**Final Project Report**

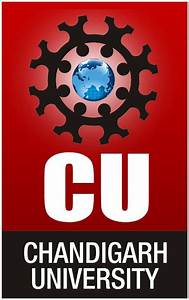
**On**

**AI VIRTUAL ASSISTANT**

By

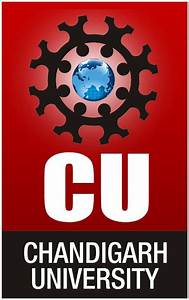
**ABHISHEK VERMA AND VIKHYAT JALOTA**

*In partial fulfillment of requirements for the award of the degree*

**BACHELOR OF ENGINEERING  
COMPUTER SCIENCE AND ENGINEERING  
  
**

*Under the guidance of*

LEEZA SHARMA  
MA’AM

Chandigarh University ****

**College Name**

**Department of**

**ABSTRACT**

Abstract This report discusses ways in which new technology could be harnessed to create an intelligent Virtual Personal Assistant (VPA) with a focus on user-based information. It will look at examples of intelligent programs with natural language processing that are currently available, with different categories of support, and examine the potential usefulness of one specific piece of software as a VPA. This engages the ability to communicate socially through natural language processing, holding (and analysing) information within the context of the user. It is suggested that new technologies may soon make the idea of virtual personal assistants a reality.

Experiments conducted on this system, combined with user testing, have provided evidence that a basic program with natural language processing algorithms in the form of a VPA, with basic natural language processing and the ability to function without the need for other type of human input (or programming) may already be viable.

Voice based personal assistant is a useful tool for search, for reminders, and to write notes just by speaking it up. Intelligent Window assistant is used to create voice apps for intelligent assistant. When user needs to open any other application, he/she can use the command open E.g Open Notepad. File explorer, google chrome, this will help to open the applications. When user wants to write some massages, he/she can use command write.

**CHAPTER 1 : INTRODUCTION**

* 1. **INTRODUCTION**

A virtual assistant, also called AI assistant or digital assistant, is an application program that [understands natural language](https://www.techtarget.com/searchenterpriseai/feature/Natural-language-processing-drives-conversational-AI-trends) voice commands and completes tasks  
for the user. Such tasks, historically performed by a personal assistant or secretary, include taking dictation, reading text or email messages aloud, looking up phone numbers, scheduling, placing phone calls and reminding the end user about appointments. Popular virtual assistants currently include Amazon [Alexa](https://www.techtarget.com/whatis/definition/Alexa-Voice-Services-AVS), Apple's [Siri](https://www.techtarget.com/searchmobilecomputing/definition/Siri), Google Assistant and Microsoft's [Cortana](https://www.techtarget.com/searchenterprisedesktop/definition/Cortana) -- the digital assistant built into Windows Phone 8.1 and Windows 10.

Though this definition focuses on the digital form of virtual assistants, the term *virtual assistant*, or virtual personal assistant, is also commonly used to describe contract workers who work from home doing administrative tasks typically performed by executive assistants or secretaries.

Virtual assistants can also be contrasted with another type of consumer-facing AI programming, called smart advisers. Smart adviser programs are subject-oriented, while virtual assistants are task-oriented.

Virtual assistants are typically cloud-based programs that require internet-connected devices and/or applications to work. Three such applications are Siri on Apple devices, Cortana on Microsoft Devices and Google Assistant on Android devices.

There are also devices dedicated to providing virtual assistance. The most popular ones are available from Amazon, Google and Microsoft. To use the Amazon Echo virtual assistant, called Alexa, users call out the wake word, "Alexa." A light on the device signals to the user it is ready to receive a command, which typically involves simple language requests, such as "what is the weather today," or "play pop music." Those requests are processed and stored in Amazon's cloud.

The technologies that power virtual assistants require massive amounts of data, which feeds artificial intelligence (AI) platforms, including machine learning, [natural language processing](https://www.techtarget.com/searchenterpriseai/definition/natural-language-processing-NLP) and [speech recognition](https://www.techtarget.com/searchcustomerexperience/definition/speech-recognition) platforms. As the end user interacts with a virtual assistant, the AI programming uses sophisticated [algorithms](https://www.techtarget.com/whatis/definition/algorithm) to learn from data input and become better at predicting the end user's needs.

**1.2 AIM**

Voice based personal assistant is a useful tool for search, for reminders, and to write notes just by speaking it up. Intelligent Window assistant is used to create voice apps for intelligent assistant. When user needs to open any other application, he/she can use the command open E.g Open Notepad. File explorer, google chrome, this will help to open the applications. When user wants to write some massages, he/she can use command write.

Main objective of building personal assistant software (a virtual assistant) is usingsemantic data sources available on the web, user generated content and providing knowledgefrom knowledge databases.

The main purpose of an intelligent virtual assistant is to answerquestions that users may have. This may be done in a business environment, for example, onthe business website, with a chat interface. On the mobile platform, the intelligent virtualassistant is available as a call-button operated service where a voice asks the user “What can Ido for you?” and then responds to verbal input.Virtual assistants can tremendously save you time. We spend hours in online researchand then making the report in our terms of understanding. JIA can do that for you. Provide atopic for research and continue with your tasks while JIA does the research. Another difficulttask is to remember test dates, birthdates or anniversaries. It comes with a surprise when youenter the class and realize it is class test today. Just tell JIA in advance about your tests andshe reminds you well in advance so you can prepare for the test.One of the main advantages of voice

searches is their rapidity. In fact, voice is reputedto be four times faster than a written search: whereas we can write about 40 words per minute,we are capable of speaking around 150 during the same period of time15. In this respect, theability of personal assistants to accurately recognize spoken words is a prerequisite for them to be adopted by consumers.

**1.3 EXISTING SYSTEM**

Purpose of virtual assistant is to being capable of voice interaction, music playback, making to-do lists, setting alarms, streaming podcasts, playing audiobooks, and providing weather, traffic, sports, and other real-time information, such as news. Virtual assistants enable users to speak natural language voice commands in order to operate the device and its apps. There is an increased overall awareness and a higher level of comfort demonstrated specifically by millennial consumers. In this ever-evolving digital world where speed, efficiency, and convenience are constantly being optimized, it’s clear that we are moving towards less screen interaction.

Voice assistants will continue to offer more individualized experiences as they get better at differentiating between voices. However, it’s not just developers that need to address the complexity of developing for voice as brands 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 maintaining a user experience that is consistent within the coming years as complexity becomes more of a concern. This is because the visual interface with voice assistants is missing. Users simply cannot see or touch a voice interface.

The mass adoption of artificial intelligence in users’ everyday lives is also fueling the shift towards voice. The number of IoT devices such as smart thermostats and speakers are giving voice

assistants more utility in a connected user’s life. Smart speakers are the number one way we are seeing voice being used. Many industry experts even predict that nearly every application will integrate voice technology in some way in the next 5 years. The use of virtual assistants can also enhance the system of IoT (Internet of Things).Twenty years from now, Microsoft and its competitors will be offering personal digita lassistants that will offer the services of a full-time employee usually reserved for the rich and famous.

**1.4 PROPOSED SYSTEM**

Call someone from my contacts list Launch an application on my iPhone, Send a text message to someone, Set up a meeting on my calendar for 9am tomorrow, Set an alarm for 5am tomorrow morning, Play a specific song in my iTunes library

SIRI does not maintain a knowledge database of its own and its understanding comesfrom the information captured in domain models and data models

ReQall is personal assistant software that runs on smartphones running Apple iOS orGoogle Android operating system. It helps user to recall notes as well as tasks within alocation and time context. It records user inputs and converts them into commands, andmonitors current stack of user tasks to proactively suggest actions while considering anychanges in the environment. It also presents information based on the context of the user, aswell as filter information to the user based on its learned understanding of the priority of that information. Will take some time to put all of the to-do items in – you could spend more time putting the entries in than actually doing the revision.

**1.5 FEASIBILITY STUDY**

A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The study begins by classifying the problem definition. Feasibility is to determine if it’s worth doing. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system. A search for alternatives is analyzed carefully. There are 3 parts in feasibility study.

1) Operational Feasibility

2) Technical Feasibility

3) Economical Feasibility

**1.5.1 OPERATIONAL FEASIBILITY**

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes.To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviours are to be realised. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

**1.5.2 TECHNICAL FEASIBILITY**

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on outline design of system requirements in terms of input, processes, output, fields, programs and procedures. This can be qualified in terms of volume of data, trends, frequency of updating inorder to give an introduction to the technical system. The application is the fact that it has been developed on windows XP platform and a high configuration of 1GB RAM on Intel Pentium Dual core processor. This is technically feasible .The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

**1.5.3 ECONOMICAL FEASIBILITY**

Establishing the cost-effectiveness of the proposed system i.e. if the benefits do not outweigh the costs then it is not worth going ahead. In the fast paced world today there is a great need of online social networking facilities. Thus the benefits of this project in the current scenario make it economically feasible. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis.

1.7.1 INTRODUCTION

This section includes the overall view of the project i.e. the basic problem definition and the general overview of the problem which describes the problem in layman terms. It also specifies the software used and the proposed solution strategy.

1.7.2 SOFTWARE REQUIREMENTS SPECIFICATION

This section includes the Software and hardware requirements for the smooth running of the application.

1.7.3 DESIGN & PLANNING

This section consists of the Software Development Life Cycle model. It also contains technical diagrams like the Data Flow Diagram and the Entity Relationship diagram.

1.7.4 IMPLEMENTATION DETAILS

This section describes the different technologies used for the entire development process of the Front-end as well as the Back-end development of the application.

1.7.5 RESULTS AND DISCUSSION

This section has screenshots of all the implementation i.e. user interface and their description.

1.7.6 SUMMARY AND CONCLUSION

This section has screenshots of all the implementation i.e. user interface and their description.

**CHAPTER 2 : SOFTWARE REQUIREMENTS SPECIFICATION**

**2.1 Hardware Requirements**

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements. 1.3.1.1 Hardware Requirements For The Project Processor: Intel dual Core i3 RAM: 4 GB Hard Disk: 128 GB System Specification: 64-bit operating system, x32-based Processor

**2.2 Software Requirements**

|  |  |  |
| --- | --- | --- |
| **Number** | **Description** | **Type** |
| 1 | Operating System | Windows XP / Windows |
| 2 | Language | Python |
|  |  |  |
| 3 | IDE | Visual Code |
| 4 | Backend | Python Packages |

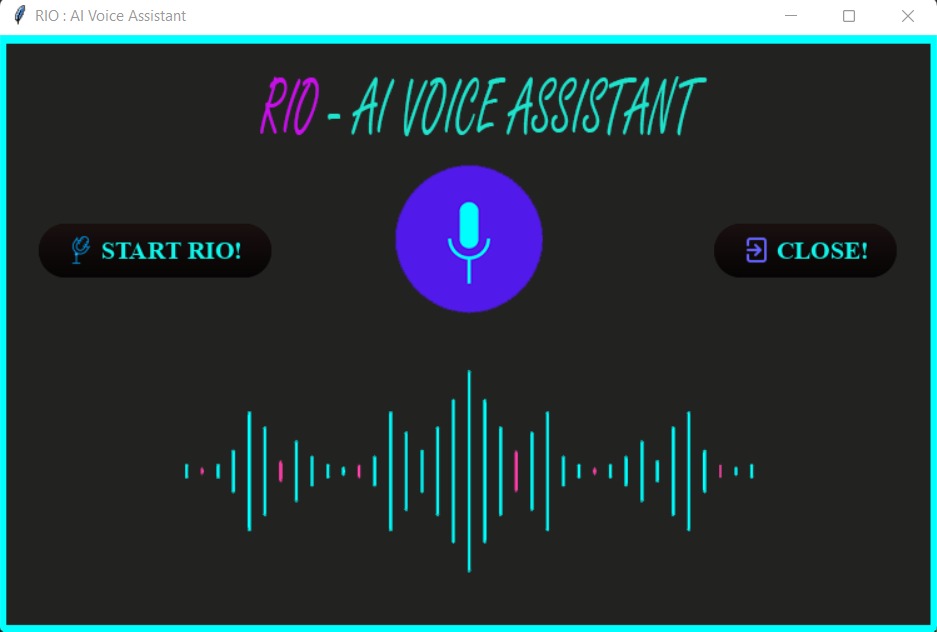
Software Installed: Visual Studio Code Machine Learning Libraries: Numpy, Pandas, Seaborn, Matplotlib, and Sklearn

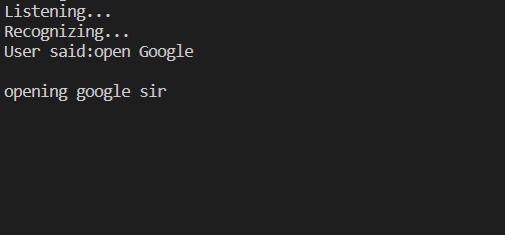
**CHAPTER 3 : DESIGN & PLANNING**

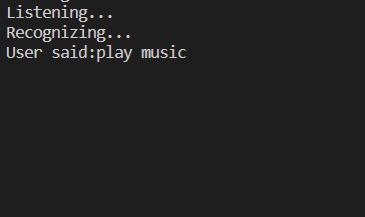
1.PROBLEM FORMULATION As technology continues its rapid advance across all industries around the world, humans become lazy and mostly relay on machines. Virtual assistant with affordability is really beginning to see the benefits. So, this AI voice Assistant helps you to streamline administrative systems and processes, as well as boost the humans overall operations. It makes the task more easier as well as it saves a lot of time.

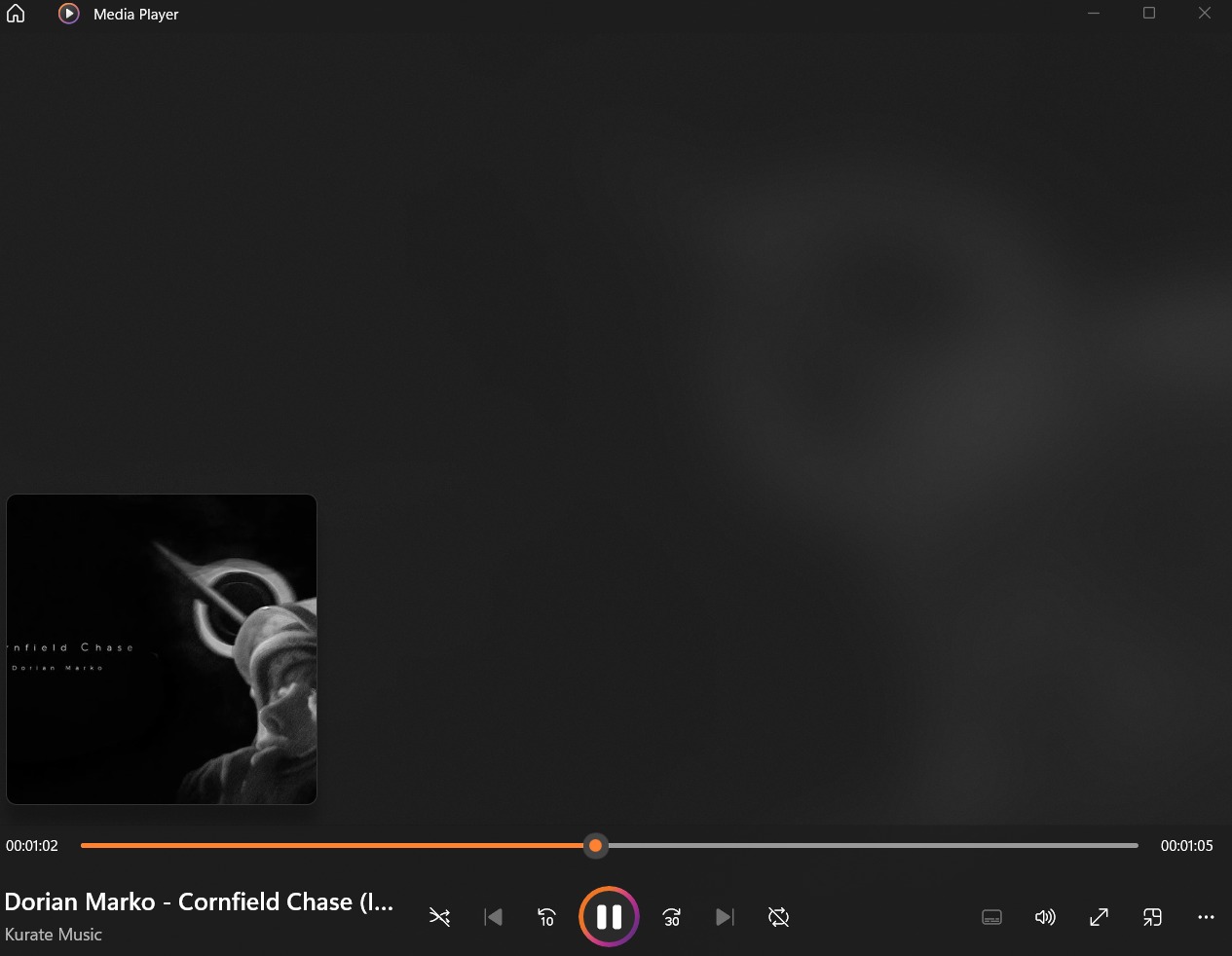
2.RESEARCH OBJECTIVES The proposed research is aimed to carry out work leading to the development of a machine learning model as well as libraries which converts voice into commands. The proposed aim will be achieved by dividing the work into following objectives: 1. To understand and explore various types of libraries used in voice commands existing in open-source software. 2. To study and analyze various datasets techniques that are suitable for our Machine Learning algorithm. 3. To design and develop the machine learning model for the vulnerable. 4. To verify and validate the proposed system. 4. METHODOLOGY We all members decided to make this project because we have basic knowledge of Pythons libraries and Machine Learning , we decided to use this concept to build our project and the following are the methods we used: 1.) We made a program which is user friendly and easy to use. 2.) Use the concepts of some python libraries (e. g, pyttsx3) and Machine Learning. 3.) The work of finding and solving problems, to find new improvements and implementation was done. 4.) The focus was to make a program which have all essential features as Virtual Assistant. 5.) Keeping all things in mind , the program was compiled

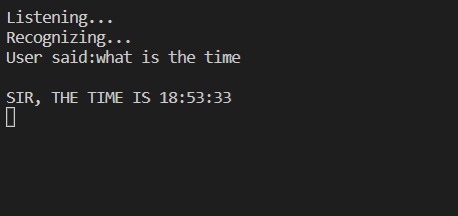
**3.2 SCREENSHOT**

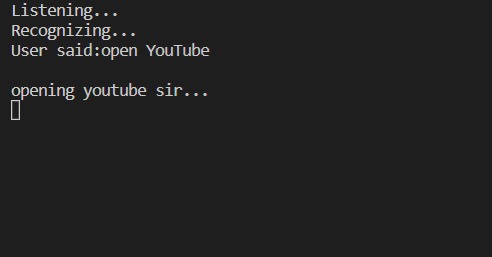


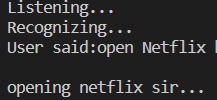
****

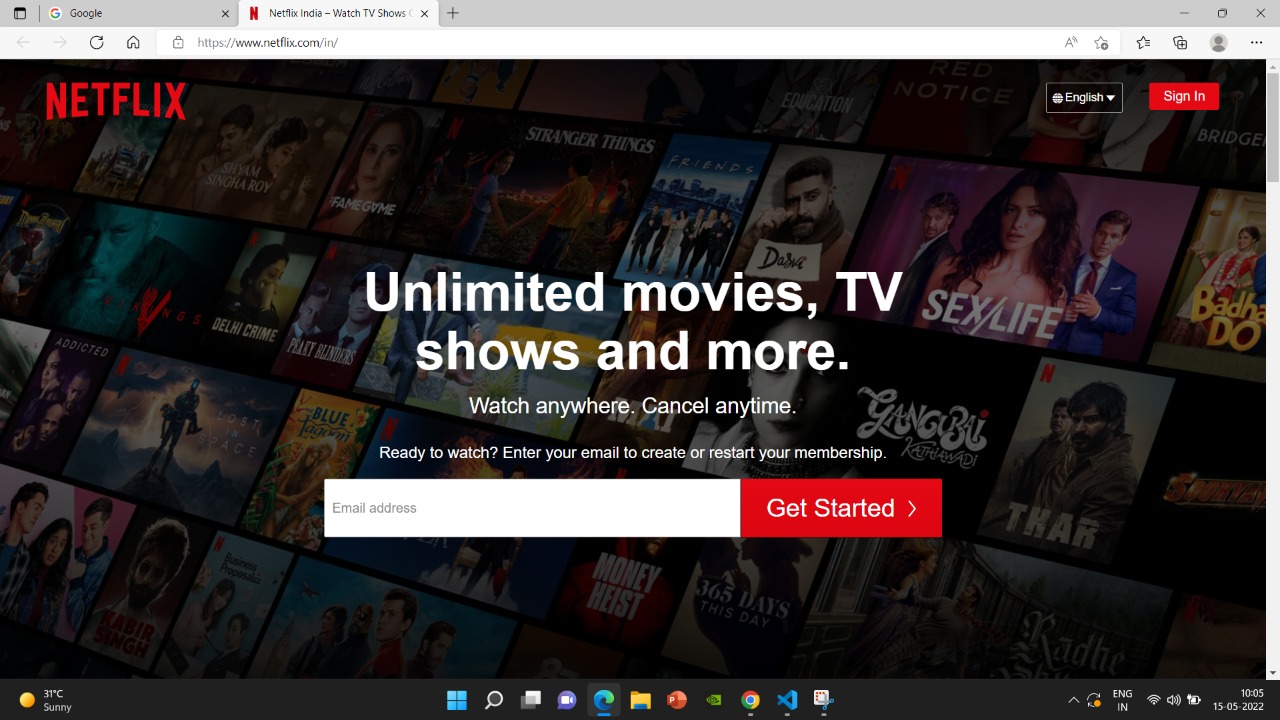
****

****

****

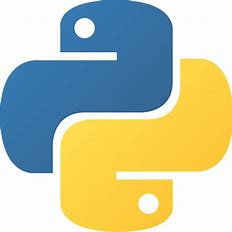
****

****

****

**CHAPTER 4 : IMPLEMENTATION DETAILS**

In this Section we will do Analysis of Technologies to use for implementing the project.

****

**4.1.1 PYTHON**

**Python** is a [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [interpreted](https://en.wikipedia.org/wiki/Interpreter_(computing)), [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language). Its design philosophy emphasizes [code readability](https://en.wikipedia.org/wiki/Code_readability) with the use of [significant indentation](https://en.wikipedia.org/wiki/Off-side_rule).[[32]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-7-32)

Python is [dynamically-typed](https://en.wikipedia.org/wiki/Type_system#DYNAMIC) and [garbage-collected](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigm), including [structured](https://en.wikipedia.org/wiki/Structured_programming) (particularly [procedural](https://en.wikipedia.org/wiki/Procedural_programming)), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) and [functional programming](https://en.wikipedia.org/wiki/Functional_programming). It is often described as a "batteries included" language due to its comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library).

[Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) began working on Python in the late 1980s as a successor to the [ABC programming language](https://en.wikipedia.org/wiki/ABC_(programming_language)) and first released it in 1991 as Python 0.9.0.[[35]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-35) Python 2.0 was released in 2000 and introduced new features such as [list comprehensions](https://en.wikipedia.org/wiki/List_comprehension), [cycle-detecting](https://en.wikipedia.org/wiki/Cycle_detection) garbage collection, [reference counting](https://en.wikipedia.org/wiki/Reference_counting), and [Unicode](https://en.wikipedia.org/wiki/Unicode) support. Python 3.0, released in 2008, was a major revision that is not completely [backward-compatible](https://en.wikipedia.org/wiki/Backward_compatibility) with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020.

Python consistently ranks as one of the most popular programming languages

Python's large standard library [[118]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-86-118) provides tools suited to many tasks, and is commonly cited as one of its greatest strengths. For Internet-facing applications, many standard formats and protocols such as [MIME](https://en.wikipedia.org/wiki/MIME) and [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) are supported. It includes modules for creating [graphical user interfaces](https://en.wikipedia.org/wiki/Graphical_user_interface), connecting to [relational databases](https://en.wikipedia.org/wiki/Relational_database), [generating pseudorandom numbers](https://en.wikipedia.org/wiki/Pseudorandom_number_generator), arithmetic with arbitrary-precision decimals,[[119]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-88-119) manipulating [regular expressions](https://en.wikipedia.org/wiki/Regular_expression), and [unit testing](https://en.wikipedia.org/wiki/Unit_testing).

Some parts of the standard library are covered by specifications—for example, the [Web Server Gateway Interface](https://en.wikipedia.org/wiki/Web_Server_Gateway_Interface) (WSGI) implementation wsgiref follows PEP 333[[120]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-89-120)—but most are specified by their code, internal documentation, and [test suites](https://en.wikipedia.org/wiki/Test_suite). However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

**4.1.2  
Machine Learning Libraries**



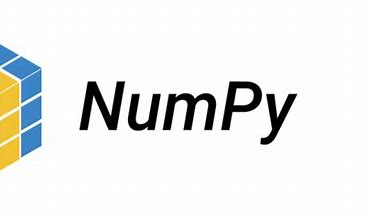
**Machine learning** (**ML**) is a field of inquiry devoted to understanding and building methods that 'learn', that is, methods that leverage data to improve performance on some set of tasks.[[1]](https://en.wikipedia.org/wiki/Machine_learning#cite_note-1) It is seen as a part of [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence). Machine learning algorithms build a model based on sample data, known as [training data](https://en.wikipedia.org/wiki/Training_data), in order to make predictions or decisions without being explicitly programmed to do so.[[2]](https://en.wikipedia.org/wiki/Machine_learning#cite_note-2) Machine learning algorithms are used in a wide variety of applications, such as in medicine, [email filtering](https://en.wikipedia.org/wiki/Email_filtering), [speech recognition](https://en.wikipedia.org/wiki/Speech_recognition), and [computer vision](https://en.wikipedia.org/wiki/Computer_vision), where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.[[3]](https://en.wikipedia.org/wiki/Machine_learning#cite_note-tvt-3)

A subset of machine learning is closely related to [computational statistics](https://en.wikipedia.org/wiki/Computational_statistics), which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of [mathematical optimization](https://en.wikipedia.org/wiki/Mathematical_optimization) delivers methods, theory and application domains to the field of machine learning. [Data mining](https://en.wikipedia.org/wiki/Data_mining) is a related field of study, focusing on [exploratory data analysis](https://en.wikipedia.org/wiki/Exploratory_data_analysis) through [unsupervised learning](https://en.wikipedia.org/wiki/Unsupervised_learning).[[5]](https://en.wikipedia.org/wiki/Machine_learning#cite_note-5)[[6]](https://en.wikipedia.org/wiki/Machine_learning#cite_note-6) Some implementations of machine learning use data and [neural networks](https://en.wikipedia.org/wiki/Neural_networks) in a way that mimics the working of a biological brain. In its application across business problems, machine learning is also referred to as [predictive analytics](https://en.wikipedia.org/wiki/Predictive_analytics).

Learning algorithms work on the basis that strategies, algorithms, and inferences that worked well in the past are likely to continue working well in the future. These inferences can be obvious, such as "since the sun rose every morning for the last 10,000 days, it will probably rise tomorrow morning as well". They can be nuanced, such as "X% of [families](https://en.wikipedia.org/wiki/Family_(biology)) have geographically separate species with color variants, so there is a Y% chance that undiscovered [black swans](https://en.wikipedia.org/wiki/Black_swan_theory) exist".

Machine learning programs can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step.

The discipline of machine learning employs various approaches to teach computers to accomplish tasks where no fully satisfactory algorithm is available. In cases where vast numbers of potential answers exist, one approach is to label some of the correct answers as valid. This can then be used as training data for the computer to improve the algorithm(s) it uses to determine correct answers. For example, to train a system for the task of digital character recognition, the [MNIST](https://en.wikipedia.org/wiki/MNIST_database) dataset of handwritten digits has often been used.

**  
4.1.3 Numpy**

**NumPy** (pronounced [/ˈnʌmpaɪ/](https://en.wikipedia.org/wiki/Help:IPA/English) ([*NUM-py*](https://en.wikipedia.org/wiki/Help:Pronunciation_respelling_key)) or sometimes [/ˈnʌmpi/](https://en.wikipedia.org/wiki/Help:IPA/English)[[4]](https://en.wikipedia.org/wiki/NumPy#cite_note-4)[[5]](https://en.wikipedia.org/wiki/NumPy#cite_note-5) ([*NUM-pee*](https://en.wikipedia.org/wiki/Help:Pronunciation_respelling_key))) is a [library](https://en.wikipedia.org/wiki/Library_(computing)) for the [Python programming language](https://en.wikipedia.org/wiki/Python_(programming_language)), adding support for large, multi-dimensional [arrays](https://en.wikipedia.org/wiki/Array_data_structure) and [matrices](https://en.wikipedia.org/wiki/Matrix_(mathematics)), along with a large collection of [high-level](https://en.wikipedia.org/wiki/High-level_programming_language) [mathematical](https://en.wikipedia.org/wiki/Mathematics) [functions](https://en.wikipedia.org/wiki/Function_(mathematics)) to operate on these arrays.[[6]](https://en.wikipedia.org/wiki/NumPy#cite_note-Nature-6) The ancestor of NumPy, Numeric, was originally created by [Jim Hugunin](https://en.wikipedia.org/wiki/Jim_Hugunin) with contributions from several other developers. In 2005, [Travis Oliphant](https://en.wikipedia.org/wiki/Travis_Oliphant) created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is [open-source software](https://en.wikipedia.org/wiki/Open-source_software) and has many contributors. NumPy is a [NumFOCUS](https://en.wikipedia.org/w/index.php?title=NumFOCUS&action=edit&redlink=1" \o "NumFOCUS (page does not exist)) fiscally sponsored project.

NumPy targets the [CPython](https://en.wikipedia.org/wiki/CPython" \o "CPython) [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) of Python, which is a non-optimizing [bytecode](https://en.wikipedia.org/wiki/Bytecode) [interpreter](https://en.wikipedia.org/wiki/Interpreter_(computing)). [Mathematical algorithms](https://en.wikipedia.org/wiki/List_of_algorithms#Computational_mathematics) written for this version of Python often run much slower than [compiled](https://en.wikipedia.org/wiki/Compiler) equivalents due to the absence of compiler optimization. NumPy addresses the slowness problem partly by providing multidimensional arrays and functions and operators that operate efficiently on arrays; using these requires rewriting some code, mostly [inner loops](https://en.wikipedia.org/wiki/Inner_loop), using NumPy.

Using NumPy in Python gives functionality comparable to [MATLAB](https://en.wikipedia.org/wiki/MATLAB) since they are both interpreted,[[21]](https://en.wikipedia.org/wiki/NumPy#cite_note-21) and they both allow the user to write fast programs as long as most operations work on [arrays](https://en.wikipedia.org/wiki/Array) or matrices instead of [scalars](https://en.wikipedia.org/wiki/Scalar_(computing)). In comparison, MATLAB boasts a large number of additional toolboxes, notably [Simulink](https://en.wikipedia.org/wiki/Simulink), whereas NumPy is intrinsically integrated with Python, a more modern and complete [programming language](https://en.wikipedia.org/wiki/Programming_language). Moreover, complementary Python packages are available; SciPy is a library that adds more MATLAB-like functionality and [Matplotlib](https://en.wikipedia.org/wiki/Matplotlib) is a [plotting](https://en.wikipedia.org/wiki/Plot_(graphics)) package that provides MATLAB-like plotting functionality. Internally, both MATLAB and NumPy rely on [BLAS](https://en.wikipedia.org/wiki/Basic_Linear_Algebra_Subprograms) and [LAPACK](https://en.wikipedia.org/wiki/LAPACK) for efficient [linear algebra](https://en.wikipedia.org/wiki/Linear_algebra) computations.

Python [bindings](https://en.wikipedia.org/wiki/Language_binding) of the widely used [computer vision](https://en.wikipedia.org/wiki/Computer_vision) library [OpenCV](https://en.wikipedia.org/wiki/OpenCV) utilize NumPy arrays to store and operate on data. Since images with multiple channels are simply represented as three-dimensional arrays, indexing, [slicing](https://en.wikipedia.org/wiki/Array_slicing#1991:_Python) or [masking](https://en.wikipedia.org/wiki/Mask_(computing)#Image_masks) with other arrays are very efficient ways to access specific pixels of an image. The NumPy array as universal data structure in OpenCV for images, extracted [feature points](https://en.wikipedia.org/wiki/Interest_point_detection), [filter kernels](https://en.wikipedia.org/wiki/Kernel_(image_processing)) and many more vastly simplifies the programming workflow and [debugging](https://en.wikipedia.org/wiki/Debugger).

****

**4.2.1 Pandas**

**pandas** is a [software library](https://en.wikipedia.org/wiki/Software_library) written for the [Python programming language](https://en.wikipedia.org/wiki/Python_(programming_language)) for data manipulation and [analysis](https://en.wikipedia.org/wiki/Data_analysis). In particular, it offers [data structures](https://en.wikipedia.org/wiki/Data_structure) and operations for manipulating numerical tables and [time series](https://en.wikipedia.org/wiki/Time_series). It is [free software](https://en.wikipedia.org/wiki/Free_software) released under the [three-clause BSD license](https://en.wikipedia.org/wiki/3-clause_BSD_license).[[2]](https://en.wikipedia.org/wiki/Pandas_(software)#cite_note-2) The name is derived from the term "[**pan**el **da**ta](https://en.wikipedia.org/wiki/Panel_data)", an [econometrics](https://en.wikipedia.org/wiki/Econometrics) term for [data sets](https://en.wikipedia.org/wiki/Data_set) that include observations over multiple time periods for the same individuals.[[3]](https://en.wikipedia.org/wiki/Pandas_(software)#cite_note-3) Its name is a play on the phrase "Python data analysis" itself.[[4]](https://en.wikipedia.org/wiki/Pandas_(software)#cite_note-4) [Wes McKinney](https://en.wikipedia.org/wiki/Wes_McKinney) started building what would become pandas at [AQR Capital](https://en.wikipedia.org/wiki/AQR_Capital) while he was a researcher there from 2007 to 2010

* DataFrame [object](https://en.wikipedia.org/wiki/Object-oriented_programming) for data manipulation with integrated indexing.
* Tools for reading and writing data between in-memory [data structures](https://en.wikipedia.org/wiki/Data_structure) and different [file formats](https://en.wikipedia.org/wiki/File_format).
* Data alignment and integrated handling of missing data.
* Reshaping and pivoting of data sets.
* Label-based slicing, fancy indexing, and subsetting of large data sets.
* Data structure column insertion and deletion.
* Group by engine allowing split-apply-combine operations on data sets.
* Data set merging and joining.
* Hierarchical axis indexing to work with high-dimensional data in a lower-dimensional data structure.
* Time series-functionality: Date range generation[[6]](https://en.wikipedia.org/wiki/Pandas_(software)#cite_note-6) and frequency conversions, moving window [statistics](https://en.wikipedia.org/wiki/Statistics), moving window [linear regressions](https://en.wikipedia.org/wiki/Linear_regression), date shifting and lagging.
* Provides data filtration.

The library is highly optimized for performance, with critical code paths written in [Cython](https://en.wikipedia.org/wiki/Cython" \o "Cython) or [C](https://en.wikipedia.org/wiki/C_(programming_language)).

**4.2.2 Seaborn**



**Seaborn**is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.

One has to be familiar with **[Numpy](https://medium.com/coderbyte/numpy-python-f8c8f2bbd13e)** and[**Matplotlib**](https://levelup.gitconnected.com/matplotlib-python-ecc7ba303848) and [**Pandas**](https://levelup.gitconnected.com/pandas-python-e69f4829fee1) to learn about Seaborn.

Seaborn offers the following functionalities:

1. Dataset oriented API to determine the relationship between variables.
2. Automatic estimation and plotting of linear regression plots.
3. It supports high-level abstractions for multi-plot grids.
4. Visualizing univariate and bivariate distribution



**Matplotlib**

.

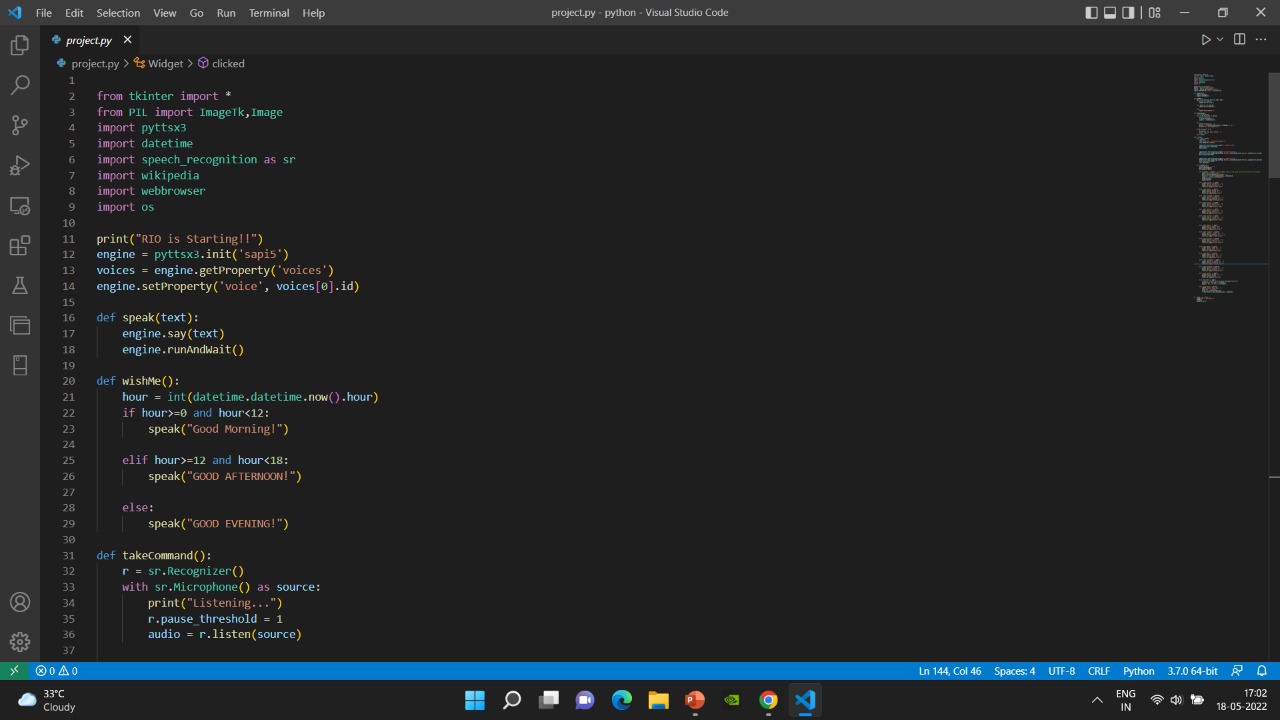
Human minds are more adaptive for the visual representation of data rather than textual data. We can easily understand things when they are visualized. It is better to represent the data through the graph where we can analyze the data more efficiently and make the specific decision according to data analysis. Before learning the matplotlib, we need to understand data visualization and why data visualization is important.

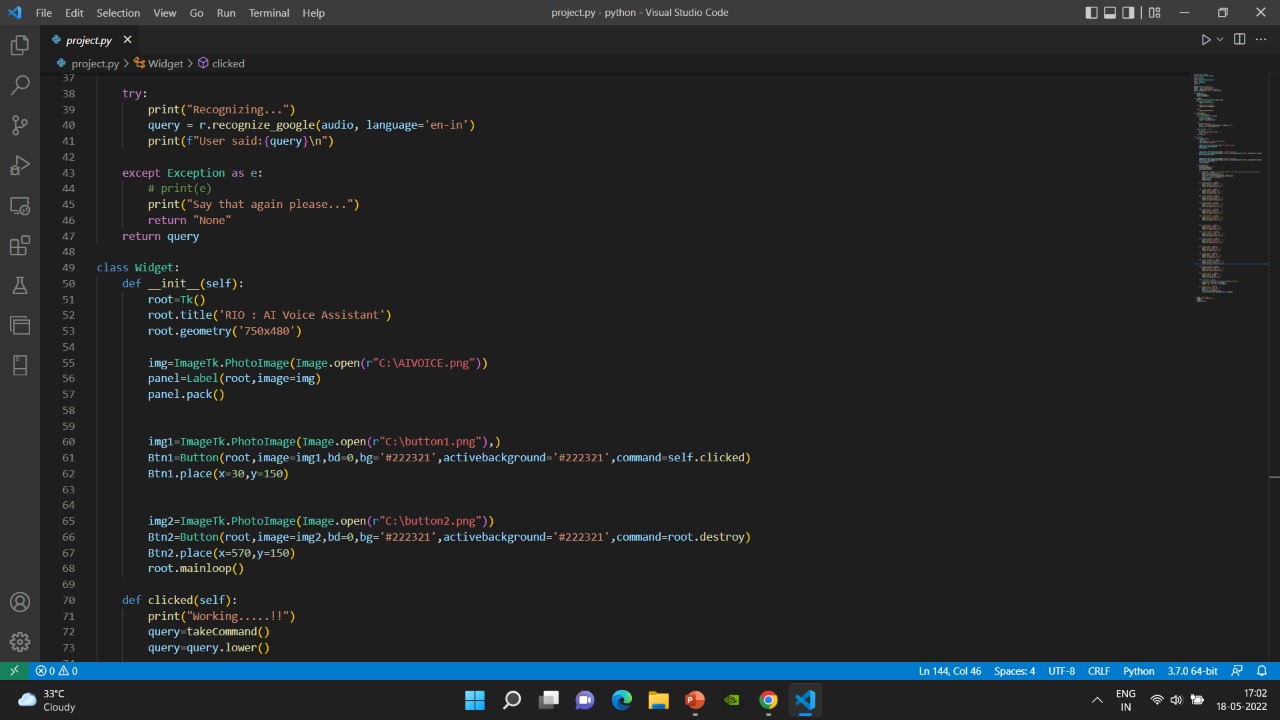
Graphics provides an excellent approach for exploring the data, which is essential for presenting results. Data visualization is a new term. It expresses the idea that involves more than just representing data in the graphical form (instead of using textual form).

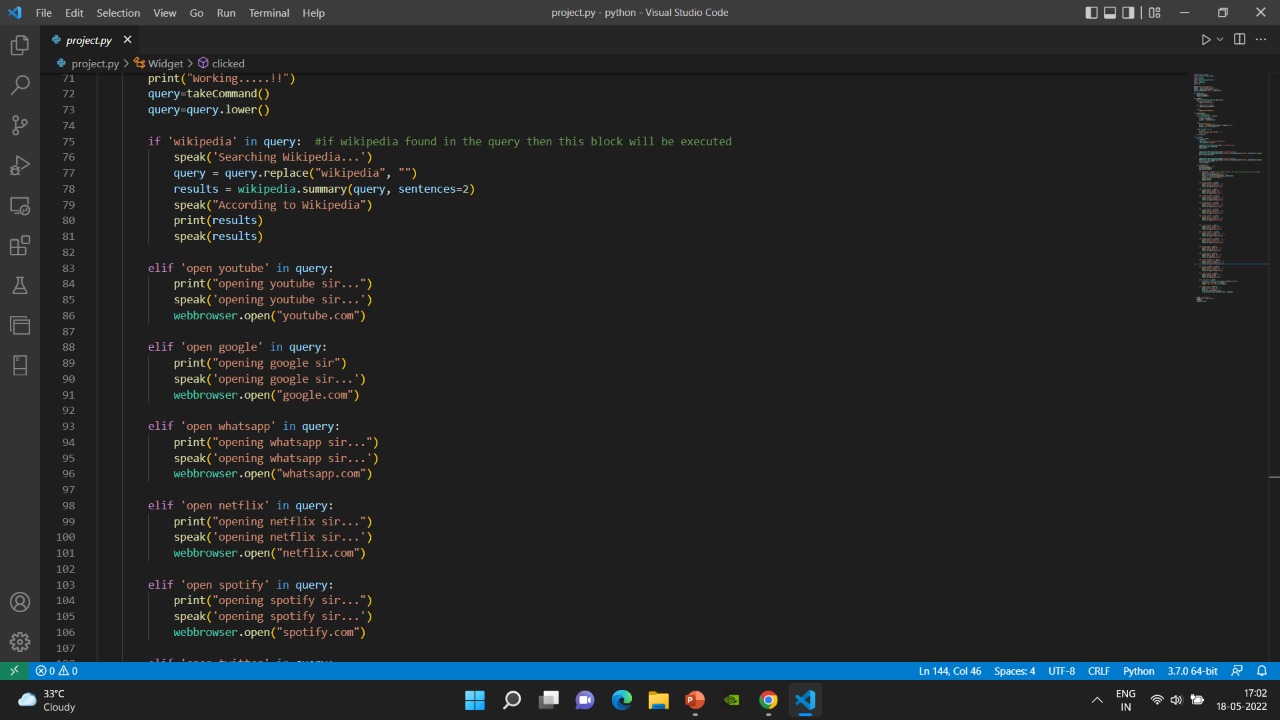
This can be very helpful when discovering and getting to know a dataset and can help with classifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts. The static does indeed focus on quantitative description and estimations of data. It provides an important set of tools for gaining a qualitative understanding.

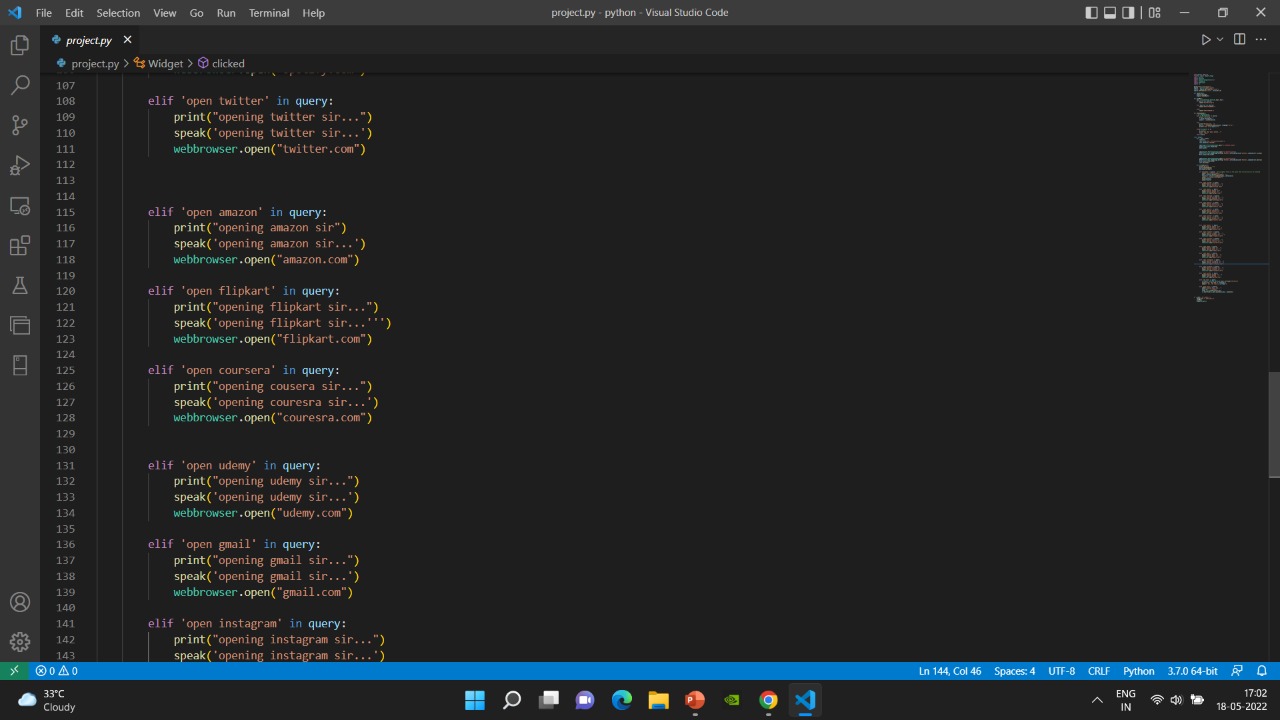
🕑 5-15 minutes — To Create your file with — FileMakr

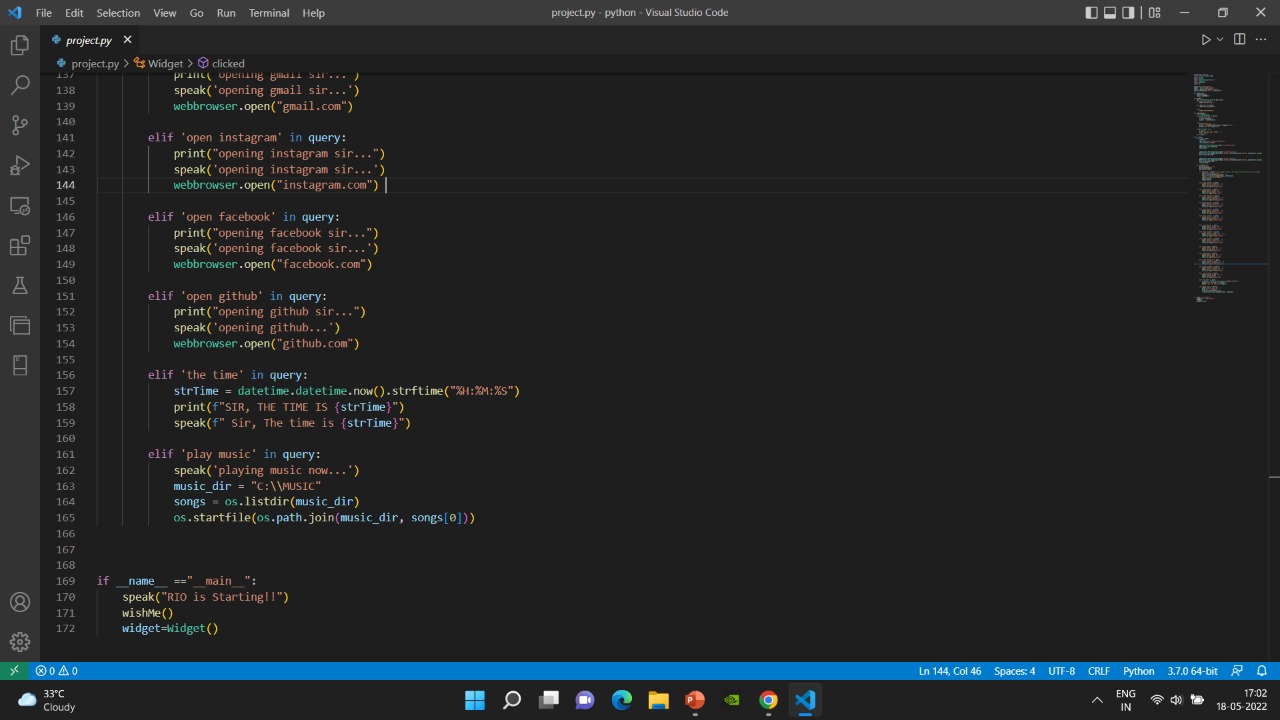
**CODE**

****

****

****

****

****

**CHAPTER 5 : TESTING AND IMPLEMENTATION**

The term implementation has different meanings ranging from the conversation of a basic application to a complete replacement of a computer system. The procedures however, are virtually the same. Implementation includes all those activities that take place to convert from old system to new. The new system may be totally new replacing an existing manual or automated system or it may be major modification to an existing system. The method of implementation and time scale to be adopted is found out initially. Proper implementation is essential to provide a reliable system to meet organization requirement.

**5.1 : UNIT TESTING**

**5.1.1 Introduction**

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. It forms the basis for component testing. Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

**5.1.2 Benifits**

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.

**1) Find problems early :**Unit testing finds problems early in the development cycle. In test-driven development (TDD), which is frequently used in both extreme programming and scrum, unit tests are created before the code itself is written. When the tests pass, that code is considered complete. The same unit tests are run against that function frequently as the larger code base is developed either as the code is changed or via an automated process with the build. If the unit tests fail, it is considered to be a bug either in the changed code or the tests themselves. The unit tests then allow the location of the fault or failure to be easily traced. Since the unit tests alert the development team of the problem before handing the code off to testers or clients, it is still early in the development process.

**2 ) Facilitates Change :**Unit testing allows the programmer to refactor code or upgrade system libraries at a later date, and make sure the module still works correctly (e.g., in regression testing). The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified. Unit tests detect changes which may break a design contract.

**3 ) Simplifies Integration :**Unit testing may reduce uncertainty in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier.

**4 ) Documentation :**Unit testing provides a sort of living documentation of the system. Developers looking to learn what functionality is provided by a unit, and how to use it, can look at the unit tests to gain a basic understanding of the unit's interface (API).Unit test cases embody characteristics that are critical to the success of the unit. These characteristics can indicate appropriate/inappropriate use of a unit as well as negative behaviors that are to be trapped by the unit. A unit test case, in and of itself, documents these critical characteristics, although many software development environments do not rely solely upon code to document the product in development.

**5.2 : INTEGRATION TESTING**

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

**5.2.1 Purpose**

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e., assemblages (or groups of units), are exercised through their interfaces using black-box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e., unit testing. The overall idea is a "building block" approach, in which verified assemblages are added to a verified base which is then used to support the integration testing of further assemblages.Software integration testing is performed according to the software development life cycle (SDLC) after module and functional tests. The cross-dependencies for software integration testing are: schedule for integration testing, strategy and selection of the tools used for integration, define the cyclomatical complexity of the software and software architecture, reusability of modules and life-cycle and versioning management.Some different types of integration testing are big-bang, top-down, and bottom-up, mixed (sandwich) and risky-hardest. Other Integration Patterns[2] are: collaboration integration, backbone integration, layer integration, client-server integration, distributed services integration and high-frequency integration.

*Technologies*

1. Let’s say you want to create your own virtual assistant like Siri. How would you go about making it? Your first and possibly least difficult option would be to integrate Siri into your application directly. Siri, Cortana, and Google Assistant are three well-known examples of AI assistants that many developers integrate into their applications. In 2016, Apple Inc. announced SiriSDK, a development kit that allowed programmers to integrate functions of their own apps as “Tasks” that Siri could perform. SiriSDK uses “Intents” as labels for user intentions and associates Intents with custom classes and properties.
2. If your company doesn’t want to rely on existing AI assistant options, you’d need an expert team of AI engineers to build your own solution. Let’s dive into the key AI technologies behind intelligent virtual assistants.
3. *SPEECH-TO-TEXT (STT) AND TEXT-TO-SPEECH* (TTS)
4. If we’re talking about intelligent virtual assistants, they at the very least require Speech-to-text (STT) and Text-to-speech (TTS) capabilities.
5. Speech-to-text allows apps to convert human speech into digital signals. This is how it works. When you speak, you create a series of vibrations. Using an analog-to-digital converter (ACD) the software converts them into digital signals and extracts sounds, then segments them and matches them to existing phonemes. Phonemes are the smallest unit of a language capable of distinguishing the sound shells of different words. Based on complex mathematical models, the system compares these phonemes with individual words and phrases and creates a text version of what you said.
6. Text-to-speech does the opposite. This technology translates text into voice output. TTS is a computer simulation of human speech from text using machine learning. The system must go through three steps to convert text to voice. First, the system needs to convert text to words, then perform phonetic transcription and then convert transcription to speech.
7. *Speech-to-text (STT*) and Text-to-speech (TTS) are used in virtual assistant technology to ensure smooth and efficient communication between users and applications. To turn a basic voice assistant with static commands into a proper AI assistant, you also need to give the program the ability to interpret user requests with intelligent tagging and heuristics.
8. *COMPUTER VISION (CV)*
9. Computer vision is an AI technology that extracts meaningful information from visual inputs like digital images or videos. CV is an integral part of creating visual virtual assistants. These assistants can respond with creator-generated videos, not just sounds, which greatly enhances the user experience.
10. Computer vision allows the system to recognize body language which is a significant part of communication. Visual virtual assistants powered by this technology use a camera that stores data and utilizes real-time face detection to catch when someone is looking at the screen, this sends a signal to the rest of the system, which converts the user’s speech into text.
11. CV can also greatly increase the accuracy of speech recognition by comparing what the user has said verbally to the movement of the user’s face and mouth.

### NOISE CONTROL

Noise control is another critical feature for voice assistant accuracy. While many smartphones include software-based noise control and suppression features, you can’t count on this being the case for all of your customers. To compensate for a lack of onboard noise suppression software, top-shelf Bluetooth headsets also include hardware noise suppression, but once again there are no guarantees that your AI assistant is going to be able to detect what your customers are

.saying in a busy train car. By integrating in-house noise control packages, you minimize the risk of misunderstanding voice queries.

### *SPEECH COMPRESSION*

Your AI assistant will also need to at least temporarily store voice information for processing unless you’re going to fill up the customer’s hard drive locally with voice data. Speech compression is critical, but developers toe a fine line with compression. It’s possible to compress an audio file so much that substantial amounts of fidelity are lost, making it difficult or impossible to recover what was said during the processing. Compression technology is rapidly improving, but when developing your voice assistant, audio codecs and compression solutions merit a thorough investigation.

### *NATURAL LANGUAGE PROCESSING (NLP)*

Once you have the voice data, the AI assistant needs to process and interpret the data with Natural Language Processing (NLP) and then execute the requested command. NLP simplifies the speech recognition process. While many AI kits are pre-trained on countless hours of voice samples, you’d still need enough data from customers to adjust for precision for your use cases. If your AI assistant is going to respond verbally, you’ll need speech synthesis such as Google Cloud’s top-of-the-line solution, which produces realistic and clear voices.

However, speech processing is not enough to derive a person’s actual intent and maintain a normal conversation.

The request still needs to be interpreted right, and that’s when Natural Language Understanding comes into play.

### *NATURAL LANGUAGE UNDERSTANDING (NLU)*

Natural Language Understanding (NLU) is a different approach to Natural Language Processing and is considered by most computer and data scientists to be a subtopic of NLP. While NLP methods parse, tokenize, and standardize natural language into a standardized structure for command processing, NLU interprets the natural language without standardizing it and derives meaning from queries by identifying the context. In short terms, NLP processes grammar, structure, and compensates for the user’s spelling errors while NLU examines the actual intent behind the query.

### *NATURAL LANGUAGE GENERATION (NLG)*

Natural language generation produces natural language output. Thanks to this technology, users receive a human-like response from virtual assistants and chatbots. Models and techniques used for NLG can be different and depend on the goals of the project and development approaches. One of the simplest approaches is a template system that can be used for texts that have a predefined structure and require only a small amount of data to be filled in. This approach allows such gaps to be automatically filled in with data retrieved from a row in a spreadsheet, a record in a database table, and so on.

Another approach is dynamic NLG which does not require the developer to write code for each edge case and enables the system to react on its own. This is a more advanced type of natural language generation that relies on machine learning algorithms.

### *DEEP LEARNING*

Chatbots that utilize text-based responses only are substantially less complicated than voice assistants. Because you don’t have to then convert speech into text for interpretation, you remove a lot of tooling from the equation when constructing a chatbot. Next-gen text generation such as GPT-3 is capable of producing not only responses to basic queries, but entire news stories from a “seed”. Deep learning makes it happen.

Virtual assistants and chatbots powered by deep learning algorithms learn from their data and human-to-human dialogue. Chatbots that utilize deep learning examine existing

interactions between customers and support staff and create paired messages and responses and compensate for the user’s typos and grammatical errors.

### *AUGMENTED REALITY (AR)*

Augmented reality allows you to overlay 3D objects in the real world for an immersive experience. AR-based mobile chatbots and AR avatars are great examples of using this technology. For example, Arcade created a mobile AR Avatar Chatbot called Miss Perkins for the Ragged School Museum of East London. This assistant serves as a guide for museum visitors and quizzes them ensuring an interactive user experience.

Another example of an intelligent AR chatbot was developed for the Vienna Museum of Technology. The creators also used [mobile AR](https://mobidev.biz/services/augmented-reality). The functionality of the chatbot includes conducting tours and answering user questions about specific display items in the text, images, videos, and audio formats.

The rise of the Metaverse and VR technology leads to the logical conclusion of virtual assistants: 3D AI avatars. Combined with artificial intelligence, AR virtual assistants become more functional, bypassing the limitations of existing AR tools. For example, deep learning allows IVAs to capture user behavior in real-time to drive neural networks that automatically train and improve virtual assistant performance.

### GENERATIVE ADVERSARIAL NETWORKS (GANS)

Being algorithmic architectures that use neural networks, Generative Adversarial Networks create new instances of synthetic data. GANs consist of real image samples and generators fed into discriminators to generate a realistic 3D face for AI avatars and 3D assistants.

The technology has been utilized in many video games and other products to create true-to-life human figures. GANs can also be utilized to turn still images into full-depth 3D images. Perhaps the most advanced integration of AI avatars so far is Nvidia’s Omniverse Avatar Project Maxine, which creates a photorealistic real-time animation of a human face speaking a text-to-speech sample.

**5.3 : SOFTWARE VERIFICATION AND VALIDATION**

**5.3.1 Introduction**

In software project management, software testing, and software engineering, verification and validation (V&V) is the process of checking that a software system meets specifications and that it fulfills its intended purpose. It may also be referred to as software quality control. It is normally the responsibility of software testers as part of the software development lifecycle. Validation checks that the product design satisfies or fits the intended use (high-level checking), i.e., the software meets the user requirements.This is done through dynamic testing and other forms of review.Verification and validation are not the same thing, although they are often confused. Boehm succinctly expressed the difference between

* Validation : Are we building the right product?
* Verification : Are we building the product right?

According to the Capability Maturity Model (CMMI-SW v1.1)

Software Verification: The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.

Software Validation: The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements.

In other words, software verification is ensuring that the product has been built according to the requirements and design specifications, while software validation ensures that the product meets the user's needs, and that the specifications were correct in the first place. Software verification ensures that "you built it right". Software validation ensures that "you built the right thing". Software validation confirms that the product, as provided, will fulfill its intended use.

From Testing Perspective

* Fault – wrong or missing function in the code.
* Failure – the manifestation of a fault during execution.
* Malfunction – according to its specification the system does not meet its specified functionality

Both verification and validation are related to the concepts of quality and of software quality assurance. By themselves, verification and validation do not guarantee software quality; planning, traceability, configuration management and other aspects of software engineering are required.Within the modeling and simulation (M&S) community, the definitions of verification, validation and accreditation are similar:

* M&S Verification is the process of determining that a ⦁ computer model, simulation, or federation of models and simulations implementations and their associated data accurately represent the developer's conceptual description and specifications.
* M&S Validation is the process of determining the degree to which a model, simulation, or federation of models and simulations, and their associated data are accurate representations of the real world from the perspective of the intended use(s).

**5.3.2 Classification of Methods**

In mission-critical software systems, where flawless performance is absolutely necessary, formal methods may be used to ensure the correct operation of a system. However, often for non-mission-critical software systems, formal methods prove to be very costly and an alternative method of software V&V must be sought out. In such cases, syntactic methods are often used.

**5.3.3 Test Cases**

A test case is a tool used in the process. Test cases may be prepared for software verification and software validation to determine if the product was built according to the requirements of the user. Other methods, such as reviews, may be used early in the life cycle to provide for software validation.

**5.4 : Black-Box Testing**

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well.

**5.4.1 Test Procedures**

Specific knowledge of the application's code/internal structure and programming knowledge in general is not required. The tester is aware of what the software is supposed to do but is not aware of how it does it. For instance, the tester is aware that a particular input returns a certain, invariable output but is not aware of how the software produces the output in the first place.

**5.4.2 Test Cases**

Test cases are built around specifications and requirements, i.e., what the application is supposed to do. Test cases are generally derived from external descriptions of the software, including specifications, requirements and design parameters. Although the tests used are primarily functional in nature, non-functional tests may also be used. The test designer selects both valid and invalid inputs and determines the correct output, often with the help of an oracle or a previous result that is known to be good, without any knowledge of the test object's internal structure.

**5.5 : White-Box Testing**

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT). White-box testing can be applied at the unit, integration and system levels of the software testing process. Although traditional testers tended to think of white-box testing as being done at the unit level, it is used for integration and system testing more frequently today. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it has the potential to miss unimplemented parts of the specification or missing requirements.

**5.5.1 Levels**

**1 ) Unit testing :**White-box testing is done during unit testing to ensure that the code is working as intended, before any integration happens with previously tested code. White-box testing during unit testing catches any defects early on and aids in any defects that happen later on after the code is integrated with the rest of the application and therefore prevents any type of errors later on.

**2 ) Integration testing :**White-box testing at this level are written to test the interactions of each interface with each other. The Unit level testing made sure that each code was tested and working accordingly in an isolated environment and integration examines the correctness of the behaviour in an open environment through the use of white-box testing for any interactions of interfaces that are known to the programmer.

**3 ) Regression testing :**White-box testing during regression testing is the use of recycled white-box test cases at the unit and integration testing levels.

**5.5.2 Procedures**

White-box testing's basic procedures involves the tester having a deep level of understanding of the source code being tested. The programmer must have a deep understanding of the application to know what kinds of test cases to create so that every visible path is exercised for testing. Once the source code is understood then the source code can be analyzed for test cases to be created. These are the three basic steps that white-box testing takes in order to create test cases:

* Input involves different types of requirements, functional specifications, detailed designing of documents, proper source code, security specifications. This is the preparation stage of white-box testing to layout all of the basic information.
* Processing involves performing risk analysis to guide whole testing process, proper test plan, execute test cases and communicate results. This is the phase of building test cases to make sure they thoroughly test the application the given results are recorded accordingly.
* Output involves preparing final report that encompasses all of the above preparations and results.

**5.5.3 Advantages**

White-box testing is one of the two biggest testing methodologies used today. It has several major advantages:

* Side effects of having the knowledge of the source code is beneficial to thorough testing.
* Optimization of code by revealing hidden errors and being able to remove these possible defects.
* Gives the programmer introspection because developers carefully describe any new implementation.
* Provides traceability of tests from the source, allowing future changes to the software to be easily captured in changes to the tests.
* White box testing give clear, engineering-based, rules for when to stop testing.

**5.5.5 Disadvantages**

Although white-box testing has great advantages, it is not perfect and contains some disadvantages:

* White-box testing brings complexity to testing because the tester must have knowledge of the program, including being a programmer. White-box testing requires a programmer with a high level of knowledge due to the complexity of the level of testing that needs to be done.
* On some occasions, it is not realistic to be able to test every single existing condition of the application and some conditions will be untested.
* The tests focus on the software as it exists, and missing functionality may not be discovered.

**5.6 : SYSTEM TESTING**

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

**CHAPTER 6 : ADVANTAGES**

1 ) They will stop single direction discussions and exhausting repetitive answers by answering FAQ. If you think these metadata questions are trivial chat bots reduce customer service costs by 30% when deployed in commercial sales marketing funnels. There are dozens of articles and academic research efforts validating the benefits to teachers when they use chat bots to support basic question answering – and they are right.

2) They can quickly devour a big chunk of information and take care of difficult issues so they can be understood—right away. Through natural language processing (NLP) algorithms Chat bots can answer complex questions on the first try that can take several minutes of explanation and repetition to disambiguate for a human. NLP is a complex process that yields simplicity in it’s outcome.

3 ) Easy to use, available 24/7 and fun. Call your instructor at 3 in the morning to find out your next due assignment – I dare you.

4) Students can ask questions privately that they might be embarrassed to ask face to face. Instructors understand the importance of impression management to today’s students who swim in social media presentation like water. How many bits of information are lost if a student thinks they have a “dumb” question?

5) Student impulse to ask questions may be increased. If a student can ask a question at any time without disrupting or interfering with anyone else – do you think they might ask more questions?

6) They give extensive criticism to coaches. Students can say whatever they want to their bot without fear of offending anyone.

7) Because the bots are always in development the discussion is always fresh and timely. Some teachers get into a routine – and appear dated or out of touch to their students as a result

8 ) They quickly can give visual answers – and a picture is worth a thousand words. Imagine being able to immediately show a picture related to the required answer.

9) They are intended to communicate like a human. In this way they can draw in students with cleverness and insight. Dialog design for a bot can be playful unique and compelling.

10) Students realize they can return and pose extra inquiries at anytime. The bot doesn’t become impatient with a topic simple becasue it has been previously discussed. They never say “Haven’t I already discussed with with you?” or “I have other students to attend to right now”

11) They can assist students with language difficulties as they can look at the responses and answers a few times, without feeling awkward. Even the best Instructors can become frustrated by 2nd language communication difficulties.

12) They deal with cell phone popularity by appearing on them. A bot never says “Students please put away your phones.” 1 ) They will stop single direction discussions and exhausting repetitive answers by answering FAQ. If you think these metadata questions are trivial chat bots reduce customer service costs by 30% when deployed in commercial sales marketing funnels. There are dozens of articles and academic research efforts validating the benefits to teachers when they use chat bots to support basic question answering – and they are right.

**CHAPTER 7 : CONCLUSION**

Usually, user needs to manually manage multiple sets of applications to complete one task. For example, a user trying to make a travel plan needs to check for airport codes for nearby airports and then check travel sites for tickets between combinations of airports to reach the destination. There is need of a system that can manage tasks effortlessly. We already have multiple virtual assistants. But we hardly use it. There are number of people who have issues in voice recognition. These systems can understand English phrases but they fail to recognize in our accent. Our way of pronunciation is way distinct from theirs. Also, they are easy to use on mobile devices than desktop systems. There is need of a virtual assistant that can understand English in Indian accent and work on desktop system. When a virtual assistant is not able to answer questions accurately, it’s because it lacks the proper context or doesn’t understand the intent of the question. Its ability to answer questions relevantly only happens with rigorous optimization, involving both humans and machine learning. Continuously ensuring solid quality control strategies will also help manage the risk of the virtual assistant learning undesired bad behaviors. They require large amount of information to be fed in order for it to work efficiently. Virtual assistant should be able to model complex task dependencies and use these models to recommend optimized plans for the user. It needs to be tested for finding optimum paths when a task has multiple sub-tasks and each sub-task can have its own sub-tasks. In such a case there can be multiple solutions to paths, and the it should be able to consider user preferences, other active tasks, priorities in order to recommend a particular plan.

**BIBLIOGRAPHY**

* https://www.tutorialspoint.com/pythonlib
* https://www.javatpoint.com
* https://www.w3schools.com
* https://harrywithcode.com