College Enquiry Chat bot

Software Engineering (Osmania University)

College Enquiry Chat bot

A project report submitted for the partial fulfillment of the award of degree of

Bachelor of Science

By

MOHAMMED ARIF UDDIN

121420467027 Bsc (Mscs)

SYED SHAH IFTEQUAR UDDIN

121420468026 Bsc (Mpcs)

Under the guidance of

Ms. SWATHI CHINTAVATLA

Mr. RAMA KRISHNA

St. Joseph's Degree &PG College

(Re-Accredited by NAAC with 3rd Cycle)

(Affiliated to Osmania University)

Hyderabad – 500 029

2022-2023

CERTIFICATE

This is to certify that this project entitled “**College Enquiry Chat bot”** is a bonafide work carried out by **MOHAMMED ARIF UDDIN** bearing Hall Ticket No: **121420467027 and SYED SHAH IFTEQUAR UDDIN** bearing Hall Ticket No: **121420468026** in **Bachelor of Science** and submitted to **St. Joseph’s Degree & PG College** in partial fulfillment of the requirements for the award of Bachelor of Science.

Project Guide H.O.D External Examiner

<u>DECLARATION</u>

The current study “**College Enquiry Chat bot**” has been carried out under supervision of Guides: **SWATHI CHINTAVATLA, RAMA KRISHNA**, Department of Computer Science, St. Joseph’s Degree & PG College, and Hyderabad. We hereby declare that the present study that has been carried out by us, us during May 2023 is original and no part of it has been carried out prior to this date.

Date:

Signature of Candidates:

MOHAMMED ARIF UDDIN

SYED SHAH IFTEQUAR UDDIN

ACKNOWLEDGEMENT

We feel ourselves honored and privileged to place our warm salutation to our college **St. Joseph's Degree & PG College** and department of **Bachelor of Science** which gave us the opportunity to have expertise in engineering and profound technical knowledge.

We would like to convey thanks to our project guideS **SWATHI CHINTAVATLA** and **RAMA KRISHNA** for their regular guidance and constant encouragement and we are extremely grateful to them for their valuable suggestions and unflinching co-operation throughout project work.

With Regards and Gratitude

MOHAMMED ARIF UDDIN
121420468026 BSC (MSCS)

SYED SHAH IFTEQUAR UDDIN
121420468026 BSC (MPCS)

ABSTRACT

College Enquiry Chat bot is simple web application which aims to provide the information regarding college. It can be an upgraded form of our college's WEBkiosk. After some improvements and some additions this project can be fully embedded into the working site of the college. The chatbot created here is a web based application which used Natural Language Processing Libraries and Artificial Intelligence Markup Language to have conversations with humans. .It is a simple bot which answers the queries regarding the college. It uses hardcoded phrases so that the conversation is continued. In the future, NLP can be implemented to understand what a user is saying and give the solutions to his problems. Natural Language Processing is a field of computer science sub- branch artificial intelligence which is concerned with interactions between computer and humans.

INDEX

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

College Enquiry Chat bot is simple web application which aims to provide the information regarding college. The information can be in the form of teachers or student's GPA or the various activities in the college. It can be an upgraded form of our college's webkiosk. After some improvements and some additions this project can be fully embedded into the working site of the college.

The chatbot created here is a web based application which used Natural Language Processing Libraries and Artificial Intelligence Markup Language to have conversations with humans. "Eliza" and "Cleverbot" are some of the web applications which have been created in the past. Like "Eliza", the responses of this chatbot are programmed up to some extent. This is because of the fact that it is a simple bot which answers the queries regarding the college. Since the curriculum of the college keeps on changing, there has to be database which can be edited and upgraded from time to time.

So far, what I have achieved is a sample program which processes the response of the users by using simple parsing and substituting them into premade templates. It also uses hardcoded phrases so that the conversation is continued.

In the future, NLP can be implemented to understand what a user is saying and give the solutions to his problems. Natural Language Processing is a field of computer science sub-branch artificial intelligence which is concerned with interactions between computer and humans. Some the field inside NLP are Natural Language Understanding(NLU) and Natural Language Generation(NLG). College Enquiry ChatBot is a web application which uses artificial intelligence concepts to have conversations with humans. Some of the similar web applications built in the past are "Eliza", "Cleverbot" etc. This report will revolve around the concept of NLP and AIML along with the work committed to build Eliza. Further, we will also see the various problems and complications that arise while developing these applications and how these can be managed to make them better.

The sample application is developed using Python Kernel and XML's Artificial Intelligence Markup Language(AIML) along with a database file which stores the name, e-mail, and password to tell the GPA of a student. It is accessed using MYSQL. The front end of the project is designed using HTML, CSS and Javascript.

The inspiration to build this project came from the working our college's webkiosk. It is possible that the chatbot can be connected to the college database using the webkiosk's API but it included the implementation of JSON.

The construction of the project is similar to Eliza. Since it was first of its kind and an open source, it provided an idea of how these programs work. It worked on an algorithm based on substitution. Another creation named Cleverbot was much efficient than Eliza, but since it is not an open source and its algorithm is also very complex, it is not much of an importance here. However, if its algorithm is studied, it would create an application which would be very complex and help to broaden the scope of ChatBot.

CHAPTER 2

LITERATURE SURVEY

2. LITERATURE SURVEY

1.1 Components of ChatBot Application

The main components of the ChatBot are:

- UI
- Back-end

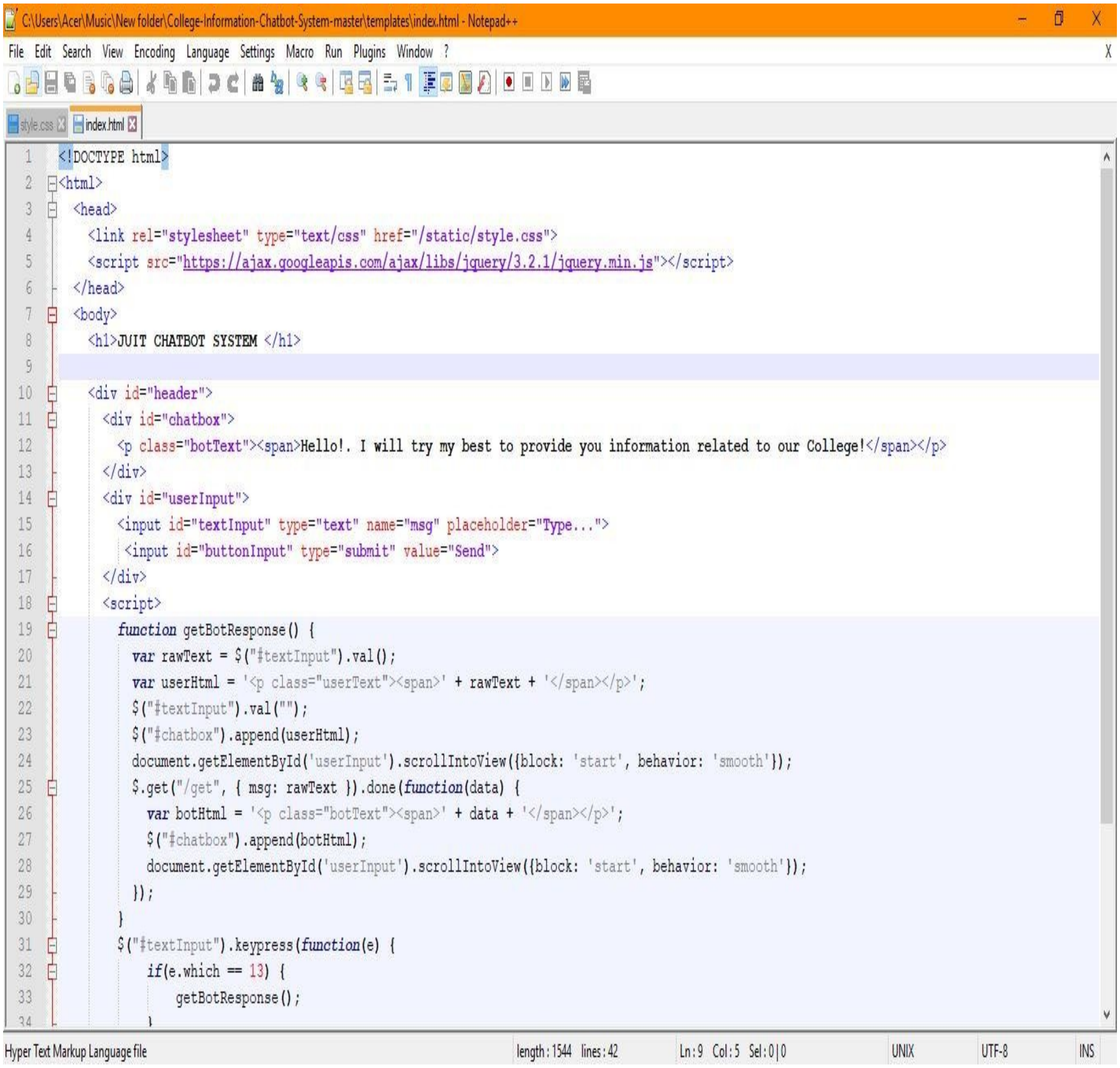
UI: The user interface is simple with not much colors. It is kept as simple as possible so as to make it look like a college chatbot. It consists of a text box at the bottom where the user may write the queries. A “send” button is placed in order to send the query to the bot. The UI is created using HTML, CSS and Javascript.

Background Working:

Parsing and Substitution: Whenever a user types a query, it is passes on to a class which parses the input and then substitutes words and phrases with other words and phrases so that a statement which is grammatically correct can be generated. It is carried out using XML and Python.

Natural Language Processing: NLP is required so that the data which is parsed can be “understood” by the application. For e.g. a user’s humour, feelings, names and places mentioned in the input. NLP is not implemented in the project but can be implemented if needed in the future.

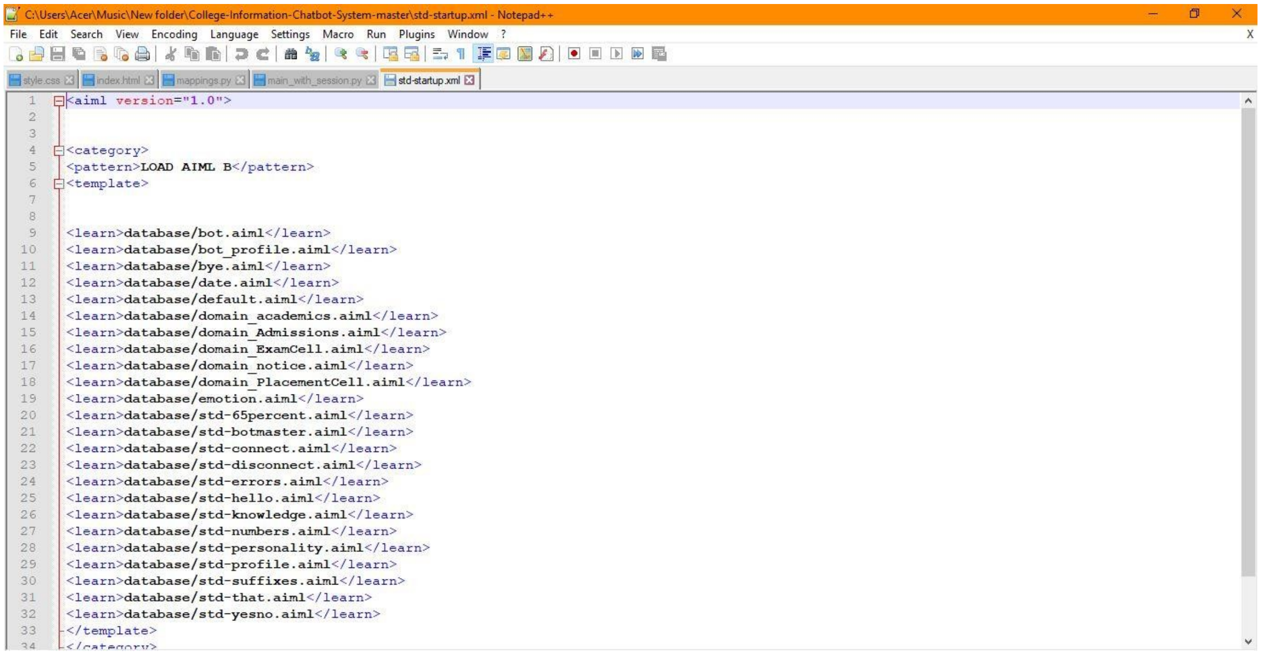
Database: There are various database files in the database folder with .aiml extension. These are the files which contain the various patterns of the conversations. Also a .db file is present which stores the name, e-mail and GPA of a student. It is stored offline because the college webkiosk’s API was not available. Also it demanded the implementation of JSON.


C:\Users\Acer\Music\New folder\College-Information-Chatbot-System-master\templates\index.html - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
style.css
index.html
Hyper Text Markup Language file
length : 1544 lines : 42
Ln : 9 Col : 5 Sel : 0 | 0
UNIX
UTF-8
INS

```
<!DOCTYPE html>
<html>
 <head>
  <link rel="stylesheet" type="text/css" href="/static/style.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
 </head>
 <body>
  <h1>JUIT CHATBOT SYSTEM </h1>

  <div id="header">
   <div id="chatbox">
    <p class="botText"><span>Hello!. I will try my best to provide you information related to our College!</span></p>
   </div>
   <div id="userInput">
    <input id="textInput" type="text" name="msg" placeholder="Type...">
     <input id="buttonInput" type="submit" value="Send">
   </div>
   <script>
    function getBotResponse() {
     var rawText = $("#textInput").val();
     var userHtml = '<p class="userText"><span>' + rawText + '</span></p>';
     $("#textInput").val("");
     $("#chatbox").append(userHtml);
     document.getElementById('userInput').scrollIntoView({block: 'start', behavior: 'smooth'});
     $.get("/get", { msg: rawText }).done(function(data) {
      var botHtml = '<p class="botText"><span>' + data + '</span></p>';
      $("#chatbox").append(botHtml);
      document.getElementById('userInput').scrollIntoView({block: 'start', behavior: 'smooth'});
     });
    }
    $("#textInput").keypress(function(e) {
      if(e.which == 13) {
        getBotResponse();
      }
```

Fig 1(a): The HTML file index.html

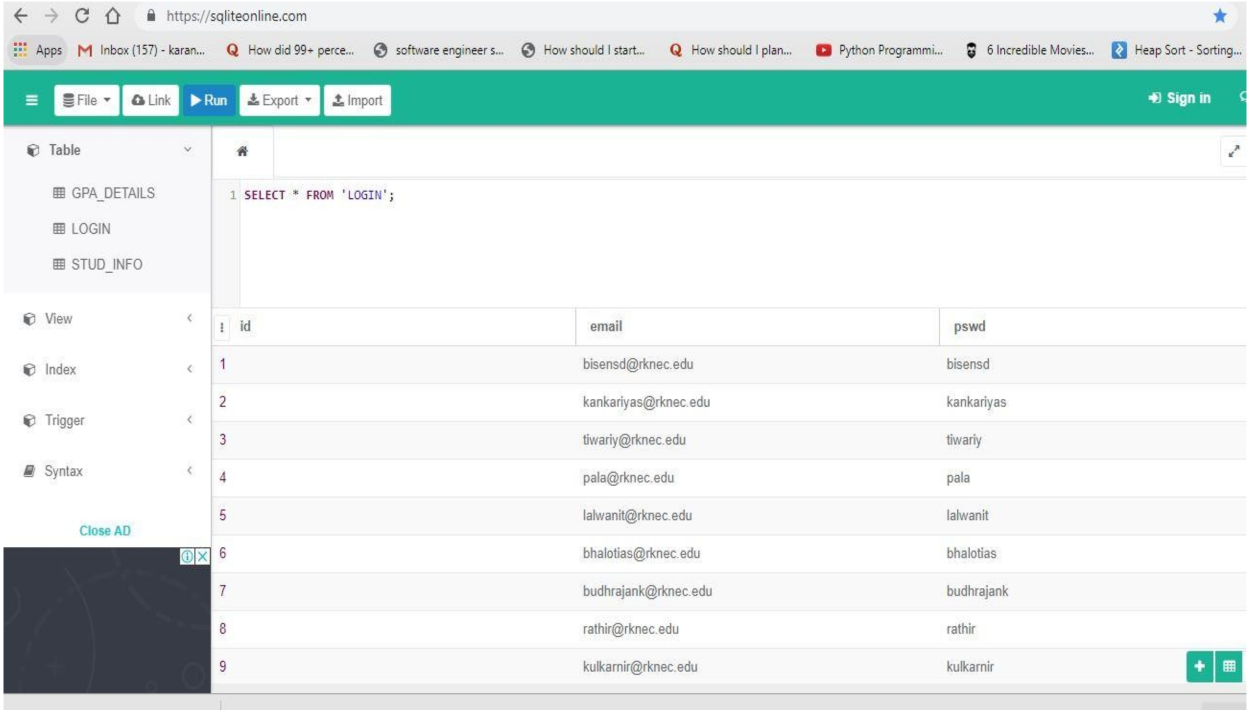

C:\Users\Acer\Music\New folder\College-Information-Chatbot-System-master\std-startup.xml - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
style.css
index.html
mappings.py
main_with_session.py
std-startup.xml

```
<aiml version="1.0">


<category>
<pattern>LOAD AIML B</pattern>
<template>


<learn>database/bot.aiml</learn>
<learn>database/bot_profile.aiml</learn>
<learn>database/bye.aiml</learn>
<learn>database/date.aiml</learn>
<learn>database/default.aiml</learn>
<learn>database/domain_academics.aiml</learn>
<learn>database/domain_Admissions.aiml</learn>
<learn>database/domain_ExamCell.aiml</learn>
<learn>database/domain_notice.aiml</learn>
<learn>database/domain_PlacementCell.aiml</learn>
<learn>database/emotion.aiml</learn>
<learn>database/std-65percent.aiml</learn>
<learn>database/std-botmaster.aiml</learn>
<learn>database/std-connect.aiml</learn>
<learn>database/std-disconnect.aiml</learn>
<learn>database/std-errors.aiml</learn>
<learn>database/std-hello.aiml</learn>
<learn>database/std-knowledge.aiml</learn>
<learn>database/std-numbers.aiml</learn>
<learn>database/std-personality.aiml</learn>
<learn>database/std-profile.aiml</learn>
<learn>database/std-suffixes.aiml</learn>
<learn>database/std-that.aiml</learn>
<learn>database/std-yesno.aiml</learn>
</template>
</category>
```

Fig 1(b): Startup file std-startup.xml



```
SELECT * FROM 'LOGIN';
```

id	email	pswd
1	bisensd@rknec.edu	bisensd
2	kankariyas@rknec.edu	kankariyas
3	tiwariy@rknec.edu	tiwariy
4	pala@rknec.edu	pala
5	lalwanit@rknec.edu	lalwanit
6	bhalotias@rknec.edu	bhalotias
7	budhrajank@rknec.edu	budhrajank
8	rathir@rknec.edu	rathir
9	kulkarnir@rknec.edu	kulkarnir

Fig 1(c): Database file contents opened in online browser

1.2 Eliza and the Doctor Script

Many programs and applications have been developed based on this concept. One of the first was Eliza. It is famous because it is an open source program.

ELIZA makes the use of primitive natural language processing. It operates on user's responses to scripts, in which the most famous was the DOCTOR, a simulation of psychotherapist. It provided a basic human-like interaction with almost no information about human thought or emotion. It was written at MIT between 1964 and 1966.

A simple conversation example:
User: My head hurts.
DOCTOR: Why do you say your head hurts?
User: My mother hates me.
DOCTOR: Who else in the family hates you?

ELIZA uses simple pattern matching techniques, but was in talks when the creator explained how it actually works. It was one of the first chatterbots in existence. It comprised of many scripts, each of which processed the responses of the users by simple parsing and by the substitution of them into premade templates and also there were hardcore phrases to as to continue conversations.

The creator of ELIZA never expected to create a fully working model consisting of NLP, it was rather a simple toy. What DOCTOR actually did was that it used a lot of data from the statements from the humans in order to compose the responses by using simple templates. It actually used an if-else pattern of using a certain template for a certain trigger. It was first implemented on SLIP language (an extension to FORTRAN).

ELIZA was important in revolutionizing the computing industry because such an idea of interaction was not a popular fancy.

A simple conversation transcript of ELIZA would look like this:

I wish to know about you.

Why do you say i might want to know
me? To write my report
What makes you believe to write your report?

I do not understand your question, please clarify
Earlier you said i would like to know me?
Yes

Can you elaborate on that?
I wish to know ELIZA
How does it feel to want?
Want what?
Maybe your plans have something to do with
this. Er, what plans?
Is it because of your life that you say

I would appreciate it if you would
continue. I will.

As it concluded here that ELIZA has a habit of parsing the responses of the user to frame the next question. Also after the conversation is continued these responses become predictable. Same is the case with the ChatBot created here. Since the data of templates is not much, the responses after some time become predictable.

1.3 Cleverbot

Cleverbot is also a web application which converses with humans and uses artificial intelligence algorithms. It was created be British scientist Rollo Carpenter. The difference between Eliza and Cleverbot is that its unique algorithm learns from humans and remembers words within its AI.

There is also a point to note that the bot's responses are programme. It learns from human input; The UI of my project is somewhat similar to Cleverbot where the human types into the box below the logo of Cleverbot and finds all the matching keywords or phrases matching the input. It responds to the input of the human by searching its saved conversations. It also responds to the particular input by finding how a human responded to that input when it was asked, but thtat happens in part or in full.

Because of its complex algorithm structure, it is constantly learning. It's data size is also increasing. Due to this it appears to display a degree of "intelligence". It's software updates are constantly checked and in 2014 it was upgraded to use GPU serving techniques.

Cleverbot also participated in a Turing Test in 2011 organised by IIT Guwahati. Cleverbot was judged to be 59.3% human. The software which participated in the test had to process 1 or 2 simultaneous requests, but it was noted that Cleverbot handled 80000 people at once.

1.4 Problems with Cleverbot

Despite the complex algorithm of CleverBot some problems were noted:

It doesn’t take anyone too many sentences for it to fail a Turing test, and figure out it was saying things that had been said to it many times.

It can't reference back than a single sentence.
It has no core identity.
It often answers a "why" question with a "where" answer.

The reason for this is that the CleverBot stores the responses that people give to it in return for that it says. It is also noted that it simply says the answers to someone else and records their answer.

The condition for a machine to hold a proper conversation with anyone is that it would need to design a system that "understands" what its hearing and saying properly so that it can hold a co-operative conversation. Making that happen is hard. The greatest obstacle to it is unraveling the jumbled up mess of complexity that words and ideas are made up of.

1.5 AIML

In Python, using AIML package, artificial intelligence ChatBots are easy to write. Its stands for Artificial Intelligence Markup Language. Basically it is just simple XML.

It was developed by Richard Wallace. He made a bot named ALICE. AIML is a form of XML that helps to define rules for patter matching and determining responses.

There are some phases to implement AIML which are used in this project:

# Create a standard Startup File
# Creating an AIML File

# Installing Python AIML Module
# Creating a Python Program

Some of the important tags used in AIML documents are:

# <aiml> - It defines the beginning and end of the AIML document.

# <category> -It defines the basic knowledge in Chatbot's knowledge base.
# <pattern> - It defines the pattern to match the user's input to the ChatBot.
# <template> - defines the response of the ChatBot to user's input

One of the very important questions that arises during the implemtation of AIML is that whether it counts as Artificial Intelligence or not.

A simple answer to this is that it more like an impression of intelligence rather than actual intelligence. It involves basic scripting which is done in XML and there are no learning models.

1.6 NLP

NLP is a part of artificial intelligence which deals with human languages. It has the following structure:

# Application
# NLP layer
# Knowledge Base
# Data Storage

NLP is divided into two very important components:

# **Natural Language Understanding**: It is mostly used to map inputs to useful representations. It is also helpful in analyzing different aspects of the language.

# **Natural Language Generation**: It is generally used text planning, sentence planning, and text realization.

NLP is implemented using a library in Python named NLTK.
There are some steps followed in NLP:

# **Tokenization**: It is the process to break a complex sentence into words. Also, the importance of each word is understood with respect to the sentence. It also helps to produce a structural description on an input sentence.

# **Stemming**: It is the process in which words are normalized into its base form or root form.

# **Lemmatization**: It is the process in which grouping of different inflected forms of a word is done. It also roots several words into one common root but the output of Lemmatization is a proper word.

# **Stop Words**: These are some of the words which are helpful to make a sentence meaningful but do not help in NLP.

# **Parts of Speech**: It is an inbuilt library containing the various parts of speech. For e.g.:

CD-
Cardinal
Number #
NN- Noun
Singular
# NNS- Noun Plural; etc.

# **Named Entity Recognition**: It helps to identify the particular entity name. For e.g. movie, monetary value, organization, location or quantities.

# Chunking: It is a process of picking up individual pieces of information and grouping them into bigger pieces.

Since NLP is a little hard topic to grasp, a smaller demo is implemented using the above processes. I have implemented NLP on Anaconda and the above processes have been implemented using various built-in functions.

However, all the above steps which make NLP complete are not implemented in this project structure. It might be a future task.

CHAPTER 3

SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in systems engineering, information systems and software engineering, is the process of creating or altering systems, and the models and methodologies that people use to develop these systems. In software engineering the SDLC concept underpins many kinds of software development methodologies.

3.1 EXISTING SYSTEM

In the earlier days students had to visit the college to enquire about details like courses, fee structure, admission process as well as long process for both parents as well as students. Now a days there are many changes occurred in the education system with help of advanced technology. Everything is happening over the internet without any trouble. In those days for enquiring about courses we have to visit the college, but as the days are passing away its completely changing. Collecting the course details, fee structure manually will be a big procedure and it also needs a manpower. For reducing that manpower and avoid such difficulties and time consuming many devices or systems were emerged day by day

3.1.1 DISADVANTAGES OF EXISTING SYSTEM

- More time consuming
- Delay in response
- In the existing system we have only limited number of predefined queries.
- It cannot understand specific problems and cannot perform task for the client.

3.2 PROPOSED SYSTEM

The objective of this application is to propose a chatbot enquiry for students to communicate with the colleges. By using artificial intelligence, the system answers the queries asked by the students. The chatbot mainly consists of core and interface, where it mainly access the core in Natural language processing technologies are here used for parsing, tokenizing, stemming and filtering the content of the complaint.

To further develop the proposed system- college chatbot we can use any programming language that supports object-oriented concept, but we use python as it is the most happening language and user friendly. Software we use python compiler. we develop the artificial neural network algorithm in the python language on python compiler. And further integrate it with database using python compiler..

3.2.1 ADVANTAGES OF PROPOSED SYSTEM

- Faster processing when compared to existing one.
- It takes less time to respond.
- It gives response in the form of queries rather than options.
- It provides 24/7 service

CHAPTER 4
IMPLEMENTATION

4. IMPLEMENTATION

What is Python :-

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

Advantages of Python :-

Let’s see how Python dominates over other languages.

1. Extensive Libraries

Python downloads with an extensive library and it *contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more.* So, we don’t have to write the complete code for that manually.

2. Extensible

As we have seen earlier, Python can be **extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities** to our code in the other language.

4. Improved Productivity

The language's simplicity and extensive libraries render programmers **more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

6. Simple and Easy

When working with Java, you may have to create a class to print **'Hello World'**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and **code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

8. Object-Oriented

This language supports both the **procedural and object-oriented** programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

9. Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you **download Python** for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to **code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

Advantages of Python Over Other Languages

1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.

3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and **machine learning**, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

Disadvantages of Python

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

1. Speed Limitations

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in **slow execution**. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

The reason it is not so famous despite the existence of Brython is that it isn't that secure.

3. Design Restrictions

As you know, Python is **dynamically-typed**. This means that you don't need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what's that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can **raise run-time errors**.

4. Underdeveloped Database Access Layers

Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python's database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

5. Simple

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

History of Python : -

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python.Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners[1], Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

What is Machine Learning : -

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of *building models of data*.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models *tunable parameters* that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain.Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

Categories Of Machine Leaning :-

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into *classification* tasks and *regression* tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as *clustering* and *dimensionality reduction.* Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven't surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, "to make decisions, based on data, with efficiency and scale".

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can't do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

Challenges in Machines Learning :-

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

Quality of data − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

Time-Consuming task − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

Lack of specialist persons − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

No clear objective for formulating business problems − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

Issue of overfitting & underfitting − If the model is overfitting or underfitting, it cannot be represented well for the problem.

Curse of dimensionality − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

Difficulty in deployment − Complexity of the ML model makes it quite difficult to be deployed in real life.

Applications of Machines Learning :-

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

- Emotion analysis
- Sentiment analysis
- Error detection and prevention
- Weather forecasting and prediction
- Stock market analysis and forecasting
- Speech synthesis
- Speech recognition
- Customer segmentation
- Object recognition
- Fraud detection
- Fraud prevention
- Recommendation of products to customer in online shopping

How to Start Learning Machine Learning?

Arthur Samuel coined the term **"Machine Learning"** in 1959 and defined it as a **"Field of study that gives computers the capability to learn without being explicitly programmed".**

And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to Indeed, Machine Learning Engineer Is The Best Job of 2019 with a *344%* growth and an average base salary of **$146,085** per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let's get started!!!

How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don't know these, never fear! You don't need a Ph.D. degree in these topics to get started but you do need a basic understanding.

(a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

(b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!!
Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

(c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as Keras, TensorFlow, Scikit-learn, etc.

So if you want to learn ML, it's best if you learn Python! You can do that using various online resources and courses such as **Fork Python** available Free on GeeksforGeeks.

Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It's best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

(a) Terminologies of Machine Learning

- **Model –** A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
- **Feature –** A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
- **Target (Label) –** A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
- **Training –** The idea is to give a set of inputs(features) and it's expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
- **Prediction –** Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

(b) Types of Machine Learning

- **Supervised Learning –** This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.

- **Unsupervised Learning –** This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
- **Semi-supervised Learning –** This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
- **Reinforcement Learning –** This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

Advantages of Machine learning :-

1. Easily identifies trends and patterns -

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

2. No human intervention needed (automation)

With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

3. Continuous Improvement

As **ML algorithms** gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

Disadvantages of Machine Learning :-

1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

4. High error-susceptibility

Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

Python Development Steps : -

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system.
Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked.Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a

full garbage collector and it was supporting unicode.Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

- Print is now a function
- Views and iterators instead of lists
- The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
- There is only one integer type left, i.e. int. long is int as well.
- The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
- Text Vs. Data Instead Of Unicode Vs. 8-bit

Purpose :-

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Modules Used in Project :-

Tensorflow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

Numpy

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. **Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Install Python Step-by-Step in Windows and Mac :

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

How to Install Python on Windows and Mac :

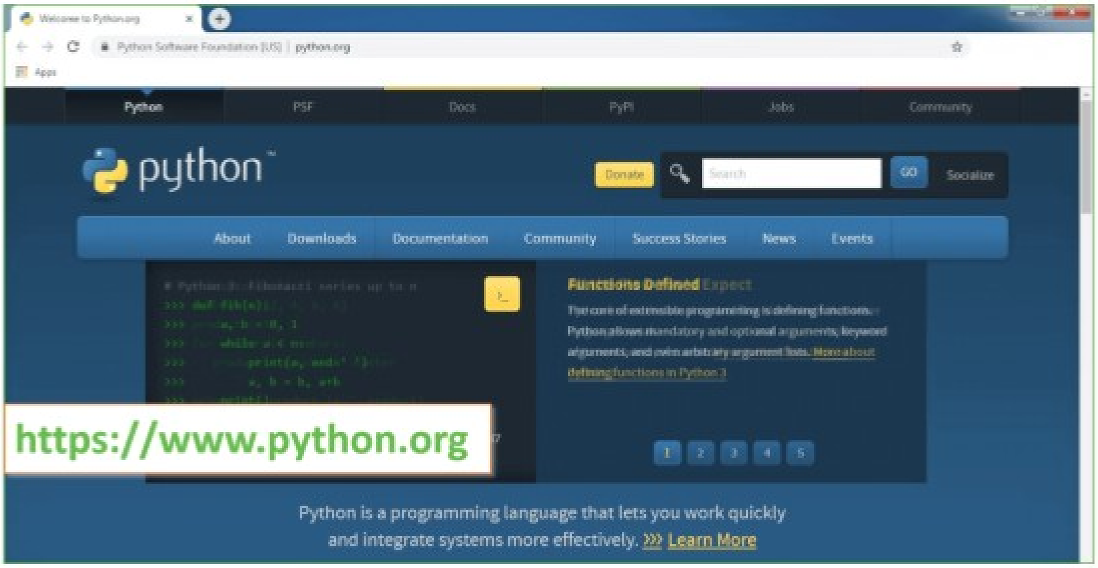
There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheatsheet here.The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

Download the Correct version into the system

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: **https://www.python.org**



```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
```

Functions Defined

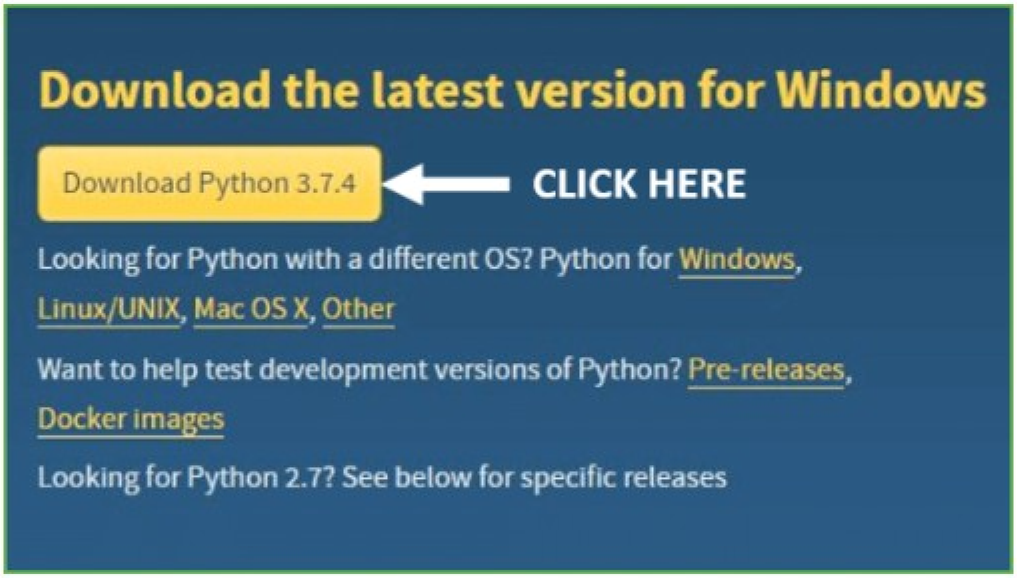
The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. More about defining functions in Python 3

https://www.python.org

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.



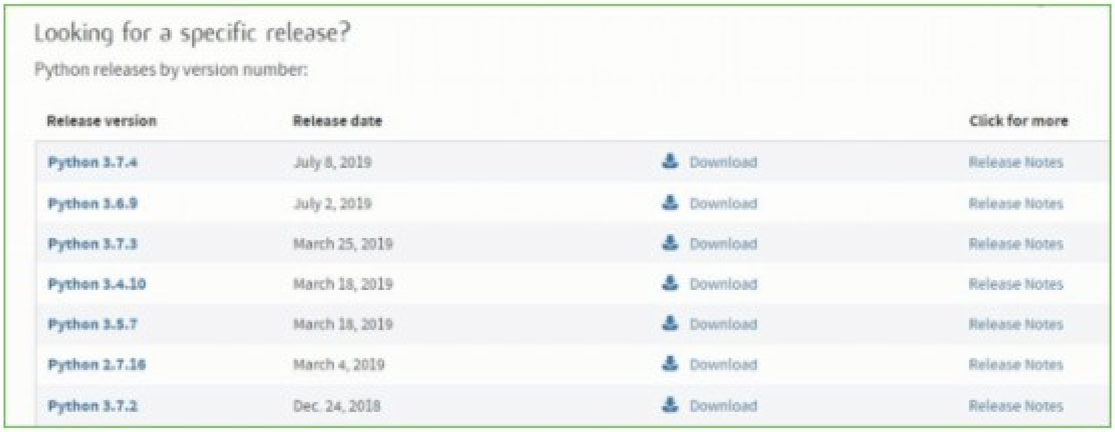
Download the latest version for Windows

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Pre-releases, Docker images

Looking for Python 2.7? See below for specific releases

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4



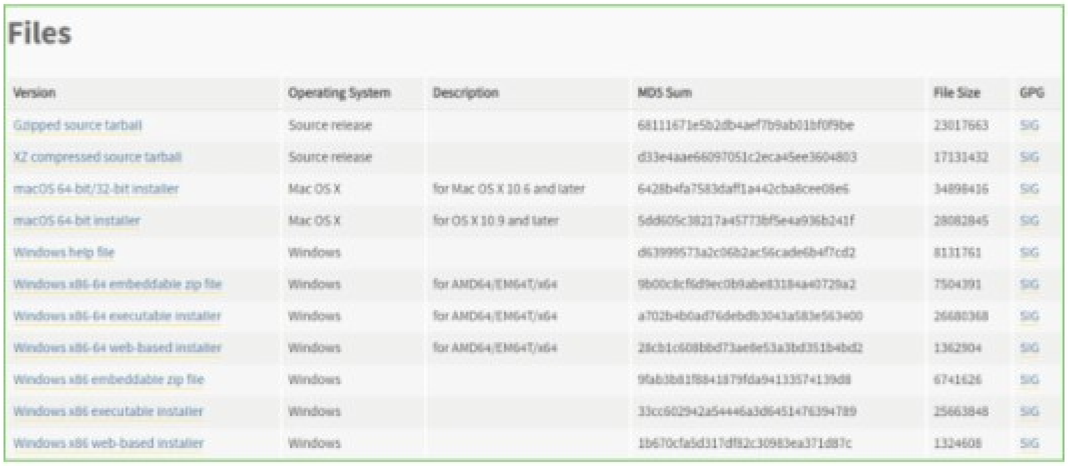
Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.



Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e5b2db4aef7b9ab01bf0f9be	23017663	SIG
XZ compressed source tarball	Source release		d33e4aae66097051c2eca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daff1a442cba8cee08e6	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b241f	28082845	SIG
Windows help file	Windows		d63999573a2c06b2ac56cade6b4f7cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b00c8cf6d9ec0b9abe83184a40729a2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0ad76debdb3043a583e563400	26680368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28cb1c608bbd73ae8e53a3bd351b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab3b81f8841879fda94133574139d8	6741626	SIG
Windows x86 executable installer	Windows		33cc602942a54446a3d6451476394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa5d317df82c30983ea371d87c	1324608	SIG

- To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

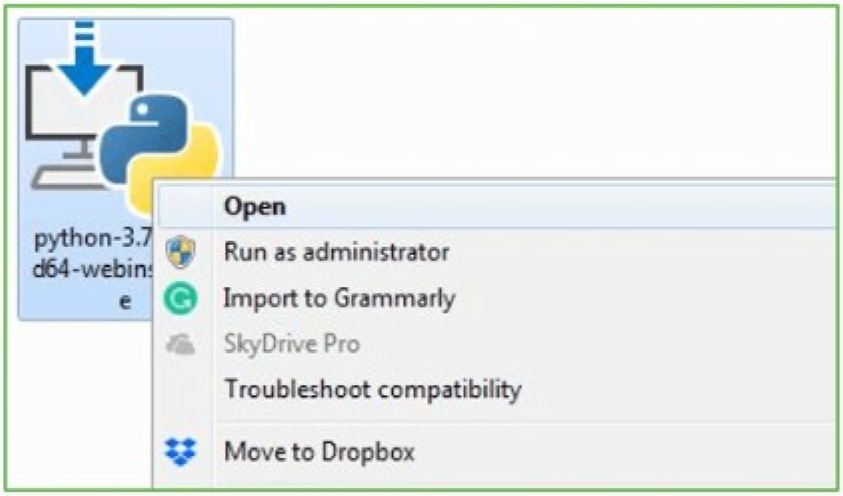
•To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

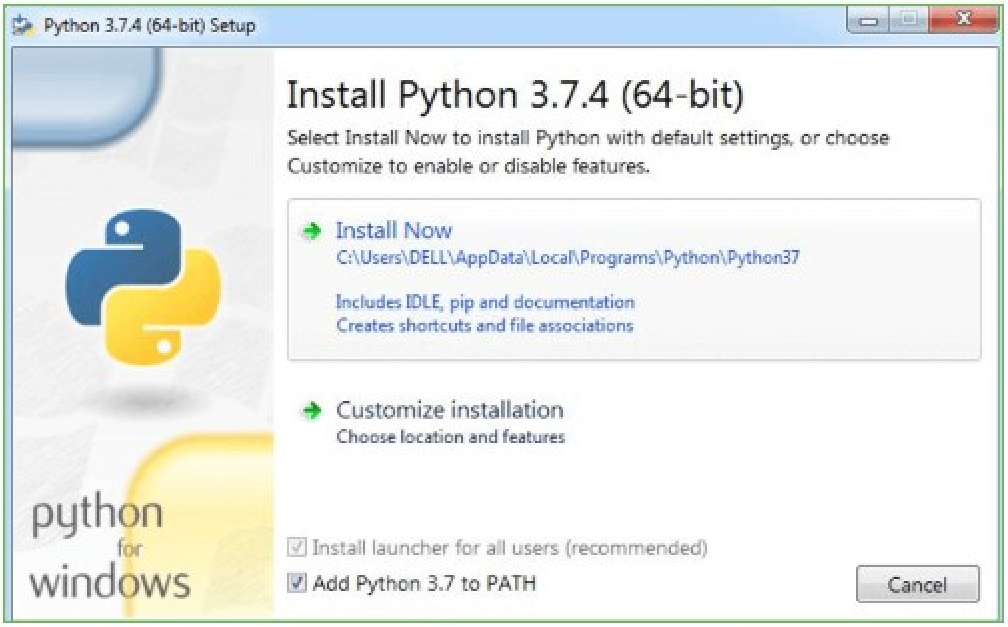
Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.



Step 2: Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.



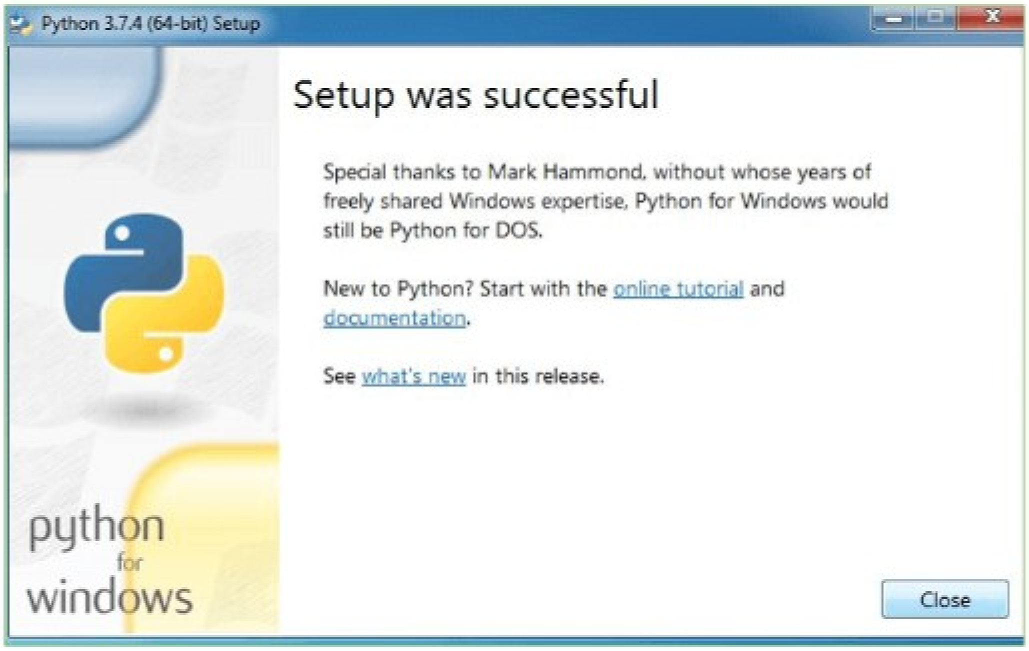
Python 3.7.4 (64-bit) Setup

Install Python 3.7.4 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\DELL\AppData\Local\Programs\Python\Python37
Includes IDLE, pip and documentation
Creates shortcuts and file associations
Customize installation
Choose location and features
☑ Install launcher for all users (recommended)
☑ Add Python 3.7 to PATH
Cancel
python for windows

Step 3: Click on Install NOW After the installation is successful. Click on Close.



Python 3.7.4 (64-bit) Setup

Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

See what's new in this release.

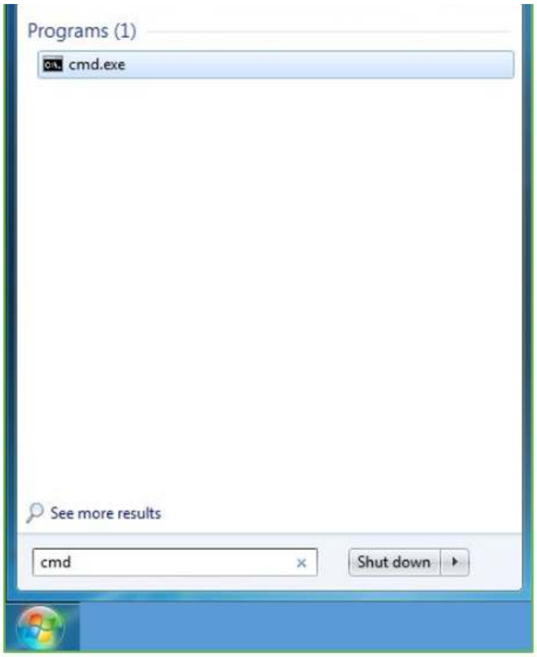
With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the Python Installation

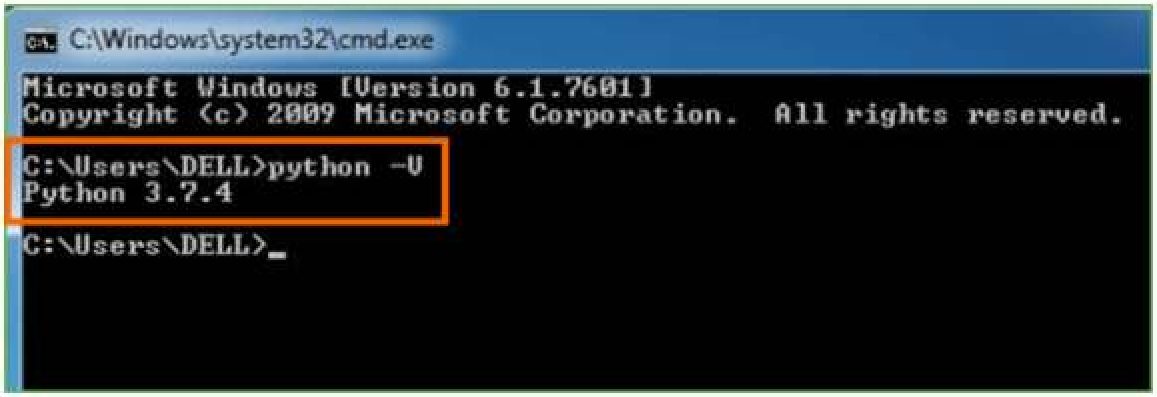
Step 1: Click on Start

Step 2: In the Windows Run Command, type "cmd".



Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type **python –V** and press Enter.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\DELL>python -V
Python 3.7.4

C:\Users\DELL>_
```

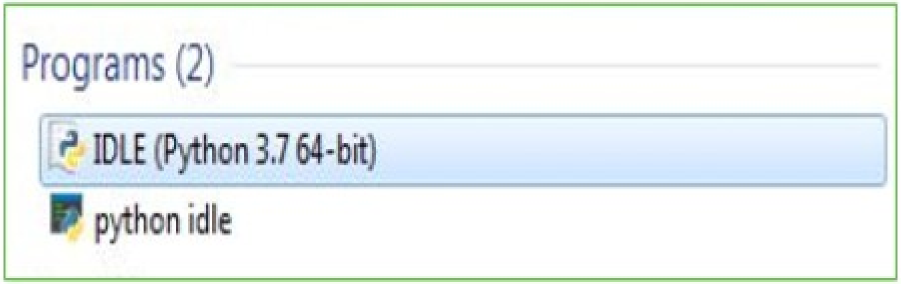
Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works

Step 1: Click on Start

Step 2: In the Windows Run command, type "python idle".



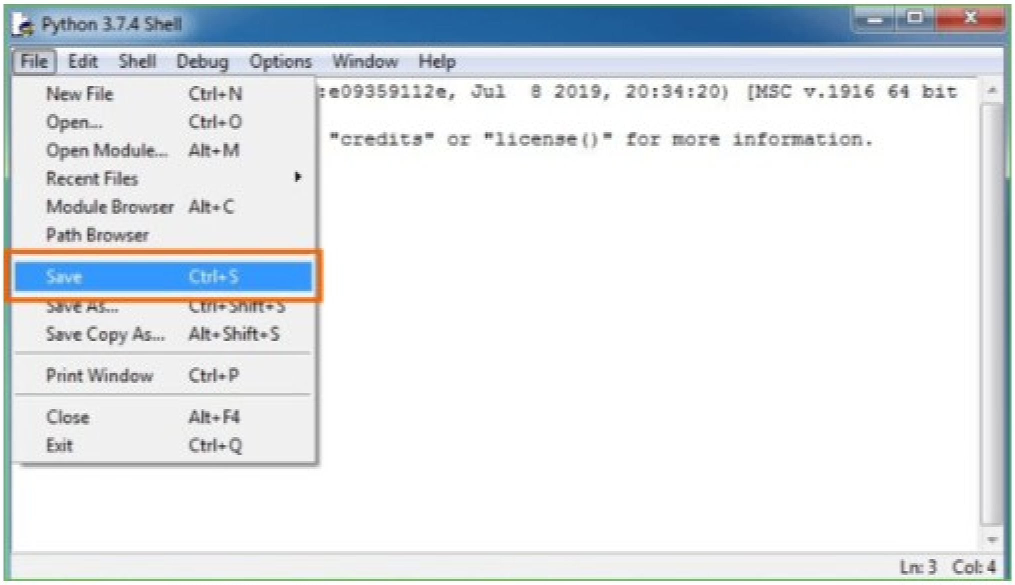
Programs (2)

IDLE (Python 3.7 64-bit)

python idle

Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

Step 4: To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



```
:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit

 "credits" or "license()" for more information.
```

Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g. **enter print**

CHAPTER 5

SOFTWARE REQUIREMENT SPECIFICATION

5. SOFTWARE REQUIREMENT SPECIFICATION

5.1 Requirements Specification:

Requirement Specification provides a high secure storage to the web server efficiently. Software requirements deal with software and hardware resources that need to be installed on a serve which provides optimal functioning for the application. These software and hardware requirements need to be installed before the packages are installed. These are the most common set of requirements defined by any operation system. These software and hardware requirements provide a compatible support to the operation system in developing an application.

5.1.1 HARDWARE REQUIREMENTS:

The hardware requirement specifies each interface of the software elements and the hardware elements of the system. These hardware requirements include configuration characteristics.

- ➢ System : Pentium IV 2.4 GHz.
- ➢ Hard Disk : 100 GB.
- ➢ Monitor : 15 VGA Color.
- ➢ Mouse : Logitech.
- ➢ RAM : 1 GB.

5.1.2 SOFTWARE REQUIREMENTS:

The software requirements specify the use of all required software products like data management system. The required software product specifies the numbers and version. Each interface specifies the purpose of the interfacing software as related to this software product.

- Operating system : Windows XP/7/10
- Coding Language: Python 3.7

5.2 FUNCTIONAL REQUIREMENTS:

The functional requirement refers to the system needs in an exceedingly computer code engineering method.

The key goal of determinant "functional requirements" in an exceedingly product style and implementation is to capture the desired behavior of a software package in terms of practicality and also the technology implementation of the business processes.

1.Load Dataset:

Load data set using pandas read_csv() method.

2.Split Data Set:

Split the data set to two types. One is train data test and another one is test data set.

3.Train data set:

Train data set will train our data set using fit method.

4.Test data set:

Test data set will test the data set using algorithm.

5.Predict data set:

Predict() method will predict the results.

5.3 NON FUNCTIONAL REQUIREMENTS

All the other requirements which do not form a part of the above specification are categorized as Non-Functional needs. A system perhaps needed to gift the user with a show of the quantity of records during info. If the quantity must be updated in real time, the system architects should make sure that the system is capable of change the displayed record count at intervals associate tolerably short interval of the quantity of records dynamic. Comfortable network information measure may additionally be a non-functional requirement of a system.

The following are the features:

- Accessibility

- Availability
- Backup
- Certification
- Compliance
- Configuration Management
- Documentation
- Disaster Recovery
- Efficiency(resource consumption for given load)
- Interoperability

5.4 PERFORMANCE REQUIREMENTS

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.
The requirement specification for any system can be broadly stated as given below:

- The system should be able to interface with the existing system
- The system should be accurate
- The system should be better than the existing system

The existing system is completely dependent on the user to perform all the duties.

5.5 Feasibility Study:

Preliminary investigation examines project feasibility; the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All systems are feasible if they are given unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operation Feasibility

Economical Feasibility

5.5.1 Technical Feasibility

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipments have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?

Are there technical guarantees of accuracy, reliability, ease of access and data security?

5.5.2 Operational Feasibility

User-friendly

Customer will use the forms for their various transactions i.e. for adding new routes, viewing the routes details. Also the Customer wants the reports to view the various transactions based on the constraints. These forms and reports are generated as user-friendly to the Client.

Reliability

The package wills pick-up current transactions on line. Regarding the old transactions, User will enter them in to the system.

Security

The web server and database server should be protected from hacking, virus etc

Portability

The application will be developed using standard open source software (Except Oracle) like Java, tomcat web server, Internet Explorer Browser etc these software will work both on Windows and Linux o/s. Hence portability problems will not arise.

Availability

This software will be available always.

Maintainability

The system uses the 2-tier architecture. The 1st tier is the GUI, which is said to be front-end and the 2nd tier is the database, which uses My-Sql, which is the back-end.

The front-end can be run on different systems (clients). The database will be running at the server. Users access these forms by using the user-ids and the passwords.

5.5.3 Economic Feasibility

The computerized system takes care of the present existing system's data flow and procedures completely and should generate all the reports of the manual system besides a host of other management reports.

It should be built as a web based application with separate web server and database server. This is required as the activities are spread throughout the organization customer wants a centralized database. Further some of the linked transactions take place in different locations.

CHAPTER 6

METHODOLOGY

6. Methodology

SDLC (Software Development Life Cycle) – Umbrella Model

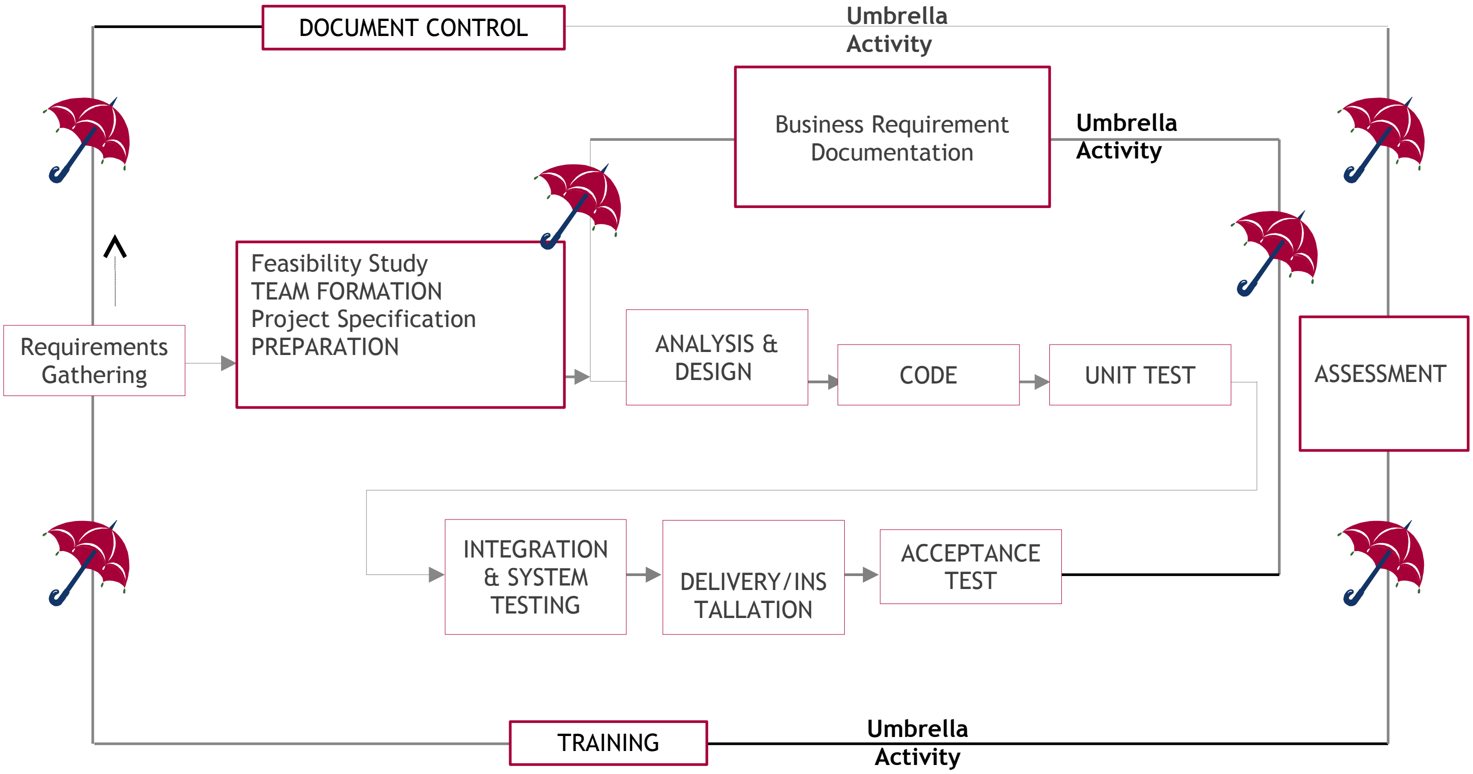


Fig no. 6.1 Umbrella model

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

Requirements Gathering Stage

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.

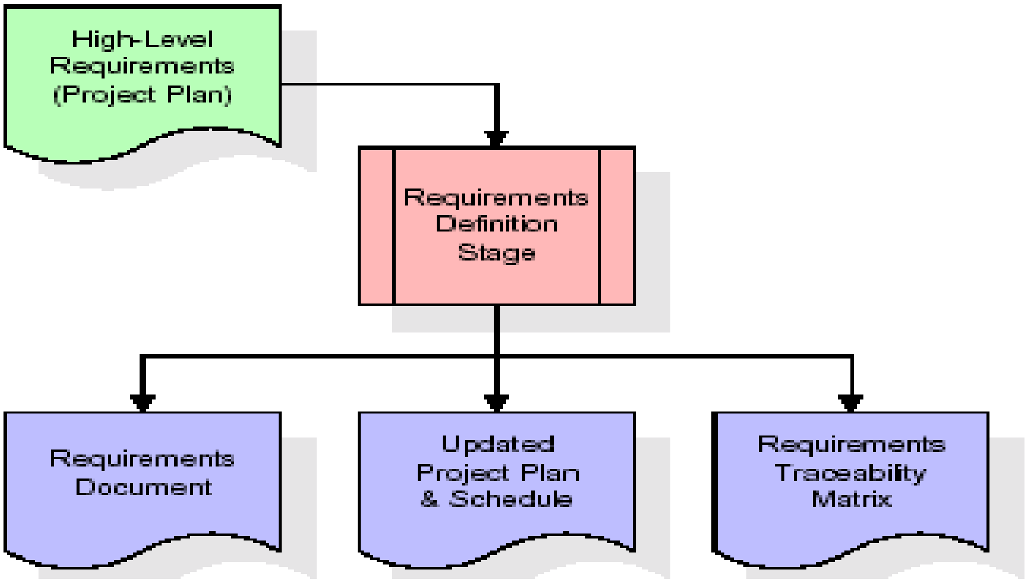

High-Level Requirements (Project Plan)
Requirements Definition Stage
Requirements Document
Updated Project Plan & Schedule
Requirements Traceability Matrix

Fig no. 6.2 Requirements Gathering stage

These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are not included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

Feasibility study is all about identification of problems in a project, number of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.

Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

Analysis Stage

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.

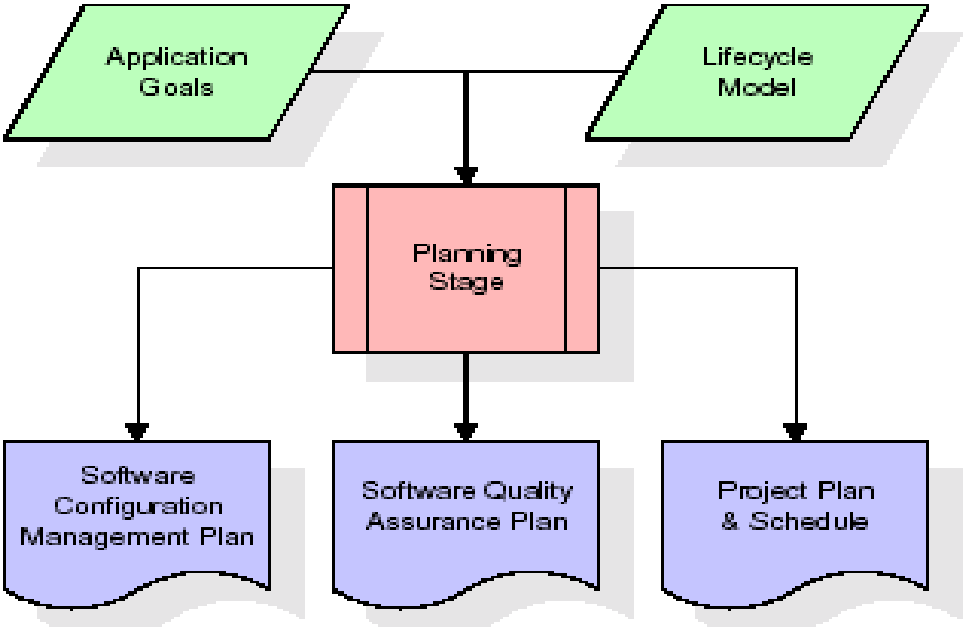

Application Goals
Lifecycle Model
Planning Stage
Software Configuration Management Plan
Software Quality Assurance Plan
Project Plan & Schedule

Fig no. 6.3 Analysis stage

The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

Designing Stage

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

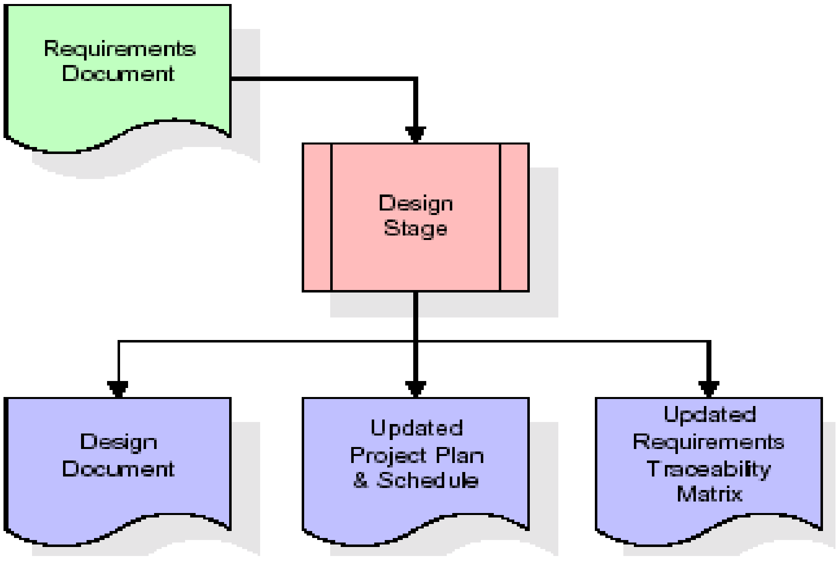

Requirements Document
Design Stage
Design Document
Updated Project Plan & Schedule
Updated Requirements Traceabilty Matrix

Fig no. 6.4 **Designing stage**

When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

Development (Coding) Stage

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.

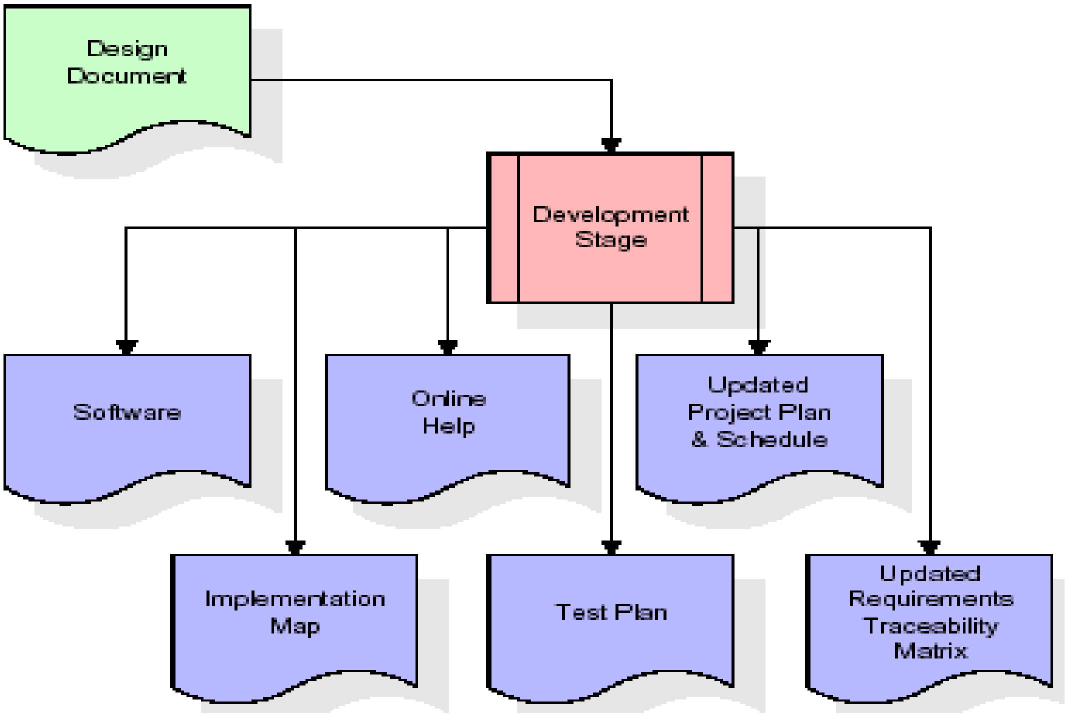
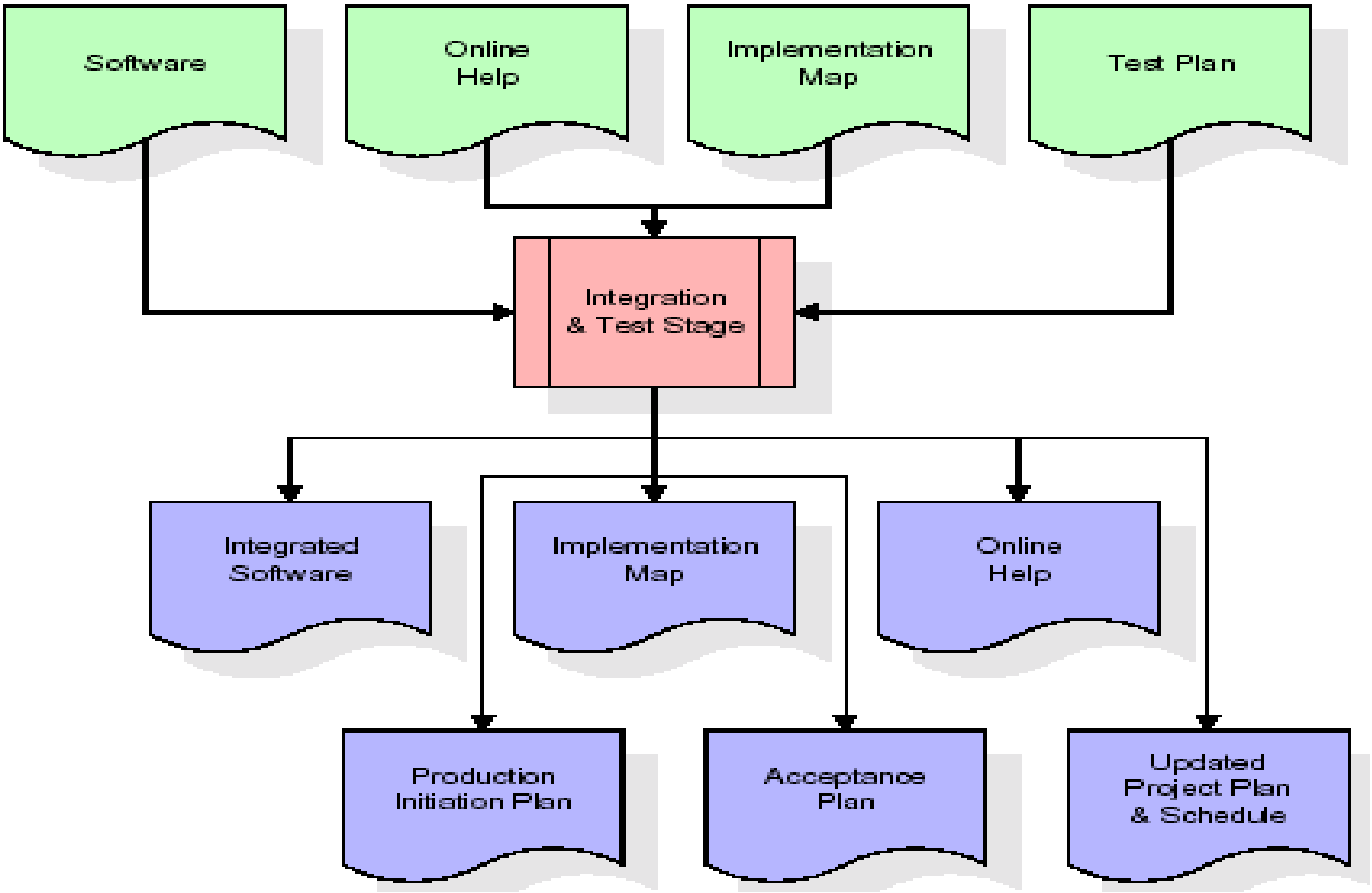


Fig no. 6.5 Coding stage

Integration & Test Stage

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

Fig no. 6.6 Integration and Testing Stage



Installation & Acceptance Test

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.

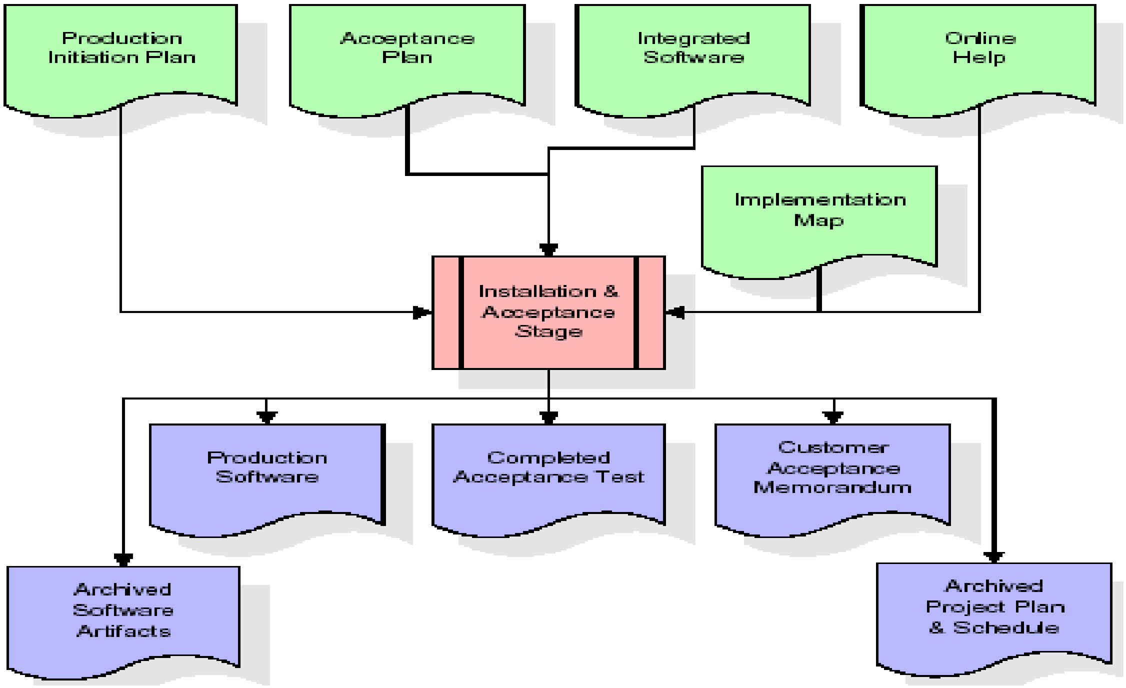

Production Initiation Plan
Acceptance Plan
Integrated Software
Online Help
Implementation Map
Installation & Acceptance Stage
Production Software
Completed Acceptance Test
Customer Acceptance Memorandum
Archived Software Artifacts
Archived Project Plan & Schedule

Fig no. 6.7 Installation

Maintenance

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category.

CHAPTER 7

SYSTEM DESIGN & UML DESIGN

7. System Design

7.1 SYSTEM ARCHITECTURE

The purpose of the design phase is to arrange an answer of the matter such as by the necessity document. This part is that the opening moves in moving the matter domain to the answer domain. The design phase satisfies the requirements of the system. The design of a system is probably the foremost crucial issue warm heartedness the standard of the software package. It's a serious impact on the later part, notably testing and maintenance.

The output of this part is that the style of the document. This document is analogous to a blueprint of answer and is employed later throughout implementation, testing and maintenance. The design activity is commonly divided into 2 separate phases System Design and Detailed Design.

System Design conjointly referred to as top-ranking style aims to spot the modules that ought to be within the system, the specifications of those modules, and the way them move with one another to supply the specified results.

At the top of the system style all the main knowledge structures, file formats, output formats, and also the major modules within the system and their specifications square measure set. System design is that the method or art of process the design, components, modules, interfaces, and knowledge for a system to satisfy such as needs. Users will read it because the application of systems theory to development.

Detailed Design, the inner logic of every of the modules laid out in system design is determined. Throughout this part, the small print of the info of a module square measure sometimes laid out in a high-level style description language that is freelance of the target language within which the software package can eventually be enforced.

In system design the main target is on distinguishing the modules, whereas throughout careful style the main target is on planning the logic for every of the modules.

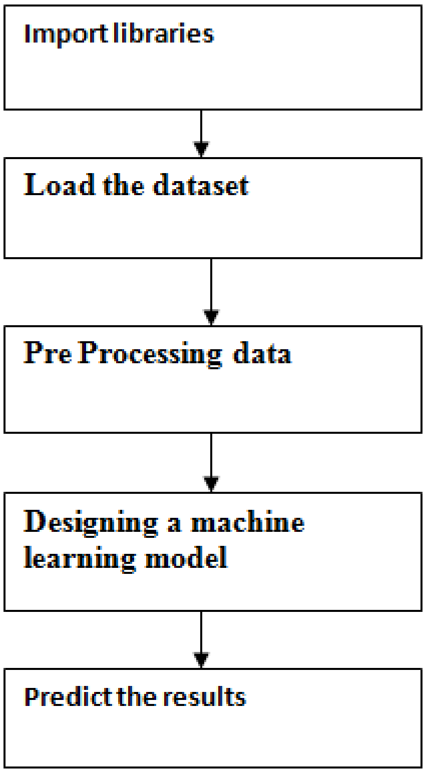


Figure 7.1: Architecture diagram

7.3 UML DIAGRAMS

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

User Model View

This view represents the system from the user's perspective. The analysis representation describes a usage scenario from the end-users perspective.

Structural Model view

In this model the data and functionality are arrived from inside the system. This model view models the static structures.

Behavioral Model View

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

Implementation Model View

In this the structural and behavioral as parts of the system are represented as they are to be built.

5.3.1 USE CASE DIAGRAM

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

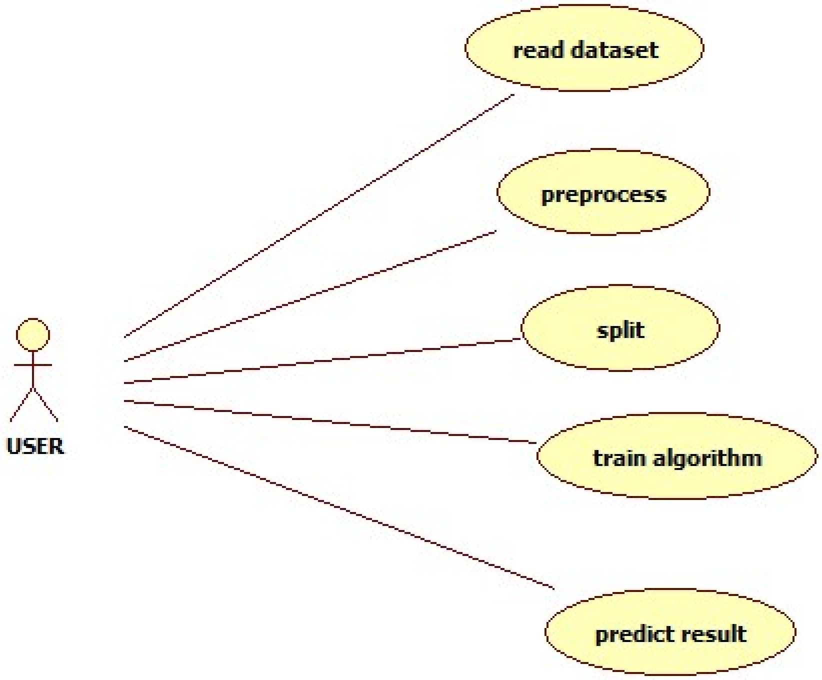


Figure 7.3.1 Use Case Diagram

5.3.2 SEQUENCEDIAGRAM

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

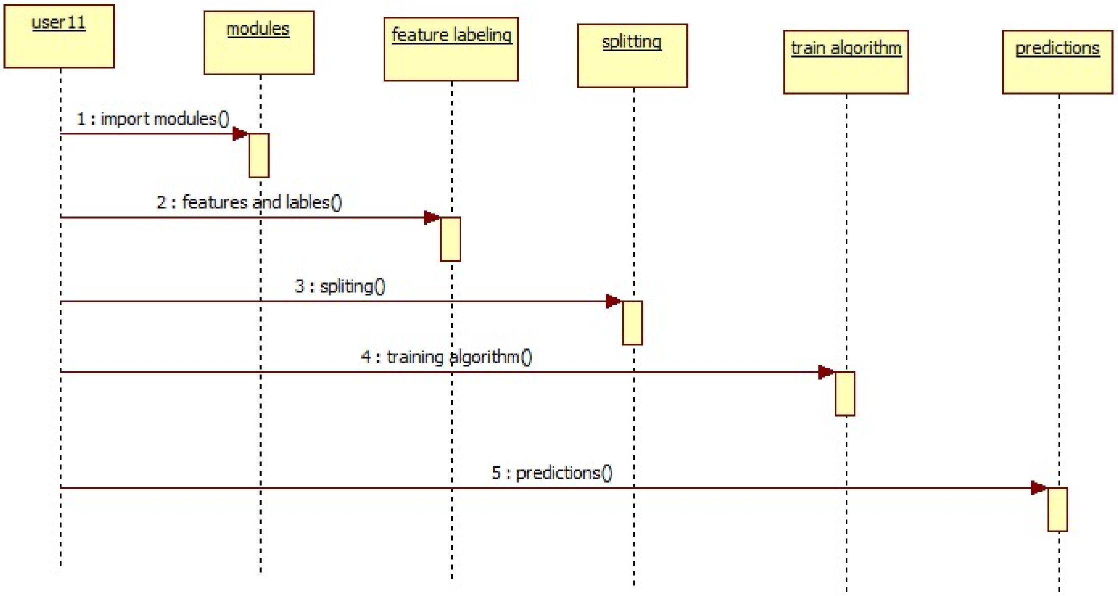


Figure 7.3.2: Sequence diagram

5.3.3 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

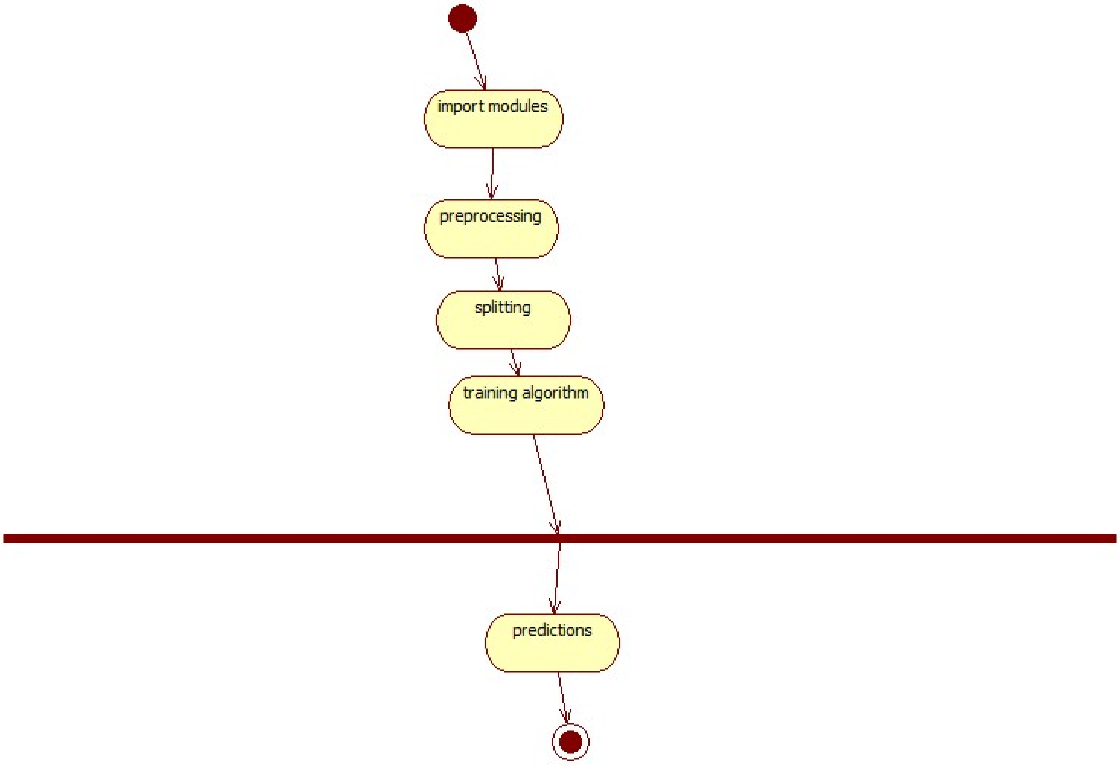


Figure 7.3.3: Activity Diagram

CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

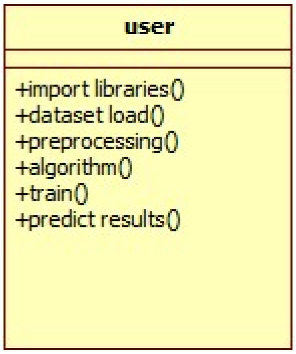


Figure 7.3.4: Class Diagram

CHAPTER 8

SYSTEM TESTING

8. TESTING

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. The following is the description of the testing strategies, which were carried out during the testing period.

8.1 SYSTEM TESTING

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

8.2 MODULE TESTING

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

8.3 INTEGRATION TESTING

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system

8.4 ACCEPTANCE TESTING

When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

8.5 TEST CASES:

Test Case Id	Test Case Name	Test Case Desc.	Test Steps			Test Case Status	Test Priority
			Step	Expected	Actual		
01	Upload the tasks dataset	Verify either file is loaded	If dataset is not uploaded	It cannot display the file	File is loaded which	High	High

		or not		loaded message	displays task waiting time		
02	Upload patients dataset	Verify either dataset loaded or not	If dataset is not uploaded	It cannot display dataset reading process completed	It can display dataset reading process completed	low	High
03	Preprocessing	Whether preprocessing on the dataset applied or not	If not applied	It cannot display the necessary data for further process	It can display the necessary data for further process	Medium	High
04	Prediction Random Forest	Whether Prediction algorithm applied on the data or not	If not applied	Random tree is not generated	Random tree is generated	High	High
05	Recommendation	Whether predicted data is displayed or not	If not displayed	It cannot view prediction containing patient data	It can view prediction containing patient data	High	High
06	Noisy	Whether	If graph is	It does	It shows	Low	Mediu

	Records Chart	the graph is displayed or not	not displayed	not show the variations in between clean and noisy records	the variations in between clean and noisy records		m

TABLE 8.5.1 TESTCASES

CHAPTER 9

SCREENS

9. SCREEN SHOTS

CHAPTER 10

CONCLUSION

10. CONCLUSION

What we can conclude from the above situation and usage is that greater the database and more the models and use cases, the better is the reaction produced for the client. In any case, the issues are many. For adjusting more uses, the scope changes from the investigation of AI to language examine.

We additionally need to recollect that we are taking a shot at a cell phone. The NLP is very broad and maybe utilizing them on a server and isolating this application into customer and server side application can fathom the speed issue as when we do that the speed of the program won't be restricted to the equipment.

We live in a time of intelligent technology. Our watches let us know the time, however they likewise remind us to work out. Our telephones prescribe the best places to eat, and our PCs foresee our inclinations, helping us to do our everyday work all the more productively.

All things considered, these advanced collaborators show just a modest bit of Artificial Intelligence (AI).

Google's Google Assistant or Apple's Siri is a new form of AI which is used by millions of people every day.The vast majority of the purchaser level computerized reasoning applications we're communicating with are customized to utilize gathered information put away in their databases to improve a response to inputs, which prompts a superior reaction inside foreordained parameters.

ELIZA, while itself a moderately straightforward framework, is significant from the perspective of understanding human insight and theoretical machine knowledge as it incites us to ask what knowledge really is and what can be viewed as authentic insight or "unique" deliberateness in a machine. It additionally influences us to consider the measure of what is seen as canny conduct of an operator lies.

In spite of the fact that the discussions are linguistically right, they don't pass on an important discussion as the program doesn't appear to have the capacity to comprehend whatever is being said by the client and henceforth the client never feels associated while chatting with Eliza or Cleverbot. There is by all accounts no "memory" and not in any case a feeling of character.

Such a machine is of no utilization to utilize.

Be that as it may in the event that we build up a program which can hold every one of the records and the "recollections" of the client, his sentiments the names and every one of the things which matter to him/her, we can truly approached assemble something which can assist a client when he is focused and act like an instructor.

This can be accomplished with the accompanying two segments which we plan on actualizing later on.

NLP is required with the goal that the parsed information can be "comprehended" by the application, things, for example, a client's temperament, his sentiments, if there is a funniness in the announcement, the names and places referenced in the info. Different things, and so forth.

Database: A Database is required with the goal that the parsed information can be put away by the application, things, for example, a client's mind-set, his sentiments, and his state of mind previously. What's more, on the off chance that he is feeling also worried, a pressure call or some other careful steps can be taken. Different things of significance to the client can likewise be put away.

For example, names and places. Different things, and so forth. A Database can be utilized with the goal that the client can feel that he is associated with somebody and not conversing with a machine which overlooks each time whatever the client says.

Future Enhancements:

The 21st century has seen the advancement in technology and its impact on the people. In many ways, the technology innovation has helped to make the lives of the people easier.

It is widely known that information is carried out from one person to another. When someone does not know about something, he asks the other person. That way the information is carried on. Keeping this fact in mind when the computers came into existence, some of the scholars and scientists came up with the idea of making them answer the problems of the people. By creating this portal, the users could get their problems answered.

Google started with a simple web application which answered the queries of the people but that was something different from one on one talk. So it introduced the concept of Google Assistant through which every user could talk. Not only the user can talk but it is equipped with top notch features today.

You might have heard the statement “Something that works perfectly doesn’t mean it can’t be upgraded”. Keeping this thought in mind, this project can also be upgraded to a much higher extent according to the need.

- One of the major additions that can be done is the introduction of Voice API in the near future. The users will give the queries through their voice and the system will give text and voice output both.

REFERENCES

Published Papers:

1. Weizenbaum, Joseph. "ELIZA - A Computer Program for the Study of Natural Language Communication between Man and Machine," Communications of the Association for Computing Machinery 9 (1966).

2. Suchman, Lucy A. Plans and Situated Actions: The problem of human-machine communication (Cambridge University Press, 1987).

3. Sagar Pawar, Omkar Rane, Ojas Wankhede and Pradnya Mehta. “A Web Based College Enquiry Chatbot with Results”, (IJIRSET, April 2018).

4. Pratike Salve, Vishruta Patil, Vyankatesh Gaikwad and Prof. Girish Wadhwa: “College Enquiry Chatbot”. (IJRTCC March 2017).

5. Tarun Lalwani, Shashank Bhalotia, Ashish Pal, Shreya Bisen, Prof. Vasundhra Rathod: “Implementation of a ChatBot system using AI and NLP”.

Outline Links:

1. https://www.tutorialspoint.com/aiml/aiml_introduction.htm

2.https://blog.recime.io/using-aiml-and-nlp-to-create-a-conversation-flow-for-your-chatbot-fea63d09b2e6

3. https://www.quora.com/How-would-you-compare-NLP-NLU-to-AIML

4. https://www.devdungeon.com/content/ai-chat-bot-python-aiml

5.https://www.quora.com/Does-AIML-count-as-artificial-intelligence

6.https://towardsdatascience.com/develop-a-nlp-model-in-python-deploy-it-with-flask-step-by- step-744f3bdd7776

7. https://sqliteonline.com/

.

Bibliopgraphy:

1. Al-Sabbagh, A. E., & Al-Sabbagh, M. (2019). A comprehensive study of chatbots: History, classification, design consideration, challenges, applications, future directions. Journal of King Saud University-Computer and Information Sciences, 31(4), 408-422.
2. Chollet, F. (2018). Deep learning with Python. Manning Publications.
3. Django. (n.d.). Retrieved April 30, 2023, from https://www.djangoproject.com/
4. Flask. (n.d.). Retrieved April 30, 2023, from https://flask.palletsprojects.com/en/2.1.x/
5. Kaul, N., Purohit, P., & Singh, D. (2019). A review of chatbot technology. International Journal of Advanced Research in Computer Science, 10(2), 1-8.
6. Python. (n.d.). Retrieved April 30, 2023, from https://www.python.org/
7. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).

Sample Code

```
from spacy.lang.en import English
import numpy
from flask import Flask, render_template, request
import json
import pickle
import os
import time
import tensorflow as tf
from tensorflow.keras import layers, models, regularizers
from voc import voc
import random
nlp = English()
tokenizer = nlp.Defaults.create_tokenizer(nlp)
PAD_Token=0
app = Flask(__name__)
model= models.load_model('mymodel.h5')
with open("mydata.pickle", "rb") as f:
    data = pickle.load(f)
def predict(ques):
    ques= data.getQuestionInNum(ques)
    ques=numpy.array(ques)
   # ques=ques/255
    ques = numpy.expand_dims(ques, axis = 0)
    y_pred = model.predict(ques)
    res=numpy.argmax(y_pred, axis=1)
    return res
def getresponse(results):
    tag= data.index2tags[int(results)]
    response= data.response[tag]
    return response
def chat(inp):
    while True:
        inp_x=inp.lower()
        results = predict(inp_x)
        response= getresponse(results)
        return random.choice(response)
@app.route("/")
def home():
    return render_template("index.html")
@app.route("/get")
def get_bot_response():
    userText = request.args.get('msg')
    time.sleep(1)
    return str(chat(userText))
if __name__ == "__main__":
        app.run()
import numpy
import json
import tensorflow as tf
from tensorflow.keras import layers, models, regularizers
import pickle
from voc import voc
def splitDataset(data):
    x_train=[ data.getQuestionInNum(x) for x in data.questions]
    y_train=[data.getTag(data.questions[x]) for x in data.questions]
    return x_train,y_train
with open("intents.json") as file:
    raw_data = json.load(file)




data=voc()
for intent in raw_data["intents"]:
    tag=intent["tag"]
    data.addTags(tag)
    for question in intent["patterns"]:
        ques=question.lower()
        data.addQuestion(ques,tag)
_train,y_train=splitDataset(data)
x_train=numpy.array(x_train)
y_train=numpy.array(y_train)
#normalize
#x_train=x_train/255
```

```
#reshape ytrain'''
y_train = y_train.reshape((len(y_train), 1))
encoder = OneHotEncoder(sparse=False)
y_train=encoder.fit_transform(y_train)'''
#intialising the ANN
model = models.Sequential()
# adding first layer
model.add(layers.Dense(units = 12, input_dim = len(x_train[0])))
model.add(layers.Activation('relu'))
#adding 2nd hidden layer
model.add(layers.Dense(units = 8))
model.add(layers.Activation('relu'))
#adding output layer
model.add(layers.Dense(units = 38))
model.add(layers.Activation('softmax'))
# Compiling the ANN
model.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
# Fitting the ANN model to training set
model.fit(x_train, y_train, batch_size = 10, epochs = 100)
model.save('mymodel.h5')
#removing questions from data as its not needed it will be entered by user
#we need other info to decode prediction to text so save it inpickle
data.questions={}
# save answers from json to pickle
```

```
for intent in raw_data["intents"]:
    tag=intent["tag"]
    response=[]
    for resp in intent["responses"]:
        response.append(resp)
    data.addResponse(tag,response)


with open('mydata.pickle', 'wb') as handle:
    pickle.dump(data, handle)
# predecting the test set Results
x_test=numpy.array(x_train[0])
img = numpy.expand_dims(x_test, axis = 0)
y_pred = model.predict(img)
p=numpy.argmax(y_pred, axis=1)
```