning experiences, aiding educators, and gaining valuable insights into learner behavior. However, the literature lacks comprehensive studies analyzing evidence-based techniques of recent chatbot-learner interactions in education. A systematic review of 36 papers was conducted to evaluate the landscape. Findings revealed that most chatbots were designed on web platforms, focusing primarily on subjects like computer science, languages, and general education, with some presence in fields like engineering and mathematics. These chatbots largely functioned as teaching or peer agents, following predetermined conversational paths, and occasionally adopting personalized or collaborative learning strategies. Evaluation through experiments indicated improvements in learning outcomes and subjective satisfaction. Challenges identified encompassed limitations in dataset training and insufficient adherence to usability heuristics, highlighting areas for improvement in future research endeavors.

2.2 PROPOSED SYSTEM

This proposal outlines the development of Govt Assist, a society-focused chatbot designed to streamline the process of applying for government ID cards and accessing personalized public welfare schemes. Leveraging AI, Govt Assist will interact with users via a web-based platform, offering an intuitive chat interface to guide them through various governmental procedures with ease. Upon inputting basic demographics such as age and gender, users will be instantly matched with schemes tailored specifically to their profile. The system will utilize a comprehensive database of government schemes, ensuring up-to-date and relevant information is provided. The chatbot will guide users step-by-step in completing applications for ID cards and registering for eligible schemes, simplifying paperwork and reducing errors. Furthermore, GovT will incorporate adaptive learning algorithms to improve response accuracy over time based on user interactions. This feature ensures that the chatbot evolves continually, enhancing user experience and satisfaction. In addressing identified challenges from prior research, the development of GovAssist will focus on robust training datasets and strict adherence to usability heuristics, ensuring a user-friendly and effective service. This system not only aims to facilitate easier access to government services but also promotes greater inclusivity and efficiency in public administration.

CHAPTER 3

REQUIREMENTS AND SPECIFICATION

3.1 INTRODUCTION

The Society Support Navigator (SSN) is a pioneering initiative aimed at utilizing chatbot technology to revolutionize the accessibility of support services within society. With the proliferation of digital platforms, there exists an opportunity to create a streamlined solution that empowers individuals to navigate support systems with ease. The SSN seeks to capitalize on this opportunity by providing a user-friendly interface that guides users through interactive conversations, enabling them to articulate their needs and preferences effectively. By integrating artificial intelligence and natural language processing capabilities, the SSN can understand user queries and provide personalized recommendations for relevant support services. Furthermore, the SSN aims to foster inclusivity by ensuring that individuals from diverse backgrounds and circumstances have equal access to support resources. Through a comprehensive database of support services spanning various sectors and geographic locations, the SSN strives to address the multifaceted needs of its users. By adhering to the specified functional and non-functional requirements, the development team aims to create a robust and reliable platform that empowers individuals to take control of their well-being and access the assistance they need in a timely manner.

Additionally, the Society Support Navigator (SSN) endeavors to bridge the gap between individuals and support services by providing a platform that prioritizes user empowerment and efficiency. Beyond merely offering a directory of resources, the SSN integrates a recommendation engine fueled by machine learning algorithms. This feature enables the chatbot to adapt and improve its suggestions over time based on user feedback and interactions, ensuring that recommendations remain relevant and beneficial.

Moreover, the SSN recognizes the importance of security and privacy in handling sensitive user information. To address these concerns, stringent security measures will be implemented to safeguard user data and uphold confidentiality. By adhering to industry best

practices and regulatory standards, the SSN aims to instill trust and confidence in its users, fostering a safe environment for seeking support.

3.2 SOFTWARE REQUIREMENTS

- ❖ Processor: intel core i5
- ❖ Ram 16
- ❖ Internal Space: 1 TB

3.3 TECHNOLOGIES USED

- ❖ Front-end : Html, css , java script
- ❖ Back-end : Python, flask
- ❖ Database :My sql

3.4 NATURAL LANGUAGE PROGRAM

Natural Language Processing (NLP) lies at the heart of the Society Support Navigator (SSN), enabling it to understand and respond to user queries in a conversational manner. NLP technology equips the SSN with the ability to comprehend the nuances of human language, allowing users to interact with the chatbot in a way that feels natural and intuitive.

Through advanced algorithms and linguistic models, NLP enables the SSN to parse user input, extract relevant information, and generate appropriate responses. This capability enables the chatbot to interpret user intentions, regardless of variations in language, syntax, or phrasing, facilitating seamless communication between the user and the system.

Furthermore, NLP plays a crucial role in enhancing the personalization and effectiveness of the SSN. By analyzing user interactions and feedback, the chatbot can adapt its responses and recommendations to better meet the individual needs and preferences of each user. This

personalized approach not only improves user satisfaction but also increases the likelihood of users finding the support services they require



FIG 3.1

In summary, NLP serves as the foundation upon which the Society Support Navigator is built, enabling it to understand, process, and respond to user input effectively. By harnessing the power of NLP, the SSN transforms the user experience, making it more intuitive, efficient, and tailored to the individual needs of each user.

HISTORY

The history of Natural Language Processing (NLP) can be traced back to the mid-20th century, with its roots in the fields of linguistics, computer science, and artificial intelligence. Here's a brief overview of key milestones in the evolution of NLP:

Early Foundations (1950s-1960s)

The groundwork for NLP was laid in the 1950s with the development of early computational linguistics and machine translation systems.

In 1950, Alan Turing published the paper "Computing Machinery and Intelligence," which proposed the Turing Test as a measure of a machine's ability to exhibit intelligent behavior indistinguishable from that of a human.

The Georgetown-IBM Experiment in 1954 demonstrated automatic translation of Russian sentences into English using IBM's experimental machine translation system, marking an early milestone in machine translation research.

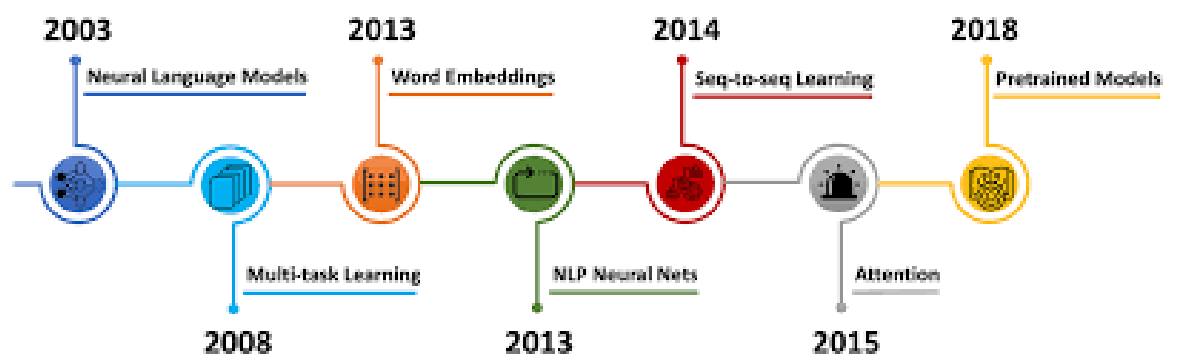


FIG 3.2

Rule-Based Systems (1960s-1970s):

During the 1960s and 1970s, researchers focused on rule-based approaches to NLP, where linguistic rules were manually crafted to analyze and generate natural language text.

Projects such as SHRDLU (developed by Terry Winograd in the late 1960s) demonstrated the ability to understand and respond to natural language queries within restricted domains, such as blocks world. The development of formal grammars, such as Chomsky's transformational generative grammar, influenced early NLP research and provided theoretical frameworks for analyzing natural language syntax.

Statistical Methods (1980s-1990s):

The 1980s saw a shift towards statistical approaches to NLP, driven by advances in computational power and data availability.

Researchers began to explore techniques such as Hidden Markov Models (HMMs) and statistical language models to address tasks like speech recognition, part-of-speech tagging, and machine translation.

In the 1990s, the introduction of the Penn Treebank dataset and the development of statistical parsing algorithms led to significant progress in syntactic parsing and language understanding.

Neural Networks and Deep Learning (2000s-Present):

The early 2000s witnessed a resurgence of interest in neural network-based approaches to NLP, fueled by the development of more powerful computational resources and large-scale datasets.

Breakthroughs in deep learning, particularly the introduction of recurrent neural networks (RNNs) and later, transformers, revolutionized various NLP tasks such as language modeling, named entity recognition, sentiment analysis, and machine translation.

Pretrained language models like BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pretrained Transformer) achieved state-of-the-art performance across a wide range of NLP benchmarks, sparking rapid progress in the field.

Contemporary Developments and Challenges:

Recent years have seen continued advancements in NLP, with a focus on areas such as multimodal language understanding, commonsense reasoning, and low-resource languages.

Challenges remain in achieving deeper semantic understanding, handling linguistic nuances, and addressing biases present in language data and models.

Ethical considerations, including privacy, fairness, and transparency, have become increasingly important as NLP technologies are deployed in real-world applications.

Overall, the history of NLP reflects a journey of interdisciplinary collaboration, technological innovation, and ongoing challenges in understanding and processing human language. As NLP continues to evolve, it holds the promise of enabling more natural and intuitive interactions between humans and machines, with applications spanning from virtual assistants and chatbots to language translation and sentiment analysis.

3.5 NLP PLATFORM

Numerous Natural Language Processing (NLP) platforms have emerged to facilitate the understanding, processing, and generation of human language. These platforms offer a variety of tools and services tailored to different NLP tasks, catering to the needs of developers, researchers, and businesses alike. Google Cloud Natural Language API stands out with its comprehensive suite of tools for sentiment analysis, entity recognition, and syntax analysis, all accessible through a user-friendly interface. Meanwhile, Amazon Comprehend, as a fully managed NLP service, provides a seamless experience for key phrase extraction, sentiment analysis, and entity recognition, coupled with support for multiple languages. On the Microsoft Azure Text Analytics front, its robust capabilities for sentiment analysis, entity recognition, and language detection make it a compelling choice for businesses seeking reliable NLP solutions. IBM Watson Natural Language Understanding offers extensive capabilities in entity extraction, sentiment analysis, and concept tagging, with customizable models to suit specific domains. In the open-source realm, spaCy and NLTK provide flexible options for Python developers, enabling tasks such as tokenization, named entity recognition, and dependency parsing. Hugging Face Transformers rounds out the selection with its library and platform for transformer-based models, facilitating advanced tasks like text generation, sentiment analysis, and language translation. Overall, the diverse landscape of NLP platforms caters to a wide range of needs, empowering users to harness the power of language for various applications and industries.

3.6 NLP LIBRARIES

There are several powerful NLP libraries available, each offering a range of functionalities for text processing, analysis, and modeling. Here are some of the most widely used NLP libraries:

1. NLTK (Natural Language Toolkit):

NLTK is one of the oldest and most comprehensive libraries for NLP in Python.

It provides tools for tasks such as tokenization, stemming, tagging, parsing, and classification. NLTK also includes support for various corpora and lexical resources. spaCy is a modern and efficient NLP library written in Python.

It offers tools for tokenization, part-of-speech tagging, named entity recognition, dependency parsing, and more.

spaCy is known for its speed and ease of use, making it popular among researchers and developers.

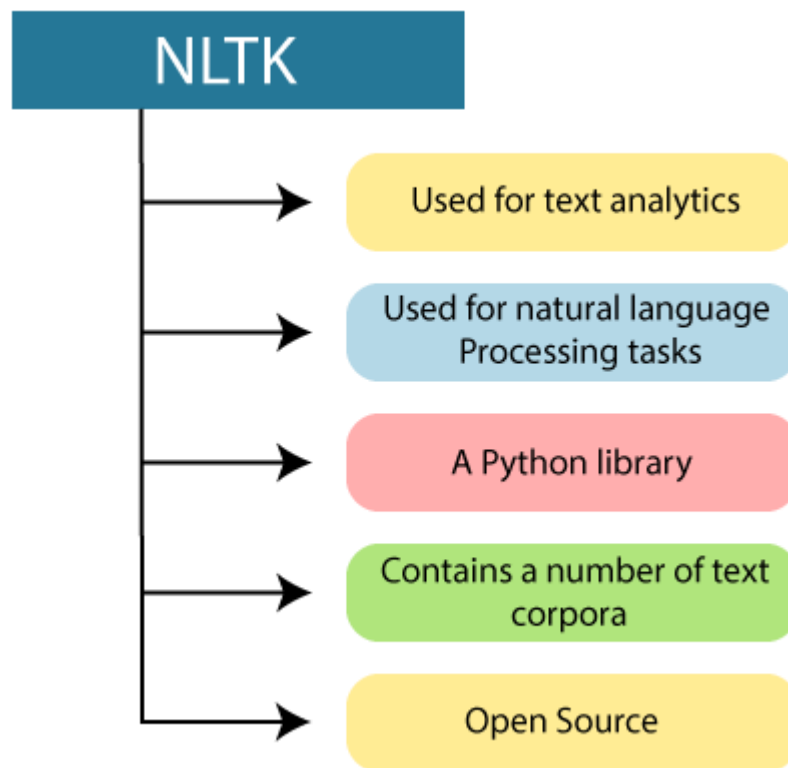


FIG 3.3

2. Gensim:

Gensim is a library for topic modeling and document similarity analysis.

It provides implementations of algorithms such as Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA), and Word2Vec.

Gensim is widely used for tasks like document clustering, semantic similarity calculation, and topic extraction.

3. TextBlob:

TextBlob is a simple and intuitive NLP library built on top of NLTK and Pattern.

It offers tools for tasks such as sentiment analysis, part-of-speech tagging, noun phrase

extraction, and language translation.

TextBlob is known for its ease of use and beginner-friendly API.

4. Stanford CoreNLP:

Stanford CoreNLP is a Java library developed by the Stanford NLP Group.

It provides tools for tasks such as tokenization, part-of-speech tagging, named entity recognition, sentiment analysis, and dependency parsing.

CoreNLP is known for its accuracy and robustness, particularly in tasks like named entity recognition and dependency parsing.

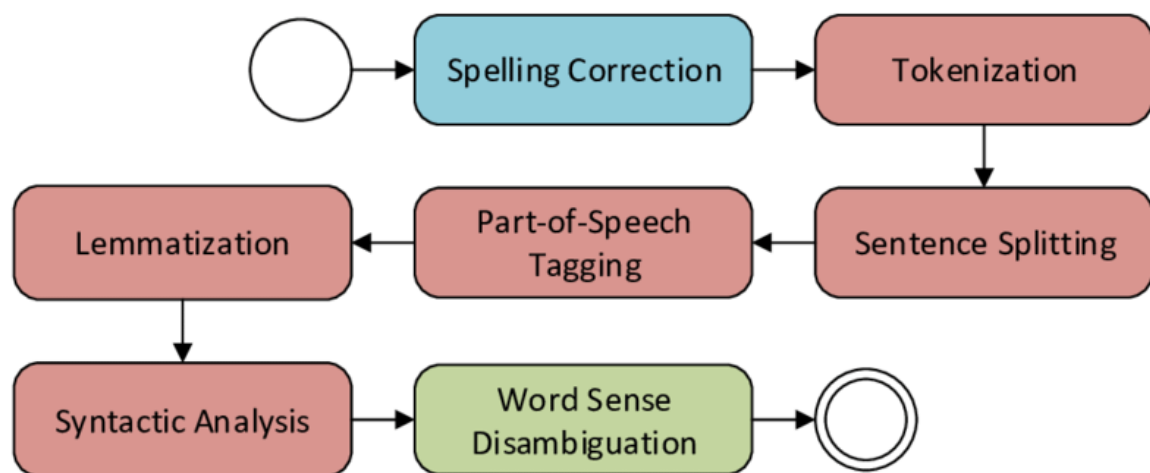


FIG 3.4

5. AllenNLP:

AllenNLP is a deep learning library for NLP tasks developed by the Allen Institute for AI.

It offers pre-built models and tools for tasks such as text classification, semantic role labeling, coreference resolution, and more. AllenNLP is built on top of PyTorch and provides a modular and extensible framework for building custom NLP models.

6. Transformers (from Hugging Face):

Transformers is a library for state-of-the-art NLP models developed by Hugging Face.

It provides pre-trained models such as BERT, GPT, and RoBERTa for tasks like text classification, language modeling, question answering, and more.

Transformers offers easy integration with PyTorch and TensorFlow and allows for fine-tuning and customization of pre-trained models.

3.7 OPERATION OF NLP

Natural Language Processing (NLP) is a crucial branch of artificial intelligence that focuses on enabling computers to understand, interpret, and generate human language in a meaningful way. The process begins with the collection of text data, which is then subjected to various preprocessing steps such as tokenization, normalization, and removal of stopwords. After preprocessing, the text is transformed into a numerical format through methods like Bag of Words, TF-IDF, or word embeddings, making it suitable for machine learning models. These models are then trained using techniques such as supervised, unsupervised, or reinforcement learning depending on the task at hand. Once trained, NLP models can perform a range of tasks including sentiment analysis, language translation, and automatic summarization, allowing for sophisticated interactions between computers and human language, thereby enhancing applications like chatbots, recommendation systems, and language translation services.

CHAPTER 4

SYSTEM DESIGN

4.1 ARCHITECTURE DIAGRAM

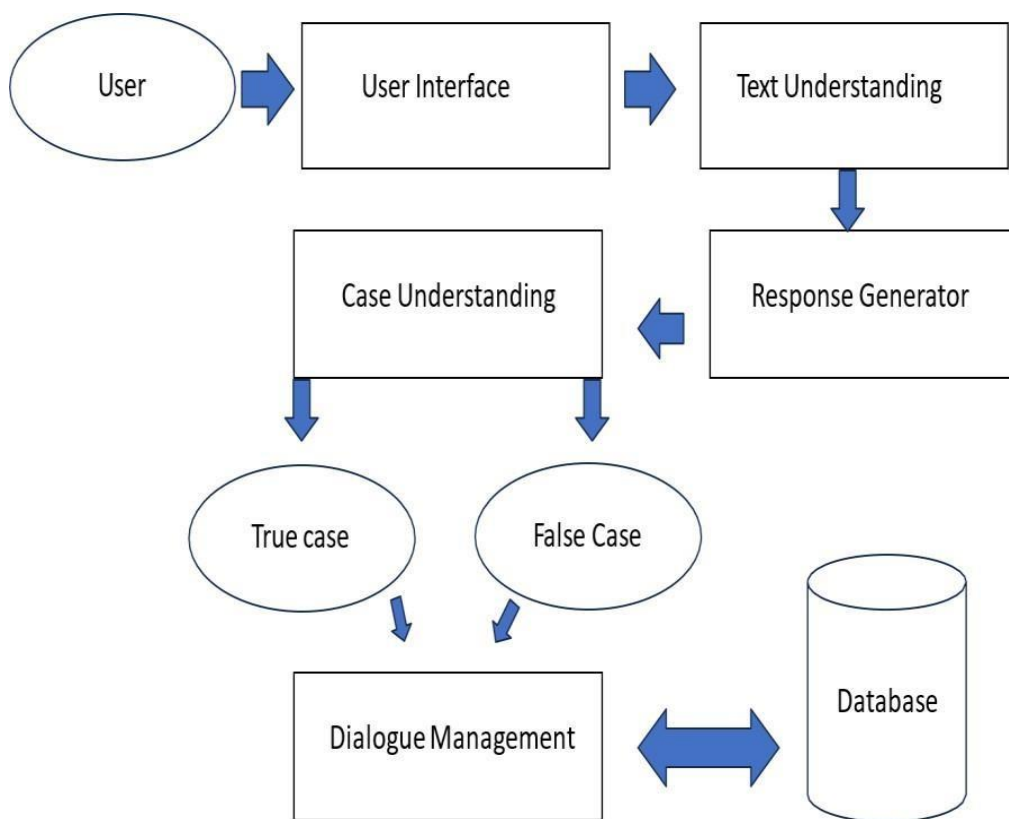


FIG 4.1

4.2 WORKING AGENT



FIG 4.2

CHAPTER 5

SCOPE

5.1 PROJECT SCOPE

The Society Support Navigator Chatbot project is designed to create a versatile platform for individuals grappling with societal issues, offering them tailored guidance and resources through intuitive conversational AI technology. By focusing on user needs and leveraging sophisticated natural language processing capabilities, the chatbot will serve as a reliable companion for users seeking assistance with a wide range of challenges. Through comprehensive information provision, a robust resource directory, and a seamless referral system, the chatbot aims to empower users to make informed decisions and access relevant support services. Moreover, prioritizing user privacy and confidentiality, the chatbot will ensure a secure environment for users to seek assistance anonymously if desired. Through iterative development and ongoing evaluation, the project aims to continually enhance the chatbot's functionality and effectiveness in meeting the evolving needs of its users and the broader community.

5.2 PROJECT PURPOSE

The Society Support Navigator Chatbot project aims to serve as a pivotal tool in addressing societal challenges by providing accessible and personalized support to individuals in need. Its primary purpose lies in empowering users through timely guidance, information, and resources tailored to their unique circumstances. By leveraging chatbot technology, the project ensures that support is readily available around the clock, transcending traditional barriers of accessibility. Moreover, the chatbot's conversational interface fosters a supportive environment, where users feel understood and valued. Through efficient resource utilization and a commitment to user privacy, the project strives to maximize the impact of existing support infrastructure while safeguarding user confidentiality. Continuous improvement is ingrained in the project's ethos, with ongoing evaluation and iteration aimed at enhancing the chatbot's responsiveness and relevance. Ultimately, the Society Support Navigator Chatbot project seeks to promote community well-being and resilience by empowering individuals to navigate societal challenges with confidence and support.

5.3 PERSPECTIVE

From an individual standpoint, the Society Support Navigator utilizing a chatbot offers a personalized lifeline for those navigating societal challenges. Users find empowerment in accessing tailored guidance and resources, fostering a sense of agency in addressing their specific needs. The platform's accessibility ensures that assistance is available around the clock, transcending geographical barriers and offering support to individuals regardless of their circumstances. The provision for anonymous engagement safeguards user privacy, creating a safe space for seeking help without fear of judgment. From a community perspective, the chatbot optimizes resource utilization by efficiently connecting individuals with relevant services, thereby maximizing community impact. It promotes inclusivity by reaching marginalized populations and contributes to overall community well-being by addressing prevalent societal issues. At a societal level, the chatbot drives positive social change by reducing disparities in access to support services and offering valuable insights into emerging trends and needs. In essence, the Society Support Navigator using a chatbot offers a transformative perspective on societal support, empowering individuals, strengthening communities, and fostering a more resilient society.

5.4 SYSTEM FEATURES

The Society Support Navigator utilizing a chatbot incorporates a range of system features tailored to address societal challenges effectively. At its core, the chatbot offers a user-friendly interface, facilitating seamless interactions for individuals seeking support. Natural language processing capabilities enable the chatbot to understand user queries and provide relevant responses, ensuring a conversational and intuitive user experience. A comprehensive database of resources and services lies at the heart of the system, categorized by type and location, allowing the chatbot to offer tailored guidance and referrals based on user needs. Anonymity options provide users with the choice to seek assistance discreetly, fostering a safe and non-judgmental environment. Multilingual support ensures accessibility to a diverse range of users, transcending language barriers. Integration with external systems, such as databases and APIs, enables real-time access to up-to-date information and resources. A feedback mechanism allows users to provide input, ensuring continuous improvement of the system's functionality and responsiveness. Overall, these system features empower the Society Support Navigator chatbot to effectively address societal challenges, providing personalized support and guidance to individuals in need.

5.5 DESIGN AND IMPLEMENTATION

Designing and implementing a society support navigator chatbot involves several steps. Here's a high-level outline of the process:

- Define Objectives and Scope
- Identify Key Features
- Choose a Platform
- Design User Experience
- Develop Backend Functionality
- Implement Frontend Interface
- Integrate External APIs and Databases
- Test and Iterate
- Deploy and Monitor
- Provide Support and Updates

5.5 OTHER NOTIFICATIONAL REQUIREMENTS

In addition to the technical aspects, implementing a society support navigator chatbot requires addressing various non-technical considerations to ensure its effectiveness, safety, and ethical use. Here are some nonfictional requirements:

1. Ethical Guidelines:

Adhere to ethical principles in the development and deployment of the chatbot, ensuring it respects user privacy, autonomy, and confidentiality.

Avoid bias in the chatbot's responses and recommendations, especially concerning sensitive issues such as mental health, gender, race, and socio-economic status.

2. Accessibility:

Ensure the chatbot is accessible to users with disabilities by following accessibility standards such as WCAG (Web Content Accessibility Guidelines).

Provide alternative means of interaction for users who may have difficulty using text-based interfaces, such as voice input or screen readers.

3. Data Privacy and Security:

Implement robust data protection measures to safeguard user data, including encryption, access controls, and secure storage practices.

Obtain explicit consent from users before collecting and processing personal information, and provide transparent information about how their data will be used.

4. Cultural Sensitivity:

Consider cultural diversity and sensitivity when designing the chatbot's responses and recommendations to ensure they are inclusive and respectful of different cultural norms and values.

Provide support in multiple languages and dialects to accommodate users from diverse linguistic backgrounds.

5. Legal Compliance:

Ensure compliance with relevant laws and regulations governing the use of chatbots and the provision of support services, such as healthcare regulations (e.g., HIPAA in the United States) and data protection laws (e.g., GDPR in Europe).

Obtain necessary permissions or licenses for accessing and sharing information about support services and crisis resources.

6. User Support and Feedback Mechanisms:

Provide clear channels for users to seek help or report issues encountered while interacting with the chatbot, such as customer support email, chat support, or feedback forms.

Actively solicit feedback from users to improve the chatbot's performance and address any shortcomings or areas for enhancement.

7. Collaboration with Stakeholders:

Collaborate with relevant stakeholders, including government agencies, non-profit organizations, mental health professionals, and community leaders, to ensure the chatbot's recommendations align with established best practices and resources.

Establish partnerships with support service providers to facilitate seamless referrals and access to additional support options.

8. Continuous Improvement and Evaluation:

Implement mechanisms for monitoring and evaluating the chatbot's performance and impact, including metrics such as user satisfaction, engagement, and outcomes.

Use feedback and analytics data to iteratively improve the chatbot's functionality, content, and user experience over time.

9. Crisis Response and Intervention:

Provide clear guidelines and protocols for handling users in crisis, including escalation procedures to connect them with emergency services or trained professionals.

Ensure the chatbot is equipped to recognize signs of distress and provide appropriate support, including empathy and encouragement, while encouraging users to seek help from qualified professionals when necessary.

10. Transparency and Accountability:

Be transparent about the capabilities and limitations of the chatbot, including its role as a support navigator and the boundaries of its expertise.

Establish accountability mechanisms to address any issues or concerns arising from the chatbot's use, including mechanisms for redress and oversight.

Addressing these nonfictional requirements alongside the technical aspects is essential for developing a society support navigator chatbot that effectively meets the needs of its users while upholding ethical and legal standards.

5.6 INFORMATION DISSEMINATION

Chatbots designed to provide services to society play a crucial role in facilitating efficient communication and access to information across various sectors such as healthcare, education, and government. These chatbots are programmed to deliver instant, reliable information about services, handle inquiries, and guide users through complex processes like application submissions or appointment bookings. They can also serve in emergency response by providing real-time updates and essential guidance, thus enhancing public safety. Furthermore, chatbots are employed to gather feedback and conduct surveys, helping organizations to improve service delivery based on user input. Overall, the integration of chatbots in public services not only improves access and responsiveness but also significantly enhances user engagement and satisfaction by providing a 24/7 support system.

CHAPTER 6

CODING

6.1 APP.PY

```
from flask import Flask, render_template, request, jsonify

from chat import get_response

app = Flask(__name__)

@app.get("/")
def index_get():
    return render_template("base.html")

@app.post("/predict")
def predict():
    text = request.get_json().get("message")
    # TODO : Error handling
    response = get_response(text)
    message = {"answer": response}
    return jsonify(message)

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

6.2 CHAT.PY

```
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 = "Sam"

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)

```

6.3 MODEL.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
```

6.4 NLTK_UTILS.PY

```
import numpy as np
import nltk
# nltk.download('punkt')
from nltk.stem.porter import PorterStemmer
stemmer = PorterStemmer()

def tokenize(sentence):
    """
    split sentence into array of words/tokens
    a token can be a word or punctuation character, or number
    """
    return nltk.word_tokenize(sentence)

def stem(word):
    """
    stemming = find the root form of the word
    examples:
    words = ["organize", "organizes", "organizing"]
    words = [stem(w) for w in words]
    -> ["organ", "organ", "organ"]
    """
```

```
return stemmer.stem(word.lower())
```

```
def bag_of_words(tokenized_sentence, words):
    """
    return bag of words array:
    1 for each known word that exists in the sentence, 0 otherwise
    example:
    sentence = ["hello", "how", "are", "you"]
    words = ["hi", "hello", "I", "you", "bye", "thank", "cool"]
    bog = [ 0, 1, 0, 1, 0, 0, 0]
    """

    # stem each word
    sentence_words = [stem(word) for word in tokenized_sentence]
    # initialize bag with 0 for each word
    bag = np.zeros(len(words), dtype=np.float32)
    for idx, w in enumerate(words):
        if w in sentence_words:
            bag[idx] = 1

    return bag
```

6.5 TRAIN.PY

```
import numpy as np
import random
import json

import torch
import torch.nn as nn
from torch.utils.data import Dataset, DataLoader

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

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

all_words = []
tags = []
xy = []
# loop through each sentence in our intents patterns
for intent in intents['intents']:
    tag = intent['tag']
    # add to tag list
    tags.append(tag)
    for pattern in intent['patterns']:
        # tokenize each word in the sentence
        w = tokenize(pattern)
```

```

    # add to our words list
    all_words.extend(w)
    # add to xy pair
    xy.append((w, tag))

# stem and lower each word
ignore_words = ['?', '!', '!']
all_words = [stem(w) for w in all_words if w not in ignore_words]
# remove duplicates and sort
all_words = sorted(set(all_words))
tags = sorted(set(tags))

print(len(xy), "patterns")
print(len(tags), "tags:", tags)
print(len(all_words), "unique stemmed words:", all_words)

# create training data
X_train = []
y_train = []
for (pattern_sentence, tag) in xy:
    # X: bag of words for each pattern_sentence
    bag = bag_of_words(pattern_sentence, all_words)
    X_train.append(bag)
    # y: PyTorch CrossEntropyLoss needs only class labels, not one-hot
    label = tags.index(tag)
    y_train.append(label)

X_train = np.array(X_train)
y_train = np.array(y_train)

# Hyper-parameters
num_epochs = 1000
batch_size = 8
learning_rate = 0.001
input_size = len(X_train[0])
hidden_size = 8
output_size = len(tags)
print(input_size, output_size)

class ChatDataset(Dataset):

    def __init__(self):
        self.n_samples = len(X_train)
        self.x_data = X_train
        self.y_data = y_train

    # support indexing such that dataset[i] can be used to get i-th sample
    def __getitem__(self, index):
        return self.x_data[index], self.y_data[index]

```

```

# we can call len(dataset) to return the size
def __len__(self):
    return self.n_samples

dataset = ChatDataset()
train_loader = DataLoader(dataset=dataset,
                           batch_size=batch_size,
                           shuffle=True,
                           num_workers=0)

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

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

# Loss and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)

# Train the model
for epoch in range(num_epochs):
    for (words, labels) in train_loader:
        words = words.to(device)
        labels = labels.to(dtype=torch.long).to(device)

        # Forward pass
        outputs = model(words)
        # if y would be one-hot, we must apply
        # labels = torch.max(labels, 1)[1]
        loss = criterion(outputs, labels)

        # Backward and optimize
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

    if (epoch+1) % 100 == 0:
        print(f'Epoch [{epoch+1}/{num_epochs}], Loss: {loss.item():.4f}')

print(f'final loss: {loss.item():.4f}')

data = {
    "model_state": model.state_dict(),
    "input_size": input_size,
    "hidden_size": hidden_size,
    "output_size": output_size,
    "all_words": all_words,
    "tags": tags
}

```

```
FILE = "data.pth"  
torch.save(data, FILE)  
  
print(f'training complete. file saved to {FILE}')
```

CHAPTER 7

OUTPUT

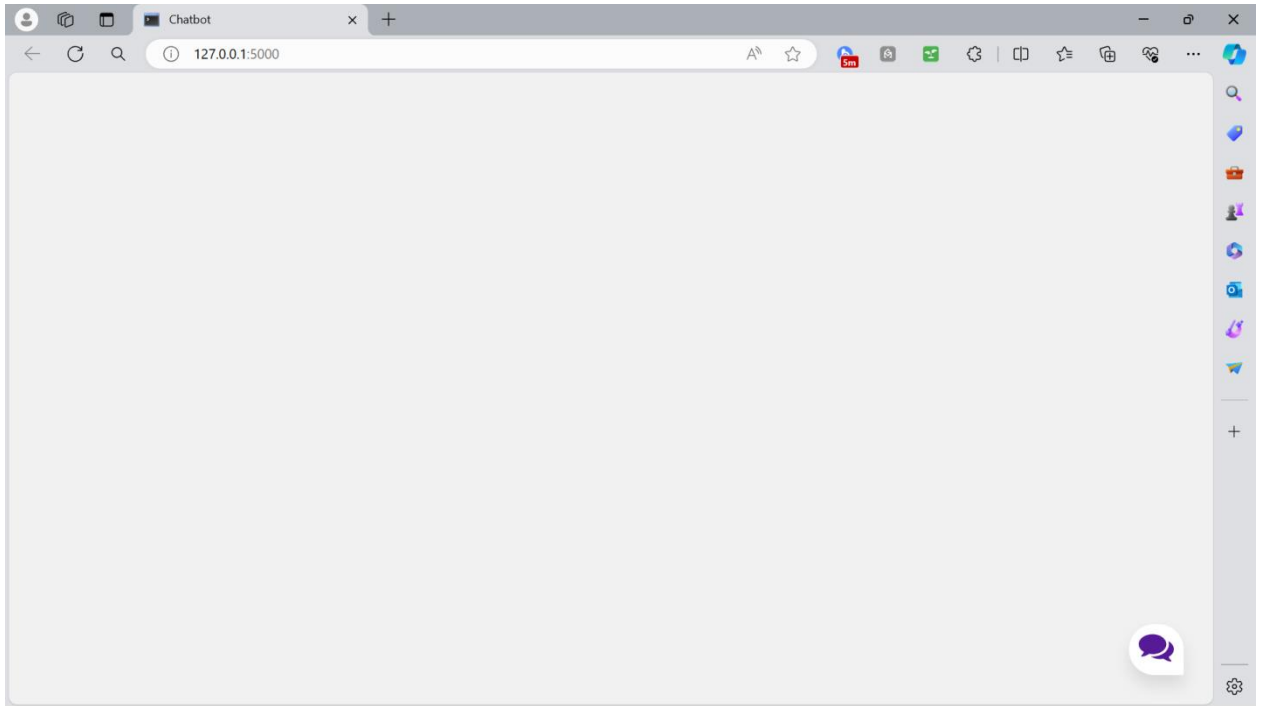


FIG 7.1

INTERFACE

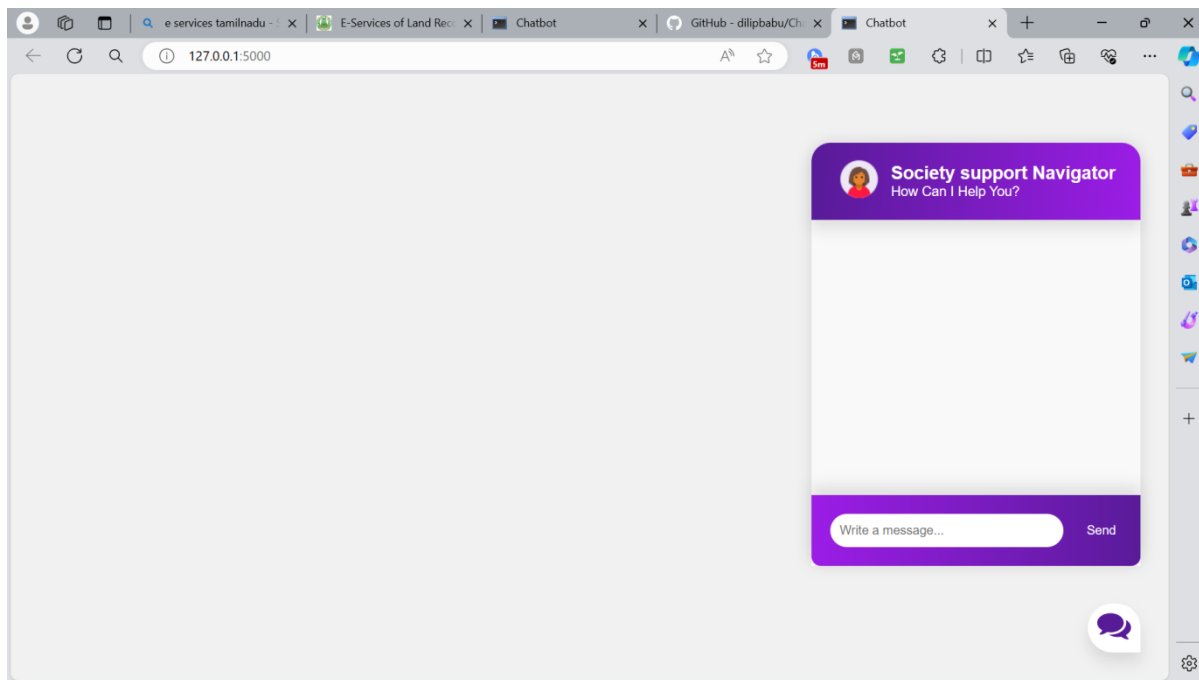


FIG 7.2

CHATBOT INTERFACE

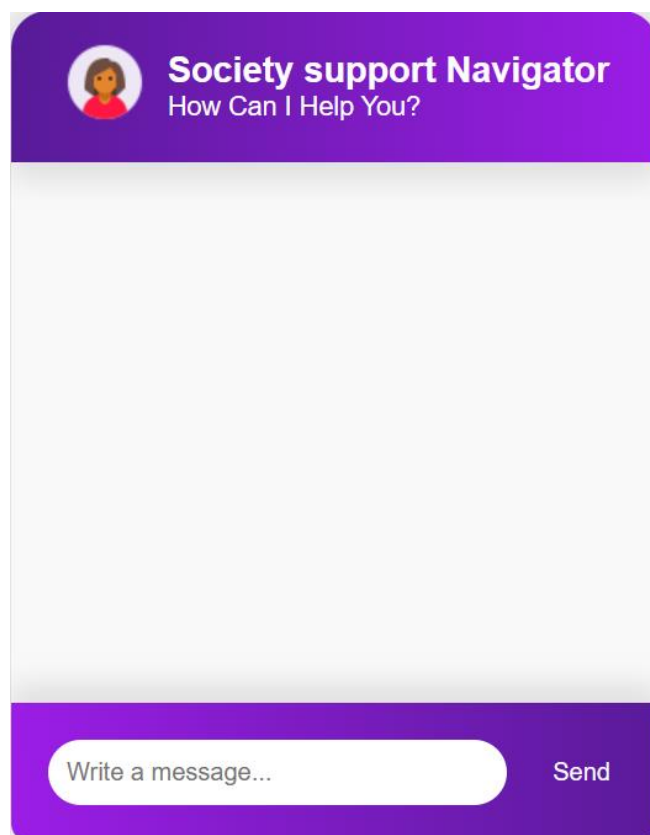


FIG 7.3 CHAT INDEX

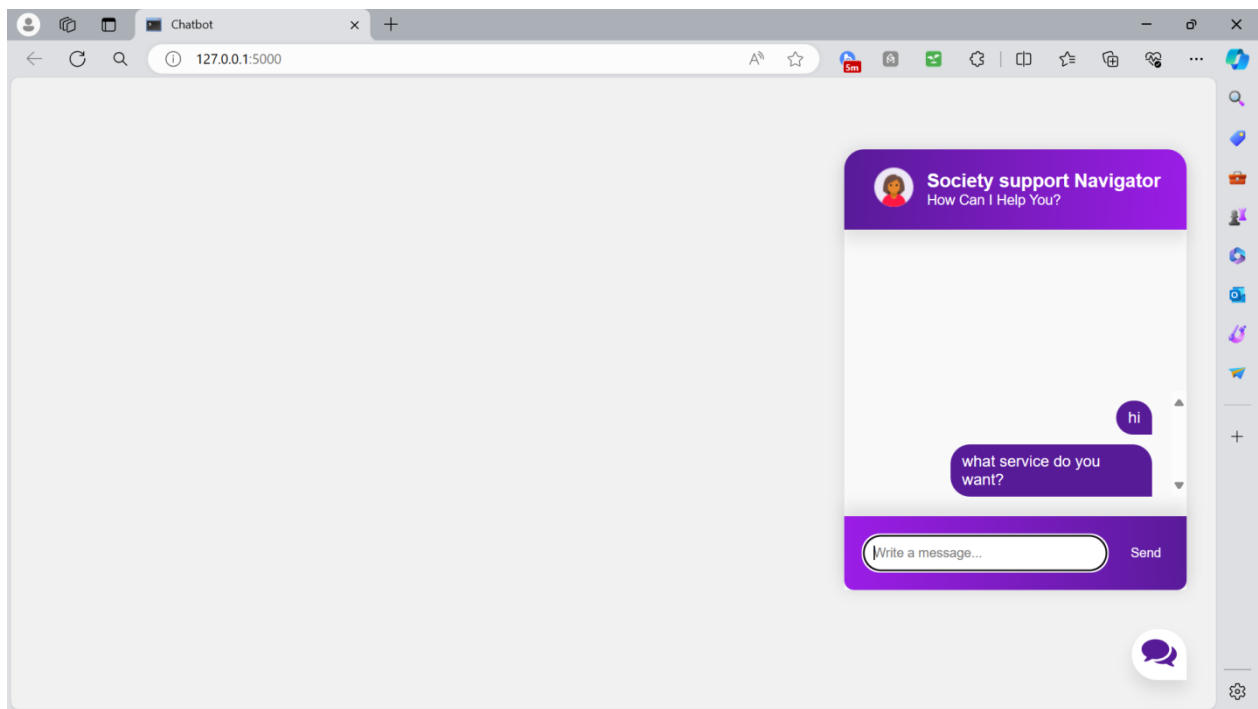


FIG 7.4
WORKING MODEL

CHAPTER 8

CONCLUSION

CONCLUSION AND FEATURE

The development of a society support navigator chatbot represents a significant step forward in leveraging technology to address societal challenges. By seamlessly integrating advanced AI capabilities with ethical considerations and user-centric design principles, this chatbot emerges as a beacon of support for individuals navigating through various societal issues. Its pivotal features include the ability to retrieve accurate and timely information about support services, offering personalized recommendations based on individual needs, and providing empathetic responses to those in distress. Moreover, it is equipped with protocols for crisis intervention, ensuring swift and appropriate responses to users in critical situations. Accessibility remains a cornerstone, with provisions made for users with disabilities, while stringent data privacy measures uphold user confidentiality and trust. Cultural sensitivity underscores every interaction, fostering inclusivity and respect for diverse backgrounds.

Compliance with legal standards is paramount, ensuring that the chatbot operates within the bounds of applicable regulations. User support mechanisms and collaboration with stakeholders further enhance its effectiveness and accountability. In conclusion, the society support navigator chatbot emerges as a potent force for positive change, empowering individuals to access resources, find solace, and navigate the complexities of modern life with resilience and support.

REFERENCE

1. Noble, J.M., Zamani, A., Gharaat, M., Merrick, D., Maeda, N., Foster, A.L., Nikolaidis, I., Goud, R., Stroulia, E., Agyapong, V.I. and Greenshaw, A.J., 2022. Developing, implementing, and evaluating an artificial intelligence–guided mental health resource navigation chatbot for health care workers and their families during and following the COVID- 19 pandemic: protocol for a cross-sectional study. *JMIR Research Protocols*, 11(7), p.e33717.
2. Kocielnik, R., Agapie, E., Argyle, A., Hsieh, D.T., Yadav, K., Taira, B. and Hsieh, G., 2019. HarborBot: a chatbot for social needs screening. In *AMIA Annual Symposium Proceedings* (Vol. 2019, p. 552). American Medical Informatics Association.
3. Hasan, I., Rizvi, S., Jain, S. and Huria, S., 2021, March. The AI enabled chatbot framework for intelligent citizen-government interaction for delivery of services. In *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 601-606). IEEE.
4. Ehrenpreis, M. and DeLooper, J., 2022. Implementing a chatbot on a library website. *Journal of Web Librarianship*, 16(2), pp.120-142.
5. Väänänen, K., Hiltunen, A., Varsaluoma, J. and Pietilä, I., 2020. CivicBots—Chatbots for supporting youth in societal participation. In *Chatbot Research and Design: Third International Workshop, CONVERSATIONS 2019, Amsterdam, The Netherlands, November 19–20, 2019, Revised Selected Papers 3* (pp. 143-157). Springer International Publishing.
6. Fadhil, A. and Gabrielli, S., 2017, May. Addressing challenges in promoting healthy lifestyles: the al-chatbot approach. In *Proceedings of the 11th EAI international conference on pervasive computing technologies for healthcare* (pp. 261-265).
7. Panda, S. and Chakravarty, R., 2022. Adapting intelligent information services in libraries: A case of smart AI chatbots. *Library Hi Tech News*, 39(1), pp.12-15.
8. Pereira, J. and Díaz, Ó., 2019. Using health chatbots for behavior change: a mapping study. *Journal of medical systems*, 43, pp.1-13.

9. Schario, M.E., Bahner, C.A., Widenhofer, T.V., Rajaballey, J.I. and Thatcher, E.J., 2022. Chatbot-assisted care management. *Professional Case Management*, 27(1), pp.19-25.

10. Nurshatayeva, A., Page, L.C., White, C.C. and Gehlbach, H., 2020. Proactive student support using artificially intelligent conversational chatbots: The importance of targeting the technology. *EdWorking paper*, Annenberg University <https://www.edworkingpapers.com/sites/default/files/ai20-208.pdf>.

