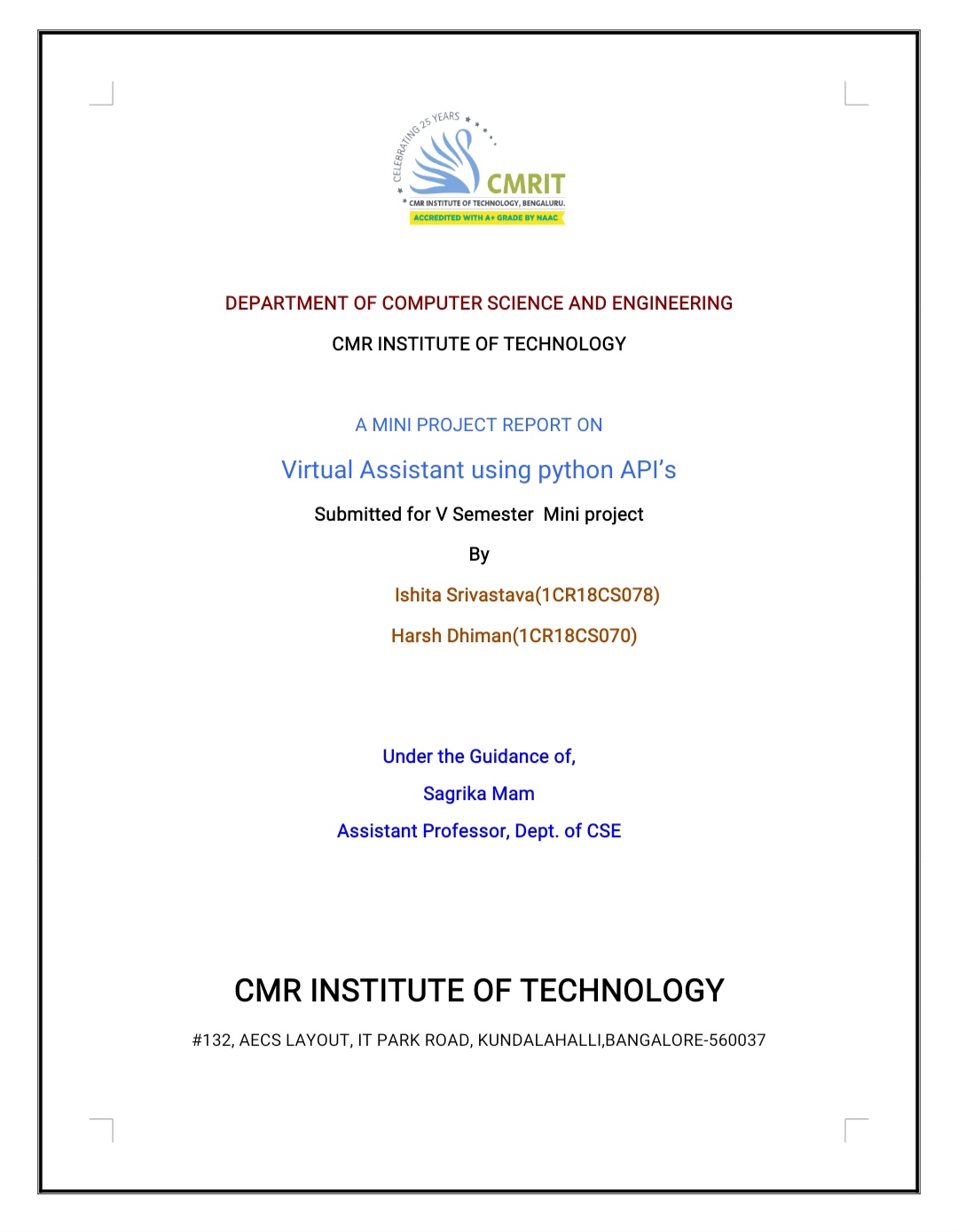
****

**CHAPTER 1**

**INTRODUCTION**

* 1. **Problem Statement**

Personal Assistant that understands speech as well as input and is capable of performing tasks other than conversing. It performs basic functionalities with required internet connection. It can perform tasks as follows:

* Find Places on Google MAP.
* Search queries on Google Search Engine.
* Play Music Online.
* Current Date and Time.
* Weather report of any City/State across the globe.
* Play Youtube videos.
* Sense of Humour (Providing Jokes).
* Reboot/Shutdown the Operating System.
* Close the Program on Command.
  1. **Relevance of the project**

Today, we can ask virtual assistants like Amazon Alexa, Apple’s Siri, Google Now to perform simple tasks like, “What’s the weather”, “Remind me to take pills in the morning”, etc. in natural language. The next evolution of natural language interaction with virtual assistants is in the form of task automation such as “turn on the air conditioner whenever the temperature rises above 30 degrees Celsius”, or “if there is motion on the security camera after 10pm, call Bob”.

* 1. **Objective**

The goal of this project is to learn about NLP applications and try to about the concept behind them. This virtual assistant is not very vast as compared to the other virtual assistants made Google, Facebook, Amazon etc. But it will give brief knowledge about the NLP application used by this project. It will basically listens to the person who is operating it and do the task according to the query entered by the user.

**CHAPTER 2**

**SYSTEM REQUIREMENTS**

**2.1 Functional Requirements**

**2.2 Hardware and Software Requirements**

## Section 2.2.1 Hardware Requirements

* **Processors:** Intel Core i3
* **Processor Speed:**1.9 GHz
* **RAM:**2 GB
* **Speaker**: For getting response
* **MIC :** For input
* **Storage:** 1 GB

## Section 2.2.2 Software Requirements

* **Operating System:**Windows or Linux or Mac OS
* **.Net Framework 4.6 or later**
* **Python 3 or Higher**
* **“espeak” for Linux Operating system**

**CHAPTER 3**

**DESIGN AND IMPLEMENTATION**

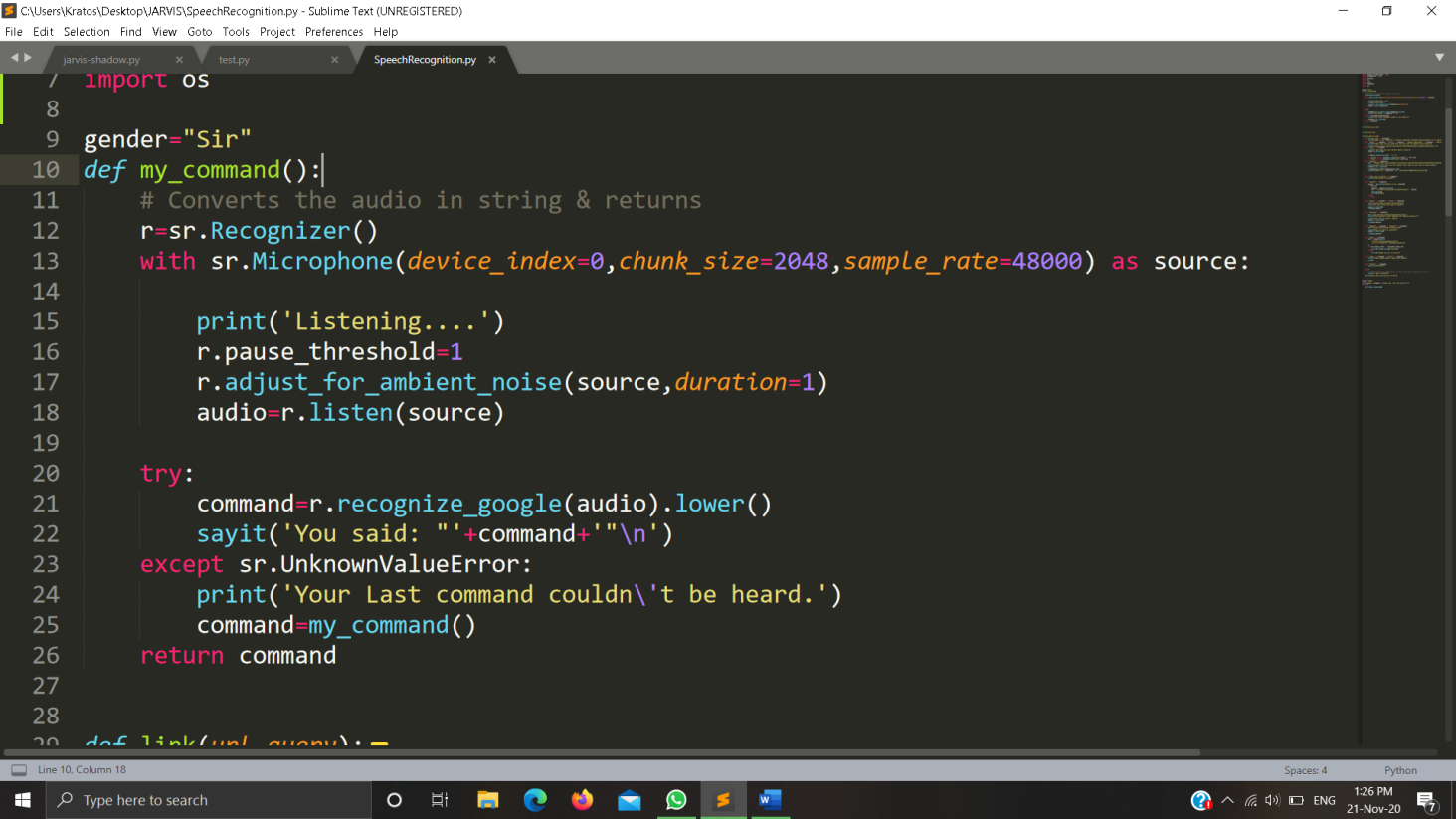
**3.1 Design**

* Speech Recognition – Google
* Text to Speech – Google, pyttsx3
* JSON Parsing
* Regex
* OS – Interacting with System
* Time – System Time

**3.2 Implementation**

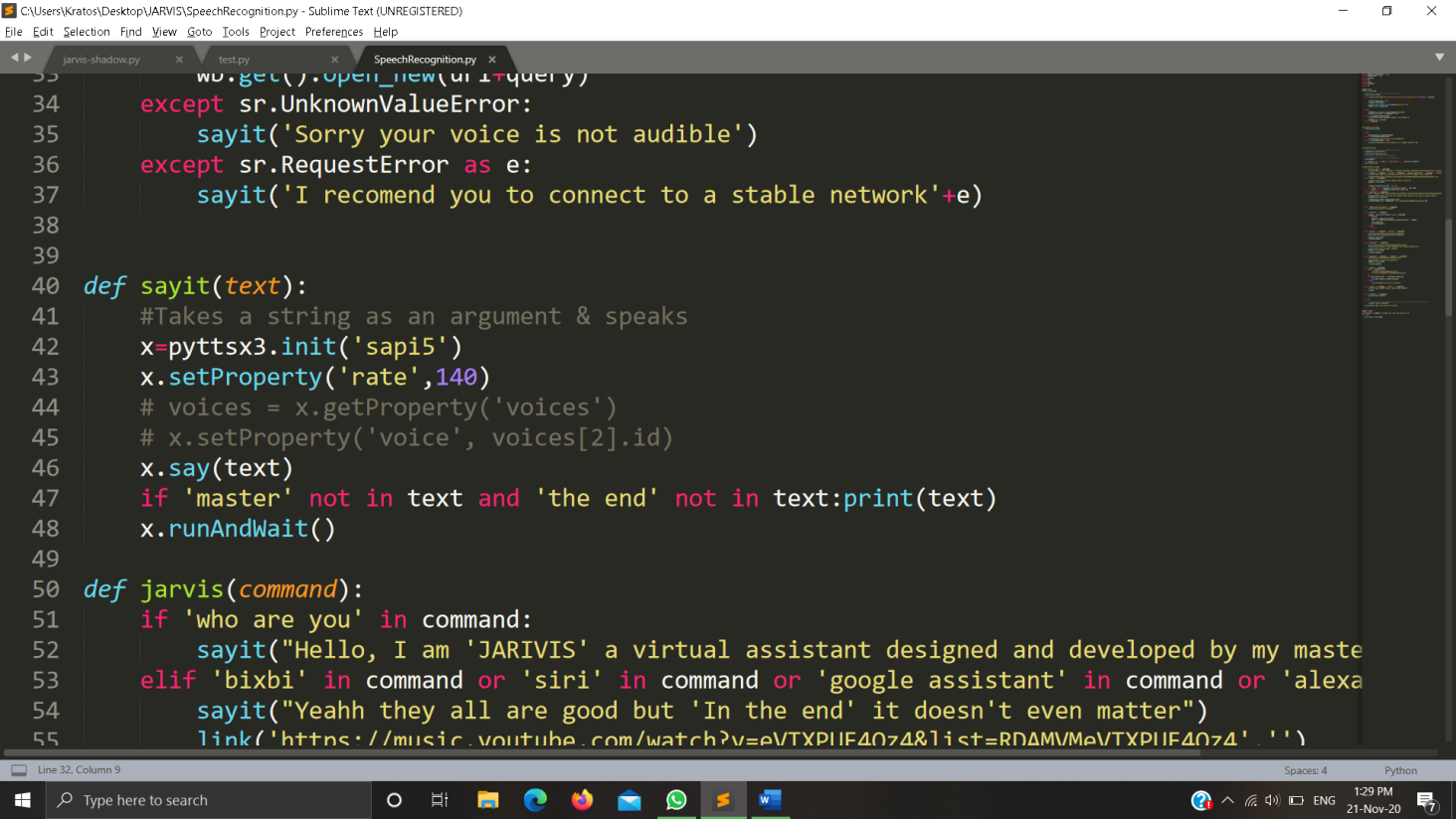
**My\_command():**

Without fail converts the audio into text and returns the string.



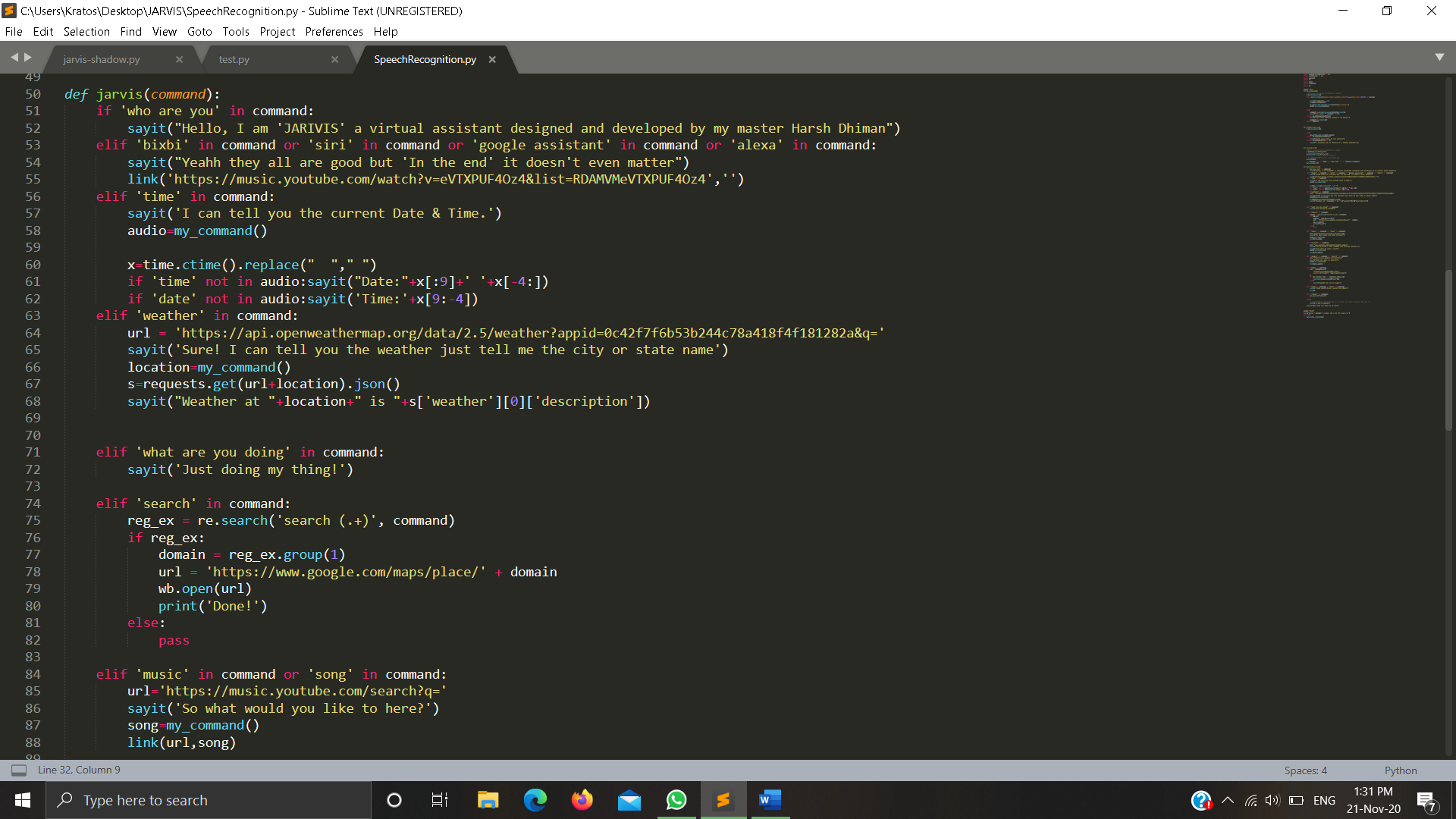
**Sayit():**

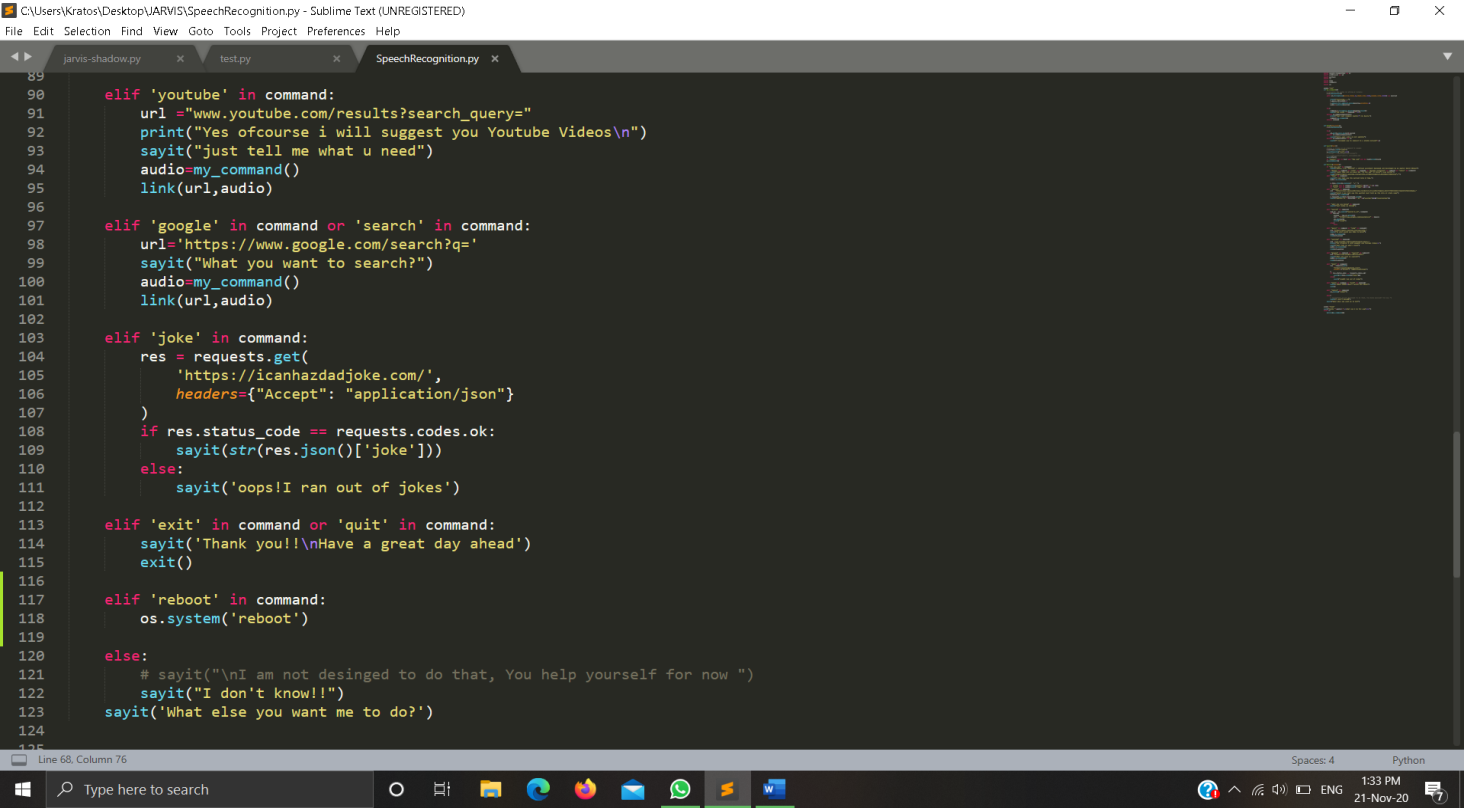
A helper function which converts the string passed as argument into audio.



**JARVIS(command):**

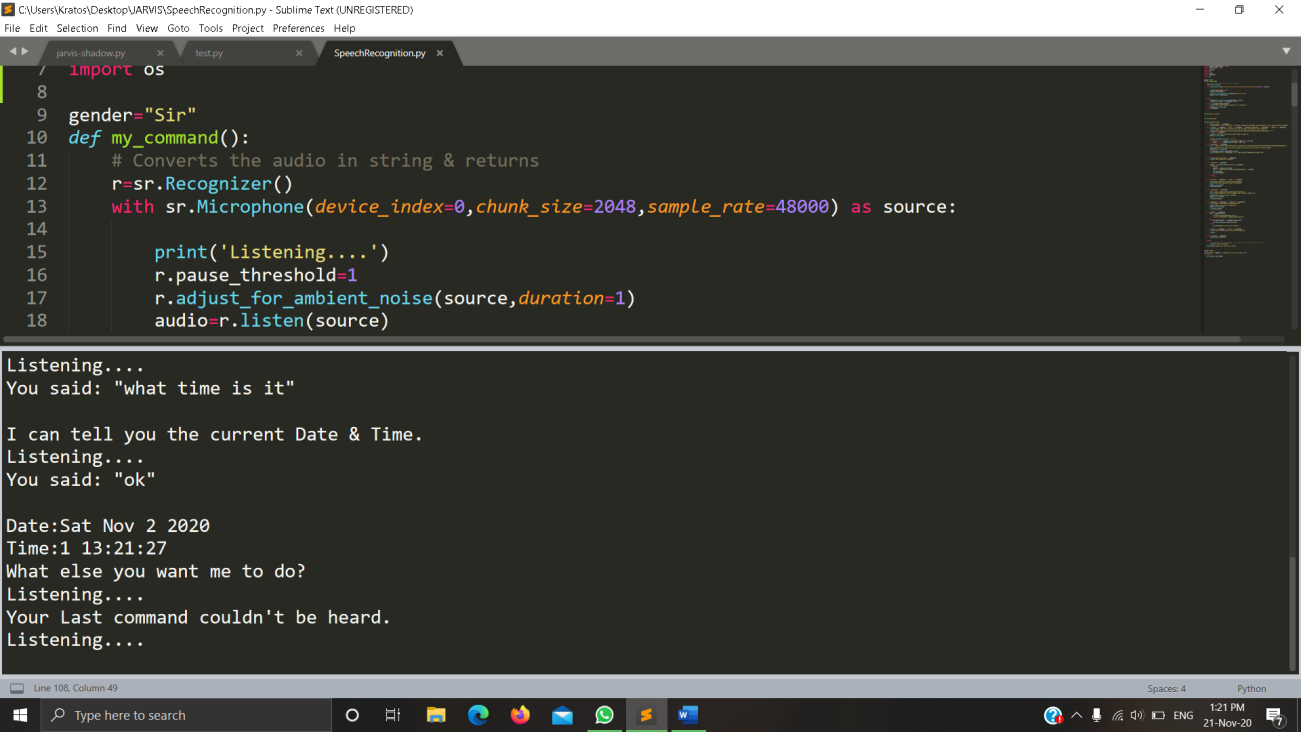
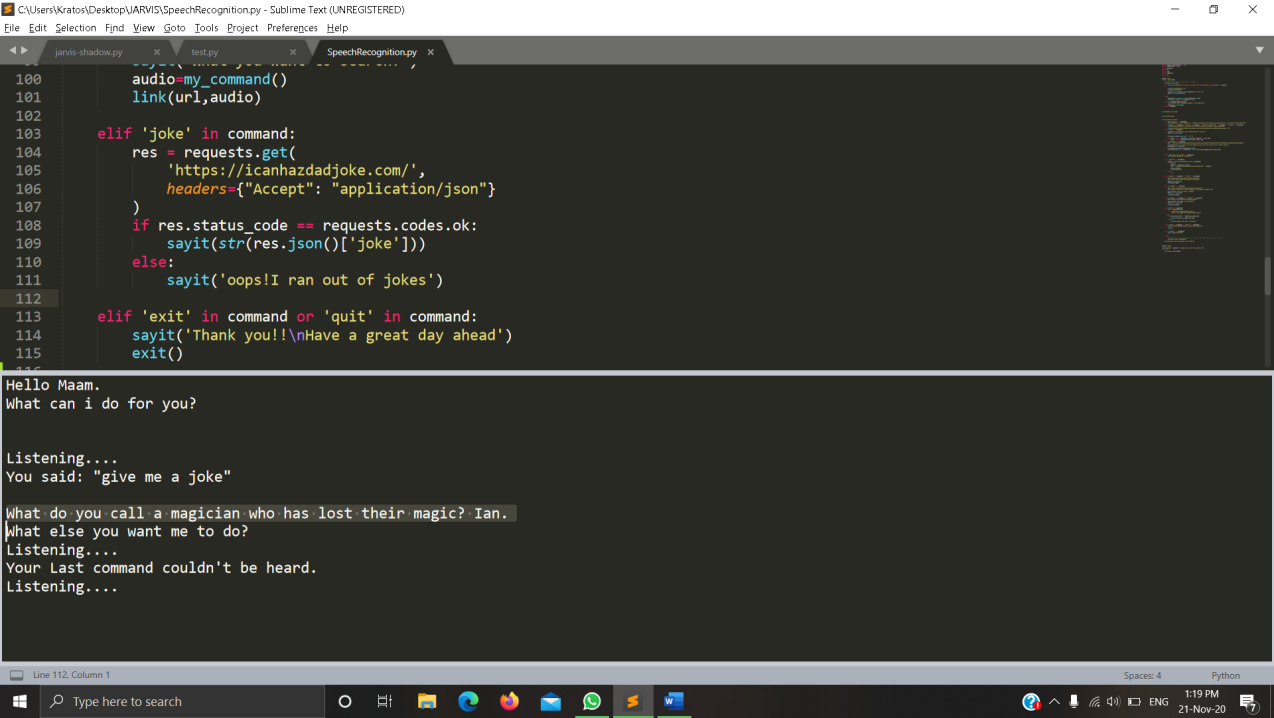
Main function where all the queries are sent & helper functions are called.

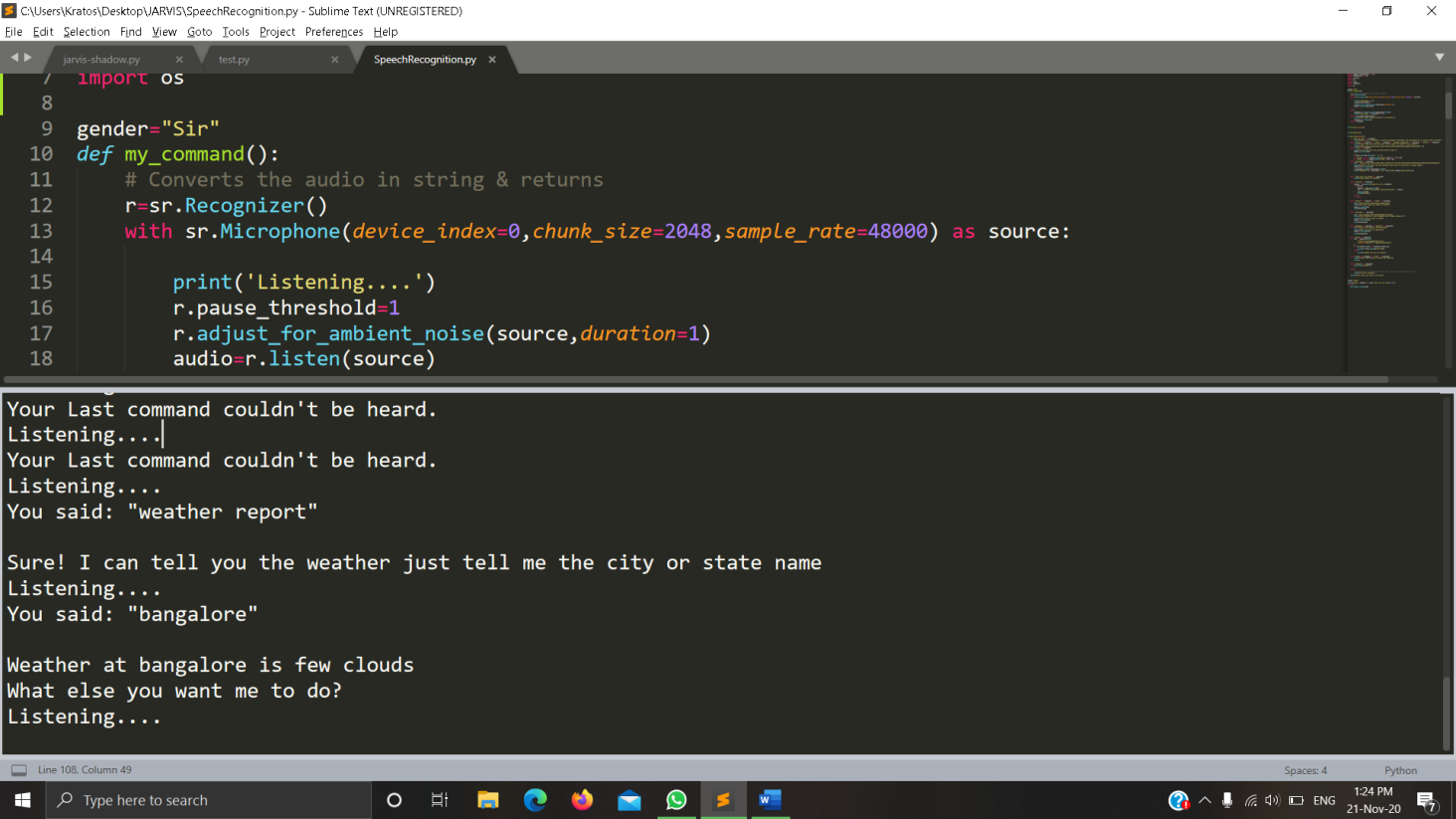




**CHAPTER 4**

**TESTING AND DEPLOYMENT**





**CHAPTER 5**

**CONCLUSION**

* Personal Assistant to do basic functionalities.
* Ask the basic queries.
* More functionality can be added in the future such as to do the functionalities like the other API’s are providing for example – Google , Amazon , Apple etc.
* Highly Scalable.

**REFERENCES**

* Giovanni Campagna, Rakesh Ramesh, Silei Xu, Michael Fischer, and Monica S. Lam. Almond: The architecture of an open, crowdsourced, privacy-preserving, programmable virtual assistant. In Proceedings of the 26th International World Wide Web Conference (WWW-2017), 2017.
* Monica S. Lam, Giovanni Campagna, Jiwon Seo, and Michael Fischer. A distributed open social platform for mobile devices. In Proceedings of the International Conference on Mobile Software Engineering and Systems, MOBILESoft ’16, pages 173–174, New York, NY, USA, 2016. ACM.
* Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In Proceedings of the 53nd Annual Meeting of the Association for Computational Linguistics (ACL-15), pages 1470–1480, 2015.
* Yushi Wang, Jonathan Berant, and Percy Liang. Building a semantic parser overnight. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL-15), pages 1332–1342, 2015.

LINKS:

* <https://console.developers.google.com/>
* <https://pypi.org/project/pyttsx3/>
* <https://www.geeksforgeeks.org/python-text-to-speech-by-using-pyttsx3/>
* <https://developers.google.com/youtube/v3>