

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belgaum-590 014, Karnataka.



A

Internship Report

On

“Friday The Virtual Assistant”

Submitted in the partial fulfilment of the requirements for the award of the Degree of

BACHELOR OF ENGINEERING IN INFORMATION SCIENCE AND ENGINEERING

Submitted by

**Ravichandra G
(1EW18IS404)**

Under the Guidance of

**Mrs. Padmavathi S
Asst Prof, Dept. of ISE, EWIT**



DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

EAST WEST INSTITUTE OF TECHNOLOGY

BANGALORE - 560 091

2022-2023

EAST WEST INSTITUTE OF TECHNOLOGY

Sy. No.63,Off. Magadi Road, Vishwaneedam Post, Bangalore - 560 091

(Affiliated To Visvesvaraya Technological University, Belgaum)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the mini project work entitled “**Friday The Virtual Assistant**” presented by **Ravichandra G(1EW18IS404)**, bonafied students of **EAST WEST INSTITUTE OF TECHNOLOGY**, Bangalore in partial fulfilment for the award of **Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University**, Belgaum during the year **2022-2023**. It is certified that all corrections/suggestions indicated have been incorporated in the report. The mini project has been approved as it satisfies the academic requirements in respect of mini project prescribed for the said degree.

Signature of Guide

Mrs. Padmavathi S

Asst. Prof, Dept. of ISE

EWIT, Bangalore

Signature of HOD

Dr Suresh M B

Prof & Head, Dept. of ISE

EWIT, Bangalore

Signature of Principal

Dr. K Channakeshavalu

Principal

EWIT, Bangalore

Name of the Examiners

External Viva

Signature with date

1. _____

1.

1. _____

2. _____

2.

2. _____

EAST WEST INSTITUTE OF TECHNOLOGY

Sy. No.63,Off. Magadi Road, Vishwaneedam Post, Bangalore - 560 091(Affiliated
To Visvesvaraya Technological University, Belgaum)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING



DECLARATION

I, **Ravichandra G**, student of Seventh semester BE, in the Department of Information Technology, **East West Institute of Technology**, Bangalore hereby declare that the Internship entitled “**Friday The Virtual Assistant**” has been carried out by me and submitted in partial fulfilment of the course requirements for the award of degree in **Bachelor of Engineering** in **Information Science and Engineering** discipline of **Visvesvaraya Technology University**, Belgaum during the academic year 2022-2023. Further, the matter embodied in the dissertation has not been submitted previously by anybody for the award of any degree or diploma to any other university.

PLACE: Bangalore
DATE:

STUDENT NAME: RAVICHANDRA G
USN: 1EW18IS404

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped me in carrying out this project work. We would like to take this opportunity to thank them all.

First and foremost we would like to thank **Dr. K Channakeshavalu**, Principal, EWIT, Bangalore, for his moral support towards completing our project work.

We would like to thank, **Dr. Suresh M B**, Professor and Head of Department of ISE, EWIT, Bangalore, for his valuable suggestions and expert advice.

We deeply express our sincere gratitude to our guide, **Mrs. Padmavathi S**, Assistant Professor of Department of ISE, EWIT, Bangalore for her able guidance throughout the project work and guiding me to organise the report in a systematic manner.

We thank my Parents, and all the Faculty members of Department of Information science & Engineering for their constant support and encouragement.

Last, but not the least, we would like to thank our peers and friends who provided us with valuable suggestions to improve our mini project.

Ravichandra G
1EW18IS404

ABSTRACT

In this modern era, day to day life become smarter & interlinked with technology. We already know some voice assistant like google, Siri etc. Now in our voice assistant system named Friday, it can act as your smart friend, get the news headlines, tell a joke, search the internet and get answers to questions. This project works on speech input & give output through speech & text on screen.

This assistant attaches with the world wide web to provide result that the user requires. Natural language processing algorithm helps machines to engage in communication using natural human language in many forms.

III

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ACKNOWLEDGEMENT	i
	ABSTRACT	ii
	CONTENTS	iii
Chapter 1	INTRODUCTION	1
	1.1 Introduction to Voice Assistants	
Chapter 2	COMPANY PROFILE	3
	2.1 Company Profile	
	2.2 Mission	
	2.3 Vision	
Chapter 3	SURVEY OF TECHNOLOGY	4
	3.1 Python	
	3.2 Pyttsx	
	3.3 Speech Recognition	
	3.4 Requirement and Analysis	
Chapter 4	PROBLEM STATEMENT	5
	4.1 Problem Definition	
	4.2 Motivation	
Chapter 5	SYSTEM REQUIREMENTS	6
	5.1 Hardware	
	5.2 Software	
Chapter 6	FEASIBILITY STUDY	7
	6.1 Technical Feasibility	
	6.2 Operational Feasibility	
	6.3 Economic Feasibility	
	6.4 Organisational Feasibility	
	6.5 Cultural Feasibility	

Chapter 7	DESIGN METHODOLOGY	8
	7.1 System Architecture	
	7.2 Working	
	7.3 Methodology	
	7.4 Knowledge abstraction	
	7.5 Data manipulation	
	7.6 Data augmentation	
	7.7 Response generation	
	7.8 Flow of conversation and decision trees	
	7.9 Linear flow	
	7.10 Non-linear flow	
	7.11 Underfitting vs. Overfitting	
	7.12 Underfitting due to lack of entities	
	7.13 Underfitting due to the excess of entities	
	7.14 Overfitting due to unsimplified intent	
	7.15 Overfitting due to separable entities	
Chapter 8	SCOPE OF SYSTEM	15
	8.1 Advantages	
	8.2 Limitations	
	8.3 Applications	
Chapter 9	SNAPSHOTS	16
	CONCLUSION	
	REFERENCES	

CHAPTER 1

INTRODUCTION

An intelligent virtual assistant (IVA) or intelligent personal assistant (IPA) is a software agent that can perform tasks or services for an individual based on commands or questions. The term "chatbot" is sometimes used to refer to virtual assistants generally or specifically accessed by online chat.

Virtual assistant is software program that helps you ease your day-to-day tasks, such as showing weather forecasting, listing reminders, creating to-do-list etc. They can take commands as voice. Our voice assistant is designed to be used efficiently for all users. This personal assistant software improves users productivity by managing day to day tasks & providing information from online sources to users. Voice searches have dominated over text search. Web search can conduct via mobile devices have only just overtaken those carried out using a computer and the analysis are already predicting that 50% of searches will be via voice by 2023. Virtual assistant are turning out to be smarter than ever. Permit your brainy subordinate to make email work for you, notice determined, choice out significant statistics, systematise procedure, and distribute modified reply.

Virtual assistants may be integrated into many types of platforms like Amazon Alexa.

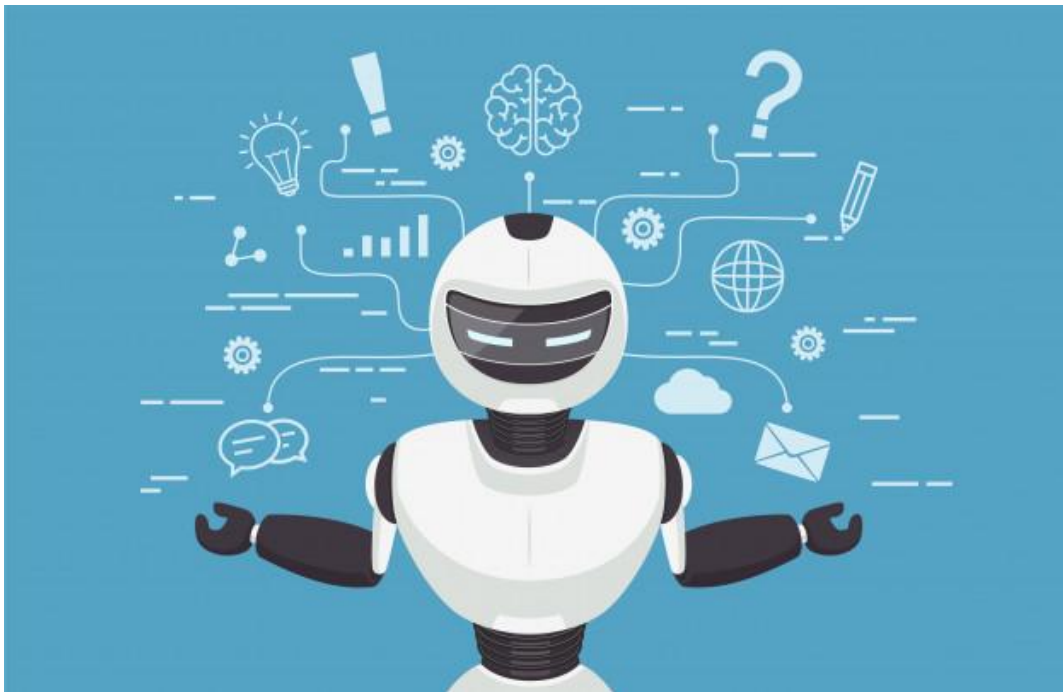


Fig 1.1: Visualisation of virtual assistant

And also into devices like smart speakers such as Amazon Echo, Google Nest and Apple HomePod.

- In instant messaging applications on both smartphones and via the Web, e.g. Facebook's (virtual assistant) on both Facebook and Facebook Messenger apps or via the Web
- Built into a mobile operating system (OS), as are Apple's Siri on iOS devices and BlackBerry Assistant on BlackBerry 10 devices, or into a desktop OS such as Cortana on Microsoft Windows OS
- Built into a smartphone independent of the OS, as is Bixby on the Samsung Galaxy S8 and Note 8.
- Within instant messaging platforms, assistants from specific organizations, such as Aeromexico's Aerobot on Facebook Messenger or Wechat Secretary on WeChat
- Within mobile apps from specific companies and other organizations, such as Dom from Domino's Pizza.
- In appliances, cars and wearable technology.

CHAPTER 2

COMPANY PROFILE

2.1 Company Profile

The Tech Fortune group was originally inceptioned in 2013 to focus only on EPC assignment of Infrastructure and Real Estate Projects. Later the Company shifted its focus towards software development, Training ,BPO, Sourcing, Food Business, Health Care and Strategic Advisory Services. The Tech Fortune group Tech Fortune was born in 2013 with an objective to create a landmark initiative by a group of highly qualified technology oriented professionals in the software domain. A Software development firm head quartered in Vijapur and operating for 5 years in with 3 offices across Karnataka. Since its inception in 2013, Tech fortune group has grown rapidly with the help of its valued Customers, Professionals & Business Associates who have been continuously contributing and monitoring the Company's business activities in the operations of Project Management, Education Consultancy, QMS and Six Sigma implementation and many other domain of expertise.

Tech Fortune Technologies, is an emerging technology organisation in the fields of business process outsourcing, software development, end-to-end ERP solutions, Artificial Intelligence, Blockchain technology with a focus on providing customised solutions to the various business needs of a diverse global clientele.

2.2 Mission

Being slow and steady, our mission is to gain the confidence of our clients and by dint of our integrity, innovation and dynamism, deliver their requirements on time with full quality thus bridging the gap between demand and delivery.

2.3 Vision

With an unyielding focus on integrity and backed by strong founders and management team, sourcing wants to make a mark in the field of IT services by applying innovation to simplify complex business processes and add value to clients business.

CHAPTER 3

SURVEY OF TECHNOLOGY

3.1 Python

Python is an OOPs (Object Oriented Programming) based, high level, interpreted programming language. It is a robust, highly useful language focused on rapid application development (RAD). Python helps in easy writing and execution of codes. Python can implement the same logic with as much as 1/5 code as compared to other OOPs languages.

Python provides a huge list of benefits to all. The usage of Python is such that it cannot be limited to only one activity. Its growing popularity has allowed it to enter into some of the most popular and complex processes like Artificial Intelligence (AI), Machine Learning (ML), natural language processing, data science etc.

3.2 Pytttx

Pytttx stands for Python Text to Speech. It is a cross-platform Python Wrapper for text to speech synthesis. It is a Python Supporting common text-to-speech engines on MacOS X, Windows and Linux. It works for both Python2.x and 3.x versions. Its main advantages is that it works offline.

3.3 Speech Recognition

This is a library for performing speech recognition, which support for several engines and API's, online and offline it supports API's like Google cloud speech API, IBM speech to text, Microsoft's Bing voice recognition etc.

3.4 Requirement and Analysis

System analysis is about complete understanding of our existing system and finding where the existing system fails. The solution is determined to resolve issues in the proposed system. It defines the system. The system is divided into smaller parts. Their functions and inter-relation of these models are studied in system analysis.

CHAPTER 4

PROBLEM STATEMENT

4.1 Problem Definition

User needs to manually accomplish numerous sets of applications to comprehensive single task. We already have multiple voice assistants but there is problem in voice reorganisation. There is need of a voice assistant that can understand English in Indian accent & work on it. Voice assistant should be able to model complex tasks, it needs to be tested for finding optimise path when a task has multiple subtasks & each sub task can have its own sub tasks.

4.2 Motivation

A voice assistant makes our life easier, saves out time & Friday Voice assistant helps user with functions that are the backbone of our day. Someone who talks with you when you are alone or feel lonely someone who manager your daily routines take care of your mental & physical health. Main motivation is the voice assistant helps adults & disables people to perform their day-to-day tasks.

Virtual assistants must provide a wide variety of services.

These include:

- Providing information such as weather, facts from e.g., Wikipedia etc.
- Set an alarm or make to-do lists and shopping lists.
- Remind you of birthdays and meetings.
- Play music from streaming services such as Spotify and Apple Music .
- Play videos , TV shows or movies on televisions, streaming from e.g., Netflix or Prime.
- Book tickets for shows, travel and movies.

CHAPTER 5

SYSTEM REQUIREMENTS

System requirements are the configuration that a system must have in order for a hardware or software application to run smoothly and efficiently. Failure to meet these requirements can result in installation problems or performance problems. The former may prevent a device or application from getting installed, whereas the latter may cause a product to malfunction or perform below expectation or even to hang or crash. System requirements are also known as minimum system requirements.

5.1 Hardware Requirements

- No less than 8 GB RAM.
- Minimum of Intel Core i5 processor or equivalent.
- Solid-state drive (SSD) as the main storage.
- Microphone
- Speaker
- Other specs relevant to your niche/tasks.

5.2 Software Requirements

- IDE such as VS Code or Pycharm
- Python 2.7 or 3.2

CHAPTER 6

FEASIBILITY STUDY

A feasibility study can help you determine whether or not you should proceed with your project. It is essential to evaluate cost and benefit of the proposed system. Four types of feasibility study or taken into consideration.

6.1 Technical feasibility

It includes finding out technologies for the project, both hardware and software. For virtual assistant, user must a microphone to convey their message and a speaker to listen when system speaks. These are very cheap nowadays and everyone generally possess them. Besides that, system needs internet connection. While using it, make sure, you have a steady internet connection. It is also not an issue in this era where almost every home or office has Wi-Fi.

6.2 Operational feasibility

It is the ease and simplicity of operation of proposed system. System does not require any special skill set for users to operate it. In fact, it is designed to be used by almost by everyone. Kids who still don't know how to write can read out problems for system and get answers.

6.3 Economic feasibility

Here, we find the total cost and benefit of the proposed system over current system. For this project, the main cost is documentation cost. User also would have to pay for microphone and speakers. Again, they're cheap and available. As far as maintenance is concerned, it won't cost too much.

6.4 Organisational feasibility

This shows the management and organisational structure of the project. This project is not built by a team. The management tasks around is to be carried out by a single person that won't create any management issues and will increase the feasibility of the project.

CHAPTER 7

DESIGN METHODOLOGY

Based on past work and assessment, the project is accomplished with understanding of an intelligent assistant capable of taking user command and analyse it and respond to the user by using voice media. Python libraries and speech recognition APIs are used to integrate the personal voice assistant python speech to text model is used.

7.1 System Architecture

The system uses speech recognition module named python-text-to-speech(pytttsx3) module. It is used to adapt user input to text. The pytttsx3 component of python has constructed in speech-to-text and text-to-speech machine. It makes system more interactive with the user.

API calls: An API is a package interface that allows dual program to interconnect with one another. API is the connection that directs the command to the provider and then returns the reply of that command.

Python Backend: the python backend receives the users query in the form of text and determine if the query is an API call or data extraction then result is sent back to python backend, at any point of time, this system is able to react with an appropriate response.

Data Extraction: It is the procedure of mining systematised statistics from the chaotic machine-readable documents. In our proposed system, we have extracted the relevant data that has been asked by user.

Text-To-Speech Module: this component is tremendously useful to persons who have trouble in interpretation. Text-to-speech engine can be implemented in a variety of languages, dialects, and advanced vocabularies by using third party libraries.

7.2 Working

Python - we are using python 3.7 and 3.9 version is not working well with TensorFlow module currently ,3.7 is more stable version of python. TensorFlow - This is an endwise open-source stage for machine learning .It is comprehensive, stretchy environment of tools, lending library and public properties that lets investigate drive the state of the art in machine learning and developers informal figure and organise machine learning motorised application

7.3 Methodology

We divide methodology in two parts: knowledge abstraction and response generation. Knowledge abstraction has to do with the analysis of course content (which we will call data). On the other hand, response generation depends on the characteristics of the data generated in the process of knowledge abstraction, and also, it depends on the features of the tools available for Dialogflow and its various integrations.

7.4 Knowledge abstraction

Knowledge abstraction involves three phases: gathering, manipulation and augmentation. These phases are mostly independent from the content of the chatbot. Data gathering - The first step is to generate a knowledge base. This step involves finding key concepts of the course and gathering information about them. Developers might want to focus on the syllabus of the course to get a general idea of the main ideas and topics. Ideally, the lecturer hands in a list of topics with its main concepts ordered in a hierarchical structure. Then, one generates a bunch of questions. These questions might come from online records (such as discussion forums, social media interaction with students or messaging applications), homework assignments, and frequently asked questions the teacher is aware of. After this process, developers can then classify this questions in categories according to the topic they relate to. These categories structure the content of the chatbot.

7.5 Data manipulation

The second step is to store this information in a database. This will enable developers to manipulate data. For instance, developers can create two spreadsheets, one with the syllabus of the course ordered by topics and accompanied by the most relevant concepts of each topic, and another spreadsheet with the set of questions found in a discussion forum together with their answers. Then, they can classify each question with its corresponding topic by looking for the keywords (aka the concepts contained in the first spreadsheet) within the question or the answer strings. This consists technically in the implementation of a classification algorithm that labels each question-answer pair. These labels must be given a degree of confidence and then checked by a human to confirm its validity. The database might contain questions with similar answers, this should mean that these questions have the same goal or purpose, so one can group them in categories, which will later evolve into Dialogflow intents. Therefore, developers should associate to each question-answer pair a label indicating the name of the intent to which they belong.

7.6 Data augmentation

Data manipulation can be taken to another level by introducing data augmentation. This can increase the number of training examples available for the natural language processing model within Dialogflow. Using the intents, one can lookup for correlations between questions and answers that belong to the same intent. These correlations can be pictured as keywords (also called entities) in common, and they can be extracted to be later used to produce more training examples. In other words, sometimes answers themselves can be used to produce questions, and these questions should belong to the same intent. Developers can generate another spreadsheet with intents with their corresponding entities. This is very useful since Dialogflow notices entities and the model learns better when certain entities appear only within a single entity

7.7 Response generation

There is a thorough description of how entities and intents work within Dialogflow, but it is precise to elaborate about how the way things are already implemented in this platform can benefit the proposed methodology. Intents are built upon three basic components: contexts, training examples and responses. Contexts serve the purpose of layering the conversation in such a way that, when they are present, they allow only certain intents to be activated. Training examples are sentences, and they are found in two categories: templates and examples. Both of them label these sentences using entities or just the plain intent in which they are contained. Responses are the outputs of intents, so each time a user entry corresponds to an intent, its response is triggered. They appear in several ways, the most common is simple text, but Dialogflow allows direct integration with apps that fit content into cards, carousels, tables, lists and other structures of content. Entities, as previously said, are keywords that may or may not be classified according to its function. There are entities that correspond to geographical locations, numbers, or dates. These are popular in the context of customer service, for example taking the order of a client. But for educational purposes not so much, so an option is to make an intent for each concept, this way we can group data resulting from students interaction with the bot not only using specific questions, but also using this broader constructs. Besides the advantage in the analytics, there is also a benefit for intent recognition, since every time an entity is detected the number of possible intents that match the query is reduced.

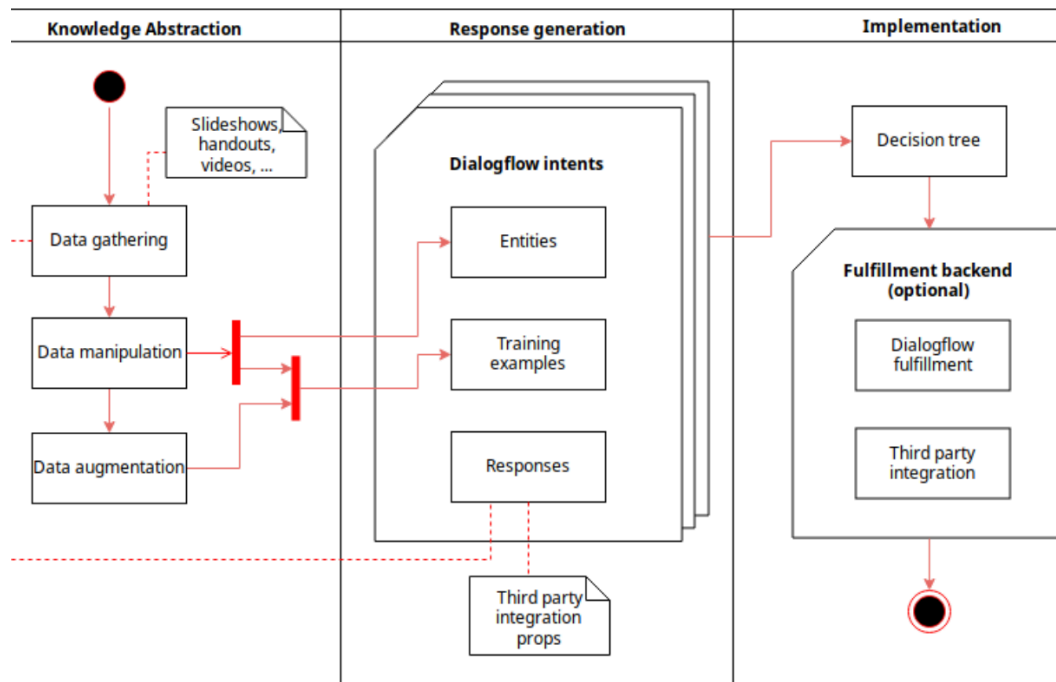


Fig. 7.1 Graphical representation for the sections in which we divide the methodology and how they are related to each other.

It is important to comment on the issues of writing a response. We know the way students watch slide shows is different from how they are supposed to interact with a virtual assistant. This difference introduces the need of content adaptation, the process of finding a media which best transmits the desired message. For instance, when outlining a procedure, we want to show a list of steps instead of a paragraph, or if it is rather a concept best described with a picture or a video, we want to be able to show this content as well as a description. We propose using Actions On Google cards or similar for short factual answers, lists or carousels for procedures, and simple responses with more specific questions.

Also, comes the question of whether the agent is going to say every written word, or is it just going to expect the user to read it. We found it was better to use a mixture of both: let the agent tell the user what it is showing, but not the actual content, because it is otherwise annoying to hear the agent speak for a long time. As an example, if the user asks for the algorithm of BFS, then the agent would say: Here is the algorithm of BFS, and then it show the algorithm as a list of steps.

7.8 Flow of conversation and decision trees

In the next two subsections, we talk about flow of conversation and how to face some problems regarding the machine learning model. These topics attempt to provide useful advice for how to implement the methodology more efficiently. Decision trees are also called conversation trees and are used to implement the structure of a virtual assistant. They are made up from nodes, where each of these contain an answer to a particular query. The task of the developer is to give each node a conditional that is activated when the user's query matches the response contained in the node (with some degree of accuracy). This last step is done with the aid of keywords and natural language recognition. In Dialogflow, nodes are equivalent to intents, figure 7.2 shows the structure of the decision tree we utilised in further detail.

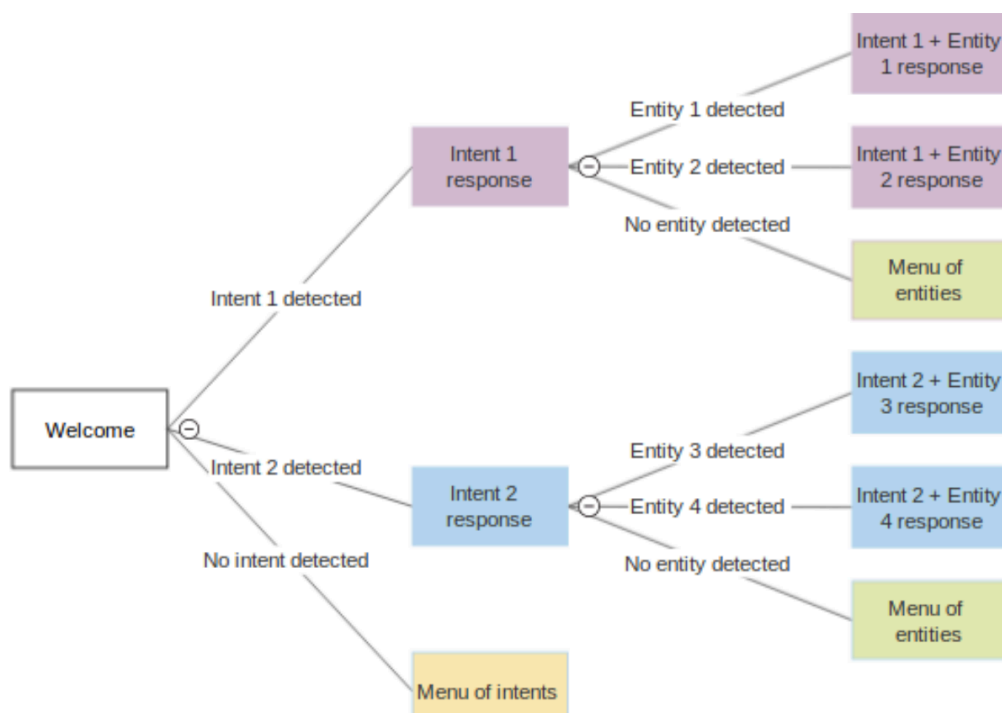


Fig.7.2 Basic structure of a proposed decision tree

The first step towards implementing a decision tree is defining a flow of conversation. The flow of conversation defines the interaction with the user. It refers to the way the agent handles bifurcations in the middle of a conversation, and how it travels through the content. We have found that flows of conversation can be categorised in linear and non-linear.

7.9 Linear flow

It is a flow of conversation that consists in two steps, question and answer. These are what we call FAQ bots and are relatively easy to implement, because the decision tree has only one branch of nodes. However, they are helpful only if an instruction manual is provided. This manual must tell what content the assistant can provide and how is the user expected to interact with it.

7.10 Non-linear flow

This flow of conversation relies heavily on suggestions and allows for exploring the decision tree much more structure deeply. The decision tree is not different at all from the implementation of the linear flow, but the key difference is that it features jumps between nodes. These jumps enable suggestions to directly link to any other node within our decision tree.

7.11 Underfitting v/s overfitting

Every machine learning application has to deal with the bias v/s variance problem. Here we present some general types of problem we have faced at the stage of implementation and testing.

7.12 Underfitting due to the lack of entities

The lack of entities can generate an underfitting problem because the agent loses the chance of identifying every parameter it needs to give a relevant answer. This problem can arise from the process of design, and is particularly problematic when intents are not specific enough.

7.13 Underfitting due to the excess of entities

Here there are so many entities declared that different intents may contain several entities in common, and thus have a strong correlation. This may lower the degree of confidence of the agent with respect to the right response.

7.14 Overfitting due to unsimplified intent

This problem has a similar nature to last one, the difference is that now it does not have to do with entities at all, but with the words used in the training examples. Also, the result is completely different because since we use overly complicated or worded sentences as training examples.

The result is that the agent is trained to recognise very specific patterns of words. The consequence is that the agent is not able to identify this intent unless the sentences have a very specific structure.

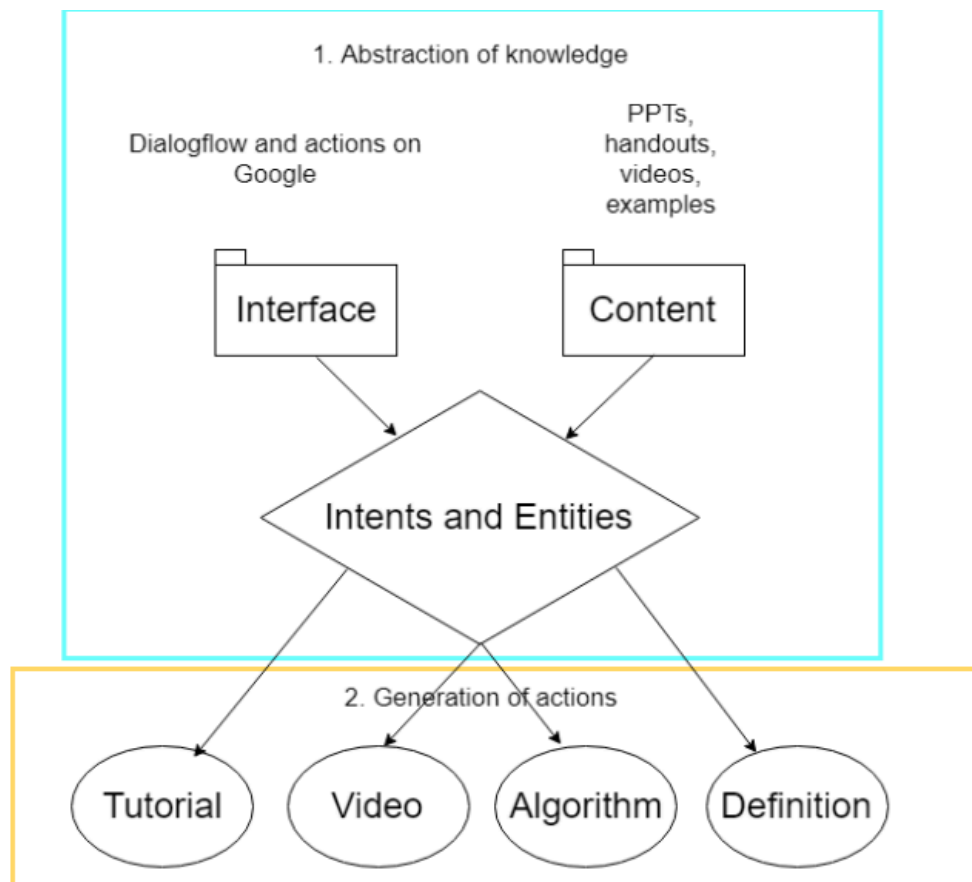


Fig. 7.3 Abstraction of knowledge and generation of actions applied to the case study

7.15 Overfitting due to separable entities

Some entities with multiple words can be separated to avoid this problem, it depends on what kind of entities we are dealing with, because there are cases in which this change is rather harmful. For example, if we want the agent to answer questions like: "what is the difference between Fast Fourier Transform and Fourier Transform?", then we have to specify that FFT is one entity and FT is another.

CHAPTER 8

SCOPE OF SYSTEM

Voice assistant will remain to proposal additional distinct involvement as they get improved at distinguishing amongst speech. however, it's not just developers that need to address the complexity of developing for voice as brand but also need to understand the capabilities of each device and integration and if it makes sense for their specific brand. They will also need to focus on mining a user experience that is consistent within the coming year as complexity becomes more concern. This is because the graphic interface with voice subordinate is absent user basically cannot realise or touch a voice assistant interface.

8.1 Advantages

- A voice assistant makes your life easier.
- It can be an expert in any field.
- A voice assistant allows you to provide more coverage in all kinds of different technologies.
- You have a smart and intelligent friend always with you.

8.2 Limitations

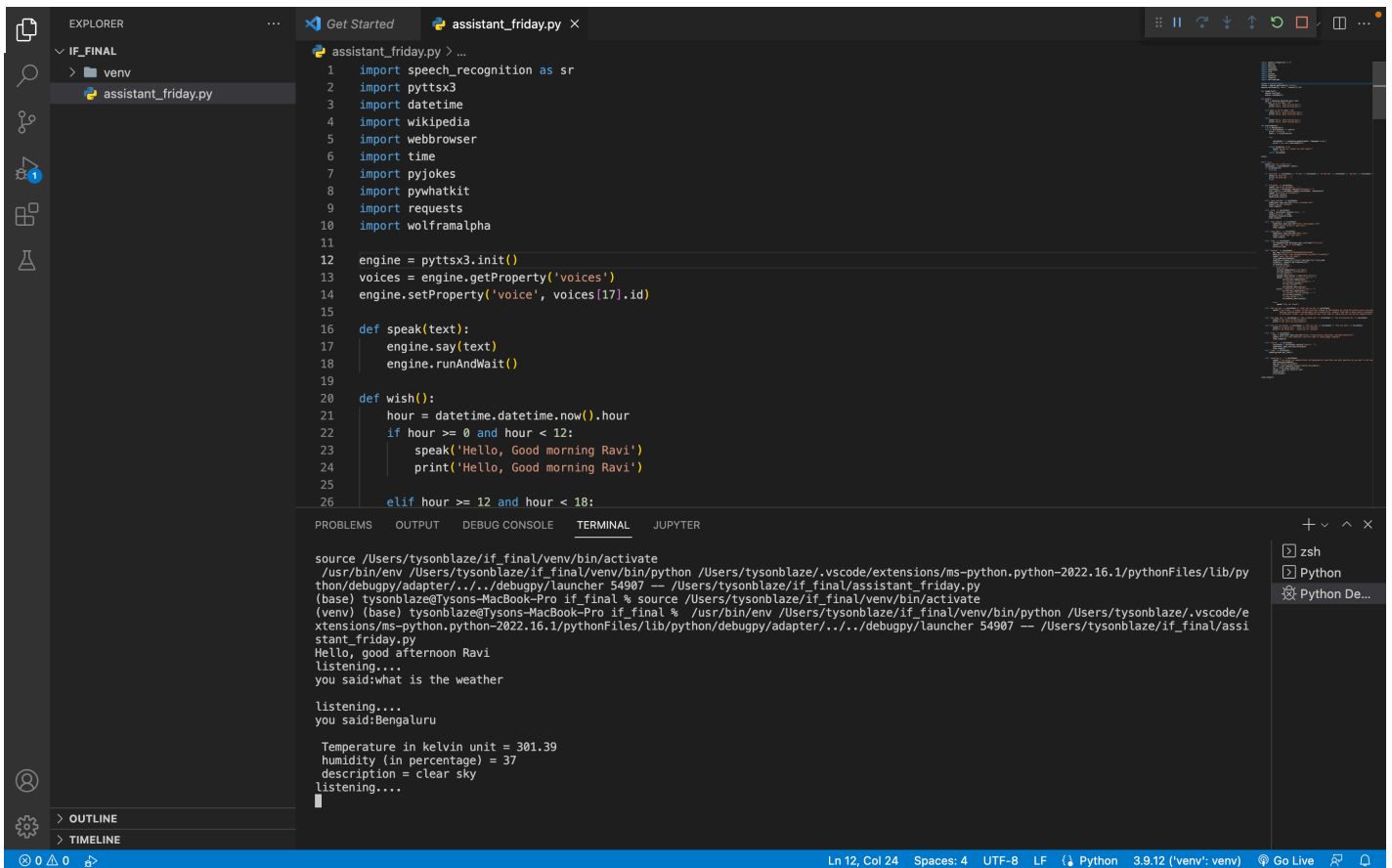
- Requires high speed internet connection.
- Mediocre sound quality
- Accent and speech recognition.
- Background noise interface

8.3 Applications

- Voice tech in healthcare
- Mobile app integration.
- Disabled's helping guide.
- Adults helping hand.

CHAPTER 9

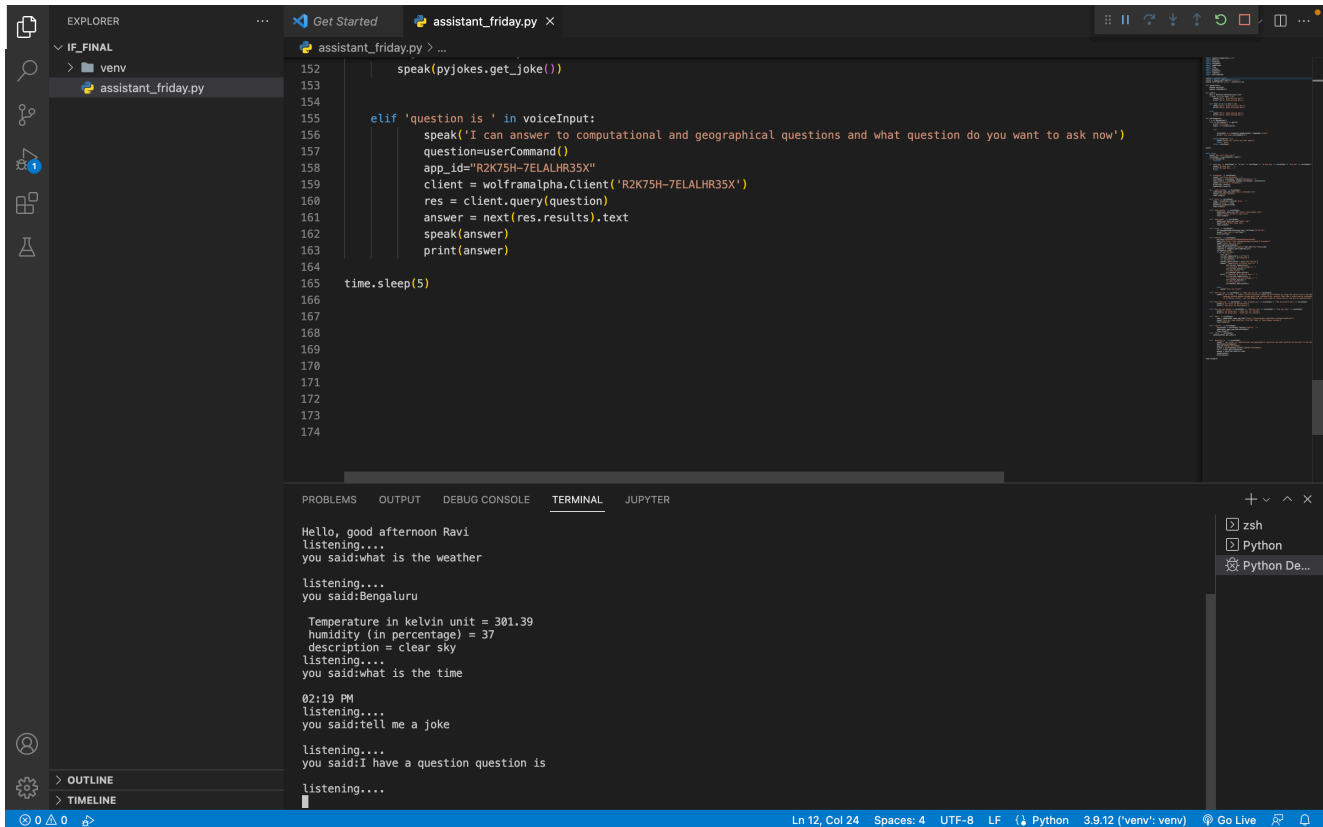
SNAPSHOTS



```
assistant_friday.py > ...
1 import speech_recognition as sr
2 import pyttsx3
3 import datetime
4 import wikipedia
5 import webbrowser
6 import time
7 import pyjokes
8 import pywhatkit
9 import requests
10 import wolframalpha
11
12 engine = pyttsx3.init()
13 voices = engine.getProperty('voices')
14 engine.setProperty('voice', voices[17].id)
15
16 def speak(text):
17     engine.say(text)
18     engine.runAndWait()
19
20 def wish():
21     hour = datetime.datetime.now().hour
22     if hour >= 0 and hour < 12:
23         speak('Hello, Good morning Ravi')
24         print('Hello, Good morning Ravi')
25
26     elif hour >= 12 and hour < 18:
27
source /Users/tysonblaze/if_final/venv/bin/activate
/usr/bin/env /Users/tysonblaze/if_final/venv/bin/python /Users/tysonblaze/.vscode/extensions/ms-python.python-2022.16.1/pythonFiles/lib/py
thon/debugpy/adapter/../../debugpy/launcher 54907 -- /Users/tysonblaze/if_final/assistant_friday.py
(base) tysonblaze@Tysons-MacBook-Pro if_final % source /Users/tysonblaze/if_final/venv/bin/activate
(venv) (base) tysonblaze@Tysons-MacBook-Pro if_final % /usr/bin/env /Users/tysonblaze/if_final/venv/bin/python /Users/tysonblaze/.vscode/e
xtensions/ms-python.python-2022.16.1/pythonFiles/lib/python/debugpy/adapter/../../debugpy/launcher 54907 -- /Users/tysonblaze/if_final/assi
stant_friday.py
Hello, good afternoon Ravi
listening...
you said:what is the weather
listening...
you said:Bengaluru

Temperature in kelvin unit = 301.39
humidity (in percentage) = 37
description = clear sky
listening...
```

Fig: 9.1 Asking the weather



The screenshot displays a Visual Studio Code editor window with a Python file named `assistant_friday.py` open. The file is located within a virtual environment named `venv`. The code in the file includes a function to speak text, a loop to listen for voice input, and a conditional statement to handle different types of questions, including a specific branch for asking for a joke. The terminal window at the bottom shows the execution of the script, with output indicating that the assistant is listening for voice input and responding to various queries, including a request for a joke.

```
152     speak(pyjokes.get_joke())
153
154
155     elif 'question is ' in voiceInput:
156         speak('I can answer to computational and geographical questions and what question do you want to ask now')
157         question=userCommand()
158         app_id="R2K75H-7ELALHR35X"
159         client = wolframalpha.Client('R2K75H-7ELALHR35X')
160         res = client.query(question)
161         answer = next(res.results).text
162         speak(answer)
163         print(answer)
164
165     time.sleep(5)
166
167
168
169
170
171
172
173
174
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Hello, good afternoon Ravi
listening...
you said:what is the weather

listening...
you said:Bengaluru

Temperature in kelvin unit = 301.39
humidity (in percentage) = 37
description = clear sky
listening...
you said:what is the time

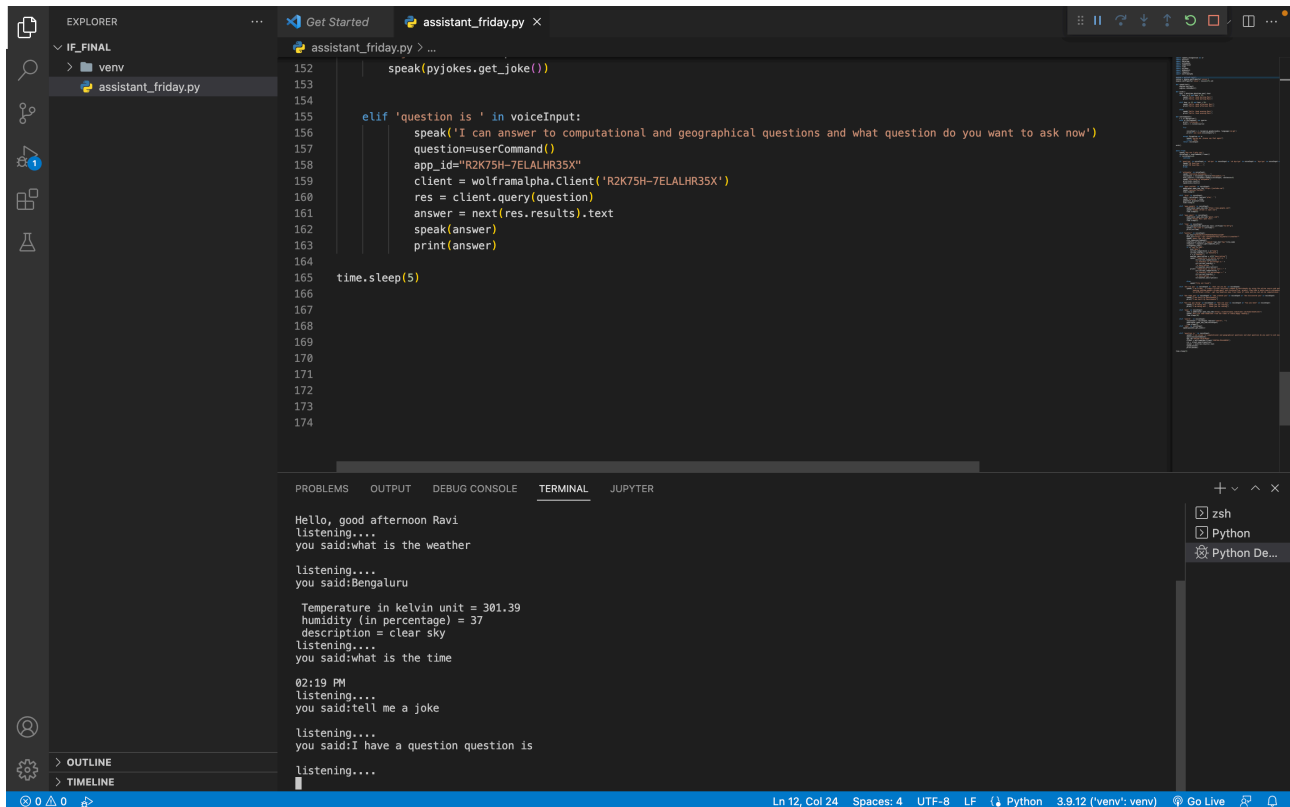
02:19 PM
listening...
you said:tell me a joke

listening...
you said:I have a question question is

listening...

Ln 12, Col 24 Spaces: 4 UTF-8 LF Python 3.9.12 ('venv': venv) Go Live

Fig: 9.2 Asking for a joke

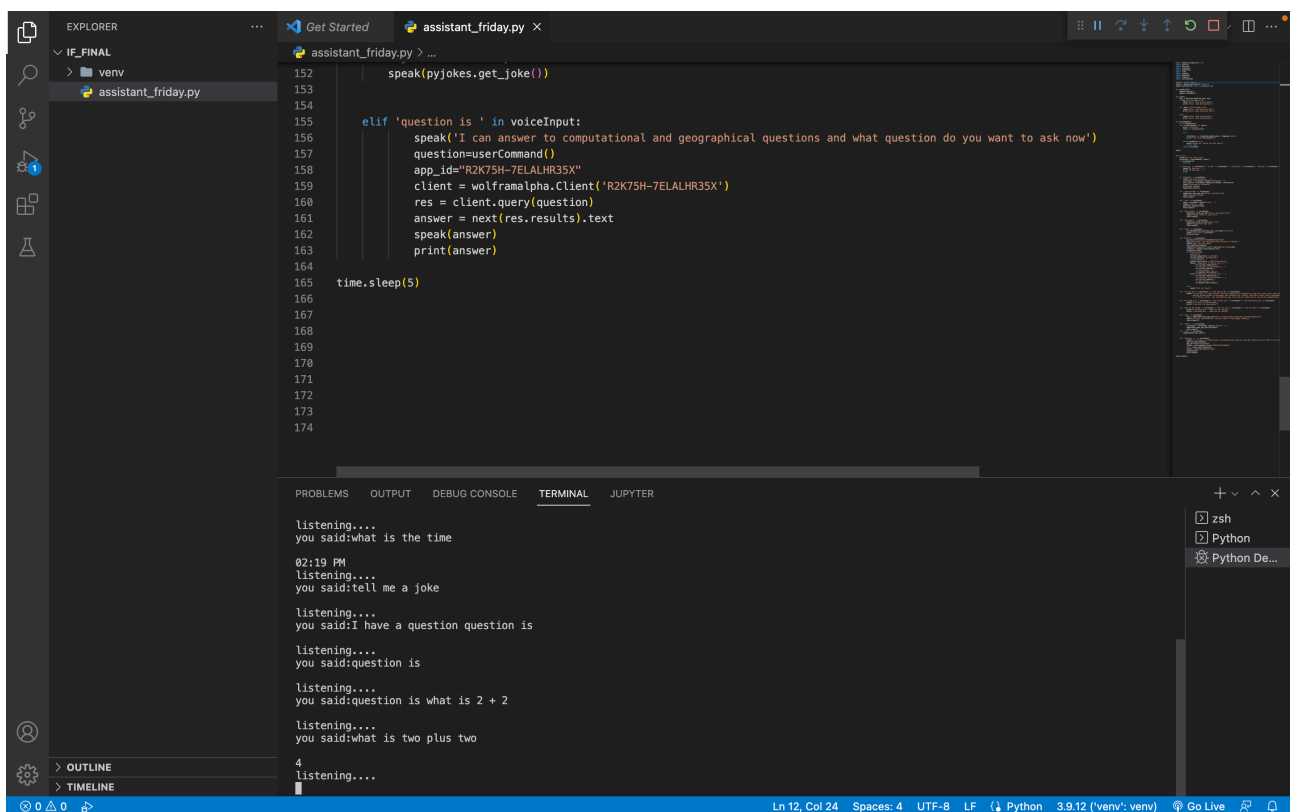


The screenshot displays the Visual Studio Code interface. The Explorer panel on the left shows a project named 'IF_FINAL' with a subdirectory 'venv' containing the file 'assistant_friday.py'. The main editor window shows the code for 'assistant_friday.py', which includes a function to speak a joke and a conditional block for handling voice input. The terminal window at the bottom shows a chat log with the following text:

```
Hello, good afternoon Ravi
listening....
you said:what is the weather
listening....
you said:Bengaluru
Temperature in kelvin unit = 301.39
humidity (in percentage) = 37
description = clear sky
listening....
you said:what is the time
02:19 PM
listening....
you said:tell me a joke
listening....
you said:I have a question question is
listening....
```

The status bar at the bottom indicates the current line and column (Ln 12, Col 24), the number of spaces (4), the encoding (UTF-8), the line feed (LF), the Python version (3.9.12), and the environment (venv).

Fig: 9.3 Asking a question



```
152     speak(pyjokes.get_joke())
153
154
155 elif 'question is ' in voiceInput:
156     speak('I can answer to computational and geographical questions and what question do you want to ask now')
157     question=userCommand()
158     app_id="R2K75H-7ELALHR35X"
159     client = wolframalpha.Client('R2K75H-7ELALHR35X')
160     res = client.query(question)
161     answer = next(res.results).text
162     speak(answer)
163     print(answer)
164
165 time.sleep(5)
166
167
168
169
170
171
172
173
174
```

Listening....
you said:what is the time

02:19 PM
Listening....
you said:tell me a joke

Listening....
you said:I have a question question is

Listening....
you said:question is

Listening....
you said:question is what is 2 + 2

Listening....
you said:what is two plus two

4
Listening....

Fig: 9.4 Getting the answer

CONCLUSION

Through this voice assistant we have automated various services using a single line command it ease most of the tasks of the users like searching the web, retrieving weather forecast details daily news and medical related queries. We mainly aim to make this project a complete server assistant and make it smart enough to act as a replacement for a general server administration. This system is designed to minimise the human efforts and control the device with just human voice.

REFERENCES

1. Head First Python, 2nd Edition O' Reilly Media.
2. AI Assistants Pieraccini Roberto.
3. Youtube Programming Hero.