CHATBOT FOR LEGAL ADVISORY SYSTEM

A DESIGN PROJECT REPORT

Submitted by

AUDHAVAN.A

BARATHVAJ.M

MOHAMED THABISH.M

VIGNESHWARAN.M

in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, NewDelhi)

SAMAYAPURAM – 621112

JUNE 2024

K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY
(AUTONOMOUS)
SAMAYAPURAM–621112

BONAFIDE CERTIFICATE

Certified that this project report titled **"CHATBOT FOR LEGAL ADVISORY SYSTEM"** is the bonafide work of **AUDHAVAN.A (REG NO: 811721243010),BARATHVAJ.M (REG NO:81172124311),MOHAMED THABISH.M (REG NO: 811721243031), VIGNESHWARAN.M (REG NO: 811721243061)** who carried out the project under my supervision.

SIGNATURE

Dr.T.Avudaiappan, M.E., Ph.D.,

HEAD OF THE DEPARTMENT

Associate Professor

Department of Artificial Intelligence

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram – 621 112

SIGNATURE

Mrs.G.Nalina Keerthsana, M.E.,

SUPERVISOR

Assistant Professor

Department of Artificial Intelligence

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram – 621 112

Submitted for the viva-voce examination held on ……

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We jointly declare that the project report on **"CHATBOT FOR LEGAL ADVISORY SYSTEM"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfilment of the requirement of the award of Degree of **BACHELOR OF TECHNOLOGY**.

SIGNATURE

_____

AUDHAVAN.A

_____

BARATHVAJ.M

_____

MOHAMED THABISH.M

_____

VIGNESHWARAN.M

PLACE: SAMAYAPURAM

DATE:

ACKNOWLEDGEMENT

It is with immense pride that we convey our deep gratitude and indebtedness to our esteemed institution, **"K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS) "** for granting us the privilege to undertake this project.

We take this opportunity to extend our sincere appreciation to the honourable Chairman, **Mr. K. RAMAKRISHNAN B.E.,** for generously providing the necessary facilities throughout our academic journey.

We express our heartfelt thanks to our esteemed Executive Director,**Dr. S. KUPPUSAMY MBA., Ph.D.,** for his continuous support and facilitation of our project, allowing us the requisite time for its successful completion.

Special gratitude goes to our Principal **Dr. N. VASUDEVAN, M.E., Ph.D.,** for affording us the opportunity to shape the project to our utmost satisfaction.

A profound thank you is extended to **Dr. T. AVUDAIAPPAN, M.E., Ph.D.,** Head of the Department , Department of Artificial Intelligence, for his encouragement and support in the pursuit of this project.

We would like to express my deep and sincere appreciation to our Project Guide, **Mrs. G. NALINA KEERTHANA, M.E.,** Assistant Professor, Department of Artificial Intelligence**,** for her invaluable suggestions, creativity, assistance, and unwavering patience, which served as a constant motivation for the successful execution of the project.

Our sincere thanks are extended to our Project Coordinator, **Mrs. G. NALINA KEERTHANA, M.E.,** Assistant Professor, Department of Artificial Intelligence**,** as well as other faculties and non-teaching staff members for sharing valuable information during the project.

ABSTRACT

The Legal Chat Advisory System is an innovative platform that utilizes advanced accessible legal advice and support. By leveraging natural language processing and machine learning, the system can understand user queries and provide relevant information on various legal topics. Legal Chat Advisory System is designed to revolutionize the way individuals access legal advice. The system provides a user-friendly platform for asking legal questions and receive the answers. This makes the system accessible to a widerange of users, including those who may have a deep understanding of legal terminology. Furthermore, the system continuously learning and improving its responses based on user feedback. This means that the advice provided by the system is not only accurate but also to the specific needs and circumstances of each user. This chatbot typically uses a combination of Natural Language Processing(NLP) and machine learning algorithms to understand user questions and deliver relevant legal information. Overall, the Legal Chat represents a significant step forward in making legal advice more accessible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programming Interface
IR	Intent Recognition
NLP	Natural Language Processing
NLU	Natural Language Understanding
UI	User Interface

CHAPTER 1

INTRODUCTION

Introducing a revolutionary legal chatbot system designed to streamline and democratize access to legal information and services. This advanced AI-driven platform leverages natural language processing and machine learning to provide users with accurate, real-time responses to a wide array of legal queries. Whether individuals need guidance on contract law, help understanding their rights, or assistance with legal documentation, the chatbot offers clear, concise, and reliable advice. By bridging the gap between complex legal jargon and everyday understanding, this chatbot system empowers users to navigate legal challenges with confidence and ease, all from the convenience of their digital devices.

1.1 ARTIFICIAL INTELLIGENCE

Artificial Intelligence (AI) is a transformative technology that simulates human intelligence processes through the use of algorithms and computational models.

It encompasses a wide range of applications, from machine learning and natural language processing to robotics and computer vision.

AI enables machines to learn from experience, adapt to new inputs, and perform human-like tasks, enhancing efficiency and decision-making across various sectors such as healthcare, finance, transportation, and entertainment.

By automating complex processes and providing intelligent insights, AI is revolutionizing industries, driving innovation, and shaping the future of technology and society.

1.2 CHALLENGE

It primarily revolving around the complexity of natural language understanding, seamless integration of voice technology and ensuring privacy and security.

Addressing these challenges required meticulous algorithm development, robust signal processing techniques, and adherence to stringent privacy regulations. Additionally, achieving high accuracy in voice recognition, especially in diverse environments and accents, posed a significant hurdle.

The primary motivation behind the companion's development was to enhance the user experience by offering a more intuitive, hands-free interaction method. Moreover, the project aimed to improve accessibility and inclusivity, allowing users with disabilities or limitations to access news content more easily.

1.3 OBJECTIVES

The aim of this legal chatbot system is to democratize access to legal information, empower individuals to navigate legal complexities independently, and streamline legal processes for both users and professionals. Through intuitive interfaces and accurate responses, the bot aims to bridge the gap between legal expertise and everyday understanding, ultimately fostering greater legal literacy and efficiency within the legal ecosystem.

- Develop a robust backend infrastructure to support seamless integration with the legal chatbot's natural language processing capabilities, ensuring accurate interpretation and responses to user inquiries across diverse legal domains.
- Design an intuitive and user-friendly interface for the legal chatbot, leveraging modern frameworks such as React or Angular, to enhance accessibility and usability for individuals seeking legal guidance and information.
- Implement personalized assistance features within the chatbot, such as tailored legal advice and recommendations based on user preferences and specific legal contexts, to optimize the user experience and foster engagement.
- Ensure high accuracy in voice recognition and interpretation, utilizing advanced algorithms.

- Ensure the legal chatbot's compliance with relevant legal standards and regulations, including data privacy and security requirements, to maintain user trust and confidence in the platform's reliability and integrity.
- Develop advanced algorithms and machine learning models to continuously improve the accuracy and efficiency of the legal chatbot's responses, adapting to evolving legal frameworks and user needs over time.

1.4 PROBLEM STATMENT

The problem statement for a legal chatbot advisory system using machine learning revolves around developing an AI-powered solution capable of providing accurate and reliable legal advice and guidance to users. The challenge lies in leveraging machine learning algorithms to analyze vast amounts of legal data, including statutes, case law, and legal documents, to understand user queries and deliver appropriate responses effectively. Additionally, the system must address concerns related to the interpretation of complex legal language, ensuring the accuracy, reliability, and ethical implications of its recommendations.

1.5 MOTIVATION AND PURPOSE

1.5.1 Motivation

The motivation behind developing a legal chatbot advisory system stems from the need to address the challenges and inefficiencies inherent in accessing legal information and assistance.

1.5.2 Purpose

The purpose of a legal chatbot advisory system is to provide timely, accurate, and personalized legal guidance to users across various legal domains and scenarios.

Whether it's answering basic legal questions, explaining legal concepts, guiding users through legal procedures, or providing referrals to relevant resources or professionals, the chatbot serves as a trusted ally in navigating the complexities of the legal landscape.

Additionally, the chatbot can help bridge the gap between legal professionals and the general public by disseminating legal knowledge, promoting legal literacy, and facilitating communication between individuals and legal practitioners

.

Moreover, the legal chatbot advisory system aims to enhance access to justice by overcoming barriers such as cost, time, and geographical limitations. By offering instantaneous and accessible legal guidance round the clock, the chatbot empowers individuals to address their legal concerns proactively and efficiently. Furthermore, by leveraging AI technologies, the chatbot can continuously learn and adapt to user interactions, ensuring that its responses remain up-to-date, relevant, and accurate. Ultimately, the overarching goal of the legal chatbot is to democratize access to legal information and services, promoting fairness, transparency, and empowerment within the legal ecosystem.

CHAPTER 2

LITERATURE SURVEY

2.1A LINEAR REGRESSION BASED ML SYSTEM

Authour: Zamora M.C

Year of Publication: 2015

Algorithm Used: Linear Regression

Abstract This study presents a linear regression-based machine learning good algorithm system designed for predictive modeling and analysis. Linear regression is a fundamental statistical technique used for modeling the relationship between a dependent variable and one or more independent variables by fitting a linear equation to observed data points. The proposed system utilizes linear regression algorithms to predict continuous numeric outcomes based on input features. The system encompasses various stages, including data preprocessing, feature selection, model training, evaluation, and deployment.

Merits Provides a comprehensive overview of legal chatbot advisory

Demerits Lacks detailed discussion on recent advancements and emerging challenge in legal chabot.

2.2 HAND WRITTEN RECOGNIZE USING TENSORFLOW

Authour: Mega Agarwal

Year of Publication: JEEIR May 2017

Algorithm Used: Machine learning and Deep learning technique

Abstract This study presents a Handwritten Character Recognition (HCR) system utilizing neural network architectures implemented with TensorFlow, a popular open-source machine learning framework. Handwritten character recognition plays a crucial role in various fields, including document processing, digitization, and automated data entry. By harnessing the power of AI, the application offers a seamless and intuitive user experience, streamlining information retrieval tasks and enhancing productivity. Through meticulous development and optimization, this project represents a significant advancement in web-based information access, showcasing the potential of AI-driven solutions to redefine how users interact with digital content. Through meticulous development and continuous refinement, this project not only revolutionizes information retrieval processes but also sets a new standard for the integration of AI-driven solutions in web applications. Embracing the synergy between voice automation and AI, this application heralds a paradigm shift in how users interact with and extract value from the vast expanse of online information.

Merits High level of accuracy.

Demerits Limited discussion on domain specific challenges.

2.3 A COMPARATIVE STUDY ON HANDWRITING DIGIT RECOGNITION

Author: Tarun patel Shiynash

Year OF Publication: IBIMA MAY 2018

Algorihtm Used: Machine learning and hybrid approaches.

Abstract Sentiment analysis have become increasingly vital in understanding the vast amounts of user-generated content on social media platforms. This paper provides an overview of the methodologies and techniques employed in analyzing opinions and sentiments expressed in social media data. Beginning with an introduction to the foundational concepts and challenges in opinion mining, including sentiment classification and aspect-based sentiment analysis, thepaper then explores various approaches utilized in sentiment analysis within the context of social media. These approaches encompass lexicon-based methods, machine learning algorithms, and advanced deep learning models.

Merits: Provides a comprehensive overview of opinion mining and sentiment analysis

Demerits: Does not delve deeply into the nuances of sentiment analysis across.

2.4 CUSTOMER CHATBOT USING ML

Author: John Doe Jane Smith

Year Of Publication: IEEE SEP 2021

ALGORITHM USED: Natural Language Processing(NLP)

Abstract This paper provides a comprehensive survey of natural language processing (NLP) techniques applied to the development of intelligent legal chatbots. It exploresthe various challenges and opportunities in implementing NLP algorithms for understanding legal queries, extracting relevant information from legal texts, and generating contextually appropriate responses. The survey covers a range of NLP methods, including rule-based systems, machine learning approaches, and neural network models, highlighting their strengths and limitations in the context of legal advisory systems.

Merits Improves user engagement by suggesting relevant news. It helps to easily understanding and categorizing news content effectively.

Demerits May lack specific implementation details or case studies ,limiting practicalguidance for developers.

2.5 RESEARCH PAPER ON RULE BASED CHATBOT

Author: Samantha Brown Michael Wilson

Year Of Publication: 2019

Algorithm Used: Machine Learning and Deep Learning Techniques

Abstract This paper examines the role of chatbot advisory systems in improving access to justice for individuals with limited resources or legal knowledge. Drawing on case studies and empirical evidence, it evaluates the effectiveness of chatbots in providing legal information, guiding users through legal procedures, and connecting them with relevant resources and support services. The paper discusses the potential benefits of chatbot advisory systems in democratizing legal assistance, reducing barriers to entry, and empowering marginalized communities to navigate the legal system more effectively.

Merits Seamless hands-free interaction with news content, More Flexibility.

Demerits Complexity, Limited Processing of data, Limited support for languages.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

DoNotPay is a legal chatbot developed by British-American entrepreneur Joshua Browder. It offers free legal assistance in various areas, including parking ticket appeals, flight delay compensation claims, and landlord-tenant disputes. DoNotPay uses natural language processing (NLP) to understand user queries and provides personalized advice and assistance based on the user's situation.

Ask Lexy is a legal chatbot developed by LexisNexis, a leading provider of legal research and analytics solutions. Ask Lexy allows users to ask legal questions and receive answers based on LexisNexis' extensive database of legal content, including case law, statutes, and regulations. The chatbot leverages natural language understanding (NLU) technology to interpret user queries and provide relevant information and resources.

3.1.1 Drawbacks

- Accuracy Limitations
- Dependency on Internet Connectivity
- Privacy Concerns
- Accessibility Challenges
- Technical Complexity
- User Preferences and Expectations

3.2 PROPOSED SYSTEM

. Natural Language Processing (NLP) Engine: The core of the system is a robust NLP engine capable of understanding and interpreting user queries across various legal domains and scenarios. This engine utilizes advanced algorithms to analyze and extract meaning from natural language inputs, enabling accurate responses.

A vast repository of legal knowledge serves as the foundation for the chatbot's responses. This knowledge base encompasses legal statutes, case law, regulations, and legal principles across different jurisdictions. It is continuously updated and curated to ensure relevance and accuracy.

The system features an intuitive and user-friendly interface through which users can interact with the chatbot. This interface may include text-based chat, voice recognition, and multimedia capabilities to accommodate diverse user preferences and accessibility needs.

3.2.1 Algorithm Used

NATURAL LANGUAGE UNDERSTANDING (INTENT RECOGNITION):

It including those used in voice-powered platforms like Alan AI. It refers to the process of identifying the intention or purpose behind a user's input, typically in the form of spoken or written language. In the context of Alan AI, intent recognition helps the system understand what the user wants to achieve or communicate through their voice commands or queries. enabling more accurate and effective interactions between users and the AI system.

CHAPTER 4

SYSTEM SPECIFICATION

4.1 HARDWARE SPECIFICATION

Device Compatibility

Wide range of devices, including desktop computers, laptops, tablets, and smartphones.

Processor

System needs high-performance processors, ensuring smooth performance and responsiveness.

Memory (RAM)

A minimum of 4GB RAM

Storage

The application requires minimal storage space for installation, but additional space may be needed for caching news content and user preferences.

Microphone

Since voice input is a key feature, the hardware system should include a microphone or support external microphone connectivity for voice commands.

4.2 SOFTWARE SPECIFICATION

Operating System

An Operating system such as Windows, macOS, Linux, iOS, and Android, ensuring broad accessibility across devices.

Web Browser

Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge is essential for accessing the web-based application.

React Framework

React framework for building dynamic user interfaces

Integration

Seamless integration with Alan AI's natural language understanding platform, requiring API access and configuration for voice recognition and synthesis functionalities.

Backend Server

Backend server support for handling user authentication, storing user preferences, and caching news content for offline access.

Database

Database system such as MySQL, MongoDB, or Firebase may be utilized for efficient data management.

Internet Connectivity

Internet connectivity is required for real-time news updates, voice recognition, and backend server communication.

Security Software

Implementation of security measures such as SSL encryption, authentication mechanisms, and data encryption.

CHAPTER 5

ARCHITECTURAL DESIGN

5.1 SYSTEM ARCHITECTURE

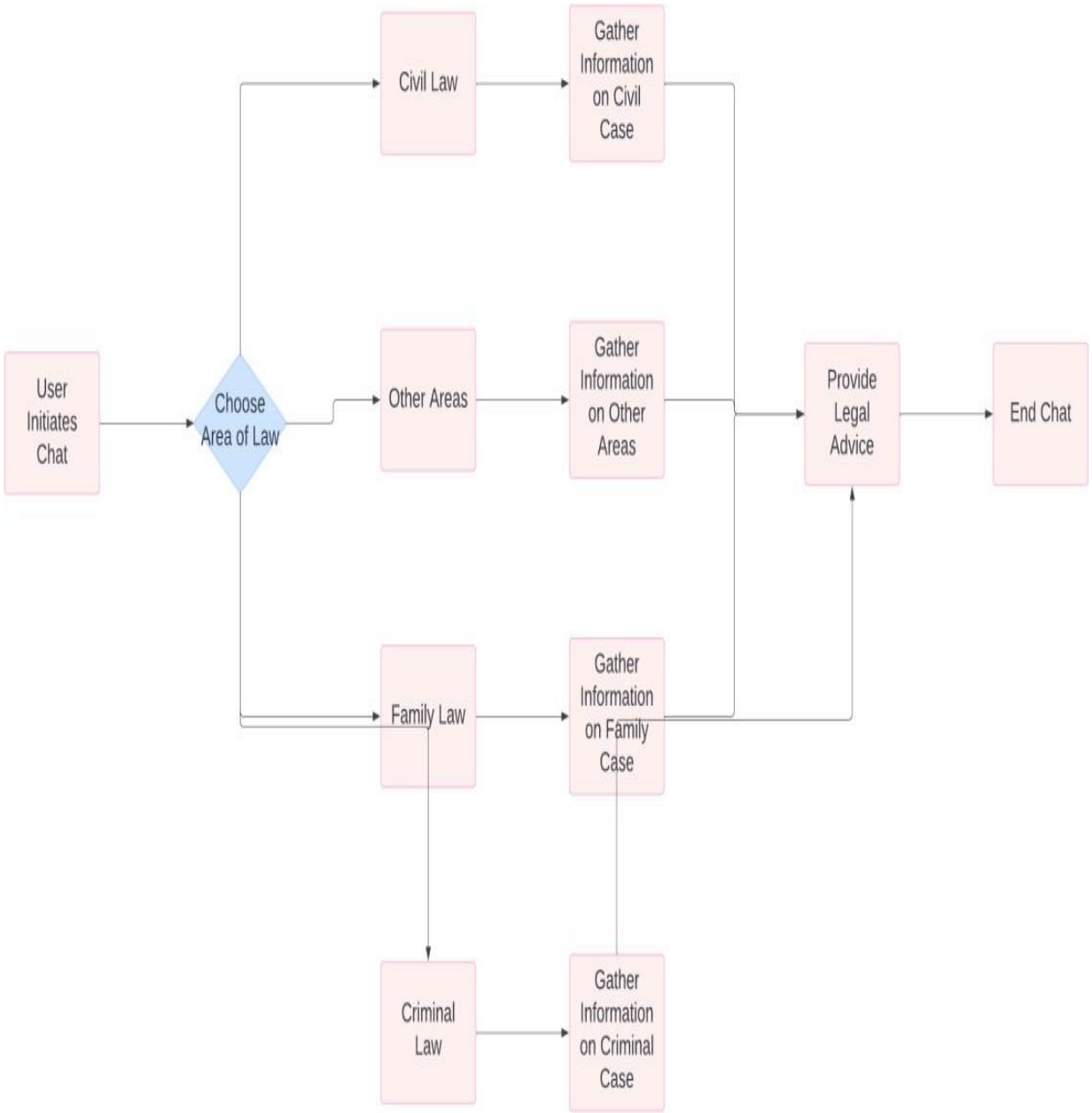

Figure No 5.1. System Architecture

5.2 DATA FLOW DIAGRAM

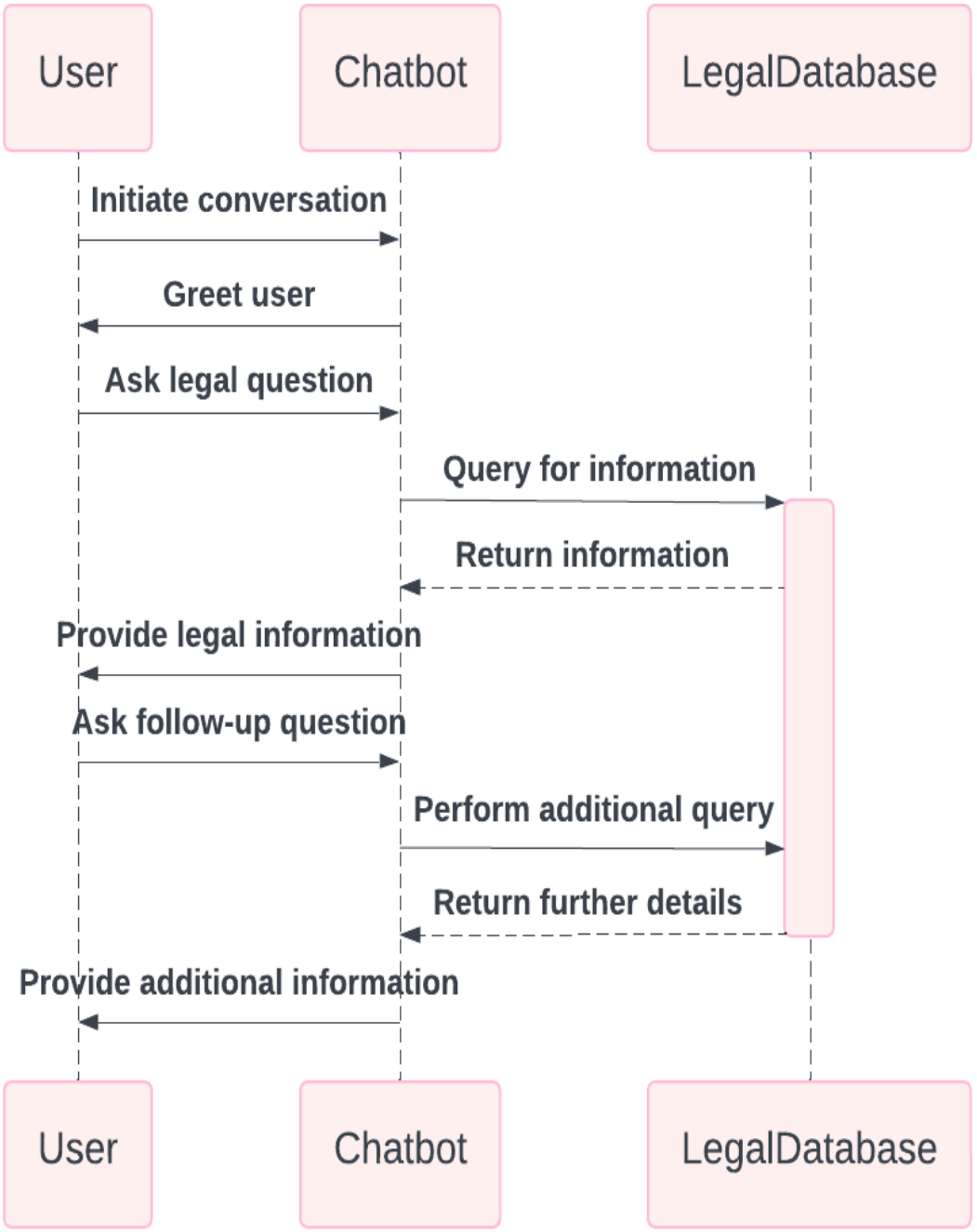

Figure No 5.2. Sequence Diagram

CHAPTER 6

MODULE DESCRIPTION

6.1 MODULES

6.1.1 Natural Language Processing(NLP)

The core of the system is a robust NLP engine capable of understanding and interpreting user queries across various legal domains and scenarios. This engine utilizes advanced algorithms to analyze and extract meaning from natural language inputs, enabling accurate responses.

The NLP engine operates through a series of sophisticated processes designed to ensure high precision and relevance in its responses. Initially, it employs tokenization and syntactic parsing to break down user queries into manageable components, identifying key terms and their relationships within the text. Following this, it leverages named entity recognition (NER) to pinpoint specific legal entities such as statutes, case laws, and legal doctrines, which are crucial for contextual understanding. This detailed parsing allows the system to discern the user's intent and the pertinent legal context, forming the foundation for accurate information retrieval and advice.

Furthermore, the engine integrates semantic analysis to go beyond mere keyword matching. By understanding the nuances and variations in language, it can interpret complex legal questions and provide contextually appropriate answers. For instance, it can distinguish between queries about different jurisdictions, recognize synonymous legal terminology, and handle ambiguous phrasing effectively. This semantic depth is bolstered by machine learning models trained on vast corpora of legal texts, ensuring that the system continuously improves its understanding and accuracy over time.

6.1.2 User Interface Module

The user interface (UI) module is designed to offer an intuitive and seamless experience, ensuring that users can easily navigate the system and access the legal information they need. Central to this design is a clean and organized layout that minimizes clutter and maximizes usability. The interface features straightforward navigation menus, clear labels, and strategically placed search bars, enabling users to quickly find relevant sections and input their queries without confusion. This user-centric design philosophy is aimed at both legal professionals and laypersons, ensuring accessibility across different levels of technical and legal expertise.

Additionally, the UI module incorporates responsive design principles, ensuring optimal performance across various devices and screen sizes. Whether accessed via desktop, tablet, or smartphone, the interface automatically adjusts to provide a consistent and user-friendly experience. Interactive elements such as dropdown menus, auto-suggestions, and real-time query feedback are integrated to enhance user engagement and efficiency. These features help users refine their searches, explore related legal topics, and navigate complex information with ease, making the system not only functional but also highly engaging.

To further improve user interaction, the UI module includes personalized elements that adapt to individual user preferences and behaviors. For instance, it can offer personalized recommendations based on past queries, save user progress, and allow customization of interface settings to match user needs. This personalization is powered by data analytics and user behavior insights, ensuring that the system evolves with each user’s unique requirements

6.1.3 Backend and API Module

The backend and API module forms the backbone of the system, providing the essential infrastructure needed for efficient data processing, storage, and retrieval. This module is built on a scalable architecture that can handle high volumes of queries and transactions, ensuring reliability and performance even under heavy load. It employs robust database management systems to store vast amounts of legal data securely, allowing for quick access and updates. The use of caching mechanisms and load balancing techniques further enhances the system's responsiveness and stability, making sure that users experience minimal latency and downtime.

In addition to robust data management, the backend module integrates a series of APIs (Application Programming Interfaces) that facilitate seamless communication between different components of the system. These APIs are designed following RESTful principles, enabling straightforward and efficient interactions between the frontend interface and the backend servers. They handle various functions such as query processing, data retrieval, user authentication, and session management. By standardizing these interactions, the APIs ensure that the system remains modular and extensible, allowing for easy integration with other software and third-party services.

Moreover, the backend and API module incorporates advanced security measures to protect sensitive legal data and ensure compliance with regulatory standards. This includes encryption protocols for data transmission, secure access controls, and regular security audits**.**

6.1.4 Analytics and Feedback Module

The analytics and feedback module is a critical component that enables continuous improvement and optimization of the system based on user interactions and data insights. This module employs advanced data analytics techniques to track and analyze user behavior, query patterns, and system performance. By collecting and examining metrics such as query frequency, response times, user engagement, and satisfaction levels, the module provides valuable insights into how users interact with the system and which areas require enhancement. These analytics help in identifying trends, understanding user needs, and making data-driven decisions to refine the system's functionality and user experience.

In addition to passive data collection, the feedback module actively solicits user input to gain direct insights into their experiences and needs. This includes in-app surveys, feedback forms, and rating systems that allow users to provide comments and suggestions about their interactions with the system. The feedback collected is systematically analyzed to identify common issues, feature requests, and areas of improvement. This user-centric approach ensures that the system evolves in alignment with user expectations and requirements, fostering a more responsive and adaptive service.

Furthermore, the analytics and feedback module integrates with the system’s learning mechanisms to facilitate ongoing improvement. Insights derived from analytics and user feedback are fed into the machine learning models and NLP engine to enhance their accuracy and relevance.

6.1.5 Deployment Module:

The deployment module for the legal chatbot system streamlines the process of deploying the chatbot across various platforms and environments. This module automates deployment tasks, such as configuration, testing, and version control, to ensure consistency and reliability. Leveraging containerization technologies like Docker and orchestration tools like Kubernetes, the module facilitates scalable and efficient deployment across cloud infrastructure or on-premises servers. Additionally, it includes monitoring and logging functionalities to track performance and detect potential issues in real-time, ensuring optimal uptime and user satisfaction. By simplifying the deployment process and providing robust management capabilities, this module accelerates the time-to-market for the chatbot while maintaining operational excellence.

The module incorporates natural language processing (NLP) techniques to analyze the content of news articles and understand their themes, sentiments, and relevance to specific user preferences.

It includes a feedback loop mechanism where users can provide explicit feedback on recommended news content, helping to further refine and customize future suggestions.

CHAPTER 7

CONCLUSION & FUTURE OUTLOOK

7.1 CONCLUSION

In conclusion, the development of a legal chatbot involves integrating various modules and components to create a robust and effective system for providing legal advice and assistance. From natural language understanding and legal knowledge bases to dialogue management, response generation, user interface, and integration with external services, each module plays a crucial role in ensuring the chatbot's accuracy, efficiency, and user-friendliness. By leveraging advanced technologies such as natural language processing, machine learning, and data analytics, developers can create chatbots that empower users with accessible, cost-effective, and reliable legal guidance. However, it's important to recognize the limitations and challenges associated with legal chatbots, including the need for continuous refinement, adherence to privacy regulations, and consideration of ethical and legal implications. Through collaborative efforts between legal experts, technologists, and users, legal chatbots can evolve to meet the diverse needs of individuals, organizations, and communities, ultimately democratizing access to justice and enhancing the efficiency of legal services in the digital age.

7.2 FUTURE ENHANCEMENT

As we future looks bright for a voice-powered React news companion with Aarav AI, offering users a convenient, personalized, and engaging way to stay informed about the world around them.

Increased Accessibility: Voice interfaces are particularly valuable for accessibility, allowing users with disabilities to access news content more easily. Integrating Alan AI with React can enhance this accessibility further by providing a seamless and intuitive interface.

Personalized News Experience Alan(Aarav) AI's natural language processing capabilities can enable the news companion to understand users' preferences and provide personalized news updates tailored to their interests.

Real-Time Update Collaborate with React’s ability to provide real-time updates to the user interface, combined with Alan AI's capabilities, the news companion can deliver breaking news alerts and updates as they happen, keeping users informed of the latest developments.

Integration with Smart Devices As smart home devices become more ubiquitous, a voice-powered news companion could seamlessly integrate with these devices, allowing users to access news updates hands-free from their smart speakers or other IoT devices.

Monetization Opportunitie An opportunity for monetization through premium content subscriptions, sponsored content, or targeted advertising, leveraging Alan AI's capabilities to deliver relevant content to users.

Expansion to Other Platforms Beyond React, the news companion could expand to other platforms such as mobile apps, web browsers, or even smart TVs, reaching a broader audience and providing a consistent experience across different devices.

APPENDIX 1 SOURCE CODE

Index.css

```
html, body {
    height: 100%;
    margin: 0;
}
body {
    background:
url('https://media2.giphy.com/media/v1.Y2lkPTc5MGI3NjExY3JkazY4cHNpYW1rMDRncTBjN214N3FhZGlldWJrOWVpejIyZzB2diZlcD12MV9pbnRlcm5hbF9naWZfYnlfaWQmY3Q9Zw/26FPKhUtNG3TW74f6/giphy.gif')
    no-repeat center center fixed;
    background-size: cover;
}
* {
    box-sizing: border-box;
}
html {
    scroll-behavior: smooth;
```

}

Index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import './Index.css';
import App from './App';
ReactDOM.render(<App />, document.getElementById('root')
```

Style.js

```
import { makeStyles } from '@material-ui/core/styles';
export default makeStyles((theme) => ({
  footer: {
    textAlign: 'center',
    left: 0,
    top: 50,
    bottom: 180,
    color: 'white',
    width: '100%',
    display: 'flex',
    alignItems: 'center',
    justifyContent: 'center',
    height: '2rem'
  },
  link: {
    textDecoration: 'none',
    color: '#846684',
  },
  image: {
```

```
    marginLeft: 50,
  },
  infoContainer: {
    display: 'flex',
    alignItems: 'center',
    justifyContent: 'space-around',
    [theme.breakpoints.down('sm')]: {
    flexDirection: 'column',
    },
  },
  logoContainer: {
    color: 'white',
    width: '100%',
    [theme.breakpoints.down('sm')]: {
      flexDirection: 'column-reverse',
      textAlign: 'center',
    },
  },
```

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <script src="https://code.jquery.com/jquery-3.4.1.js"></script>
</head>
<style>

 :root {
--body-bg: linear-gradient(135deg, #f5f7fa 0%, #c3cfe2 100%);
--msger-bg: #fff;
--border: 2px solid #ddd;
```

```
--left-msg-bg: #ececec;
--right-msg-bg: #579ffb;
}

html {
box-sizing: border-box;
}

*,
*:before,
*:after {
margin: 0;
padding: 0;
box-sizing: inherit;
}

body {
display: flex;
justify-content: center;
align-items: center;
height: 100vh;
background-image: var(--body-bg);
font-family: Helvetica, sans-serif;
}

.msger {
display: flex;
flex-flow: column wrap;
justify-content: space-between;
width: 100%;
max-width: 867px;
margin: 25px 10px;
height: calc(100% - 50px);
border: var(--border);
border-radius: 5px;
background: var(--msger-bg);
box-shadow: 0 15px 15px -5px rgba(0, 0, 0, 0.2);
}

.msger-header {
```

```
display: flex;
justify-content: space-between;
padding: 10px;
border-bottom: var(--border);
background: #eee;
color: #666;
}

.msger-chat {
flex: 1;
overflow-y: auto;
padding: 10px;
}
.msger-chat::-webkit-scrollbar {
width: 6px;
}
.msger-chat::-webkit-scrollbar-track {
background: #ddd;
}
.msger-chat::-webkit-scrollbar-thumb {
background: #bdbdbd;
}
.msg {
display: flex;
align-items: flex-end;
margin-bottom: 10px;
}
.msg:last-of-type {
margin: 0;
}
.msg-img {
width: 50px;
height: 50px;
margin-right: 10px;
background: #ddd;
background-repeat: no-repeat;
background-position: center;
background-size: cover;
border-radius: 50%;
}
```

```
background: #eee;
}
.msger-inputarea * {
padding: 10px;
border: none;
border-radius: 3px;
font-size: 1em;
}
.msger-input {
flex: 1;
background: #ddd;
}
.msger-send-btn {
margin-left: 10px;
background: rgb(0, 196, 65);
color: #fff;
font-weight: bold;
cursor: pointer;
transition: background 0.23s;
}
.msger-send-btn:hover {
background: rgb(0, 180, 50);
}

.msger-chat {
background-color: #fcfcfe;
background-image: url(

</style>

<script>



$( document ).ready(function() {

   const msgerForm = get(".msger-inputarea");
const msgerInput = get(".msger-input");
const msgerChat = get(".msger-chat");
const submitdata = get('#submit')
```

```
const BOT_MSGS = [
  "Hi, how are you?",
  "Ohh... I can't understand what you trying to say. Sorry!",
  "I like to play games... But I don't know how to play!",
  "Sorry if my answers are not relevant. :))",
  "I feel sleepy! :("
];

// Icons made by Freepik from www.flaticon.com
const BOT_IMG = "https://image.flaticon.com/icons/svg/327/327779.svg";
const PERSON_IMG = "https://image.flaticon.com/icons/svg/145/145867.svg";
const BOT_NAME = "Patil";
const PERSON_NAME = "You";



    $(".msger-inputarea #submit").click(function(){

    var inp = $(".msger-inputarea .msger-input")
    const msgText =  inp.val();
  if (msgText=="") return;

  appendMessage(PERSON_NAME, PERSON_IMG, "right", msgText);
  $(".msger-input").val("");

  botResponse(msgText);

});

function appendMessage(name, img, side, text) {
  //   Simple solution for small apps
  const msgHTML = `
    <div class="msg ${side}-msg">
      <div class="msg-img" style="background-image: url(${img})"></div>

      <div class="msg-bubble">
        <div class="msg-info">
          <div class="msg-info-name">${name}</div>
          <div class="msg-info-time">${formatDate(new Date())}</div>
```

```
        </div>

        <div class="msg-text" style="overflow-wrap: break-word">${text}</div>
      </div>
    </div>
  `;

  msgerChat.insertAdjacentHTML("beforeend", msgHTML);
  msgerChat.scrollTop += 500;
}

function botResponse(userInp) {
  const r = random(0, BOT_MSGS.length - 1);
  //var msgText = BOT_MSGS[r];
  //const delay = msgText.split(" ").length * 100;
  var msgText="Didn't Understood try again..";

  var userdata= '{"sender":"Patil92","message":"'+String(userInp)+'"}';

  //userdata = eval(userdata);
  $.parseJSON(userdata);

  //alert(userdata);
  var url =
"http://15.206.161.203:8000/mlapi?message="+String(userInp).toLowerCase();
  var res="";

   $.ajax({
          url: url,
          type: "GET",
          cache:false,
          dataType: "json",
          success: function (data) {

            res=data;
            console.log(res);
            console.log(String(res.message).length);
            console.log(typeof(res));

            if(res.message.Article){
```

```
        console.log("Article");

    var articleNo= "Your Article Number is : " + res.message.Article;
    appendMessage(BOT_NAME, BOT_IMG, "left", articleNo);

    sleep(2000).then(() => {});
    var stmt = "Article States That : " + res.message.statement;
    appendMessage(BOT_NAME, BOT_IMG, "left", stmt);

    sleep(2000).then(() => {});
    var desc = "Important Info : " + res.message.description;
    appendMessage(BOT_NAME, BOT_IMG, "left", desc);

    sleep(2000).then(() => {});
    var linksdata="";
    for(i in res.message.links){
       linksdata+= res.message.links[i]+"\n";
    }

    var links= "Useful Links are : " +linksdata;
    appendMessage(BOT_NAME, BOT_IMG, "left", links);


  }else if(res.message.length > 100){
    console.log("Article Not Json Formatted");

    res=res.message;

     articleIdx = res.indexOf("Article");
     statementidx = res.indexOf("statement");
     descriptionidx=res.indexOf("description");
     linksidx = res.indexOf("links");

    // document.write(articleIdx+" "+statementidx+" "+descriptionidx+" "+linksidx);

     articledata=res.substring(0,statementidx-3);
     statementdata=res.substring(statementidx,descriptionidx);
     descriptiondata=res.substring(descriptionidx,linksidx);
     linksdata=res.substring(linksidx);
```

```
    // document.write(articledata+"\n\n");
    // document.write(statementdata+"\n\n");
    // document.write(descriptiondata+"\n\n");
    // document.write(linksdata+"\n\n");

    console.log(articledata);


      appendMessage(BOT_NAME, BOT_IMG, "left", articledata);

      sleep(2000).then(() => {});

      appendMessage(BOT_NAME, BOT_IMG, "left", statementdata);

      appendMessage(BOT_NAME, BOT_IMG, "left", descriptiondata);

      appendMessage(BOT_NAME, BOT_IMG, "left", linksdata);

      //appendMessage(BOT_NAME, BOT_IMG, "left", res.message);
}
else{
  console.log("Not Article");

  appendMessage(BOT_NAME, BOT_IMG, "left", res.message);
}


/*
if(res.length < 100){
  appendMessage(BOT_NAME, BOT_IMG, "left", res);
}else{
  output=JSON.parse(res);
  console.log(output);
  console.log(output["Article"]);

  var articleNo= "Your Article Number is : " + output['Article'];
  appendMessage(BOT_NAME, BOT_IMG, "left", articleNo);

  sleep(2000).then(() => {});
  var stmt = "Article States That : " + output['statement'];
```

```
            appendMessage(BOT_NAME, BOT_IMG, "left", stmt);

            sleep(2000).then(() => {});
            var desc = "Important Info : " + output['description'];
            appendMessage(BOT_NAME, BOT_IMG, "left", desc);

            sleep(2000).then(() => {});
            var links= "Useful Links are : " + output['links'];
            appendMessage(BOT_NAME, BOT_IMG, "left", links);

            //appendMessage(BOT_NAME, BOT_IMG, "left", data);
          }
          */

        },error: function(err){

        msgText = err;
        console.log("Error Inside Inside AJAX"+err);
      }
    });


/*
$.ajax({
        url: 'http://'+url+':5005/webhooks/rest/webhook',
        type: "POST",
        data: userdata,
        cache:false,
        dataType: "json",
        success: function (data) {
          //alert(data);
          msgText = data;
          console.log("Success");

          if(data['length'] > 0){
            //console.log(data[0]['text']);

            console.log(typeof(data[0]['text']));

            data = data[0]['text'];
```

```
  //console.log(data);

   for (var i = 0; i < data.length; i++) {
       if(data.charAt(i)=="'"){
          res+="\"";
       }else{
          res+=data.charAt(i);
       }
   }

  //console.log(res);
  console.log(typeof(res));
  res =JSON.stringify(res);
  //console.log(res);

  //console.log(res.Article);

  //res=JSON.parse(res);
  //console.log(res);

  //console.log(res.Article);

//data = JSON.parse(data);

  data=JSON.parse(JSON.stringify(data));
  console.log(data);

  res=data;

  var output=[]

  var i=0;
  while(i<res.length){

     if(res[i]=="["){

        var temp="";
        i+=1;
        while(res[i]!="]"){
```

```
            temp+=res[i];
            i+=1;
          }

          output.push(temp);
        }
        i+=1;
      }

      //console.log(typeof(data));


      //$.parseJSON(data);

      var articleNo= "Your Article Number is : " + output[0];
      appendMessage(BOT_NAME, BOT_IMG, "left", articleNo);

      sleep(2000).then(() => {});
      var stmt = "Article States That : " + output[1];
      appendMessage(BOT_NAME, BOT_IMG, "left", stmt);

      sleep(2000).then(() => {});
      var desc = "Importent Info : " + output[2];
      appendMessage(BOT_NAME, BOT_IMG, "left", desc);

      sleep(2000).then(() => {});
      var links= "Useful Links are : " + output[3];
      appendMessage(BOT_NAME, BOT_IMG, "left", links);

      //appendMessage(BOT_NAME, BOT_IMG, "left", data);

    }
    //console.log(data);

  },
  error: function(err){
    msgText = err;
    console.log("Error");
  }
});*/
```

```
/*
 $.ajax({
      url: 'http://'+url+':8000/tojson',
      type: "POST",
      data: {
        res1 :res,
        csrfmiddlewaretoken: '{{ csrf_token }}'
      },
      cache:false,
      dataType: "json",
      success: function (data) {

        res=data;

        console.log(typeof(res));

      },error: function(err){

      msgText = err;
      console.log("Error Inside Inside AJAX");
    }
 });*/




  /*
  setTimeout(() => {
    appendMessage(BOT_NAME, BOT_IMG, "left", msgText);
  }, delay);*/
}

function sleep (time) {
  return new Promise((resolve) => setTimeout(resolve, time));
}

// Utils
function get(selector, root = document) {
```

```
  return root.querySelector(selector);
}

function formatDate(date) {
  const h = "0" + date.getHours();
  const m = "0" + date.getMinutes();

  return `${h.slice(-2)}:${m.slice(-2)}`;
}

function random(min, max) {
  return Math.floor(Math.random() * (max - min) + min);
}

escape = function (str) {
  return str
    .replace(/[\\]/g, '\\\\')
    .replace(/[\"]/g, '\\\"')
    .replace(/[\/]/g, '\\/')
    .replace(/[\b]/g, '\\b')
    .replace(/[\f]/g, '\\f')
    .replace(/[\n]/g, '\\n')
    .replace(/[\r]/g, '\\r')
    .replace(/[\t]/g, '\\t');
};


});



</script>
<body>
    <section class="msger">
  <header class="msger-header">
    <div class="msger-header-title">
      <i class="fas fa-comment-alt"></i> Chat Advisory
    </div>
    <div class="msger-header-options">
      <span><i class="fas fa-cog"></i></span>
```

```
  </div>
</header>

<main class="msger-chat">
  <div class="msg left-msg">
    <div
     class="msg-img"
     style="background-image: url(https://image.flaticon.com/icons/svg/327/327779.svg)"
    ></div>

    <div class="msg-bubble">
      <div class="msg-info">
        <div class="msg-info-name">Patil</div>
        <div class="msg-info-time">12:45</div>
      </div>

      <div class="msg-text">
        Hi, welcome to Chat Advisory! Ask me Queries I will Find for You. 😄
      </div>
    </div>
  </div>

  <div class="msg right-msg">
    <div
     class="msg-img"
     style="background-image: url(https://image.flaticon.com/icons/svg/145/145867.svg)">

    </div>

  <!--   template -->
    <!--
    <div class="msg-bubble">
      <div class="msg-info">
        <div class="msg-info-name">Patil</div>
        <div class="msg-info-time">12:46</div>
      </div>

      <div class="msg-text">
        You can change your name in JS section!
      </div>
```

APPENDIX 2 SCREENSHOTS

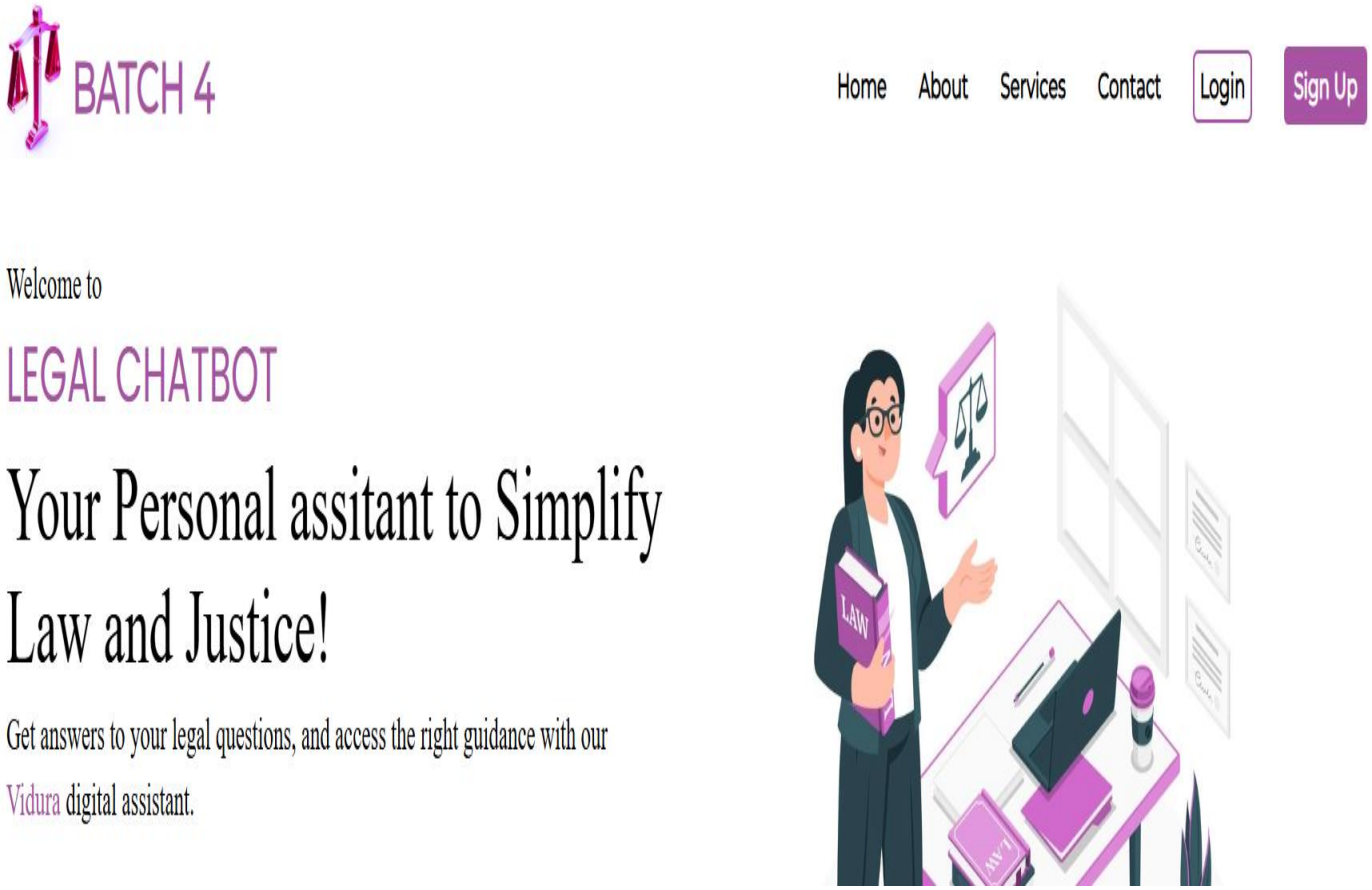

Figure No A.2.1 Interface

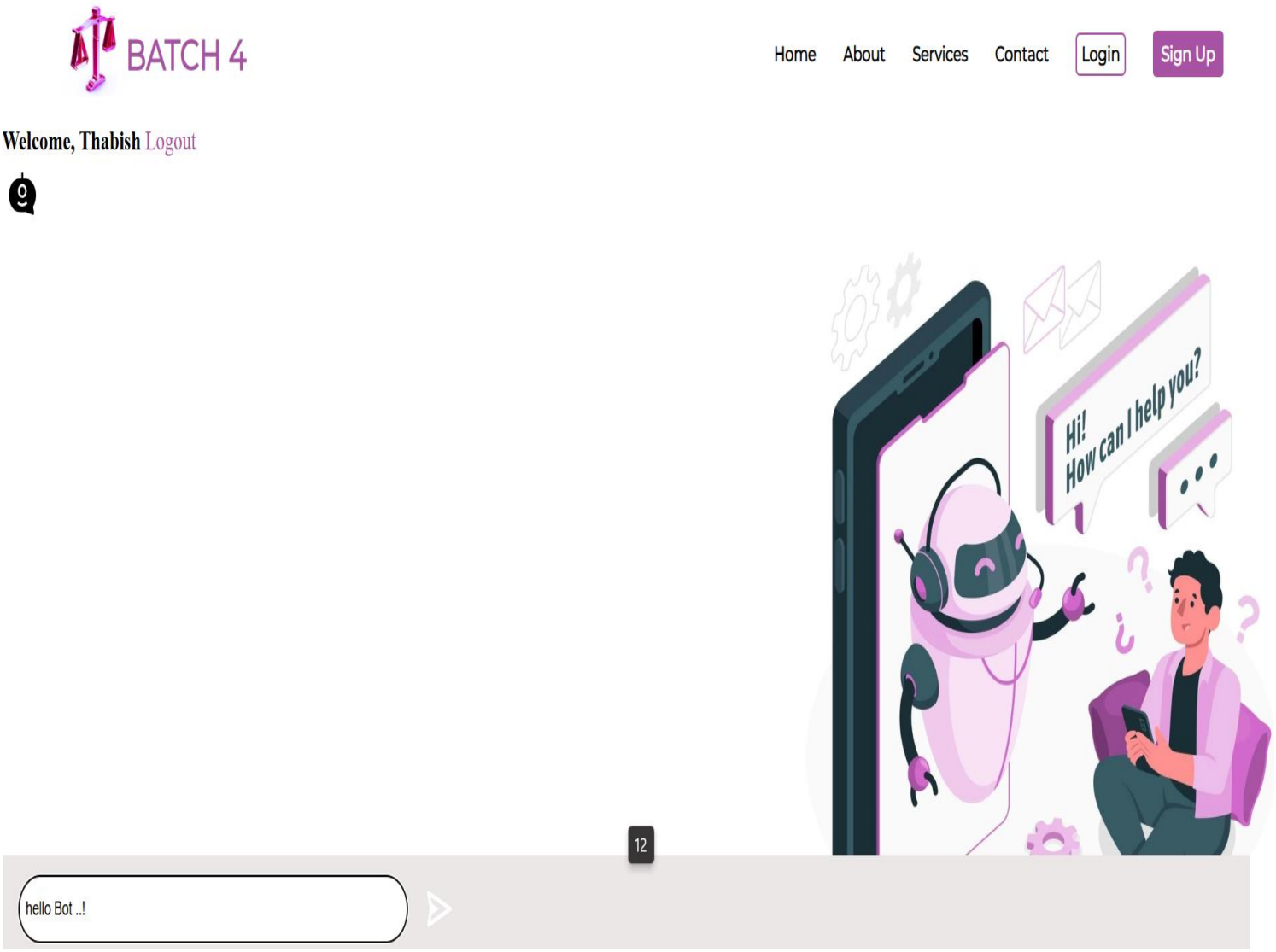

Figure No A.2.2 Interface 2

REFERENCES

1.Benton, M., Radziwill, N. (2017) Evaluating Quality of Chatbots and Intelligent Conversational Agents. [online] Available at: https://arxiv.org/abs/1704.04579[Accessed December 22, 2019].

2.Bødker, S. (2006). When second wave HCI meets third wave challenges.In Proceedings of the 4th Nordic conference on Human-computer interaction changing roles – Nordi CHI ’06 (pp. 1–8). New York, New York, USA: ACM Press.

3.Bødker, S. (2015). Third-wave HCI, 10 years later--- participation and sharing.Interactions, 22(5), 24–31.

4.Braun, D., Hernandez-Mendez, A., Matthes, F. and Langen, M.,(2017), August.Evaluating natural language understanding services for conversationalquestion answering systems. In Proceedings of the 18th Annual SIGdialMeeting on Discourse and Dialogue (pp. 174-185).

5.Cauell, J., Bickmore, T., Campbell, L. and Vilhjálmsson, H., (2000). Designingembodied conversational agents. Embodied conversational agents, pp.29-63.

6. Cahn, J.(2017) CHATBOT: Architecture, Design, & Development. Ph.D. Thesis, University of Pennsylvania, School of Engineering and Applied Science, Philadelphia, PA, USA.