

**CS3491 ARTIFICIAL INTELLIGENCE AND**

**MACHINE LEARNING**

**Project Title**

**ENHANCED AI-BASED CHATBOT FOR RIT COLLEGE**

**Submitted by**

**ABINAYA G M (953623205001)**

**ASHIKA A (953623205005)**

**BENITHA FRANCY F (953623205007)**

**JEYALAKSHMI A (953623205022)**

**In partial fulfilment for the award of the**

**degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**INFORMATION TECHNOLOGY**



**RAMCO INSTITUTE OF TECHNOLOGY**

**RAJAPALAYAM – 626 117**

# ANNA UNIVERSITY: CHENNAI 600025

# JUNE 2025

**ANNA UNIVERSITY: CHENNΑΙ 600 025**

**TABLE OF CONTENTS**

| **CHAPTER NO** | TITLE | PAGE NO |
| --- | --- | --- |
| **1** | **ABSTRACT** | 3 |
| **2** | **LIST OF ABBREVATION** | 4 |
| **3** | **LIST OF FIGURES** | 5 |
| **4** | **INTRODUCTION**  1.1 Scope of the project  1.2 Objective of the project  1.3 Review Summary | 6 |
| **5** | **LITERATURE SURVEY**  2.1 Introduction | 7 |
| **6** | **EXISTING SYSTEM**  3.1 System Model  3.2 Literature Conclusion | 8 |
| **7** | **PROPOSED WORK**  4.1 Introduction  4.2 System Framework  4.3 Proposed Methodology | 9 |
| **8** | **SYSTEM SPECIFICATION**  5.1 Software Requirement  5.1.1 Pandas 5.1.2 NumPy 5.1.3 TensorFlow 5.1.4 NLTK 5.1.5 SpeechRecognition 5.1.6 PyAudio 5.1.7 pyttsx3 5.1.8 Streamlit  5.2 Hardware Requirement  5.3 Dataset Description | 9 |
| **9** | **IMPLEMENTATION AND RESULTS** | 12 |
| **10** | **CONCLUSION AND FUTURE SCOPE**  APPENDIX – I(coding)  APPENDIX – II(References) | 13 |

**ABSTRACT**

This project presents a voice-enabled AI chatbot developed for Ramco Institute of Technology (RIT), designed to interact with users through natural language queries related to the institution. The system integrates natural language processing (NLP), machine learning, and speech recognition to provide an intelligent, voice-interactive assistant that can handle college-related queries in real time.

The chatbot uses a structured JSON-based intent dataset which contains patterns (user queries) and corresponding responses for different categories like courses offered, departments, placements, and general college info. When the user speaks a query, the system uses Google’s SpeechRecognition API to convert speech to text. The input is then preprocessed using NLTK for tokenization and lemmatization. A bag-of-words vector is generated for each input sentence and passed into a neural network classifier built using TensorFlow and Keras.

The model is trained directly within the same script using intents.json, producing a chatbot\_model.h5 file, which is used for inference after training. The chatbot uses the trained model to classify user intent and selects an appropriate response from the intent group. Once the response is chosen, it is output both as printed text and spoken audio using the pyttsx3 text-to-speech engine.

The solution runs as a fully standalone Python script that includes data preprocessing, training, inference, and interaction. It supports fallback input (manual typing) if voice input fails, making it robust and accessible. This chatbot can be customized with more college-specific data and deployed on desktops or integrated into web/mobile apps for student support.

This project showcases a practical implementation of conversational AI in an educational context and highlights the potential of combining machine learning, NLP, and speech technologies to create user-friendly and intelligent virtual assistants.

**LIST OF ABBREVATIONS**

| **S.NO** | **ABBREVATION** | **EXPANSION** |
| --- | --- | --- |
| **1** | NumPy | Numerical Python |
| **2** | TensorFlow | Tensor (multi-dimensional array) and Flow (flow of data in computation graphs) |
| **3** | NLTK | Natural Language Toolkit |
| **4** | PyAudio | Python Audio |
| **5** | pyttsx3 | Python Text-To-Speech v3 |

**LIST OF FIGURES**

| **FIGURE NO** | **NAME OF THE FIGURE** | **PAGE NO** |
| --- | --- | --- |
| **1** | Dataset Model | 10 |
| **2** | Layout of the chatbot(browser) | 11 |
| **3** | Conversation of bot and user | 12 |
| **4** | Layout of the chatbot(Network URL) | 12 |
| **5** | Training the Model with dataset | 15 |

**I INTRODUCTION**

* 1. **AIM OF THE PROJECT:**

To develop a voice-enabled intelligent chatbot system for Ramco Institute of Technology (RIT) that can interact with users through speech and text using natural language processing and machine learning, to provide accurate and automated responses to college-related queries.

* 1. **OBJECTIVE OF THE PROJECT:**
* To design and implement a user-friendly chatbot that understands both typed and spoken queries from users.
* To collect and organize domain-specific knowledge about RIT, such as courses offered, departments, placements, and faculty information, into a structured dataset.
* To preprocess the user input using NLP techniques such as tokenization and lemmatization for improved accuracy.
* To build and train a machine learning model using the Bag-of-Words approach and a neural network classifier with TensorFlow and Keras.
* To integrate SpeechRecognition for voice input and pyttsx3 for voice-based output, enabling two-way vocal communication.
* To ensure fallback mechanisms using text input in case of microphone or speech recognition failure.
* To test the chatbot across various queries and validate that appropriate intents and responses are matched accurately.
* To create a fully integrated Python-based application that handles training, intent classification, and user interaction in a single executable script.
* To enhance accessibility by supporting hands-free communication and potentially extend the chatbot to Tamil language support or web/mobile platforms.
* To demonstrate the practical use of AI and NLP in real-world academic environments for improving user experience and reducing repetitive administrative tasks.

**II LITERATURE SURVEY**

**2.1 INTRODUCTION:**

The literature survey provides a foundation for understanding the existing landscape of chatbot technologies. Various academic and industrial works have demonstrated the effectiveness of machine learning and NLP in answering domain-specific questions. In particular, college information chatbots have been developed to guide users with queries regarding courses, departments, admissions, and facilities. These systems are often trained using a dataset of intents and rely on models like neural networks for classification. However, the majority of these systems depend exclusively on text input/output and **do not explore voice interaction.**

**2.1.1 An Interactive Chatbot for College Enquiry**

**Authors** : Walaa H. Elashmawi

**Publication :** Journal of Computing and Communicationresearchgate.net, 2024

This college inquiry system can be used in any university or for any educational domain as long as it contains a dataset related to that university, either a JSON file or a CSV file,

The dataset contains many Tags and response samples, which are used to train the chatbot. The JSON file was chosen because it is well-organized and expressed in plain language that humans can readily comprehend. They began by extracting data from the university website after writing a large amount of data to maintain our current data. This step will improve chatbot's ability to answer users' inquiries quickly.

The dataset can include as many tags as necessary, containing several patterns to be compared to the user's input. To test the efficiency of the proposed chatbot system, they have experimented on Misr International University (MIU) as a case study. The team members collect the dataset with the help of certain departments in the university and the information available on the university's website.

**III EXISTING SYSTEM**

**3.1 SYSTEM MODEL:**

The users expect to have a human-like conversation with the chatbot, therefore the system was implemented using Artificial Intelligence and the sub-branches Keras and TensorFlow . These techniques are used together with Natural Language Processing to create the algorithm that processes our data and the user's input. The following section will discuss the proposed model and the components within the system.

**3.2 LITERATURE CONCLUSION:**

This paper proposed an interactive, user-friendly chatbot system for college inquiry where questions from college students and employees are encouraged to be submitted. It's often impossible to get all of the data into a single interface without the headaches of filling out multiple forms and windows. The college chatbot aims to eliminate this by providing a standardized and user-friendly interface. Students and instructors can use this interface to get answers to their inquiries in a natural language. The chatbot searches quickly and efficiently for answers to their questions and provides them with relevant connections. In the future, other algorithms, such as AIML-based bots, could be utilized. It is possible to conduct searches using voice. Users will be expected to supply voice input and written output from the system. We can apply the chatbot in other domains, such as medicine, forensics, sports, and so on, after it has been effectively implemented in the collegiate domain. It will be beneficial in many fields since we can swiftly obtain crucial information without having to arrange it.

**IV PROPOSED WORK**

**4.1 INTRODUCTION:**

This project builds upon existing research in the field of AI-based conversational agents, particularly those used in academic institutions for assisting students with common queries. The reviewed literature highlights the role of Natural Language Processing (NLP) and machine learning techniques in developing intelligent chatbots. Most of the base works reviewed provide textual interactions between the user and the system, where the user types the query and receives a text-based response. While these systems offer utility, they lack accessibility for users who may prefer or require voice-based interaction. Our project extends the capability of such systems by integrating voice recognition and speech synthesis to offer a fully conversational interface.

**V SYSTEM SPECIFICATION**

### 5.1 Software Requirements

The project uses Python libraries such as Pandas and NumPy for data handling, TensorFlow for machine learning, and NLTK for natural language processing. SpeechRecognition, PyAudio, and pyttsx3 are used for voice interaction, while Streamlit is used for building the web interface.

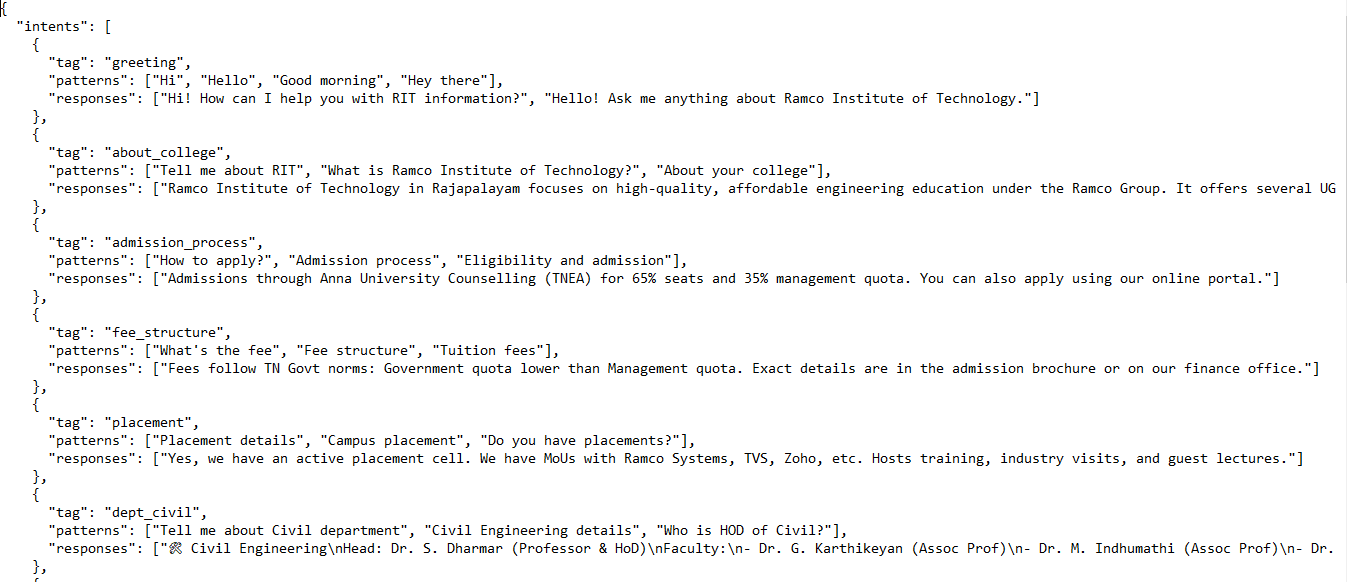
* **Pandas** Used for data manipulation and analysis, especially with structured datasets like CSV files.
* **NumPy** Supports efficient numerical operations on large multi-dimensional arrays and matrices.
* **TensorFlow** An open-source machine learning framework used for building and training neural networks.
* **NLTK** A toolkit for processing and analyzing human language data for NLP tasks like tokenization and classification.
* **SpeechRecognition** Used to convert spoken audio into text using various recognition engines.
* **PyAudio** Provides Python bindings for capturing and playing audio streams through the microphone and speaker.
* **pyttsx3** A text-to-speech library that converts text into spoken voice using offline synthesis engines.
* **Streamlit** A Python library for creating interactive web apps for machine learning and data science projects.

**5.2 Hardware Requirement**

* OS: Windows 10 or higher
* Programming Language: Python 3.10
* IDE: VS Code / Jupyter Notebook
* Libraries: TensorFlow, NLTK, NumPy, pyttsx3, SpeechRecognition, Streamlit, pickle
* Additional Tools: PyAudio (via pipwin), pipwin

**Dataset Description**

The intents.json file contains structured data used to power a college-related chatbot.It includes various **intents**, each representing a category of user queries (e.g., greetings, admissions, departments).Each intent consists of **patterns**, which are sample user messages (like "Hi", "Tell me about RIT"). It also contains **responses**, which are predefined answers the chatbot will reply with. The chatbot provides details about Ramco Institute of Technology (RIT) in Rajapalayam.It covers key topics like admission process, fee structure, placement details, and contact info.Department-specific information is included for Civil, Mechanical, ECE, CSE, IT, and AI & DS.Faculty lists and HOD names are mentioned under relevant departments.There are also intents for student achievements and farewell responses like "Bye".This file is useful for training an NLP model to respond to user queries in a Q&A format.



*(****Figno:1*** *Dataset Model)*

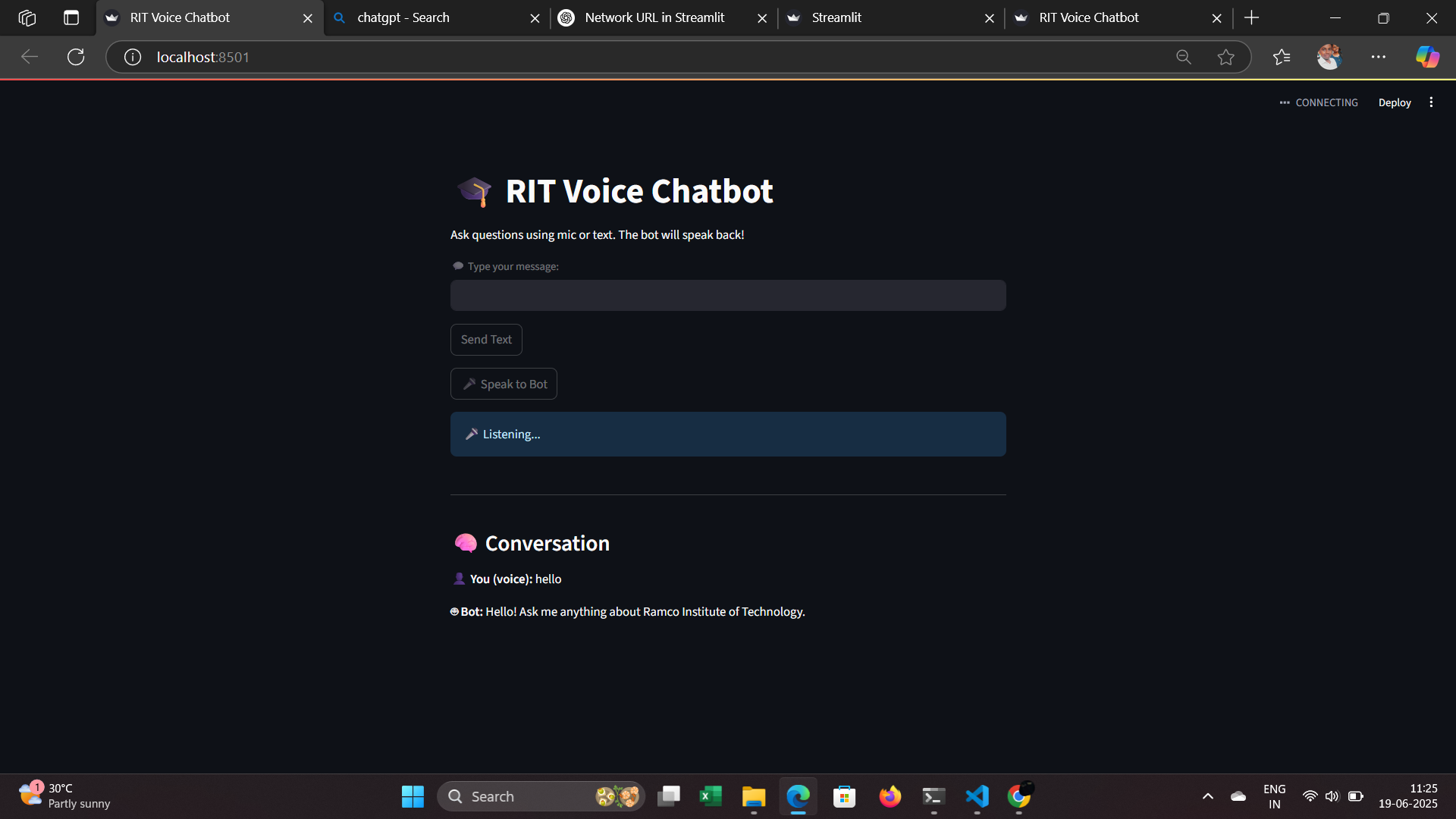
**VI IMPLEMENTATION**

The proposed voice-enabled chatbot follows a modular design. It includes:

* A speech recognition module using SpeechRecognition and PyAudio to convert voice to text.
* An NLP module using NLTK for tokenization and lemmatization.
* A machine learning module using TensorFlow for training a Sequential neural network on intent classification.
* A speech synthesis module using pyttsx3 to convert text responses into speech.
* A Streamlit front-end to provide a user-friendly browser interface.

**4.3 Proposed Methodology**

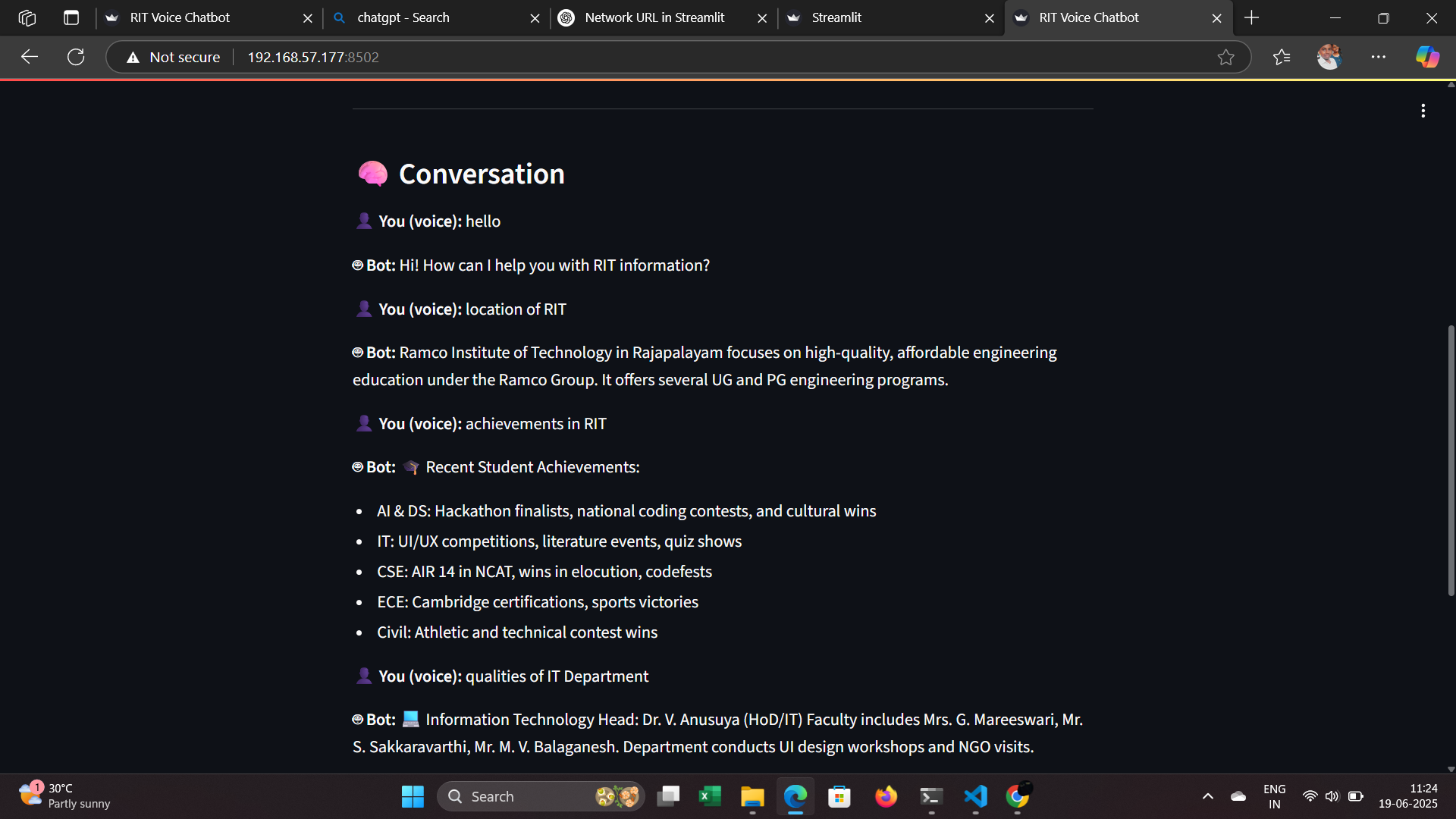
1. Collect and preprocess a dataset of intents using intents.json.
2. Tokenize, lemmatize, and convert patterns to numerical format using Bag-of-Words.
3. Train a neural network on these patterns and associated tags.
4. Save the trained model along with classes and words using Pickle.
5. Create a web-based chatbot using Streamlit that:
   * Accepts voice or typed input
   * Predicts the intent using the trained model
   * Responds with both text and speech using pyttsx3



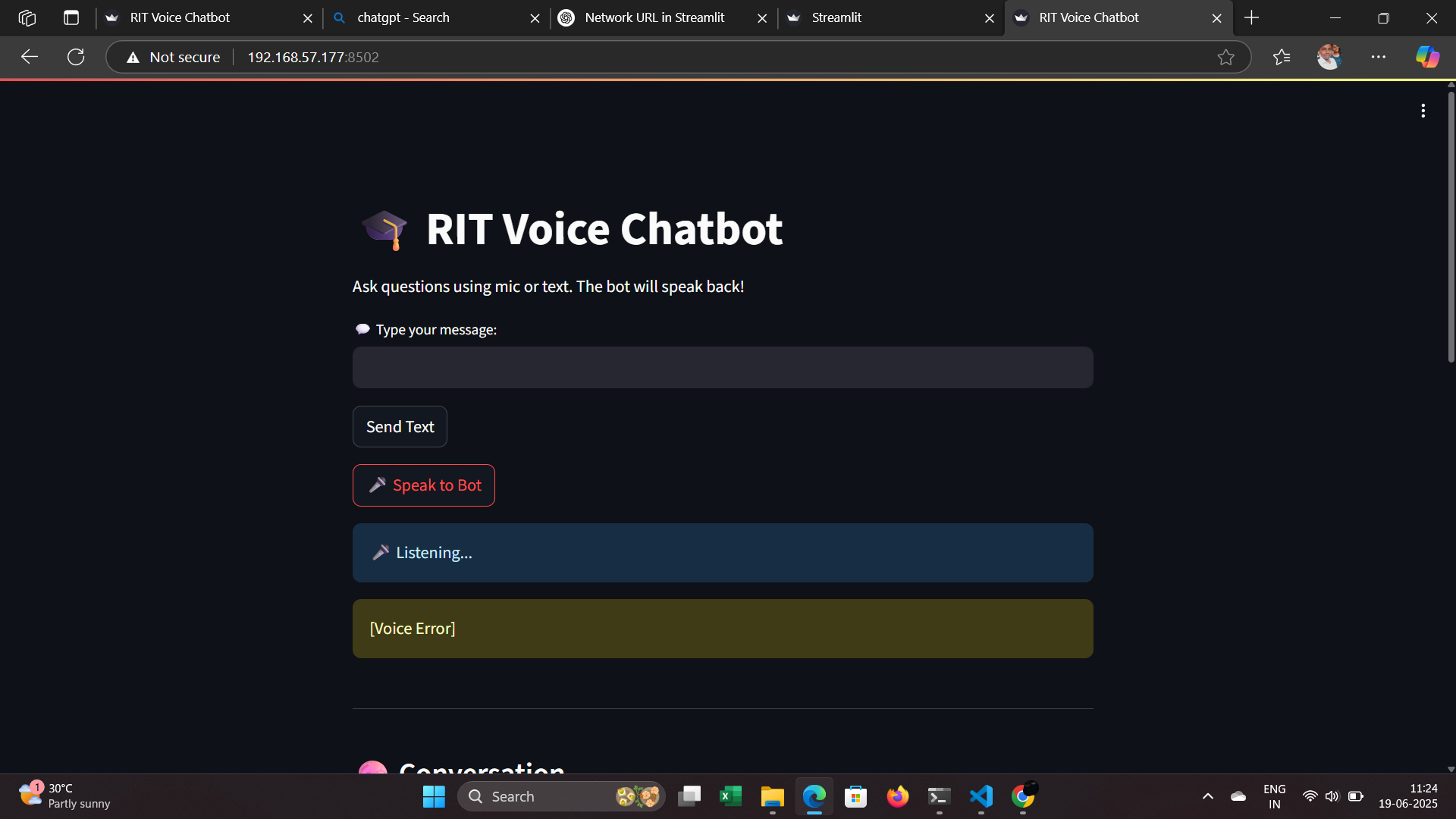
*(****Figno:2*** *layout of the chatbot(Browser))*

**VI IMPLEMENTATION RESULTS**

The system was implemented by training a model using TensorFlow and deploying it using Streamlit. The chatbot accurately predicted intents such as course information, department queries, faculty info, and general details about the college. Voice input was successfully captured and converted using SpeechRecognition, and text-to-speech responses were generated using pyttsx3. The app interface allowed users to switch between voice and text input, offering an accessible and intuitive experience. The final model was stored as chatbot\_model.h5 and integrated with the voice-based interface.

****

*(****Figno:3*** *Conversation of bot and user)*

****

*(****Figno:4*** *layout of the chatbot(Network URL))*

**VIII CONCLUSION AND FUTURE SCOPE**

**CONCLUSION**

The proposed RIT Chatbot project successfully demonstrates the integration of Natural Language Processing, Machine Learning, and Voice Interaction technologies to create an intelligent and interactive conversational system for academic environments. Unlike traditional text-only chatbots, this system supports both text and voice input/output, thereby improving accessibility, user experience, and engagement. The chatbot is capable of answering a wide range of user queries related to the institution—such as department details, course offerings, faculty, placement opportunities, and general contact information. With a trained neural network model deployed via Streamlit, users can conveniently interact through a modern browser interface. The project achieves its goal of providing a smart virtual assistant for students and stakeholders of Ramco Institute of Technology.

**FUTURE ENHANCEMENTS**

**Multilingual Support:**

* Enable the chatbot to understand and respond in Tamil or other regional languages for better inclusivity.

**Dynamic Data Integration:**

* Link the chatbot to real-time databases or college websites to fetch up-to-date information on admissions, fees, and faculty.

**Mobile Application:**

* Convert the chatbot into a mobile app using React Native or Flutter for better portability and offline usage.

**Visual Enhancements:**

* Integrate images, PDFs, or links in responses for documents like brochures, timetables, and circulars.

**Deployment on Cloud:**

* Host the chatbot on Streamlit Cloud, Heroku, or AWS to make it accessible to all students 24/7 via a public URL

**APPENDIX – I CODING**

# RIT Chatbot with Voice Input and Speech Output (All-in-One Script)

import json

import random

import numpy as np

import nltk

import pickle

import os

import speech\_recognition as sr

import pyttsx3

from nltk.stem import WordNetLemmatizer

from nltk.tokenize import TreebankWordTokenizer

from tensorflow.keras.models import Sequential, load\_model

from tensorflow.keras.layers import Dense, Dropout

from tensorflow.keras.optimizers import SGD

nltk.download('wordnet')

lemmatizer = WordNetLemmatizer()

tokenizer = TreebankWordTokenizer()

recognizer = sr.Recognizer()

engine = pyttsx3.init()

# --- Load and preprocess data ---

print("📥 Loading intents...")

with open("intents.json", encoding="utf-8") as file:

intents = json.load(file)

words = []

classes = []

documents = []

ignore\_letters = ['?', '!', '.', ',']

for intent in intents['intents']:

for pattern in intent['patterns']:

word\_list = tokenizer.tokenize(pattern)

words.extend(word\_list)

documents.append((word\_list, intent['tag']))

if intent['tag'] not in classes:

classes.append(intent['tag'])

words = [lemmatizer.lemmatize(w.lower()) for w in words if w not in ignore\_letters]

words = sorted(set(words))

classes = sorted(set(classes))

pickle.dump(words, open("words.pkl", "wb"))

pickle.dump(classes, open("classes.pkl", "wb"))

training = []

output\_empty = [0] \* len(classes)

for doc in documents:

bag = []

pattern\_words = [lemmatizer.lemmatize(word.lower()) for word in doc[0]]

for w in words:

bag.append(1 if w in pattern\_words else 0)

output\_row = list(output\_empty)

output\_row[classes.index(doc[1])] = 1

training.append([bag, output\_row])

random.shuffle(training)

training = np.array(training, dtype=object)

train\_x = np.array(list(training[:, 0]))

train\_y = np.array(list(training[:, 1]))

# --- Train model ---

print("🔧 Training model...")

model = Sequential()

model.add(Dense(128, input\_shape=(len(train\_x[0]),), activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(64, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(len(train\_y[0]), activation='softmax'))

sgd = SGD(learning\_rate=0.01, decay=1e-6, momentum=0.9, nesterov=True)

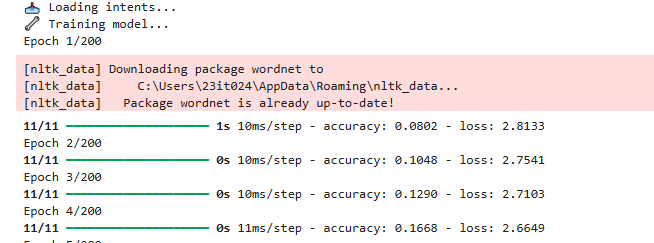
model.compile(loss='categorical\_crossentropy', optimizer=sgd, metrics=['accuracy'])

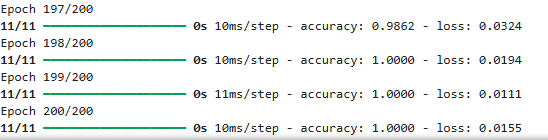
model.fit(train\_x, train\_y, epochs=200, batch\_size=5, verbose=1)

model.save("chatbot\_model.h5")

print("✅ Model trained and saved as chatbot\_model.h5")

OUTPUT:





*(****Figno:5*** *Training the model with datset)*

# --- Chatbot functions ---

def clean\_up\_sentence(sentence):

sentence\_words = tokenizer.tokenize(sentence)

sentence\_words = [lemmatizer.lemmatize(word.lower()) for word in sentence\_words]

return sentence\_words

def bow(sentence, words):

sentence\_words = clean\_up\_sentence(sentence)

bag = [0] \* len(words)

for s in sentence\_words:

for i, w in enumerate(words):

if w == s:

bag[i] = 1

return np.array(bag)

def predict\_class(sentence):

p = bow(sentence, words)

res = model.predict(np.array([p]))[0]

ERROR\_THRESHOLD = 0.25

results = [[i, r] for i, r in enumerate(res) if r > ERROR\_THRESHOLD]

results.sort(key=lambda x: x[1], reverse=True)

return [{'intent': classes[r[0]], 'probability': str(r[1])} for r in results]

def get\_response(intents\_list, intents\_json):

if intents\_list:

tag = intents\_list[0]['intent']

for intent in intents\_json['intents']:

if intent['tag'] == tag:

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

return "I'm sorry, I didn't understand."

def speak(text):

print("Bot:", text)

engine.say(text)

engine.runAndWait()

def listen():

try:

with sr.Microphone() as source:

print("🎤 Listening...")

recognizer.adjust\_for\_ambient\_noise(source, duration=0.5)

audio = recognizer.listen(source)

query = recognizer.recognize\_google(audio)

print("You said:", query)

return query

except Exception as e:

print("[Voice Failed]", e)

return input("Fallback - Type your message: ")

# --- Run chatbot with voice input/output ---

print("\n🎓 RIT Voice Chatbot is ready! Say 'quit' to exit.")

speak("Welcome to Ramco Institute of Technology. How can I help you today?")

while True:

message = listen()

if message.lower() == "quit":

speak("Goodbye! Have a great day!")

break

intents\_predicted = predict\_class(message)

response = get\_response(intents\_predicted, intents)

speak(response)

**APPENDIX – II REFERENCES**

**Authors** : Walaa H. Elashmawi

**Publication :** Journal of Computing and Communicationresearchgate.net, 2024

<https://www.researchgate.net/publication/367524393_An_Interactive_Chatbot_for_College_Enquiry>

**COLLEGE DATASET:**

[ritrjpm.ac.in](http://ritrjpm.ac.in)

**GITHUB REPOSITORY LINK:**