**CHAPTER 1**

**INTRODUCTION**

**1.1 GENERAL OVERVIEW**

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python uses whitespace indentation, rather than curly brackets or keywords, to delimit blocks. An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block. Thus, the program's visual structure accurately represents the program's semantic structure. This feature is sometimes termed the off-side rule, which some other languages share, but in most languages indentation doesn't have any semantic meaning.

* **PROBLEM STATEMENT**

A chatbot is a software application used to conduct an on-line chat conversation via text or text-to-speech, in lieu of providing direct contact with a live human agent. Designed to convincingly simulate the way a human would behave as a conversational partner, chatbot systems typically require continuous tuning and testing, and many in production remain unable to adequately converse or pass the industry standard Turing test. The term "ChatterBot" was originally coined by Michael Mauldin (creator of the first Verbot) in 1994 to describe these conversational programs.

Chatbots are typically used in dialog systems for various purposes including customer service, request routing, or for information gathering. While some chatbot applications use extensive word-classification processes, Natural Language processors, and sophisticated AI, others simply scan for general keywords and generate responses using common phrases obtained from an associated library or database.

Today, most chatbots are accessed on-line via website popups, or through virtual assistants such as Google Assistant, Amazon Alexa, or messaging apps such as Facebook Messenger or WeChat. Chatbots are typically classified into usage categories that include: commerce (e-commerce via chat), education, entertainment, finance, health, news, and productivity.

**1.3 OBJECTIVES**

The main objective of this project is to maintain and manage a chat bot which would contain all answers needed by the customer. The main objectives are given below:

* The project needs proper details of questions and the corresponding answers in a well structured form.
* Various customer can register and take part in the chatbot
* All the information is maintained by the administrators based on the information available.Any action needed can be implemented by him only.

**1.4 PURPOSE**

The main purpose of this project is to build a database that would benefit the customer in getting answer to their correspoding question.It can used as a "customer care service", Which can be fulfilled by using this project model.

The following are the advantages that are provided by the system:

* Providing answers to questions
* Viewing the "frequently asked questions".
* Cheaper than "Call centers"
* Takes less time to answer to it.

**CHAPTER 2**

**SOFTWARE AND HARDWARE REQUIREMENTS**

**2.1 SOFTWARE REQUIREMENTS**

The table 2.1 shows the software requirements for the execution of the project.

Table 2.1: Software Requirements

|  |  |
| --- | --- |
| **NAME OF THE COMPONENT** | **SPECIFICATION** |
| Operating System | Windows 7 and above |
| Language | Python |
| Browser | Any of Microsoft edge, Google Chrome,etc. |
| Scripting Language enable | HTML, CSS |

**2.2 HARDWARE REQUIREMENTS**

The table 2.2 shows the hardware requirements for the execution of the project.

Table 2.2: Hardware Requirements

|  |  |
| --- | --- |
| **NAME OF THE COMPONENT** | **SPECIFICATION** |
| Processor | Intel i5 or above processor with 1.60 GHz  Clock speed |
| RAM | Minimum 2GB |
| Hard Disk | Minimum 250GB |
| Monitor | color monitor |
| Keyboard | 104 keys |

**2.3 FRONT END TOOLS: HTML, CSS**

**Hypertext Markup Language** (**HTML**) is the standard markup language for creating web pages and web applications.

With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page.

A web page can contain headings, paragraphs, images, videos, and many other types of data. [Front-end developers](https://generalassemb.ly/coding) use HTML elements to specify what kind of information each item on a web page contains for instance. Developers also write HTML code to specify how different items relate to one another in the overall structure of the page.

**Cascading Style Sheets** (**CSS**) is a style sheet language used for describing the presentation of a document written in a markup language like HTML.

CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

CSS has a simple [syntax](https://en.wikipedia.org/wiki/Syntax) and uses a number of English keywords to specify the names of various style properties. A style sheet consists of a list of rules. Each rule or rule-set consists of one or more selectors, and a declaration block.

**2.4 BACK-END TOOL: PHP**

**PHP** is a [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language) originally designed for [web development](https://en.wikipedia.org/wiki/Web_development). It was originally created by [Rasmus HYPERLINK "https://en.wikipedia.org/wiki/Rasmus\_Lerdorf" HYPERLINK "https://en.wikipedia.org/wiki/Rasmus\_Lerdorf" HYPERLINK "https://en.wikipedia.org/wiki/Rasmus\_Lerdorf"Lerdorf](https://en.wikipedia.org/wiki/Rasmus_Lerdorf) in 1994, the PHP [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the [recursive initialism](https://en.wikipedia.org/wiki/Recursive_initialism) PHP: Hypertext Preprocessor.

PHP code may be executed with a [command line interface](https://en.wikipedia.org/wiki/Command-line_interface) (CLI), embedded into [HTML](https://en.wikipedia.org/wiki/HTML) code, or used in combination with various [web template systems](https://en.wikipedia.org/wiki/Web_template_system), web [content management systems](https://en.wikipedia.org/wiki/Content_management_system), and [web frameworks](https://en.wikipedia.org/wiki/Web_framework). PHP code is usually processed by a PHP [interpreter](https://en.wikipedia.org/wiki/Interpreter_(computing)) implemented as a [module](https://en.wikipedia.org/wiki/Plugin_(computing)) in a web server or as a [Common Gateway Interface](https://en.wikipedia.org/wiki/Common_Gateway_Interface) (CGI) executable.

PHP is a recursive acronym for "PHP: Hypertext Preprocessor".PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites. It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server. PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time. PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.

**CHAPTER 3**

**PACKAGES**

**CHATTER BOT**

ChatterBot is a Python library that makes it easy to generate automated responses to a user’s input. ChatterBot uses a selection of machine learning algorithms to produce different types of responses. This makes it easy for developers to create chat bots and automate conversations with users. For more details about the ideas and concepts behind ChatterBot see the process flow diagram.

An example of typical input would be something like this:

user: Good morning! How are you doing?

bot: I am doing very well, thank you for asking.

user: You're welcome.

bot: Do you like hats?

**NUMPY**

NumPy is a Python package which stands for ‘Numerical Python’. It is the core library for scientific computing, which contains a powerful n-dimensional array object, provide tools for integrating C, C++ etc. It is also useful in linear algebra, random number capability etc. NumPy array can also be used as an efficient multi-dimensional container for generic data. Now, let me tell you what exactly is a python numpy array.

**SQLALCHEMY**

SQLAlchemy is the Python SQL toolkit and Object Relational Mapper that gives application developers the full power and flexibility of SQL. SQLAlchemy provides a full suite of well known enterprise-level persistence patterns, designed for efficient and high-performing database access, adapted into a simple and Pythonic domain language.

**Regex**

Both patterns and strings to be searched can be Unicode strings (str) as well as 8-bit strings (bytes). However, Unicode strings and 8-bit strings cannot be mixed: that is, you cannot match a Unicode string with a byte pattern or vice-versa; similarly, when asking for a substitution, the replacement string must be of the same type as both the pattern and the search string.

**Wheel**

Wheels are the new standard of Python distribution and are intended to replace eggs. Support is offered in pip >= 1.4 and setuptools >= 0.8.This used to show the all-time most-downloaded packages. The all-time list is no longer available, and the packages in the last-365-days list will change to reflect more closely what the Python community is using.

**COMMANDS**

**Python --version**

If you have Python installed then the easiest way you can check the version number is by typing "python" in your command prompt. It will show you the version number and if it is running on 32 bit or 64 bit and some other information. For some applications you would want to have a latest version and sometimes not.

**Pip --version**

PIP is a package manager for Python packages, or modules if you like. If you have Python version 3.4 or later, PIP is included by default.A package contains all the files you need for a module.Modules are Python code libraries you can include in your project.This command is used to check the version of pip.

**Pip install chatterbot**

This command is used to install chatterbot package into our program.

**Pip install numpy**

This command is used to install numpy package into our program.

**Pip install SimpleWebSocketServer**

This command is used to install SimpleWebSocketServer package into our program.

**Pip install sqlachemy**

This command is used to install Sqlchemy package into our program.

**Pip install wheel**

This command is used to install Wheel package into our program.

**Pip install regex**

This command is used to install regex(regular expressions) package into our program.

**Python chattrainer.py**

This command is used to excute the chattrainer program(with necessary package)to see meaningful result.

**Python server.py**

This program is used to execute the server.py program.

**CHAPTER 4**

**IMPLEMENTATION**

**4.1 FRONT END CODE**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "<http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>Chat Bot</title>

<meta charset="utf-8"/>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>

<script>

var ws = new WebSocket("ws://localhost:8000");

// Close socket when window closes

$(window).on('beforeunload', function(){

ws.close();

});

ws.onerror = function(event) {

location.reload();

}

ws.onmessage = function(event) {

var message\_received = event.data;

chat\_add\_message(message\_received, false);

};

// Add a message to the chat history

function chat\_add\_message(message, isUser) {

var class\_suffix = isUser ? '\_user' : '';

var html = '\

<div class="chat\_line">\

<div class="chat\_bubble'+class\_suffix+'">\

<div class="chat\_triangle'+class\_suffix+'"></div>\

'+message+'\

</div>\

</div>\

';

chat\_add\_html(html);

}

// Add HTML to the chat history

function chat\_add\_html(html) {

$("#chat\_log").append(html);

chat\_scrolldown();

}

// Scrolls the chat history to the bottom

function chat\_scrolldown() {

$("#chat\_log").animate({ scrollTop: $("#chat\_log")[0].scrollHeight }, 500);

}

// If press ENTER, talk to chat and send message to server

$(function() {

$('#chat\_input').on('keypress', function(event) {

if (event.which === 13 && $(this).val() != ""){

var message = $(this).val();

$(this).val("");

chat\_add\_message(message, true);

ws.send(message);

}

});

});

</script>

<style>

\* {

box-sizing: border-box;

-moz-box-sizing: border-box;

-webkit-box-sizing: border-box;

}

body {

font-family: Helvetica;

}

#chat\_container {

overflow: hidden;

border-radius: 15px;

border: 1px solid black;

margin: 40px 80px 0px 80px;

}

#chat\_log {

background-color: #F3F76F;

padding: 10px;

border-bottom: 1px solid black;

overflow-y: scroll;

height: 300px;

font-size: 26px;

}

#chat\_input\_container {

padding: 10px;

}

#chat\_input {

padding: 2px;

font-size: 18px;

width: 100%;

}

.chat\_line {

overflow: hidden;

width: 100%;

margin: 2px 0 12px 0;

}

.chat\_triangle, .chat\_triangle\_user {

position: absolute;

top: 0;

width: 0;

height: 0;

border-style: solid;

left: -18px;

border-width: 0 18px 13px 0;

border-color: transparent #ffffff transparent transparent;

}

.chat\_triangle\_user {

left: auto;

right: -18px;

border-width: 13px 18px 0 0;

border-color: #234b9b transparent transparent transparent;

}

.chat\_bubble, .chat\_bubble\_user {

position: relative;

float: left;

background-color: #FFF;

margin-top: 10px;

line-height: 35px;

padding: 10px 25px 10px 25px;

margin-left: 20px;

font-size: 27px;

}

.chat\_bubble\_user {

float: right;

margin-left: 0px;

margin-right: 20px;

background-color: #234b9b;

color: #FFF;

}

</style>

</head>

<body>

<div id="chat\_container">

<div id="chat\_log">

</div>

<div id="chat\_input\_container">

<div><input id="chat\_input" /></div>

</div>

</div>

</body>

</html>

CSS we use here for styling of our webpage due to this our page will look good. Here font size is use to change the size of the font. We can give it in pixel i.e. px or in percentage %. Background color use to change the color of background, background color can by many ways like we can use # keyword or direct write the color name and so on. In overflow the content should not come out of the box...

**4.2 BACK END CODE**

**4.2.1 Chattrainer.py**

from chatterbot import ChatBot

import os

def setup():

chatbot = ChatBot('Bot',

storage\_adapter='chatterbot.storage.SQLStorageAdapter',

trainer='chatterbot.trainers.ListTrainer')

for file in os.listdir('C:/Users/Windows 8.1/chatbot/data/'):

convData = open(r'C:/Users/Windows 8.1/chatbot/data/' + file,encoding='latin-1').readlines()

chatbot.set\_trainer(ListTrainer)

chatbot.train(convData)

#print("Training completed")

setup()

ChatterBot comes with built in adapter classes that allow it to connect to different types of databases. In this tutorial, we will be using the SQLStorageAdapter which allows the chat bot to connect to SQL databases.

You can run the training process multiple times to reinforce preferred responses to particular input statements. You can also run the train command on a number of different example dialogs to increase the breadth of inputs that your chat bot can respond to.

**4.2.2 Chatdirect.py**

from chatterbot import ChatBot

def get\_response(usrText):

bot = ChatBot('Bot',

storage\_adapter='chatterbot.storage.SQLStorageAdapter',

logic\_adapters=[

{

'import\_path': 'chatterbot.logic.BestMatch'

},

{

'import\_path': 'chatterbot.logic.LowConfidenceAdapter',

'threshold': 0.70,

'default\_response': 'I am sorry, but I do not understand.'

}

],

trainer='chatterbot.trainers.ListTrainer')

bot.set\_trainer(ListTrainer)

while True:

if usrText.strip()!= 'Bye':

result = bot.get\_response(usrText)

reply = str(result)

return(reply)

if usrText.strip() == 'Bye':

return('Bye')

break

}

**4.2.3 Server.py**

from SimpleWebSocketServer import SimpleWebSocketServer, WebSocket

class ChatServer(WebSocket)

def handleMessage(self):

# echo message back to client

message = self.data

response = get\_response(message)

self.sendMessage(response)

def handleConnected(self):

print(self.address, 'connected')

def handleClose(self):

print(self.address, 'closed')

server = SimpleWebSocketServer('', 8000, ChatServer)

server.serveforever()

One class, HTTPServer, is a socketserver.TCPServer subclass. It creates and listens at the HTTP socket, dispatching the requests to a handler.

**class SimpleWebSocketServer(server\_address, RequestHandlerClass)**

This class builds on the TCPServer class by storing the server address as instance variables named server\_name and server\_port. The server is accessible by the handler, typically through the handler’s server instance variable.