**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **TITLE** | **PAGE NO.** |
| 1. | Abstract | 1 |
| 2. | Introduction | 2 |
| 3. | Description | 4 |
| 4. | Observation - project snapshots | 7 |
| 5. | Conclusion | 10 |

**ABSTRACT**

Text to speech is the process of converting text to spoken words. **Python** supports many **speech recognition** engines and APIs, including IBM text to speech.

In the era of human machine interface, speech recognition is being looked upon as highly fascinating field to achieve human computer interaction. Several applications of speech recognition have emerged over the past years including voice dialling, voice query recognition for call routing and simple data entry. Text to speech is the process of automatic determination of linguistic information conveyed by human text using a computer and reconstructing the text into spoken sentence of continuous acoustic signal, overcoming the associated noise induced disturbances.

The aim of this project is to present an overview of the application of text to speech. The system developed would be able to take textual inputs from the user, match it with the database to recognise the digits, perform the required mathematical operation and display the output in speech form.

1. **INTRODUCTION**
   1. **DEFINITION**

Here we are going to create a speaking calculator with help of artificial intelligence based on python language.

A speaking calculator is an advance calculator that inherits ability to create voice patterns from the text to speech operation .

The system developed would be able to take textual inputs from the user, match it with the database to recognise the digits, perform the required mathematical operation and display the output in speech form.

* 1. **PURPOSE**

Enable your systems to “speak” like humans:

Develop interactive calculator for disable and all and to give an ability to calculate hands free .

Customize and control pronunciation:

Deliver a seamless voice interaction that caters to your audience with control over every word

* 1. **SCOPE**

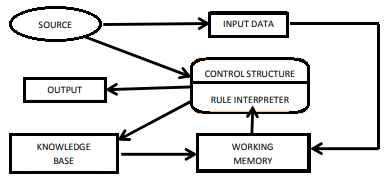
The scope of this project is the system on which the software is programmed I.e the project is developed as desktop application and work for particular computer system.

* 1. **OVERVIEW**

Speaking calculator basically has two main modues for proper functioning:

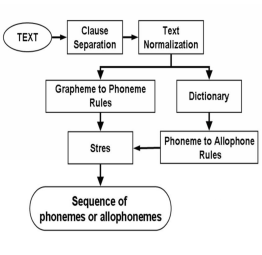
1. User - That inputs the data that is to be processed according to requirement
2. Operating System - That performs the reqired operation input by user and generates output
3. Speaker - that gives an interactive experience via speeking the output
4. **DISCRIPTION**

**2.1 Use case :**

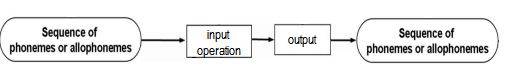


**2.2 DFD**

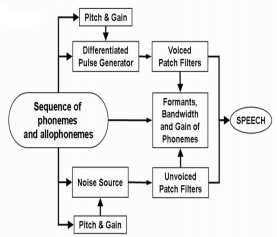
**LEVEL 0:**



**LEVEL 1:**



**LEVEL 2:**



**2.3 Speech Synthesis**

* Speech Synthesis is artificial production of human speech. A synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely “synthetic” voice output
* A computer system used for this purpose i called a speech computer or a speech synthesizer.
* A Text-to-speech (TTS) system converts normal language text into speech ;other systems render symbolic linguistic representations like phonetic transcription into speech
* A Concept-to-speech (TTS) system converts conceptual paradigm into speech

**Approaches:**

There are different approaches to speech synthesis

eg: text-to-speech and concept-to-speech

**Concept-to-speech synthesis** involves a generation component that generates a textual expression from semantic , pragmatic and discourse knowlegde .The speech signal can then be generated from expression.

**Text-to-speech synthesis** a text to be spoken is provided , it is not generated by the system. It must however be analyzed and interpreted in order to convey proper pronunciation and emphasis.

**Technique:** 

* Concatenative synthesis is based on the concatenation (or stringing together) of segments of recorded speech. generally, concatenative synthesis produces the most natural- sounding synthesized speech.
* Formant synthesis does not use human speech samples at runtime . Instead , the synthesized speech output is created using additive synthesis and an acoustic model (physical modelling synthesis). Parameters such as fundamental frequency,voicing and noise levels are varied over time to create a waveform of artificial speech.
* Articulatory synthesis refers to computational techniques for synthesizing speech based on models of the human vocaltract and the articulation processes occurring there.
* HMM-based synthesis is a synthesis method based on Hidden Markov Models, also called Statistical Parametric Synthesis. In this system, the frequency (voice source), and duration (prosody) of speech are modeled simultaneously by HMMs.
* Sinewave synthesis is a techniques for synthesizing speech by replacing the formants (main bands of energy) with pure tone whistles.

**2.4 Text-to-Speech (TTS) synthesizer :**

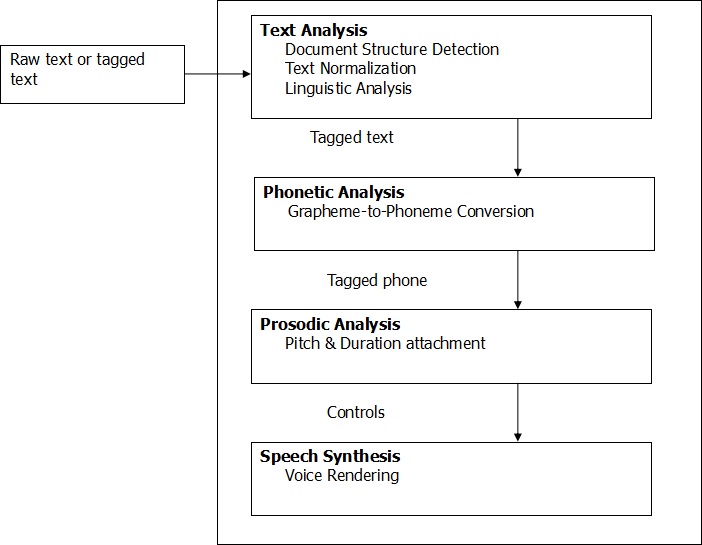
It is a computer based system that should be able to read any text aloud whether it was directly introduced in the computer by an operator or scanned and submitted to an Optical Character Recognition (OCR) system

Voice response system are application of speech synthesis technology and broadly classified in two types

1. Limited vocabulary system

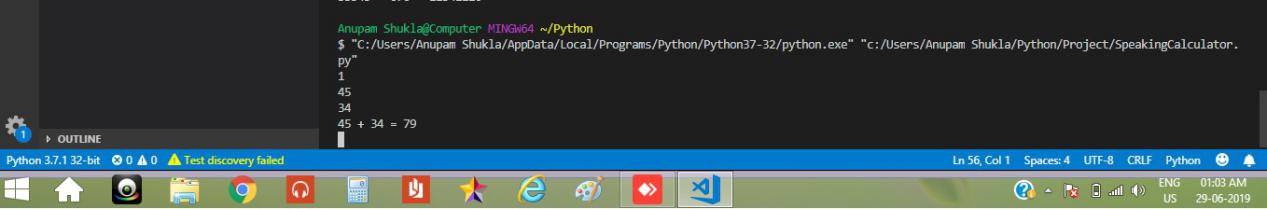
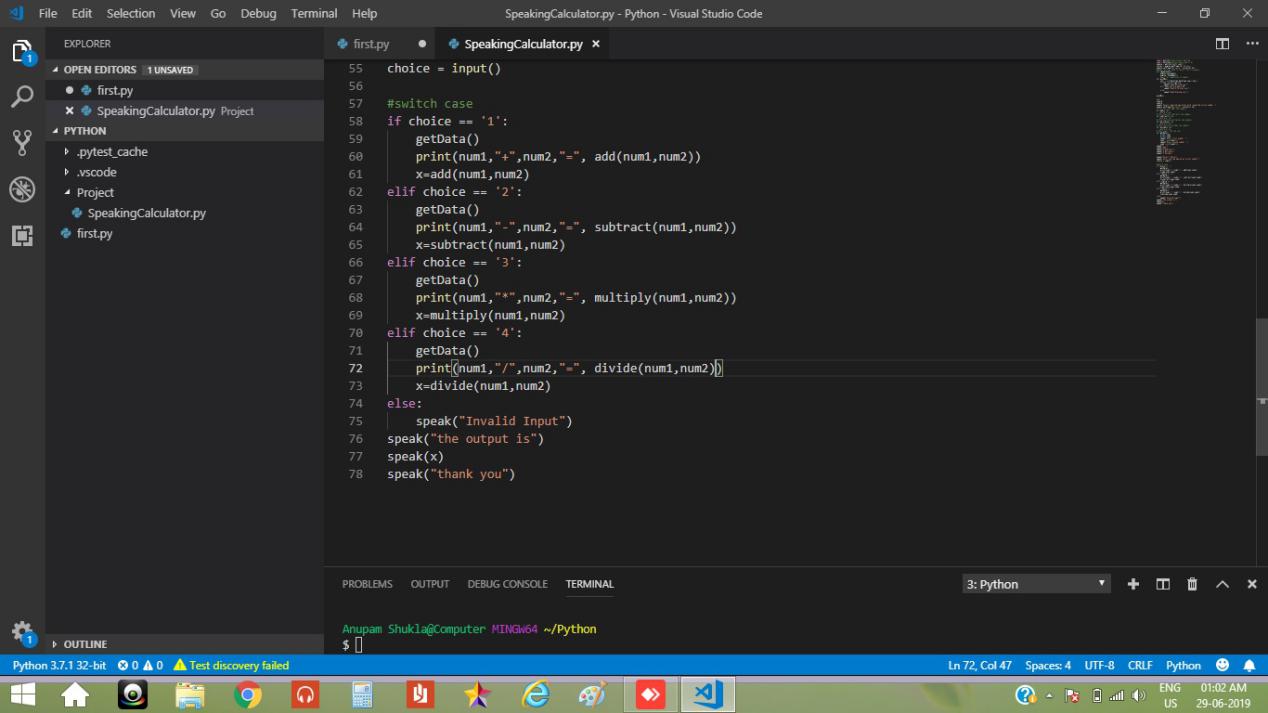
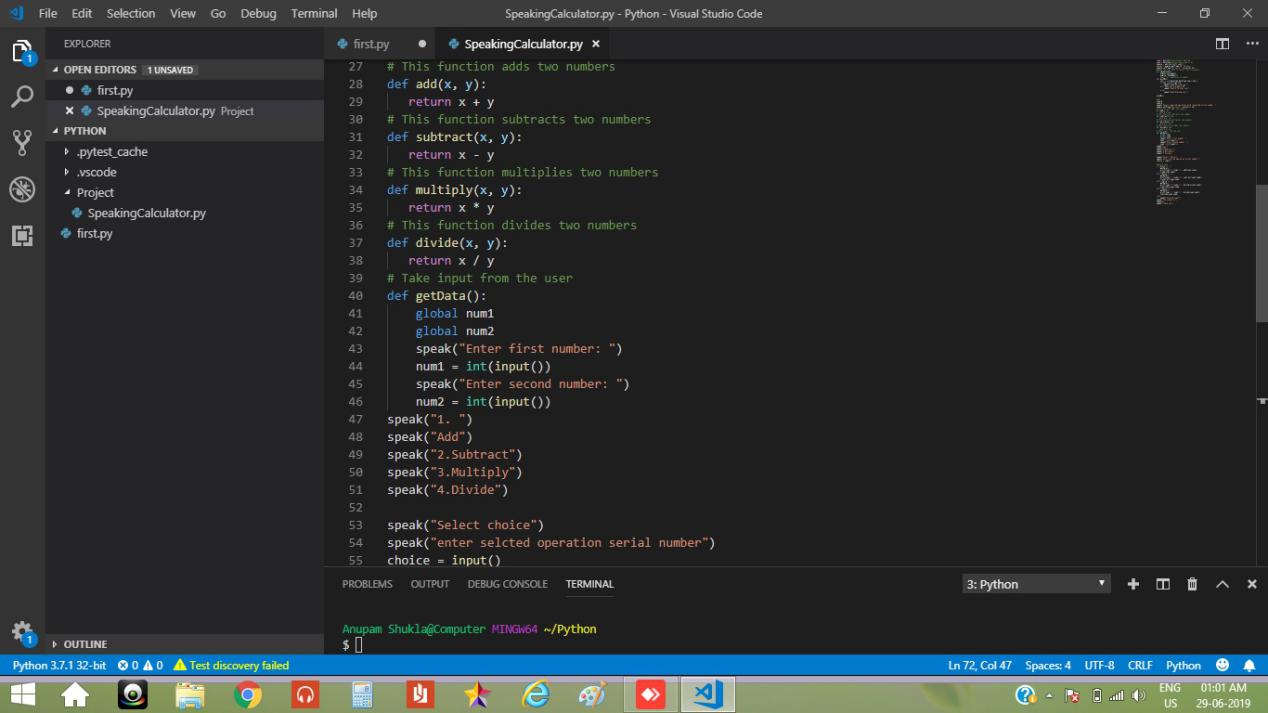
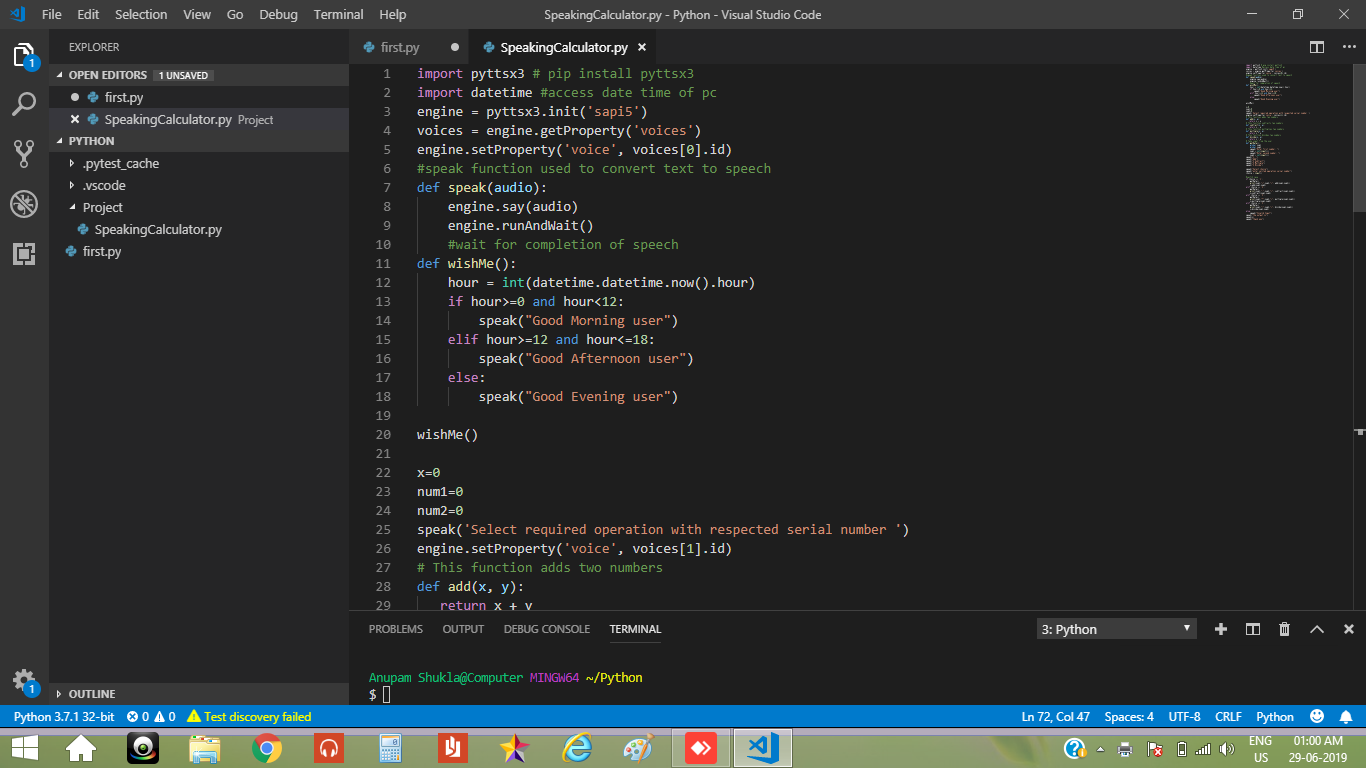
2. Unlimited vocabulary system

**Architecture of TTS system:**



1. **OBSERVATION**

**3.1 Snapshots of code:**

****

**3.2 Python - Highlights and Benefits**

* Python is an interpreted programming language, which means it doesn't require any instruction from machine languages before execution. Therefore, developers can run the program directly. As we all know, any hardware understands machine languages. So, an interpreted language like Python is ideal for any AIbased programs to run, which gives it an edge.
* Python is one of the high-level programming languages capable of handling complex scenarios like variables and arrays. It can also handle objects, complex arithmetic, Boolean expressions, and other challenging computer science ideas. Therefore, it makes the language complete and ultimately increases its popularity, making it worthy for AIbased program more often.
* Python can be used in various domains and technologies. That is why we often call it a general-purpose programming language.
* Python supports a wide range of programming paradigms such as objectoriented, imperative, functional & procedural, which is mainly possible because of two essential features - Dynamic Type System and Automatic Memory Management.
* Python is generally available to almost every operating system. What's interesting is CPython, which is the open source application derived from Python is also gathering its popularity at a rapid pace.

**3.3 Python AI Tutorial – Artificial Intelligence Tools**

1. Search and Optimization :

To intelligently search through possible solutions and use reasoning to do so is a tool for AI. For real-world problems, simple exhaustive searches rarely suffice. This is because these have really large search spaces. This leads to a slow search or one that never ends. To get around this, we can use heuristics.

1. Logic:

AI research uses different forms of logic. Propositional logics use truth functions like ‘or’ and ‘not’. The fuzzy set theory holds a degree of truth (values between 0 and 1) to vague statements. First-order logic adds quantifiers and predicates. Fuzzy logic helps with control systems to contribute vague rules.

1. Probabilistic Methods for Uncertain Reasoning:
2. We often use tools like Bayesian networks for reasoning, learning, planning, and perception. We can also use probabilistic algorithms to filter, predict, smoothen, and explain streams of data.
3. Classifiers and Statistical Learning Methods:

Classifiers and controllers work together. Consider an object. If it is shiny, the classifier knows it is a diamond. If it is shiny, the controller picks it up. But before inferring an action, a controller classifies conditions. As a function, a classifier matches patterns to find the closest match. Supervised learning puts each pattern into a predefined class.

1. Artificial Neural Networks:

ANNs are collections of nodes that are interconnected- inspired by the huge network of neurons in the human brain. Python AI Tutorial – Artificial Neural Networks Under these, we have categories like feedforward neural networks and recurrent neural networks. We will take up ANNs as a separate topic in another tutorial.

1. Evaluating Progress:

Since AI is general purpose, there is no way to find out which domains it excels in. Games are a good benchmark to assess progress.

1. **CONCLUSION**
2. This paper presents the text to speech in Artificial intelligence systems and it is important to consider the environment in which the text to speech system has to work i.e speaking calculator.
3. The grammar used by the speaker and accepted by the system, noise level, noise type, position of the microphone, and speed and manner of the user’s speech are some factors that may affect the quality of tesxt to speech.

**Challenges:**

* Despite large improvements ,Speech Synthesis can still sound a little unnatural
* The approaches to speech sysnthesis that yield the most natural peech need considerable resources in terms of data storage and processing power.
* The process of tokenizing text is rarely straightforward. There are many spellings in english which are pronounced differently based on context making it difficult for users.